



Hardware and Software
Engineered to Work Together

Oracle Database 12c: Backup and Recovery Workshop

Student Guide – Volume 1

D78850GC20

Edition 2.0 | March 2015 | D90708

Learn more from Oracle University at oracle.com/education/

Authors

Maria Billings
Donna Keesling

Technical Contributors and Reviewers

Chris Andrews
Tim Chien
Donna Cooksey
Raluca Constantin
Stefan Dolea
Gerlinde Frenzen
Joel Goodman
Daniela Hansell
Dominique Jeunot
Sean Kim
Gwen Lazenby
Naoki Kato
Olga Krakovna
Cris Pedregal
Pavan Nisankara Rao
Puneet Sangar
Ron Soltani
Jim Spiller
Branislav Valny
Harald van Breederode
Lachlan Williams

Editors

Arijit Ghosh
Malavika Jinka
Smita Kommineni

Graphic Designer

Maheshwari Krishnamurthy

Publishers

Glenn Austin
Jayanthi Keshavamurthy
Srividya Rameshkumar

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Disclaimer

This document contains proprietary information and is protected by copyright and other intellectual property laws. You may copy and print this document solely for your own use in an Oracle training course. The document may not be modified or altered in any way. Except where your use constitutes "fair use" under copyright law, you may not use, share, download, upload, copy, print, display, perform, reproduce, publish, license, post, transmit, or distribute this document in whole or in part without the express authorization of Oracle.

The information contained in this document is subject to change without notice. If you find any problems in the document, please report them in writing to: Oracle University, 500 Oracle Parkway, Redwood Shores, California 94065 USA. This document is not warranted to be error-free.

Restricted Rights Notice

If this documentation is delivered to the United States Government or anyone using the documentation on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

The U.S. Government's rights to use, modify, reproduce, release, perform, display, or disclose these training materials are restricted by the terms of the applicable Oracle license agreement and/or the applicable U.S. Government contract.

Trademark Notice

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

Contents

1 Introduction

- Objectives 1-2
- Curriculum Context 1-3
- Suggested Schedule 1-4
- Oracle Database Innovation 1-5
- Enterprise Cloud Computing 1-6
- Assessing Your Recovery Requirements 1-7
- Categories of Failure 1-10
- Data Failures 1-11
- Oracle Data Protection Solutions 1-12
- Assisting with Overview and Advice 1-13
- Complete Oracle Backup Solution 1-14
- Integrated Oracle Recovery Manager (RMAN) 1-15
- Oracle Secure Backup 1-16
- Oracle Secure Backup Cloud Module 1-17
- Oracle Data Guard: Overview 1-18
- Physical Standby Database: Redo Apply Architecture 1-20
- Oracle Active Data Guard 1-21
- Logical Standby Database: SQL Apply Architecture 1-22
- Oracle Maximum Availability Architecture: Robust and Integrated Data Protection 1-23
- Basic Workshop Architecture 1-24
- Quiz 1-25
- Summary 1-26
- Practice Overview: Exploring the Course Environment 1-27

2 Getting Started

- Objectives 2-2
- Naming the Core Components of an Oracle Database Server 2-3
- Oracle Database Server Architecture: Overview 2-4
- What You Already Know About Database Storage Architecture 2-6
- Naming Logical and Physical Database Structures 2-8
- What You Already Know About Process Architecture 2-10
- Process Structures 2-11
- Reviewing Processes 2-13

Reviewing Database Writer Process (DBWN)	2-14
Reviewing Log Writer Process (LGWR)	2-15
Reviewing Checkpoint Process (CKPT)	2-17
Reviewing System Monitor Process (SMON)	2-18
Reviewing Process Monitor (PMON)	2-19
Reviewing Archiver Processes (ARCn)	2-20
Adding Process Names	2-21
Database Log Mode	2-23
ORCL Database in ASM	2-24
Facilitating Database Management with Oracle Restart	2-26
Oracle DBA Tools	2-28
Separation of DBA Duties	2-30
Connecting to RMAN and a Target Database	2-31
Using SQL in RMAN	2-32
Quick Start: A Problem-Solution Approach	2-33
Performing Restore and Recovery of a Database in NOARCHIVELOG Mode	2-34
Quiz	2-35
Summary	2-36
Practice Overview: Getting Started	2-37

3 Configuring for Recoverability

Objectives	3-2
Types of RMAN Commands	3-3
Job Commands: Example	3-4
Configuring Persistent Settings for RMAN	3-5
Viewing Persistent Settings	3-6
Managing Persistent Settings	3-7
Specifying a Retention Policy	3-8
Recovery Window Retention Policy: Example	3-10
Quiz	3-11
Using a Fast Recovery Area	3-12
Configuring the Fast Recovery Area	3-14
Sizing the Fast Recovery Area	3-16
Fast Recovery Area Space Management	3-18
Multiplexing Control Files	3-20
Control File Autobackups	3-21
Best Practice: Multiplexing Redo Log Files	3-23
Multiplexing the Redo Log	3-24
Creating Archived Redo Log Files	3-25
Configuring ARCHIVELOG Mode	3-26
Quiz	3-27

Summary 3-28
Practice Overview: Configuring for Recoverability 3-29

4 Using the RMAN Recovery Catalog

Objectives 4-2
RMAN Repository Data Storage: Comparison of Options 4-3
Storing Information in the Recovery Catalog 4-4
Reasons to Use a Recovery Catalog 4-5
Creating the Recovery Catalog: Three Steps 4-6
Configuring the Recovery Catalog Database 4-7
Creating the Recovery Catalog Owner 4-8
Creating the Recovery Catalog 4-9
Managing Target Database Records in the Recovery Catalog 4-10
Registering a Database in the Recovery Catalog 4-11
Unregistering a Target Database from the Recovery Catalog 4-13
Recovery Catalog Resynchronization: Concepts 4-14
Manually Resynchronizing the Recovery Catalog 4-16
Using RMAN Stored Scripts 4-17
Executing RMAN Stored Scripts 4-18
Maintaining RMAN Stored Scripts 4-19
Backing Up the Recovery Catalog 4-20
Creating and Using Virtual Private Catalogs 4-21
Creating a Virtual Private Catalog (12.1.0.1) 4-22
Managing Virtual Private Catalogs 4-23
Creating a Virtual Private Catalog (12.1.0.2) 4-24
Upgrading Virtual Private Catalogs for 12.1.0.2 4-26
Quiz 4-27
Summary 4-29
Practice Overview: Using the RMAN Catalog 4-30
Practice Overview: Preparing Your Training Environment 4-31

5 Backup Strategies and Terminology

Lesson Objectives 5-2
Backup Solutions: Overview 5-3
Backup Terminology 5-4
Balancing Backup and Restore Requirements 5-5
Comparing Backup Strategies 5-7
Option 1: Full and Incremental Backups 5-8
Option 2: Incrementally Updated Disk Backups 5-9
Option 3: Offloading Backups to Physical Standby Database in Data Guard Environment 5-10

Backing Up Read-Only Tablespaces	5-11
Data Warehouse Backup and Recovery: Best Practices	5-12
Additional Backup Terminology	5-13
Creating Backup Sets	5-14
Creating Image Copies	5-15
Creating a Whole Database Backup	5-16
Quiz	5-18
Summary	5-21
Practice Overview: Developing a Backup Strategy	5-22
Case Study 1: How to Protect an OLTP Database	5-23
Case Study 2: How to Protect a DSS Database	5-24
Case Study 3: How to Protect the Recovery Catalog Database	5-25

6 Performing Backups

Objectives	6-2
RMAN Backup Types	6-3
Incrementally Updated Backups	6-5
Incrementally Updated Backups: Example	6-6
Fast Incremental Backup	6-7
Maintaining Block Change Tracking File	6-8
Monitoring Block Change Tracking	6-9
Automatic Disk-to-Disk Backup and Recovery	6-10
Oracle-Suggested Backup	6-11
Reporting on Backups	6-12
Using Dynamic Views	6-13
Managing Backups: Cross-Checking and Deleting	6-14
Quiz	6-15
Summary	6-17
Practice Overview: Creating Incremental Backups	6-18

7 Improving Your Backups

Lesson Objectives	7-2
Saving Backup Space with Unused Block Compression	7-3
Compressing Backups	7-4
Using RMAN Backup Compression	7-5
Quiz	7-6
Using a Media Manager	7-7
Configuring Backup and Restore for Very Large Files	7-9
Backing Up and Restoring Very Large Files	7-10
Creating RMAN Multisection Backups	7-11
Creating Proxy Copies	7-12

Creating Duplexed Backup Sets by Using BACKUP COPIES	7-13
Creating Backups of Backup Sets	7-14
Archival Backups: Concepts	7-15
Creating Archival Backups with RMAN	7-17
Managing Archival Database Backups	7-18
Backing Up Recovery Files	7-19
Backing Up the Control File to a Trace File	7-20
Cataloging Additional Backup Files	7-21
Backing Up ASM Disk Group Metadata	7-22
Quiz	7-23
Summary	7-25
Practice Overview: Backing Up Additional Files	7-26

8 Using RMAN-Encrypted Backups

Objectives	8-2
RMAN-Encrypted Backups	8-3
Comparing OSB and RMAN Encryption	8-4
Creating RMAN-Encrypted Backups	8-5
What Is TDE?	8-6
Using Transparent-Mode Encryption	8-7
Backing Up the Keystore	8-9
Configuring RMAN Encryption	8-10
Using Password-Mode Encryption	8-11
Using Dual-Mode Encryption	8-12
RMAN-Encrypted Backups: Considerations	8-13
Restoring Encrypted Backups	8-14
Quiz	8-15
Summary	8-16
Practice Overview: Using RMAN-Encrypted Backups	8-17

9 Diagnosing Failures

Objectives	9-2
Reducing Problem Diagnosis Time	9-3
Automatic Diagnostic Workflow	9-4
Automatic Diagnostic Repository	9-5
ADR Command-Line Tool (ADRCI)	9-6
V\$DIAG_INFO View	9-7
Interpreting RMAN Message Output	9-8
DEBUG Option	9-9
Interpreting RMAN Error Stacks	9-10
Data Recovery Advisor	9-11

Data Failure: Examples	9-14
Data Recovery Advisor RMAN Command-Line Interface	9-15
Listing Data Failures	9-16
Advising on Repair	9-18
Executing Repairs	9-19
Classifying (and Closing) Failures	9-20
Data Recovery Advisor Views	9-21
Quiz	9-22
What Is Block Corruption?	9-25
Block Corruption Symptoms: ORA-01578	9-26
How to Handle Corruption	9-27
Setting Parameters to Detect Corruption	9-28
Block Media Recovery	9-29
Prerequisites for Block Media Recovery	9-30
Recovering Individual Blocks	9-31
Best Practice: Proactive Checks	9-32
Checking for Block Corruption	9-33
Automatic Block Repair: Primary Database	9-34
Automatic Block Repair: Physical Standby Database	9-35
Summary	9-36
Practice Overview: Diagnosing Database Failure	9-37

10 Restore and Recovery Concepts

Objectives	10-2
Understanding File Loss	10-3
Data Repair Techniques	10-4
Restoring and Recovering	10-6
Using RMAN RESTORE and RECOVER Commands	10-7
Instance Failure	10-8
Understanding Instance Recovery	10-9
Phases of Instance Recovery	10-10
Tuning Instance Recovery	10-11
Using the MTTR Advisor	10-12
Media Failure	10-13
Comparing Complete and Incomplete Recovery	10-14
Complete Recovery Process	10-15
Point-in-Time Recovery Process	10-16
Recovery with RESETLOGS Option	10-18
Quiz	10-19
Summary	10-23
Practice Overview: Determining Recovery Procedures	10-24

Case Study 1 10-25
Case Study 2 10-26
Case Study 3 10-27

11 Performing Recovery I

Objectives 11-2
Ensuring Backups Are Available 11-3
Restoring in NOARCHIVELOG Mode 11-4
Recovery with Incremental Backups in NOARCHIVELOG Mode 11-5
Performing Complete Recovery 11-6
Restoring ASM Disk Groups 11-8
Restoring ASM Disk Groups: Examples 11-9
What You Already Know About Recovering Image Copies 11-10
Recovering Image Copies: Example 11-11
Performing a Fast Switch to Image Copies 11-12
Using SET NEWNAME for Switching Files 11-13
Using Restore Points 11-14
Performing Point-in-Time Recovery 11-15
Quiz 11-17
Summary 11-18
Practice Overview: Recovering from Media Failure 11-19

12 Performing Recovery II

Objectives 12-2
Recovery from Loss of Server Parameter File 12-3
Restoring the Server Parameter File from the Control File Autobackup 12-4
Loss of a Control File 12-5
Recovering from the Loss of All Control File Copies: Overview 12-6
Restoring the Control File from Autobackup 12-7
Restoring the SPFILE and the Control File 12-8
Quiz 12-9
Recovering NOLOGGING Database Objects 12-10
Loss of a Redo Log File 12-11
Log Group Status: Review 12-13
Recovering from the Loss of a Redo Log Group 12-14
Clearing a Log File 12-15
Re-creating a Password Authentication File 12-16
Recovering from a Lost Index Tablespace 12-18
Recovering a Read-Only Tablespace 12-19
Automatic Tempfile Recovery 12-20
Restoring and Recovering the Database on a New Host 12-21

Preparing to Restore the Database to a New Host	12-22
Restoring the Database to a New Host	12-23
Performing Disaster Recovery	12-27
Restoring Encrypted Backups	12-29
Quiz	12-30
Summary	12-31
Practice Overview: Performing Recoveries	12-32
Practice Overview: Using RMAN Encryption	12-33

13 RMAN and Oracle Secure Backup

Objectives	13-2
Oracle Secure Backup: Overview	13-3
Oracle Secure Backup Interface Options	13-4
Managing Data to Be Protected	13-5
Backup Pieces and Backup Images	13-6
RMAN and Oracle Secure Backup: Overview	13-7
RMAN and Oracle Secure Backup Basic Process Flow	13-8
Quiz	13-9
Starting with Oracle Secure Backup	13-10
Performing Installation Tasks	13-11
Verifying Your Installation	13-12
Securing Data and Access to the Backup Domain	13-13
Preatuthorization	13-14
Defining Retention for RMAN Backups	13-15
Media Management Expiration Policies for Automated Tape Recycling	13-16
Database Backup Storage Selector	13-17
Setting Media Management Parameters in RMAN	13-18
Summary of OSB Configuration for RMAN	13-19
Backing Up the Fast Recovery Area to Tape	13-20
Oracle Secure Backup Jobs	13-21
Displaying Log Files and Transcripts	13-22
Common obtool Commands	13-23
Quiz	13-24
Summary	13-28
Practice Overview: Performing RMAN Tape Backup and Restore	13-29

14 Using Flashback Technologies

Objectives	14-2
Flashback Technologies Error Detection and Correction	14-3
What You Already Know About Transactions and Undo	14-4
Flashback Technology	14-5

Preparing Your Database for Flashback	14-7
Guaranteeing Undo Retention	14-8
Quiz	14-9
Using Flashback Technology to Query Data	14-10
Flashback Query	14-11
Flashback Version Query	14-12
Flashback Table: Overview	14-13
Flashback Table	14-14
Flashback Table: Considerations	14-15
Flashback Transaction Query	14-16
Flashback Transaction Query: Considerations	14-17
Flashback Transaction Backout	14-18
Flashing Back a Transaction	14-19
Best Practices: Undo-based Flashback Flashback Query, Flashback Table	14-20
Flashback Drop and the Recycle Bin	14-21
Recycle Bin	14-22
Bypassing the Recycle Bin	14-23
Using Flashback Data Archives	14-24
Creating a Temporal History and Enabling Archiving	14-25
How the Flashback Data Archive Works	14-26
Collecting User Context in Temporal History	14-27
Transparent Schema Evolution	14-28
Full Schema Evolution	14-29
Temporal Validity and History	14-30
Using the PERIOD FOR Clause	14-31
Filtering on Valid-Time Columns: Example 1	14-32
Filtering on Valid-Time Columns: Example 2	14-33
Using DBMS_FLASHBACK_ARCHIVE	14-34
Quiz	14-35
Summary	14-37
Practice Overview: Using Flashback Technologies	14-38

15 Flashback Database

Objectives	15-2
Flashback Database: Continuous Data Protection	15-3
Flashback Database	15-4
Flashback Database Architecture	15-5
Configuring Flashback Database	15-6
Flashback Database: Examples	15-7
Flashback Database Considerations	15-8
Monitoring Flashback Database Information	15-9

Guaranteed Restore Points 15-11
Flashback Database and Guaranteed Restore Points 15-12
Best Practices: Flashback Database 15-14
Quiz 15-16
Summary 15-18
Practice Overview: Flashback Database 15-19

16 Transporting Data

Objectives 16-2
Transporting Data Across Platforms 16-3
Transporting Data with Minimum Down Time 16-4
Transporting a Tablespace with Image Copies 16-5
Determining the Endian Format of a Platform 16-6
Using the RMAN CONVERT Command 16-7
Quiz 16-8
Transporting Data with Backup Sets 16-9
Process Steps: 1 16-11
Process Steps: 2 16-12
Database Transport: Using Data Files 16-13
Database Transportation Procedure 16-14
Database Transportation: Conversion 16-15
Database Transportation: Example 1 16-16
Database Transportation: Example 2 16-17
Database Transportation: Considerations 16-18
Database Transport with Backup Sets: 1 16-19
Database Transport with Backup Sets: 2 16-20
Transporting Inconsistent Tablespaces 16-21
Quiz 16-22
Summary 16-24
Practice Overview 16-25

17 Performing Point-in-Time Recovery

Objectives 17-2
Point-in-Time Recovery 17-3
When to Use TSPITR 17-4
PITR Terminology 17-5
Tablespace Point-in-Time Recovery: Architecture 17-6
Preparing for PITR 17-8
Determining the Correct Target Time 17-9
Determining the Tablespaces for the Recovery Set 17-10
Identifying Objects That Will Be Lost 17-11

Performing RMAN TSPITR	17-12
Performing Fully Automated TSPITR	17-13
Improving TSPITR Performance	17-14
Performing RMAN TSPITR with an RMAN-Managed Auxiliary Instance	17-15
Performing RMAN TSPITR by Using Your Own Auxiliary Instance	17-16
Troubleshooting RMAN TSPITR	17-17
Quiz	17-18
Recovering Tables from Backups	17-19
Table Recovery: Graphical Overview	17-20
Prerequisites and Limitations	17-21
Specifying the Recovery Point in Time	17-22
Process Steps of Table Recovery: 1	17-23
Process Steps of Table Recovery: 2	17-24
Quiz	17-25
Summary	17-26
Practice Overview	17-27

18 Duplicating a Database

Objectives	18-2
Using a Duplicate Database	18-3
Choosing Database Duplication Techniques	18-4
Duplicating an Active Database with “Push”	18-5
“Push” Versus “Pull” Methods of Duplication	18-6
Duplicating a Database with a Target Connection	18-7
Duplicating a Database with Recovery Catalog Without Target Connection	18-8
Duplicating a Database Without Recovery Catalog or Target Connection	18-9
Creating a Backup-Based Duplicate Database	18-10
Creating an Initialization Parameter File for the Auxiliary Instance	18-11
Specifying New Names for Your Destination	18-12
Using the SET NEWNAME Clauses	18-13
Substitution Variables for SET NEWNAME	18-14
Specifying Parameters for File Naming	18-15
Starting the Instance in NOMOUNT Mode	18-17
Ensuring That Backups and Archived Redo Log Files Are Available	18-18
Allocating Auxiliary Channels	18-19
Understanding the RMAN Duplication Operation	18-20
Specifying Options for the DUPLICATE Command	18-22
Using Additional DUPLICATE Command Options	18-23
Substitution Variables for SET NEWNAME	18-24
Quiz	18-25

Summary 18-26
Practice Overview: Duplicating a Database 18-27

19 RMAN Troubleshooting and Tuning

Objectives 19-2
Interpreting RMAN Message Output 19-3
Using the DEBUG Option 19-4
Interpreting RMAN Error Stacks 19-5
Processing an RMAN Command 19-6
Troubleshooting with RMAN 19-7
Is There a Problem? 19-8
Diagnosing Performance Bottlenecks 19-9
Diagnosing Performance Bottlenecks: Read Phase 19-10
Is There a “Write” Problem? 19-11
Diagnosing Performance Bottlenecks: Write or Copy Phase 19-12
Using Dynamic Views to Diagnose RMAN Performance 19-13
Monitoring RMAN Job Progress 19-14
Identifying Backup and Restore Bottlenecks 19-16
Asynchronous I/O Bottlenecks 19-17
Synchronous I/O Bottlenecks 19-18
Tuning RMAN Backup Performance 19-19
Parallelization of Backup Sets 19-20
Setting LARGE_POOL_SIZE 19-22
RMAN Multiplexing 19-23
Restore and Recovery Performance: Best Practices 19-25
Quiz 19-26
Summary 19-27
No Practice 19-28

20 Workshop Overview

Objectives 20-2
Workshop Structure and Approach 20-3
Business Requirements for the Workshop Database 20-5
Diagnosing the Failures 20-6
Summary 20-7

A Your Learning

Overview A-2
Enterprise Manager Database Express Menus A-5
Request Handling in EM Express A-6
Oracle SQL Developer: Connections A-7

Oracle SQL Developer: DBA Actions A-8
Continuing Your Learning A-9
Further Information A-10
Suggested Oracle University ILT Courses A-11

B Using Enterprise Manager Cloud Control

Objectives B-2
Key Challenges for Administrators B-3
Enterprise Manager Cloud Control B-4
Cloud Control Components B-6
Components and Communication Flow B-7
Oracle Management Repository B-8
Controlling the Enterprise Manager Cloud Control Framework B-9
Starting the Enterprise Manager Cloud Control Framework B-10
Stopping the Enterprise Manager Cloud Control Framework B-11
Different Target Types B-12
Target Discovery B-13
Enterprise Manager Cloud Control B-14
User Interface B-15
Security: Overview B-16
Managing Securely with Credentials B-17
Distinguishing Credentials B-18
Quiz B-20
Practice Overview: Using Enterprise Manager Cloud Control B-21

C Cloud Computing

Cloud Based? C-2
Cloud Computing Explained C-3
Cloud Computing: Essential Characteristics C-7
Cloud Computing Service Models C-8
Cloud Computing Deployment Models C-10
Sharing the Benefits of Cloud Computing C-11
Why Implement a Cloud? C-13
Oracle's Cloud Offerings C-15
Enterprise Manager Cloud Control 12c Clouds C-16
Cloud Management Life Cycle C-18
Quiz C-20

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and Error : You are not a Valid Partner use only

1

Introduction

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Describe the curriculum context and course setup
- Assess your recovery requirements
- Describe types of database failures
- Describe Oracle backup and recovery solutions
- Describe the benefits of using Oracle Secure Backup and Oracle Data Guard
- Describe the Oracle Maximum Availability Architecture



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Curriculum Context

Before this course:

After this course, consider deepening your knowledge with specialty courses, on RAC, Data Guard, and so on.

- *Oracle Database 12c: Administration Workshop*
- *Oracle Database 12c: Install and Upgrade Workshop*

In this course, you learn to:

- Secure the availability of your database by appropriate backup and recovery strategies
- Implement backup and recovery settings and perform backup operations to disk and tape
- Diagnose and repair data failures
- Restore and recover from media and other failures
- Use Flashback Technologies and data duplication to complement backup and recovery procedures

<http://education.oracle.com>

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

- **Before** taking this course, you should ensure that you fulfill the prerequisites, which include knowledge of Oracle Database 12c, SQL, and PL/SQL (for DBA use). It is recommended that you are also familiar with Oracle Enterprise Manager Cloud Control 12c.
- With **this course**, you achieve the objectives listed in the slide.
- **After** this course, which gives you experience with backup and recovery applicable to Oracle databases in general, you can continue your education by taking courses that include backup and recovery for specific environments and options, for example:
 - *Oracle Database 12c: Managing Multitenant*
 - *Oracle Database 12c: RAC Administration*
 - *Oracle Database 12c: Data Guard Administration*
 - *Oracle Database 12c: Security*

Query <http://education.oracle.com> for up-to-date course offerings.

Suggested Schedule

Day	Lessons	Day	Lessons		
1	Unit I: Introduction and Configuration <ul style="list-style-type: none"> 1. Introduction 2. Getting Started 3. Configuring for Recoverability 4. Using the RMAN Recovery Catalog 	3	<ul style="list-style-type: none"> 11. Performing Recovery, Part 1 12. Performing Recovery, Part 2 Unit IV: Additional Technologies <ul style="list-style-type: none"> 13. RMAN and Oracle Secure Backup 14. Using Flashback Technologies 		
2	Unit II: Backup <ul style="list-style-type: none"> 5. Backup Strategies and Terminology 6. Performing Backups 7. Improving Your Backups 8. Creating RMAN-Encrypted Backups Unit III: Recovery <ul style="list-style-type: none"> 9. Diagnosing Failures 10. Restore and Recovery Concepts 	4	<ul style="list-style-type: none"> 15. Flashback Database 16. Transporting Data 17. Performing Point-in-Time Recovery 18. Duplicating a Database 19. RMAN Troubleshooting & Tuning 	5	Unit V: Real-Life Hands-on <ul style="list-style-type: none"> 20. Backup and Recovery Workshop



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

This schedule is just a very general outline. Your instructor determines the actual class schedule.

Oracle Database Innovation

... continuing with

Oracle Database 12c

... with Oracle Database 11g

... with Oracle Database 10g

Private DB Cloud

Defense in Depth

Information Lifecycle Mgt

Extreme Availability

Flex Clusters

Performance and Ease of Use

Oracle Grid Infrastructure

Real Application Testing

Automatic SQL Tuning

Fault Management

Audit Vault

Database Vault

Secure Enterprise Search

Grid Computing

Automatic Storage Mgmt

Self Managing Database

XML Database, Oracle Data Guard, RAC, Flashback Query, Virtual Private Database

Built-in Java VM , Partitioning Support, Built-in Messaging, Object Relational Support, Multimedia Support



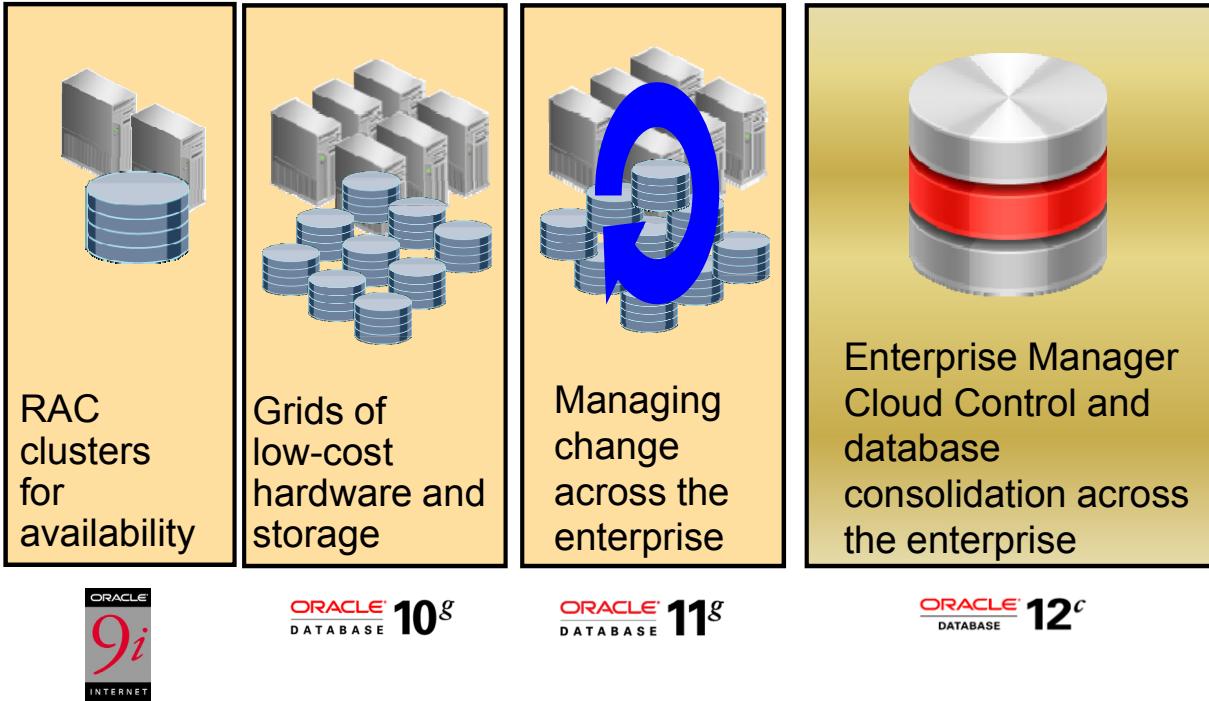
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

As a result of its early focus on innovation, Oracle has maintained the lead in the industry with a large number of trend-setting products.

Some of the marquee areas in the Oracle Database 12c release are the following:

- Private Database Cloud
- Defense in Depth including Oracle Data Redaction, Real Application Security
- Information Lifecycle Management (ILM), which includes hot/cold data classification, declarative compression and tiering, In-database Archiving, and Valid-Time Temporal
- Flex Clusters
- Extreme Availability, which includes Data Guard Far-Sync and Application Continuity
- Lower Cost Migrations
- Performance and Ease of Use, which includes “just-in-time” optimizations, attribute clustering, and zone maps for Exadata only

Enterprise Cloud Computing



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

ORACLE

Oracle Database 10g was the first database management system designed for grid computing.

Oracle Database 11g consolidates and extends Oracle's unique ability to deliver the benefits of grid computing, transforming data centers from silos of isolated system resources to shared pools of servers and storage.

Oracle Database 12c and Enterprise Manager Cloud Control are designed for cloud computing. Cloud computing creates a complete, pre-integrated, off-the-shelf private cloud solution that allows you to quickly transform the enterprise data center into a private cloud.

The key benefits are the following:

- Reduce server sprawl and improve CPU utilization by consolidating on fewer servers.
- Reduce the amount of time a DBA spends installing and configuring databases, by automating deployment of standard database configurations.
- A single console manages the entire Cloud life cycle—plan, set up, deliver, and operate.
- Prevent resource hogging by setting quotas for individual users.
- Forecast future resource needs by analyzing trending reports.
- Compute chargeback based on performance and configuration metrics.

Assessing Your Recovery Requirements

- Identify and prioritize critical data.
- Base recovery requirements on data criticality.
 - Recovery Point Objective (RPO): Tolerance for data loss
 - How frequently should backups be taken?
 - Is point-in-time recovery required?
 - Recovery Time Objective (RTO): Tolerance for down time
 - Down time: Problem identification + recovery planning + systems recovery
 - Tiered RTO per level of granularity (database, tablespace, table, row)
 - Determine backup retention policy for on-site, off-site, and long-term backups.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

To assess your recovery requirements properly, you should first determine how critical a database that is lost or unavailable is to your business. Consider the following:

- How much will the company lose per hour of down time?
- How much will be lost in productivity and labor costs if the database is down?

By quantifying these costs, you will be able to justify hardware and storage expenditures to prevent and recover from database failures.

Your next step is to classify your databases according to their criticality. As an example, a company has a large data warehousing reporting system that can tolerate 12 hours worth of lost data, because batch loads can be rerun with a few hours of down time that is acceptable to the user community. A disk and tape backup strategy may be appropriate for this type of system.

A company has a critical OLTP system that can tolerate no more than a few minutes worth of lost data and several minutes of acceptable down time, because a down database translates into \$100,000 per hour of lost revenue. For this system, a traditional backup and recovery solution will not suffice. The company needs to consider a “minimal downtime solution” such as Oracle Data Guard.

Determine your *Recovery Point Objective (RPO)*, which helps you evaluate how much data loss is acceptable. Consider the following:

- Are there some databases that require more frequent backups?
- Is adequate disk space available for archived logs (for point-in-time recovery as an example)?

Consider your *Recovery Time Objective (RTO)* to determine how much recovery time you can tolerate. Is it hours, or only minutes? Bear in mind that RTO may vary by database and/or server. You may require a combined disk and tape backup strategy for critical databases to meet more pressing RTO requirements. When you have determined your desired recovery capability from disk and tape, assess the retention period for backups.

Assessing Your Recovery Requirements

- Assess data protection requirements.
 - Physical: Disasters, outages, failures, corruptions
 - Logical: Human errors, application errors



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In addition to assessing your recovery requirements, you need to determine which failures you need to protect against. Consider physical losses of data and logical errors that can occur in your application and database.

Categories of Failure

Failures can generally be divided into the following categories:

- Statement failure
- User process failure
- Network failure
- User error
- Instance failure
- Media failure



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

- **Statement failure:** A single database operation (select, insert, update, or delete) fails.
- **User process failure:** A single database session fails.
- **Network failure:** Connectivity to the database is lost.
- **User error:** A user successfully completes an operation, but the operation (dropping a table or entering bad data) is incorrect.
- **Instance failure:** The database instance shuts down unexpectedly.
- **Media failure:** A loss of any file that is needed for database operation (that is, the files have been deleted or the disk has failed).

Data Failures

- Inaccessible components: Missing data files at the OS level, incorrect access permissions, offline tablespace
- Physical corruptions: Block checksum failures, invalid block header field values
- Logical corruptions: Inconsistent dictionary; corrupt row piece, index entry, or transaction
- Inconsistencies: Control file older or newer than the data files and online redo logs
- I/O failures: Limit on the number of open files exceeded, inaccessible channels, network or I/O error



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Included in the slide are additional types of failures you should anticipate in your environment. Throughout this course, methods to address various types of failures are presented.

Oracle Data Protection Solutions

Backup and Recovery Objective	Recovery Time Objective (RTO)	Oracle Solution
Physical data protection	Hours/Days	Recovery Manager Oracle Secure Backup
Logical data protection	Minutes/Hours	Flashback Technologies
Recovery analysis	Minimize time for problem identification and recovery planning	Data Recovery Advisor

Disaster Recovery Objective	Recovery Time Objective (RTO)	Oracle Solution
Physical data protection	Seconds/Minutes	Data Guard Active Data Guard



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Oracle provides an appropriate data protection solution depending on your backup and recovery objective and RTO:

- Oracle Recovery Manager (RMAN) is the core Oracle Database software component that manages database backup, restore, and recovery processes.
- Oracle Secure Backup (OSB) is Oracle's enterprise-grade tape backup management solution for both database and file system data.
- Oracle Database Flashback Technologies are a set of data recovery solutions that enable human errors to be reversed by selectively and efficiently undoing the effects of a mistake.
- The Data Recovery Advisor provides intelligent database problem identification and recovery capabilities.
- Data Guard and Active Data Guard enable physical standby databases to be open for read access while being kept synchronized with the production database through media recovery.

Additional information about these features and technologies are presented in this course.

Assisting with Overview and Advice

Oracle Enterprise Manager Cloud Control 12c (Cloud Control)

- Maximum Availability Architecture (MAA) Advisor
 - For best practices that eliminate or reduce down time
 - Provides your configuration status
- High Availability Console:
 - Customizable dashboard
 - Overview of events, usage, and history
 - Organized in sections: Availability, Backup/Recovery, and Data Guard



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

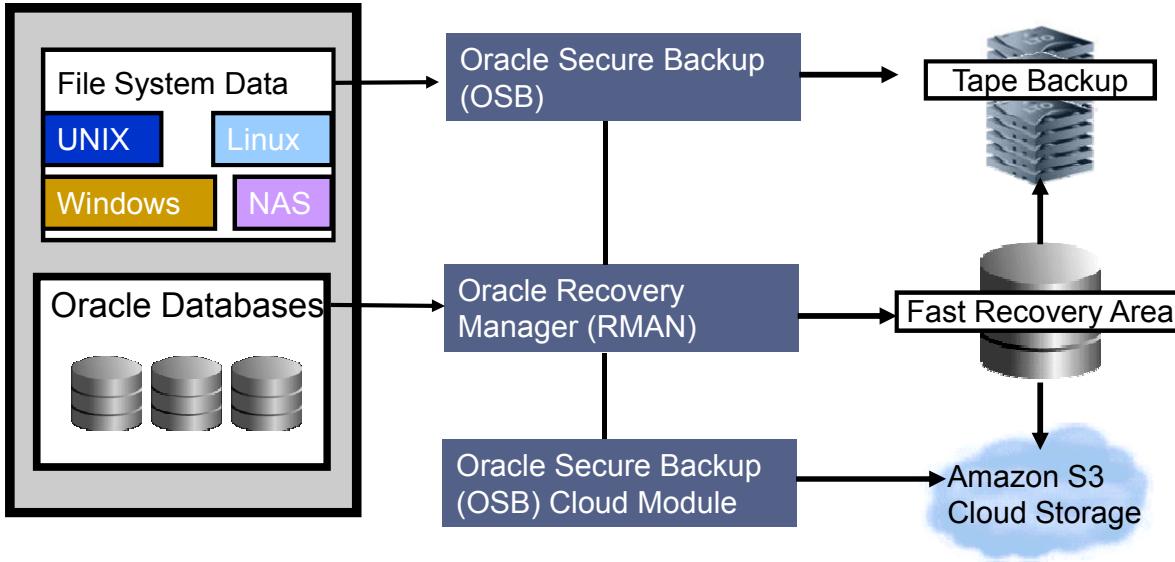
Oracle Enterprise Manager Cloud Control 12c (Cloud Control) provides the Maximum Availability Architecture (MAA) Advisor. MAA provides a fully integrated and validated High Availability (HA) architecture with operational and configuration best practices that eliminate or reduce down time.

From the database home page, navigate to Availability > MAA Advisor.

Tip: Choose “All Solutions” to display your configuration status, even if MAA is not your aim.

The High Availability Console in Cloud Control is a customizable dashboard, useful for overview and summary information. From the database home page, navigate to Availability > High Availability Console.

Complete Oracle Backup Solution



- Oracle backup and recovery for your entire IT environment
- Multiple media options: Local disk, remote cloud storage, and physical and virtual tape

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

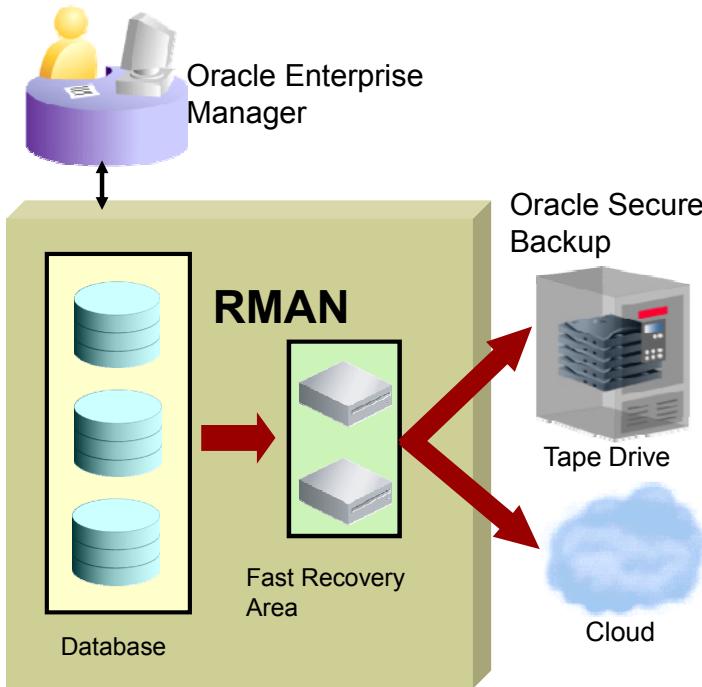
Oracle Secure Backup delivers centralized tape backup management for distributed, heterogeneous environments for your entire IT environment, by providing:

- Oracle Database integration with Recovery Manager (RMAN) supporting versions Oracle9i to Oracle Database 12c.
- File system data protection for UNIX, Windows, and Linux servers, as well as Network Attached Storage (NAS) protection via the Network Data Management Protocol (NDMP).

A key component of the Oracle disk backup strategy is the fast recovery area (FRA), a storage location on a file system or Automatic Storage Management (ASM) disk group that organizes all recovery-related files and activities for an Oracle database. All files that are required to fully recover a database from media failure can reside in the fast recovery area, including control files, archived logs, data file copies, and RMAN backups.

To meet requested Service Level Agreements (SLA), consider the use of Oracle Secure Backup (OSB) Cloud Module. With RMAN and OSB Cloud Module, you can send local disk backups directly to Amazon S3 for off-site storage.

Integrated Oracle Recovery Manager (RMAN)



- Intrinsic knowledge of database file formats and recovery procedures
 - Block validation
 - Online block-level recovery
 - Tablespace and data file recovery
 - Online, multi-streamed backup
 - Unused block compression
 - Native encryption
- Integrated disk, tape, and cloud backup leveraging the fast recovery area and Oracle Secure Backup

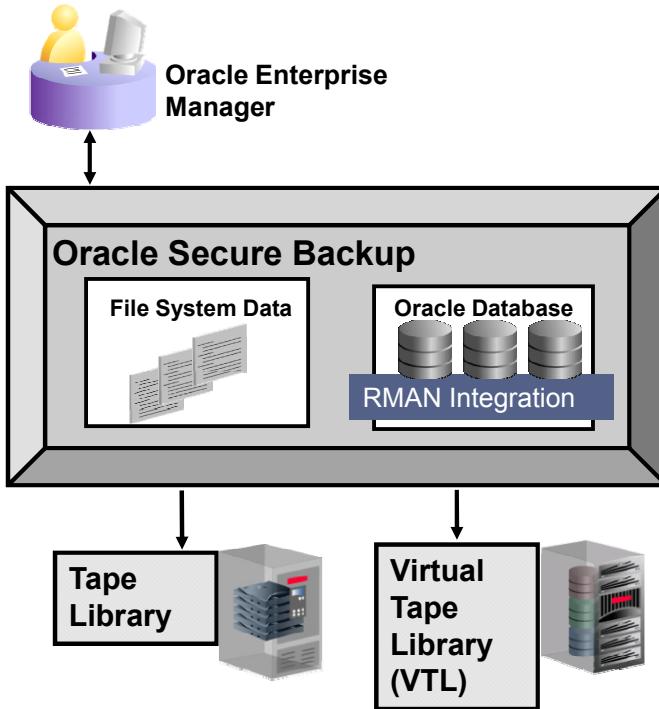
ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Oracle Recovery Manager (RMAN) is the core Oracle Database software component that manages database backup, restore, and recovery processes. RMAN maintains configurable backup and recovery policies and keeps historical records of all database backup and recovery activities. RMAN ensures that all files required to successfully restore and recover a database are included in complete database backups. Furthermore, as part of RMAN backup operations, all data blocks are verified to ensure that corrupt blocks are not propagated into the backup files.

The diagram in the slide provides one suggested configuration for backups. In this diagram, RMAN uses the fast recovery area for backups. Optionally, through Oracle Secure Backup, backups can be made to the tape drive and to the cloud. Backups can be performed to other designated locations, such as directly to disk. These options are described in subsequent lessons.

Oracle Secure Backup



- Enterprise tape backup management:
 - Heterogeneous file systems (UNIX/Linux/Windows) and NAS devices
 - Built-in Oracle integration
 - Centralized management in distributed environments

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Oracle Secure Backup is a centralized tape backup management software that protects the Oracle database and file systems in distributed UNIX, Linux, Windows, and Network Attached Storage (NAS) environments. Oracle Secure Backup provides tape backup for application files as well as the database.

Oracle Secure Backup is integrated with RMAN and provides the media management layer for RMAN backups to tape.

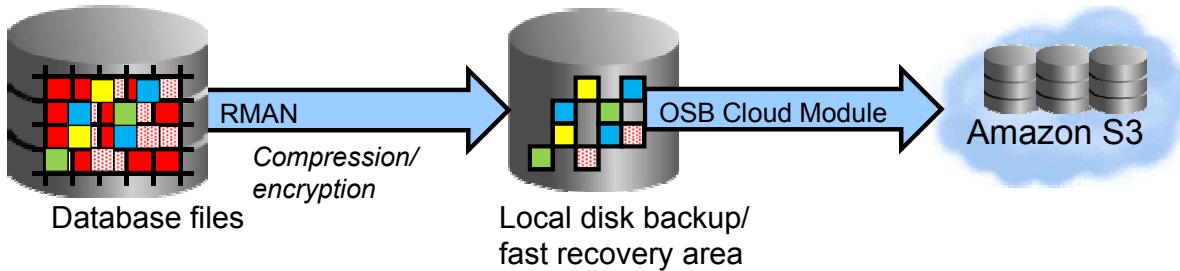
Through the use of Oracle Secure Backup, tape backup management is automated through user-defined policies throughout the life cycle of the backup tape.

From one central console, you can easily manage the distributed servers and tape devices within the backup domain.

The *Oracle Secure Backup* course provides detailed information about installing and using Oracle Secure Backup.

Note: Refer to *Oracle Secure Backup Licensing Information* for detailed information about feature availability with Oracle Secure Backup and Oracle Secure Backup Express.

Oracle Secure Backup Cloud Module



Back up databases to Amazon Cloud:

- Provides reliable off-site backups
- Complements local disk and/or tape backup
- May eliminate IT management overhead of a disaster recovery site



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

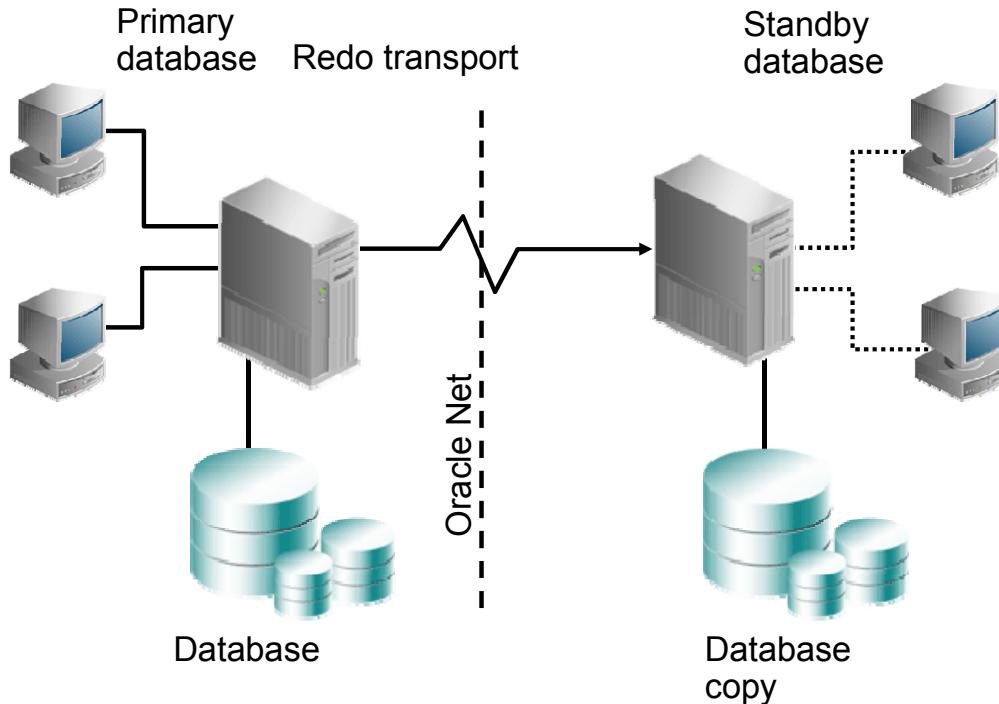
OSB Cloud Module is independent of the OSB tape solution, although it is in the same product family and fully integrated with RMAN functionality.

With RMAN and the OSB Cloud Module, you can send local disk backups directly to Amazon S3 for off-site storage. The OSB Cloud Module can also be used to stream backups directly to the Cloud. This is particularly useful when the database is running in the Cloud, using services such as Amazon Elastic Compute Cloud (EC2).

Because S3 storage is disk based, it is inherently more reliable than tape media. If your backup strategy has the need to complement local disk or tape backups with reliable off-site backups and you are trying to avoid the IT management costs of maintaining a separate disaster recovery site, you could examine Amazon's S3 up time service-level agreements to determine whether this technology covers your needs.

The OSB Cloud Module can be used for all supported versions of Oracle Database. You use familiar tools such as Enterprise Manager and RMAN to perform backups to the Cloud.

Oracle Data Guard: Overview



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

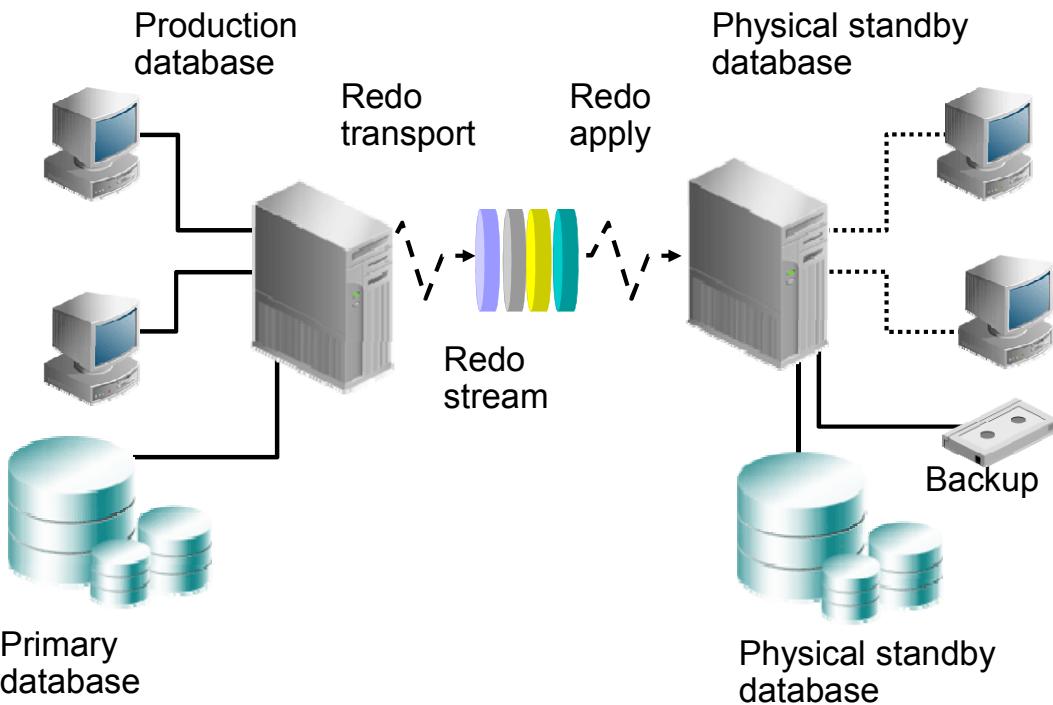
Oracle Data Guard is a central component of an integrated Oracle Database High Availability (HA) solution set that helps organizations ensure business continuity by minimizing the various kinds of planned and unplanned down time that can affect their businesses.

Oracle Data Guard is a management, monitoring, and automation software infrastructure that works with a production database and one or more standby databases to protect your data against failures, errors, and corruptions that might otherwise destroy your database. It protects critical data by providing facilities to automate the creation, management, and monitoring of the databases and other components in a Data Guard configuration. It automates the process of maintaining a copy of an Oracle production database (called a standby database) that can be used if the production database is taken offline for routine maintenance or becomes damaged.

In a Data Guard configuration, a production database is referred to as a primary database. A standby database is a copy of the primary database. Using a backup copy of the primary database, you can create from one to 30 standby databases. The standby databases, together with the primary database, make up a Data Guard configuration.

All Data Guard standby databases can enable up-to-date read access to the standby database while redo being received from the primary database is applied. This makes all standby databases excellent candidates for relieving the primary database of the overhead of supporting read-only queries and reporting.

Physical Standby Database: Redo Apply Architecture



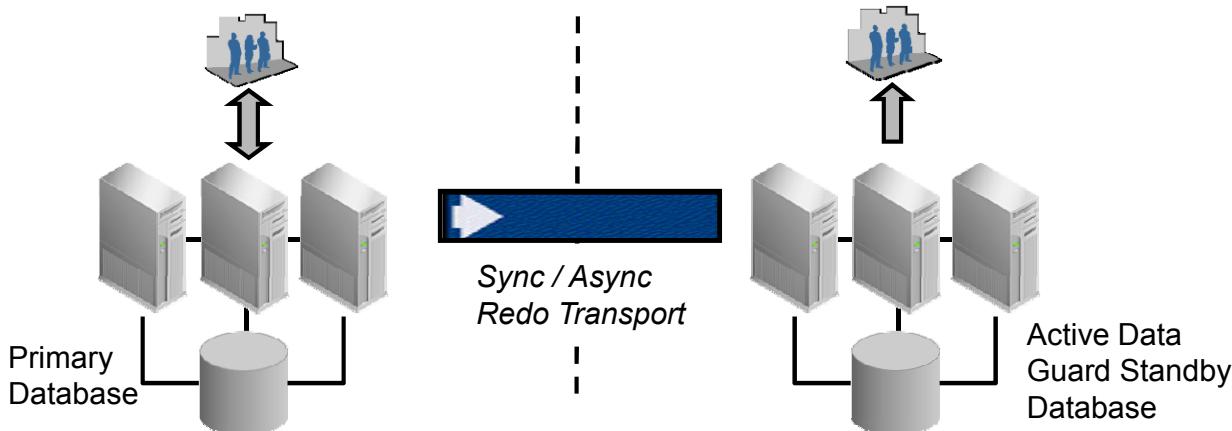
ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The Data Guard physical standby Redo Apply architecture consists of:

- A production (primary) database, which is linked to one or more standby databases (up to 30) that are identical copies of the production database. The limit of 30 standby databases is imposed by the `LOG_ARCHIVE_DEST_n` parameter. The maximum number of destinations is 31. One is used as the local archive destination, leaving the other 30 for uses such as the standby database.
Note: You can use the Cascaded Redo Log Destinations feature to incorporate more than 30 standby databases in your configuration.
- The standby database, which is updated by redo that is automatically shipped from the primary database. The redo can be shipped as it is generated or archived on the primary database. Redo is applied to each standby database by using Oracle media recovery. During planned down time, you can perform a switchover to a standby database. When a failure occurs, you can perform a failover to one of the standby databases. The physical standby database can also be used to back up the primary database.

Oracle Active Data Guard



- Data availability and data protection for Oracle Database
- Automatic block repair with zero application down time
- Standby database used for queries, reports, testing, or backups
- Up-to-date and out-of-date queries

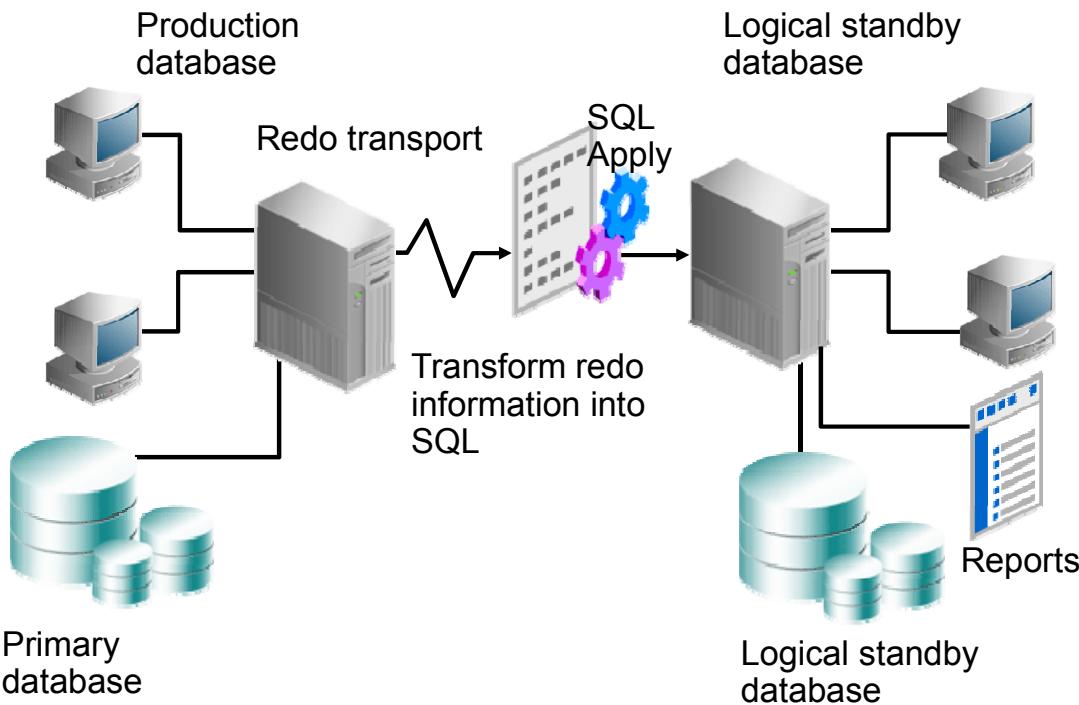
Active Data Guard allows queries against real-time data.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

- Active Data Guard enables physical standby databases to be open for read access while media recovery is being performed on them to keep them synchronized with the production database. The physical standby database is open in read-only mode while redo transport and standby apply are both active.
- Active Data Guard automatically repairs corrupt blocks online by using the active standby database.
- Normally, queries executed on active standby databases return up-to-date results. Due to potential delays in redo transport or standby apply, a standby database may “fall behind” its primary, which can cause results of queries on the standby to be out of date.
- Active Data Guard sessions can be configured with a maximum query delay (in seconds). If the standby exceeds the delay, Active Data Guard returns an error to the application, which can then retry the query or transparently redirect the query to the primary, as required.

Logical Standby Database: SQL Apply Architecture



ORACLE®

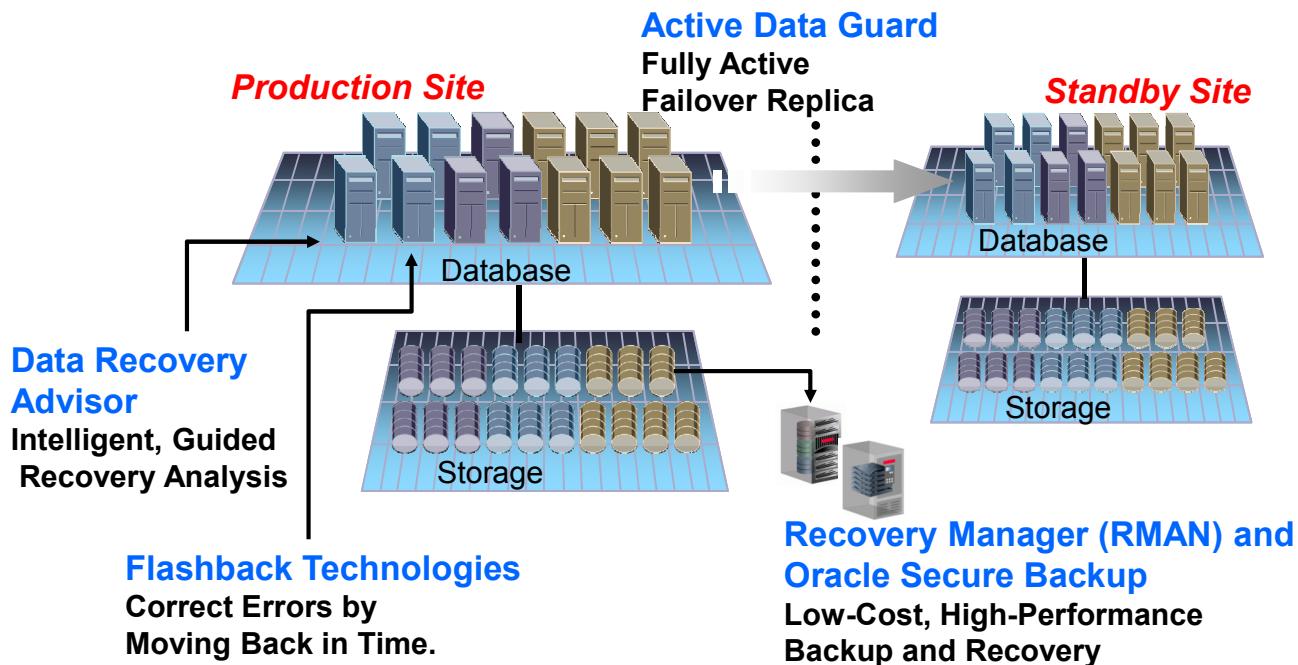
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In a logical standby database configuration, Data Guard SQL Apply uses redo information shipped from the primary system. However, instead of using media recovery to apply changes (as in the physical standby database configuration), the redo data is transformed into equivalent SQL statements by using LogMiner technology. These SQL statements are then applied to the logical standby database. The logical standby database is open in read/write mode and is available for reporting capabilities.

The logical standby database can offer protection at database level, schema level, or even object level.

A logical standby database can be used to perform rolling database upgrades, thereby minimizing down time when upgrading to new database patch sets or full database releases.

Oracle Maximum Availability Architecture: Robust and Integrated Data Protection



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

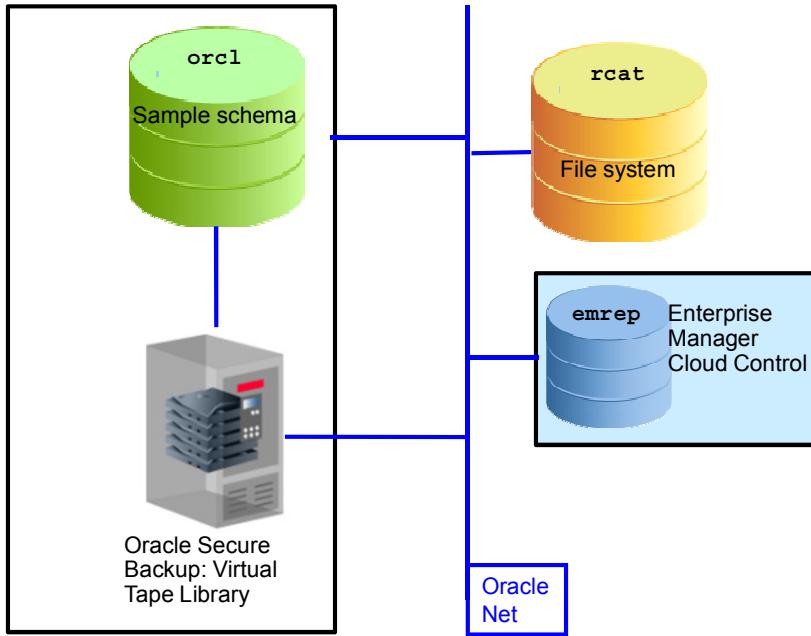
Oracle Maximum Availability Architecture (MAA) is Oracle's best practices blueprint based on proven Oracle high-availability technologies and recommendations. The goal of MAA is to achieve the optimal high-availability architecture at the lowest cost and complexity.

MAA integrates Oracle Database features for high availability including Real Application Clusters (RAC), Data Guard, GoldenGate, ASM, RMAN, and Enterprise Manager. MAA includes best practice recommendations for critical infrastructure components including servers, storage, and network. Beyond the technology, the MAA blueprint encompasses specific design and configuration recommendations that have been tested to ensure optimum system availability and reliability. Enterprises that leverage MAA in their IT infrastructure find that they can quickly and efficiently deploy applications that meet their business requirements for high availability.

RMAN may also be used with other non-Oracle tape backup products. However, Oracle Secure Backup provides database backup performance optimizations that may not be found with other products.

Detailed information about MAA is available on the Oracle Technology Network website (<http://www.oracle.com/technetwork/database/features/availability/index.html>).

Basic Workshop Architecture



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

For the hands-on practice, the instances displayed in this slide are preinstalled on a Linux OS: **orcl**, **rcat**, and **emrep** are instances of Oracle Database 12c.

During this course, you will add components to this setup.

Quiz

As a first step in data protection planning, you should determine the *criticality* of a database by evaluating which of the following?

- a. Loss of revenue due to down time
- b. Hardware maintenance costs
- c. Loss in productivity due to down time
- d. Software maintenance costs



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Answer: a, c

Summary

In this lesson, you should have learned how to:

- Describe the curriculum context and course setup
- Assess your recovery requirements
- Describe types of database failures
- Describe Oracle backup and recovery solutions
- Describe the benefits of using Oracle Secure Backup and Oracle Data Guard
- Describe the Oracle Maximum Availability Architecture



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Practice Overview: Exploring the Course Environment

This practice covers the following topics:

- Verifying that the three database instances are started
- Verifying that the listener is started
- Confirming the `LOG_MODE` of the ORCL database

Optionally, view the *Oracle Database 12c: Using SYSBACKUP Privilege and Predefined User* demonstration.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this practice, you explore your course environment.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and Error : You are not a Valid Partner use only

2

Getting Started

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

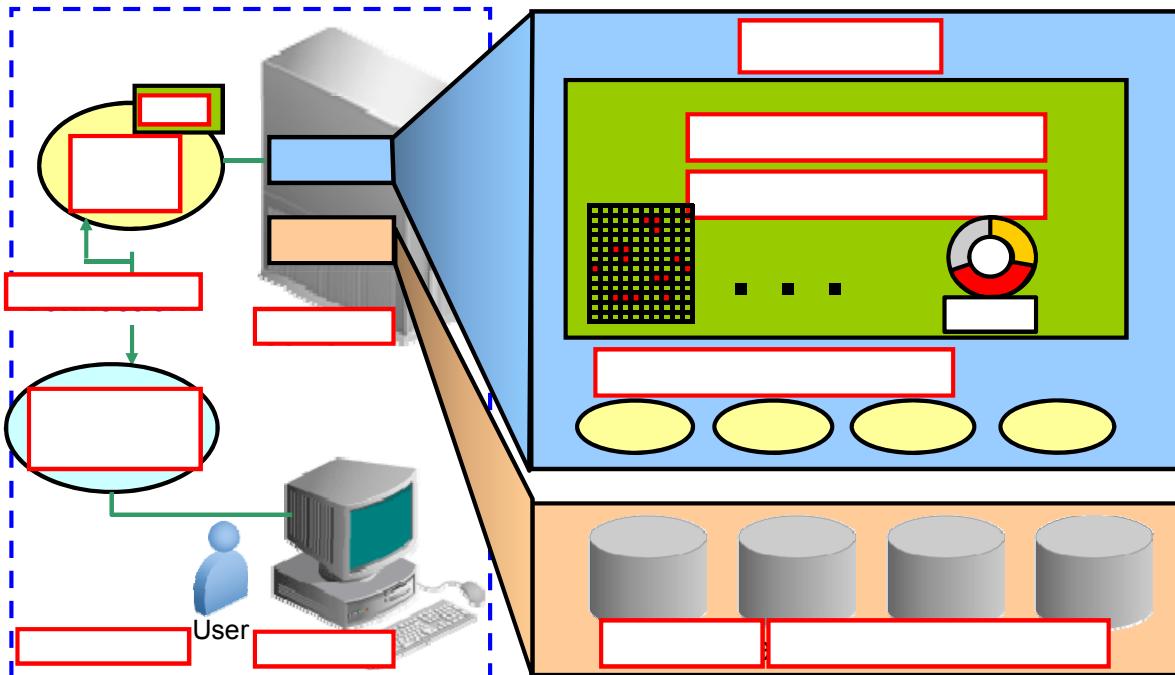
- Describe database architecture as it relates to backup and recovery
- Describe your ORCL database in NOARCHIVELOG mode
- Distinguish Oracle tools for backup and recovery
- Perform basic backup and recovery (in NOARCHIVELOG mode)



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

This lesson reviews the core concepts of the Oracle Database, which are critical for backup and recovery. Then, it provides an overview of available tools (which are used throughout this course). The lesson is followed by a hands-on practice (which might raise questions to be answered in the following lesson). You should understand how to perform backup and recovery in NOARCHIVELOG mode (and the potential shortcomings of this approach).

Naming the Core Components of an Oracle Database Server



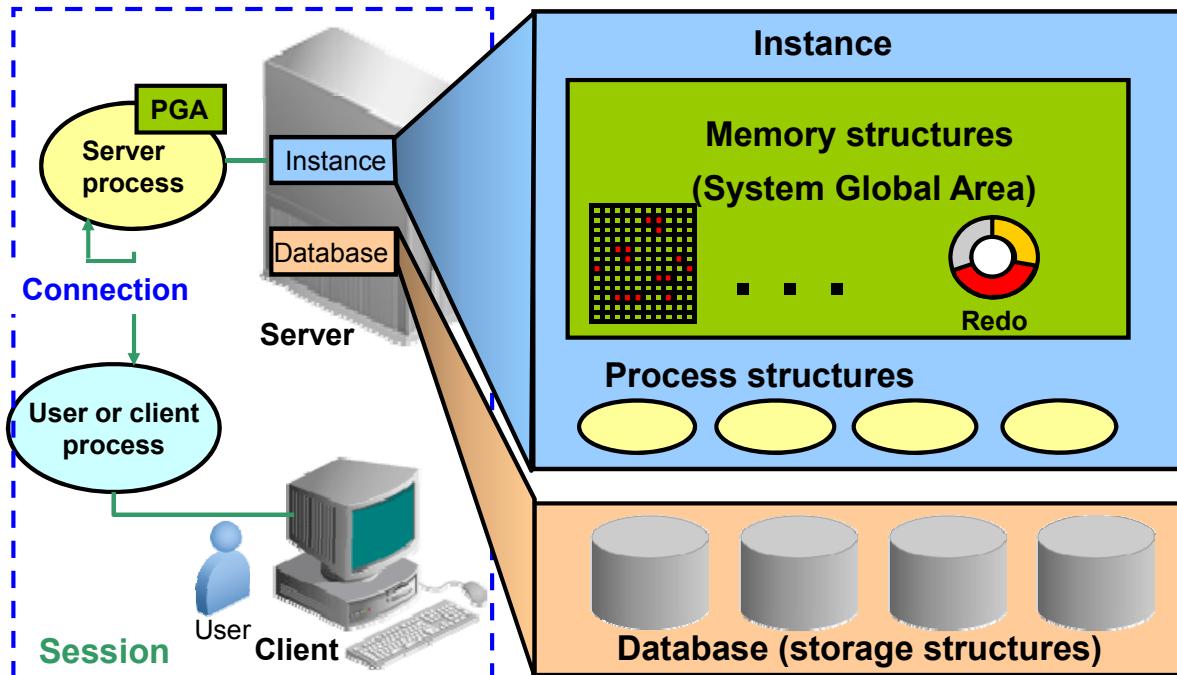
ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The following are a few sample questions to get you started by naming the core components:

1. The two main components of a basic Oracle Database system are _____ and _____.
2. The Instance consists of _____ and _____ processes.
3. The three major structures in the Oracle Database server architecture are _____, _____, and _____.
4. A session is a connection of a user to an _____ through a _____.
5. Of special interest for backup and recovery tasks is a circular _____ buffer (used for instance recovery). Instance recovery is the process of applying records in the online redo log to data files to reconstruct changes made after the most recent checkpoint.

Oracle Database Server Architecture: Overview



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

There are three major structures in the Oracle Database server architecture: memory structures, process structures, and storage structures. A basic Oracle database system consists of an Oracle database and a database instance.

The **database** consists of both physical and logical structures. Because the physical and logical structures are separate, the physical storage of data can be managed without affecting access to logical storage structures.

The **instance** consists of memory structures and background processes associated with that instance.

- Every time an instance is started, a shared memory area called the System Global Area (**SGA**) is allocated and the background processes are started. Of special interest for backup and recovery tasks is the **redo log buffer**. It caches redo information (used for instance recovery) until it can be written to the physical redo log files stored on the disk.
- **Processes** are jobs that work in the memory of computers. A process is defined as a “thread of control” or a mechanism in an operating system that can run a series of steps. After starting a database instance, the Oracle software associates the instance with a specific database. This is called *mounting the database*. The database is then ready to be opened, which makes it accessible to authorized users.

When a user starts a **transaction**—for example, a DML operation—the old data is written from the buffer cache to the undo tablespace and the new change details are in the redo log files.

Note: Oracle Automatic Storage Management (ASM) uses the concept of an instance for the memory and process components, but is not associated with a specific database.

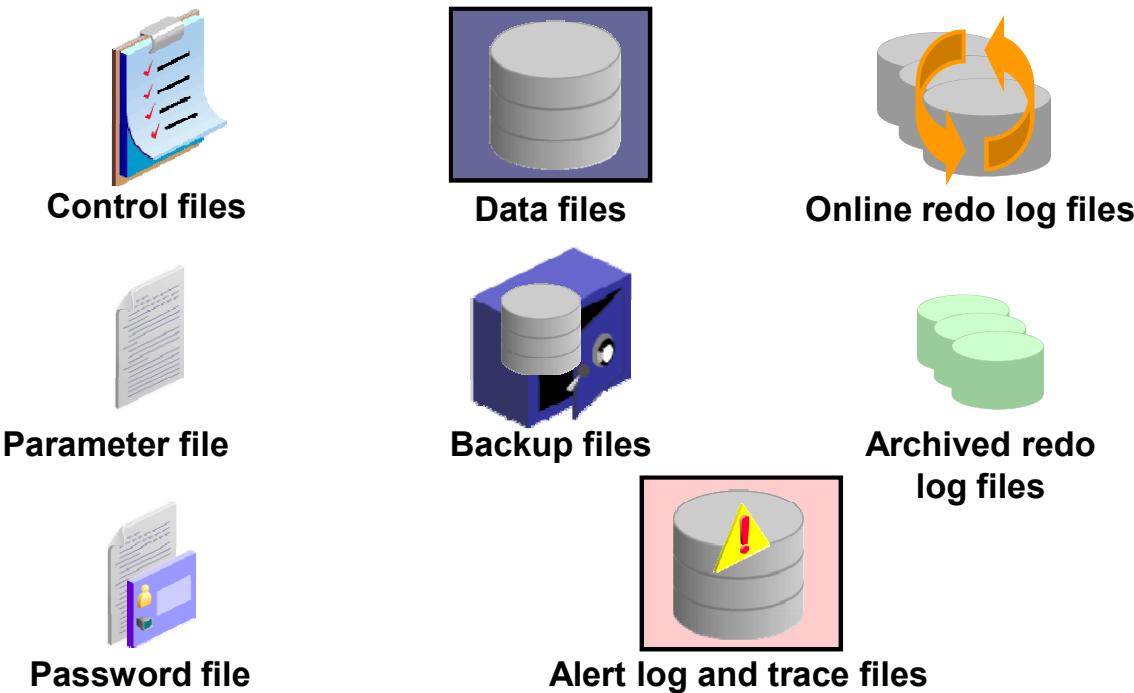
Connections and sessions are closely related to user processes but are very different in meaning.

A *connection* is a communication pathway between a user process and an Oracle Database instance. A communication pathway is established by using available interprocess communication mechanisms (on a computer that runs both the user process and Oracle Database) or network software (when different computers run the database application and Oracle Database, and communicate through a network).

A *session* represents the state of a current user login to the database instance. For example, when a user starts SQL*Plus, the user must provide a valid username and password, and then a session is established for that user. A session lasts from the time a user connects until the user disconnects or exits the database application.

Multiple sessions can be created and exist concurrently for a single Oracle database user using the same username. For example, a user with the username and password of `HR` and `oracle_4U`, respectively, can connect to the same Oracle Database instance several times.

What You Already Know About Database Storage Architecture



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The files that constitute an Oracle database are organized into the following:

- **Control files:** Contain data about the database itself (that is, physical database structure information). These files are critical to the database. Without them, you cannot open data files to access the data in the database. It can also contain metadata related to backups.
- **Data files:** Contain the user or application data of the database, as well as metadata and the data dictionary
- **Online redo log files:** Allow for instance recovery of the database. If the database server crashes and does not lose any data files, the instance can recover the database with the information in these files.

The following additional files are important to the successful running of the database:

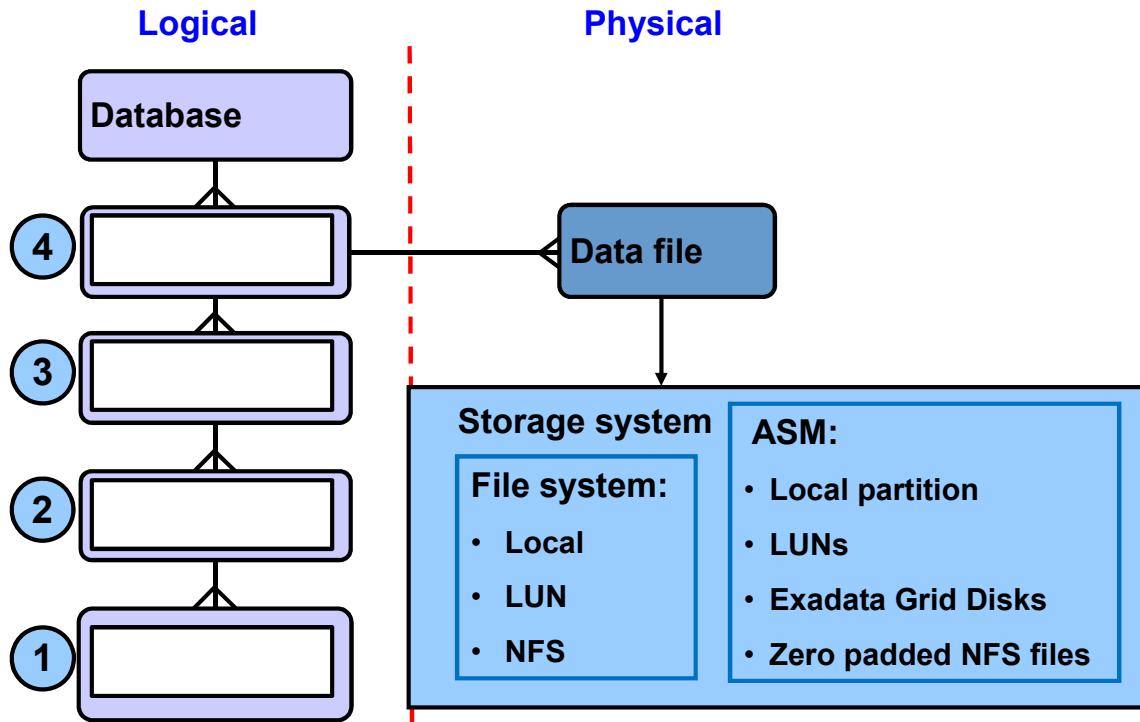
- **Parameter file:** Is used to define how the instance is configured when it starts up
- **Password file:** Allows sysdba, sysoper, sysbackup, and sysasm to connect remotely to an instance and perform administrative tasks
- **Backup files:** Are used for database recovery. You typically restore a backup file when a media failure or user error has damaged or deleted the original file.

Archived redo log files: Contain an ongoing history of the data changes (redo) that are generated by the instance. Using these files and a backup of the database, you can recover a lost data file. That is, archive logs enable the recovery of restored data files.

To monitor and diagnose use:

- **Trace files:** Each server and background process can write to an associated trace file. When an internal error is detected by a process, the process dumps information about the error to its trace file. Some of the information written to a trace file is intended for the database administrator, whereas other information is for Oracle Support Services.
- **Alert log file:** These are special trace entries. The alert log of a database is a chronological log of messages and errors. Oracle recommends that you review the alert log periodically.

Naming Logical and Physical Database Structures



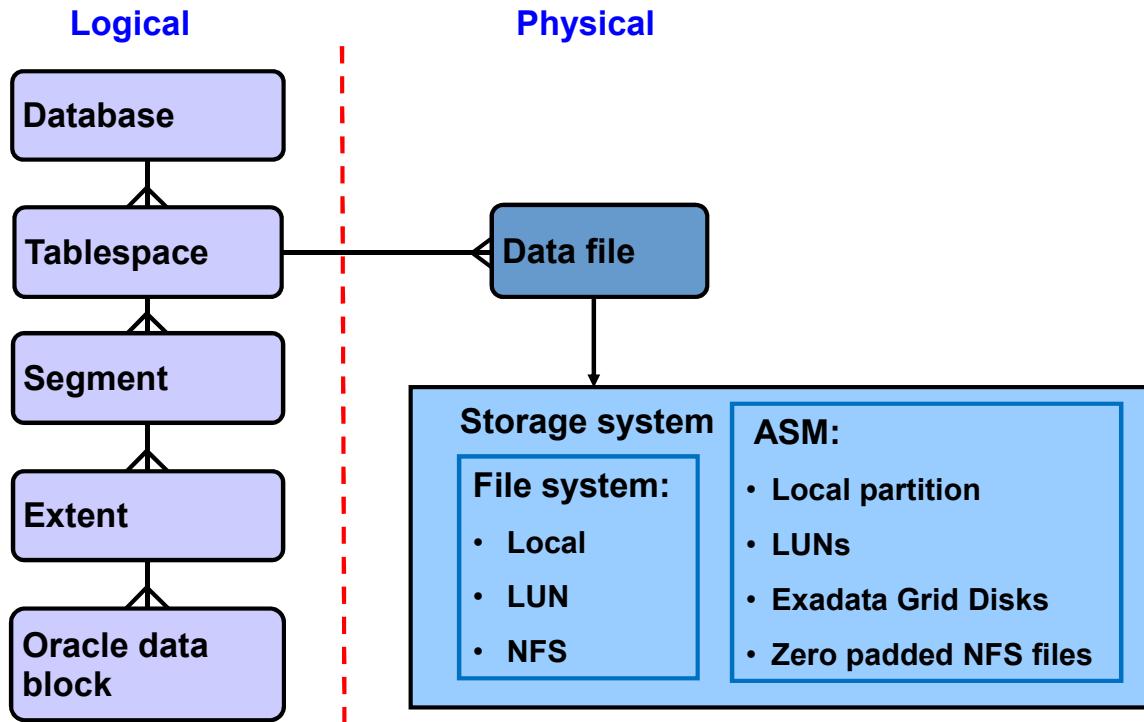
ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The database has logical structures and physical structures.

1. At the finest level of granularity, an Oracle database's data is stored in _____.
2. An _____ is a specific number of contiguous Oracle data blocks that are logically contiguous but can be physically spread out on disk (because of RAID striping and file system implementations).
3. The level of logical database storage above an _____ is called a _____.
4. A database is divided into logical storage units called _____, which group related logical structures or data files together. (Your backup and recovery strategy is likely to have special considerations for this logical storage unit.)

Naming Logical and Physical Database Structures



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The database has logical structures and physical structures.

1. At the finest level of granularity, an Oracle database's data is stored in **data blocks**.
2. An **extent** is a specific number of contiguous Oracle data blocks that are logically contiguous but can be physically spread out on disk (because of RAID striping and file system implementations).
3. The level of logical database storage above an **extent** is called a **segment**.
4. A database is divided into logical storage units called **tablespaces**, which group related logical structures or data files together. (Your backup and recovery strategy is likely to have special considerations for this logical storage unit.)

What You Already Know About Process Architecture

- User process
 - Is the application or tool that connects to the Oracle Database instance
- Database processes
 - Server process: Connects to the Oracle instance and is started when a user establishes a session
 - Background processes:
 - Are started when an Oracle instance is started
 - Perform I/O to write data from and to disk
 - Perform recovery at instance startup (if necessary)
 - Perform other tasks
- Daemon and Application processes
 - Networking listeners
 - Grid Infrastructure daemons



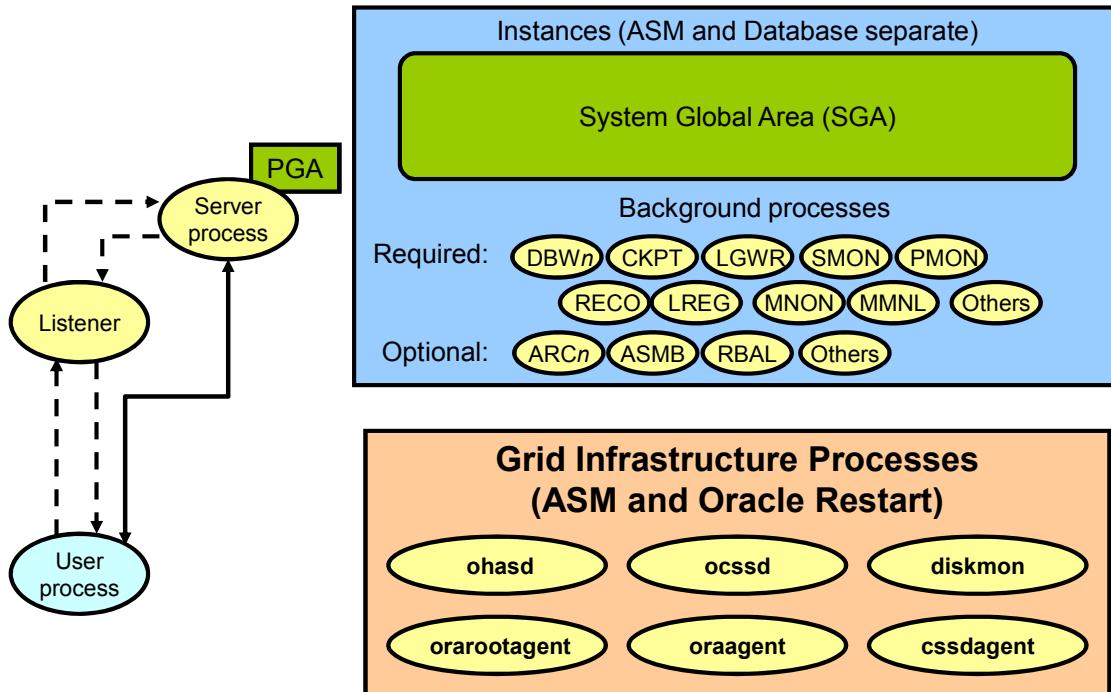
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The processes in an Oracle Database system can be divided into three major groups:

- User processes that run the application or Oracle tool code
- Oracle Database processes that run the Oracle database server code (including server processes and background processes)
- Oracle daemons and application processes that are not specific to a single database

When a user runs an application program or an Oracle tool such as SQL*Plus, the term *user process* is used to refer to the user's application. The user process may or may not be on the database server machine. Oracle Database also creates a *server process* to execute the commands issued by the user process. In addition, the Oracle server also has a set of *background processes* for an instance that interact with each other and with the operating system to manage the memory structures, asynchronously perform I/O to write data to disk, and perform other required tasks. The process structure varies for different Oracle Database configurations, depending on the operating system and the choice of Oracle Database options.

Process Structures



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Server Processes

Oracle Database creates server processes to handle the requests of user processes connected to the instance. The user process represents the application or tool that connects to the Oracle database. It may be on the same machine as the Oracle database or it may exist on a remote client and use a network to reach the Oracle database. The user process first communicates with a listener process that creates a server process in a dedicated environment.

Server processes created on behalf of each user's application can perform one or more of the following:

- Parse and run SQL statements issued through the application.
- Read necessary data blocks from data files on disk into the shared database buffers of the SGA (if the blocks are not already present in the SGA).
- Return results in such a way that the application can process the information.

Background Processes

To maximize performance and accommodate many users, a multiprocess Oracle Database system uses some additional Oracle Database processes called *background processes*. An Oracle Database instance can have many background processes.

The background processes commonly seen in non-RAC, non-ASM environments can include the following:

- Database writer process (DBW n)
- Log writer process (LGWR)
- Checkpoint process (CKPT)
- System monitor process (SMON)
- Process monitor process (PMON)
- Recoverer process (RECO)
- Listener registration process (LREG)
- Manageability monitor process (MMON)
- Manageability monitor lite process (MMNL)
- Job queue coordinator (CJQ0)
- Job slave processes (Jnnn)
- Archiver processes (ARC n)
- Queue monitor processes (QM Nn)

Other background processes may be found in more advanced configurations such as RAC. See the V\$PROCESS and V\$BGPROCESS views for more information on the background processes.

Some background processes are created automatically when an instance is started, whereas others are started as required.

Other process structures are not specific to a single database, but rather can be shared among many databases on the same server. The Grid Infrastructure and networking processes fall into this category.

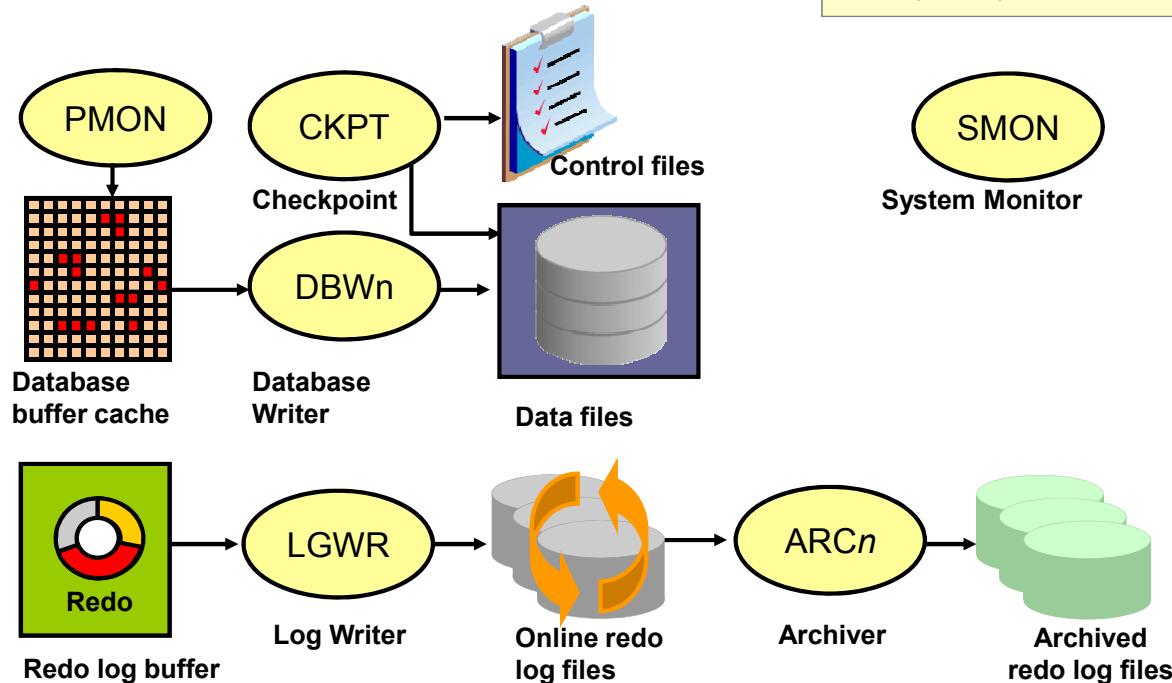
Oracle Grid Infrastructure processes on Linux and UNIX systems include the following:

- ohasd: Oracle High Availability Service daemon that is responsible to starting Oracle Clusterware processes
- ocssd: Cluster Synchronization Service daemon
- diskmon: Disk Monitor daemon that is responsible for input and output fencing for HP Oracle Exadata Storage Server
- cssdagent: Starts, stops, and checks the status of the CSS daemon, ocssd
- oraagent: Extends clusterware to support Oracle-specific requirements and complex resources
- orarootagent: A specialized Oracle agent process that helps manage resources owned by root, such as the network

Note: For a more detailed list of the background processes, view the *Oracle Database Reference* guide.

Reviewing Processes

From prerequisite course



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The topics on the following pages about Oracle background processes are important for an understanding of the appropriate backup and recovery strategies. They are from a prerequisite course and included here for the convenience of those participants who want to review them. Unless specifically requested, it is not necessary to repeat the details in class.

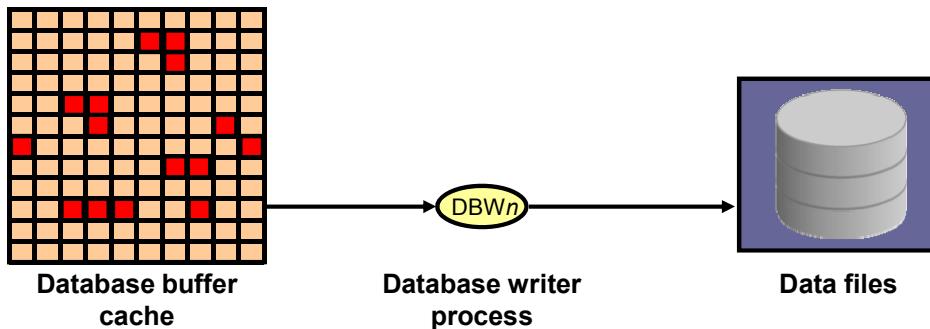
Feel free to continue on the “Adding Process Names” slide to confirm your understanding of the following topics:

- Database Writer Process (DBWn)
- Log Writer Process (LGWR)
- Checkpoint Process (CKPT)
- System Monitor Process (SMON)
- Process Monitor Process (PMON)
- Archiver Processes (ARCn)

Reviewing Database Writer Process (DBWn)

Writes modified (dirty) buffers in the database buffer cache to disk:

- Asynchronously while performing other processing
- To advance the checkpoint



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

When a buffer in the database buffer cache is modified, it is marked dirty and is added to the head of the checkpoint queue that is kept in system change number (SCN) order. This order therefore matches the order of redo that is written to the redo logs for these changed buffers.

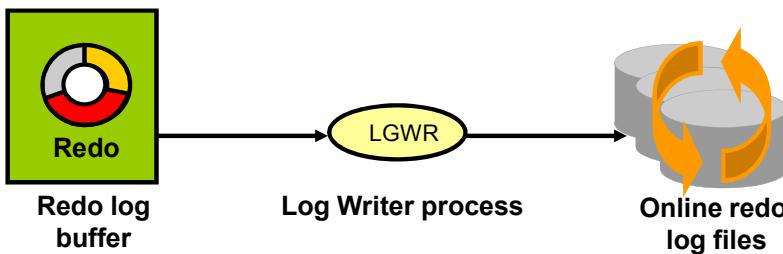
When the number of available buffers in the buffer cache falls below an internal threshold (to the extent that server processes find it difficult to obtain available buffers), DBWn writes non frequently used, modified (dirty) buffers to the data files from the tail of the LRU list so that processes can replace buffers when they need them. DBWn also writes from the tail of the checkpoint queue to keep the checkpoint advancing.

The SGA contains a memory structure that has the redo byte address (RBA) of the position in the redo stream where recovery should begin in the case of an instance failure. This structure acts as a pointer into the redo and is written to the control file by the CKPT process once every three seconds. Because the DBWn writes dirty buffers in SCN order, and because the redo is in SCN order, every time DBWn writes dirty buffers from the LRU list, it also advances the pointer held in the SGA memory structure so that instance recovery (if required) begins reading the redo from approximately the correct location and avoids unnecessary I/O. This is known as incremental checkpointing.

In all cases, DBWn performs batched (multiblock) writes to improve efficiency. The number of blocks written in a multiblock write varies by operating system.

Reviewing Log Writer Process (LGWR)

- Writes the redo log buffer to a redo log file on disk
- Writes:
 - When a user process commits a transaction
 - When the redo log buffer is one-third full
 - Before a DBW n process writes modified buffers to disk
 - Every three seconds



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

LGWR is responsible for redo log buffer management by writing the redo log buffer entries to a redo log file on disk. LGWR writes all redo entries that have been copied into the buffer since the last time it wrote.

The redo log buffer is a circular buffer. When LGWR writes redo entries from the redo log buffer to a redo log file, server processes can then copy new entries over the entries in the redo log buffer that have been written to disk. LGWR normally writes fast enough to ensure that space is always available in the buffer for new entries, even when access to the redo log is heavy. LGWR writes one contiguous portion of the buffer to disk.

LGWR writes:

- When a user process commits a transaction
- When the redo log buffer is one-third full
- Before a DBW n process writes modified buffers to disk (if necessary)
- Every three seconds

Before DBW n can write a modified buffer, all redo records that are associated with the changes to the buffer must be written to disk (the write-ahead protocol). If DBW n finds that some redo records have not been written, it signals LGWR to write the redo records to disk and waits for LGWR to complete writing the redo log buffer before it can write out the data buffers. LGWR writes to the current log group. If one of the files in the group is damaged or unavailable, LGWR continues writing to other files in the group and logs an error in the LGWR trace file and in the system alert log. If all files in a group are damaged, or if the group is unavailable because it has not been archived, LGWR cannot continue to function.

When a user issues a `COMMIT` statement, LGWR puts a commit record in the redo log buffer and writes it to disk immediately, along with the transaction's redo entries. The corresponding changes to data blocks are deferred until it is more efficient to write them. This is called a *fast commit mechanism*. The atomic write of the redo entry containing the transaction's commit record is the single event that determines whether the transaction has committed. Oracle Database returns a success code to the committing transaction, although the data buffers have not yet been written to disk.

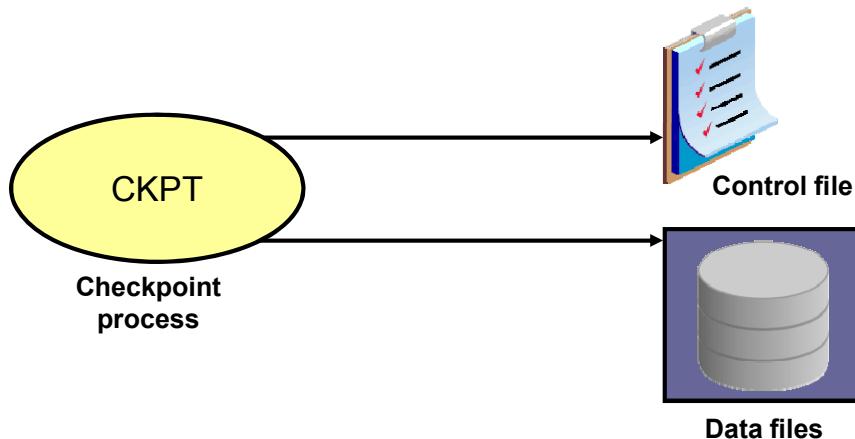
If more buffer space is needed, LGWR sometimes writes redo log entries before a transaction is committed. These entries become permanent only if the transaction is later committed. When a user commits a transaction, the transaction is assigned a system change number (SCN), which Oracle Database records along with the transaction's redo entries in the redo log. SCNs are recorded in the redo log so that recovery operations can be synchronized in Real Application Clusters and distributed databases.

In times of high activity, LGWR can write to the redo log file by using group commits. For example, suppose that a user commits a transaction. LGWR must write the transaction's redo entries to disk. As this happens, other users issue `COMMIT` statements. However, LGWR cannot write to the redo log file to commit these transactions until it has completed its previous write operation. After the first transaction's entries are written to the redo log file, the entire list of redo entries of waiting transactions (not yet committed) can be written to disk in one operation, requiring less I/O than do transaction entries handled individually. Therefore, Oracle Database minimizes disk I/O and maximizes performance of LGWR. If requests to commit continue at a high rate, every write (by LGWR) from the redo log buffer can contain multiple commit records.

Reviewing Checkpoint Process (CKPT)

Checkpoint: At which SCN does instance recovery need to begin?

- Full: Recorded in control file and data header
- Incremental: Recorded in control file



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

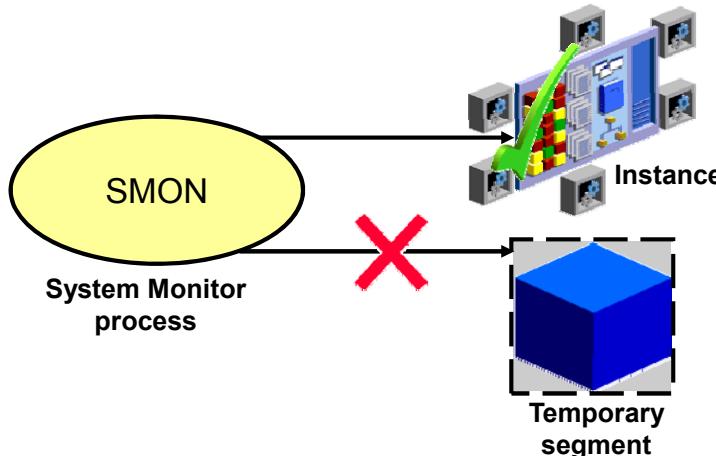
A *checkpoint* is a concept and mechanism. There are different types of checkpoints. The most important ones related to this course are the full checkpoint and the incremental checkpoint. The checkpoint position defines at which system change number (SCN) in the redo thread instance recovery would need to begin.

The SCN at which a full checkpoint occurred is stored in both the data file headers and the control file. The SCN at which the last incremental checkpoint occurred is only stored in the control file (in a structure known as the checkpoint progress record).

The CKPT process updates the control files and the headers of all data files to record the details of the checkpoint (as shown in the graphic). The CKPT process does not write blocks to disk; DBW n always performs that work. The SCNs recorded in the file headers guarantee that all changes made to database blocks prior to that SCN have been written to disk.

Reviewing System Monitor Process (SMON)

- Performs recovery at instance startup
- Cleans up unused temporary segments



ORACLE®

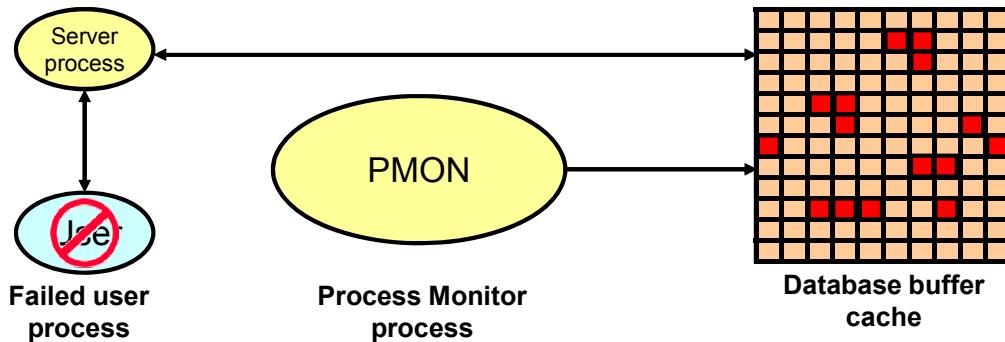
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The System Monitor process (SMON) performs recovery at instance startup if necessary. SMON is also responsible for cleaning up temporary segments that are no longer in use. If any terminated transactions were skipped during instance recovery because of file-read or offline errors, SMON recovers them when the tablespace or file is brought back online.

SMON checks regularly to see whether the process is needed. Other processes can call SMON if they detect a need for it.

Reviewing Process Monitor (PMON)

- Performs process recovery when a user process fails
 - Cleans up the database buffer cache
 - Frees resources that are used by the user process
- Monitors sessions for idle session timeout



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

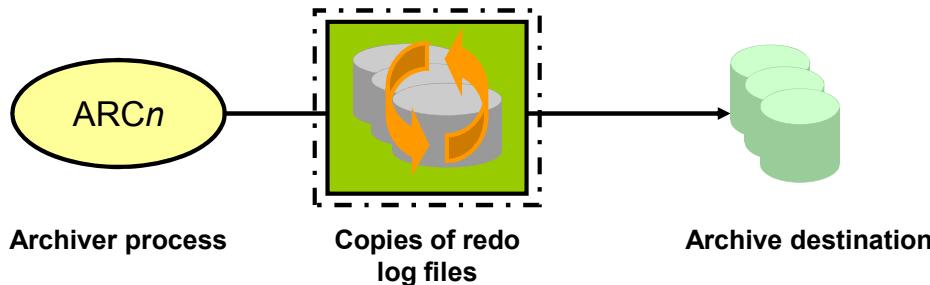
The Process Monitor process (PMON) performs process recovery when a user process fails. PMON is responsible for cleaning up the database buffer cache and freeing resources that the user process was using. For example, it resets the status of the active transaction table, releases locks, and removes the process ID from the list of active processes.

PMON periodically checks the status of dispatcher and server processes, and restarts any that have stopped running (but not any that Oracle Database has terminated intentionally).

Like SMON, PMON checks regularly to see whether it is needed. It can be called if another process detects the need for it.

Reviewing Archiver Processes (ARCn)

- Copy redo log files to a designated storage device after a log switch has occurred
- Can collect transaction redo data and transmit that data to standby destinations



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

ARCn processes are present only when the database is in ARCHIVELOG mode.

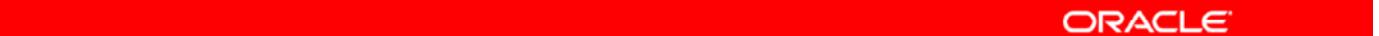
The archiver processes (ARCn) copy redo log files to a designated storage device after a log switch has occurred. This provides more opportunities for recovery, because you can save, back up, and restore all of the archive redo logs ever generated.

If you anticipate a heavy workload for archiving (such as during bulk loading of data), you can increase the maximum number of archiver processes. There can also be multiple archive log destinations. It is recommended that there be at least one archiver process for each destination. The default is to have four archiver processes.

Because the online redo log files are reused in a circular fashion, there is a protocol for controlling when one is allowed to be reused. In ARCHIVELOG mode, the database only begins writing to an online redo log file if it has been archived. This ensures that every redo log file has a chance to be archived.

Adding Process Names

1. The _____ process writes the dirty buffers to the data files.
 2. The _____ process writes the redo entries to the online redo log files.
 3. The _____ process writes checkpoint information in the control file and each data file header.
 4. The _____ process performs recovery on instance startup.
 5. The _____ processes copy redo log files to a designated storage device.
 6. The _____ process performs process recovery when a user process fails.
- A. Checkpoint process (CKPT)
 - B. System monitor process (SMON)
 - C. Log writer process (LGWR)
 - D. Archiver processes (ARC n)
 - E. Database writer process (DBW n)
 - F. Process monitor process (PMON)

The red bar spans the width of the slide content area.

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Adding Process Names

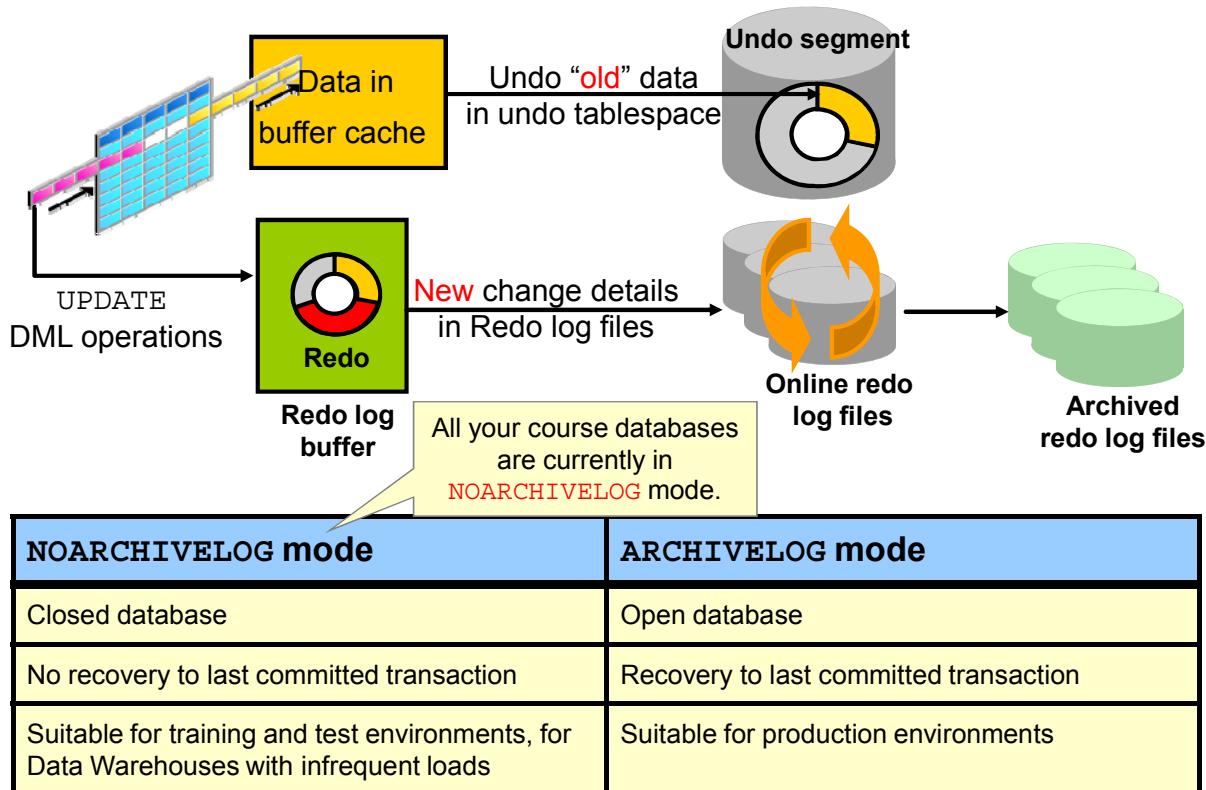
1. The **DBW n** process writes the dirty buffers to the data files.
 2. The **LGWR** process writes the redo entries to the online redo log files.
 3. The **CKPT** process writes checkpoint information in the control file and each data file header.
 4. The **SMON** process performs recovery on instance startup.
 5. The **ARC n** processes copy redo log files to a designated storage device.
 6. The **PMON** process performs process recovery when a user process fails.
- A. Checkpoint process (CKPT)
 - B. System monitor process (SMON)
 - C. Log writer process (LGWR)
 - D. Archiver processes (ARC n)
 - E. Database writer process (DBW n)
 - F. Process monitor process (PMON)



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Solution: 1E, 2C, 3A, 4B, 5D, and 6F

Database Log Mode



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

As modifications are made to data in the database, the “old” data is stored in the undo tablespace and the new change details in online redo log files (as shown in the graphic). The undo is also in the redo stream. Contents of the online redo log include uncommitted transactions, and schema and object management statements. The database maintains online redo log files to protect against data loss. Specifically, after an instance failure the online redo log files enable Oracle Database to recover committed data that it has not yet written to the data files.

Because the online redo log files are reused in a circular fashion, there is a protocol for controlling when one is allowed to be reused. In ARCHIVELOG mode, the database only begins writing to an online redo log file if it has been archived. This ensures that every redo log file has a chance to be archived.

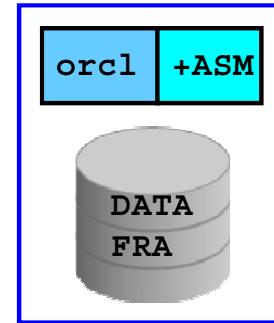
It is possible to perform any type of backup (full or incremental) of a database that is in NOARCHIVELOG mode—if, of course, the database is not open. When the database is in NOARCHIVELOG mode, you must restore all data files before executing a recovery operation. Note also that recovery is limited to the time of the last backup. The database can be recovered to the last committed transaction only when the database is in ARCHIVELOG mode.

Use cases for both log modes are included in the table above.

ORCL Database in ASM

Grid Infrastructure for a stand-alone server:

- Is installed from the clusterware media, separate from the Oracle database software
- Contains Oracle Automatic Storage Management (ASM)
- Contains Oracle Restart—a high availability solution for nonclustered databases
 - Can monitor and restart the following components:
 - Database instances
 - Oracle Net listener
 - Database services
 - ASM instance
 - ASM disk groups (DATA and FRA)
 - Oracle Notification Services (ONS/eONS) for Data Guard



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Grid Infrastructure for a stand-alone server is installed from the clusterware media, separate from the Oracle database software. It includes Oracle Automatic Storage Management and Oracle Restart. If the Oracle Grid Infrastructure is installed before the Oracle database software, databases will be created and automatically configured with Oracle Restart.

Oracle Restart is designed to improve the availability of your Oracle Database. It implements a high availability solution for single instance (nonclustered) environments only. For Oracle Real Application Cluster (Oracle RAC) environments, the functionality to automatically restart components is provided by Oracle Clusterware. Oracle Restart can monitor the health and automatically restart the following components:

- Database Instances (for example, ORCL, as shown in the slide)
- Oracle Net Listener
- Database Services
- ASM Instance (for example, +ASM)
- ASM Disk Groups (for example, DATA and FRA)
- Oracle Notification Services (ONS/eONS) for Data Guard (not included in this course)

Oracle Restart ensures that the components are started in the proper order, in accordance with component dependencies. If a component must be shut down, it ensures that the dependent components are cleanly shut down first.

Some glossary definitions (for your ease of reference):

- A **database instance** is the combination of the system global area (SGA) and background processes. An instance is associated with one and only one database. In an Oracle Real Application Clusters configuration, multiple instances access a single database simultaneously.
- An **Oracle Net listener** is a process that listens for incoming client connection requests and manages network traffic to the database.
- A **database service** is a user-created service that is managed by Oracle Clusterware. A database service may be offered on one or more RAC instances, and managed on per-instance basis (with respect to starting and stopping the service). Only services that are managed by Oracle Clusterware are able to be part of a Performance Class. Services created with the DBMS_SERVICE package are not managed by Oracle Clusterware.
- An **ASM instance** is built on the same technology as an Oracle Database instance. An ASM instance has a System Global Area (SGA) and background processes that are similar to those of Oracle Database. However, because ASM performs fewer tasks than a database, an ASM SGA is much smaller than a database SGA. ASM instances mount disk groups to make ASM files available to database instances. ASM instances do not mount databases.
- An **ASM disk groups** consists of one or more ASM disks, which are managed as a logical unit. I/O to a disk group is automatically spread across all the disks in the group.
- An **Oracle Notification Services** (ONS) is a publish-and-subscribe service for communicating information about all Fast Application Notification (FAN) events.

Facilitating Database Management with Oracle Restart

- Restarting Oracle components when the host computer restarts or after hardware or software failure
- Monitoring components and restarting them, if needed
- For single-instance environments
- Considering component dependencies:
 - Mounting disk groups and starting the ASM instance before starting the database instance
 - Soft dependency between the database instance and the listener
- Starting Oracle Restart with the `crsctl` utility
- Managing Oracle Restart components with the `srvctl` utility

```
$ srvctl stop database -d orcl -o abort
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

- With Oracle Restart, the various Oracle components are automatically restarted after a hardware or software failure or whenever your database host computer restarts.
- Oracle Restart performs periodic checks to monitor the health of these components. If a check operation fails for a component, the component is shut down and restarted.
- Oracle Restart is used in single-instance (nonclustered) environments only. For Oracle Real Application Clusters (Oracle RAC) environments, the functionality to automatically restart components is provided by Oracle Clusterware.
- Oracle Restart ensures that Oracle components are started in the proper order, considering component dependencies. For example, if database files are stored in ASM disk groups, then before starting the database instance, Oracle Restart ensures that the ASM instance is started and the required disk groups are mounted. Likewise, if a component must be shut down, Oracle Restart ensures that dependent components are cleanly shut down first.
- Oracle Restart also manages the soft dependency between database instances and the Oracle Net listener (the listener). When a database instance is started, Oracle Restart attempts to start the listener. If the listener startup fails, the database is still started. If the listener later fails, Oracle Restart does not shut down and restart any database instances.

- You start Oracle Restart with the Clusterware Control (`crsctl`) utility.
- Oracle Restart includes the Server Control (`srvctl`) utility that you use to start and stop Oracle Restart-managed components.

Note: The `srvctl` utility is located in both the `$ORACLE_HOME/bin` directory for the Grid Infrastructure software and the `$ORACLE_HOME/bin` directory for the Oracle database software. You should use the `srvctl` utility from the Oracle database software when starting the Oracle database. You should use the `srvctl` utility from the Grid Infrastructure software when starting the ASM instance or the listener.

Oracle DBA Tools

For backup- and recovery-related tasks:

- RMAN client
- SQL*Plus
- SQL Developer
- `srvctl`: For Oracle Restart components and clustered environments
- `asmcmd`: For all ASM administration tasks
- Enterprise Manager Cloud Control
 - Graphical user interface to RMAN and OSB
 - Wizards employed for many tasks
- `obtool`: Command-line interface to Oracle Secure Backup
- Enterprise Manager Database Express (EM Express)



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

- The RMAN client is a command-line tool with which you can submit RMAN commands to manage your backup environment, create backups, and perform recovery operations.
- SQL*Plus is also a command-line interface for SQL and SQL*Plus commands. It is used for many DBA tasks, including configuring persistent settings for backup and recovery.
- Oracle SQL Developer is a graphical version of SQL*Plus. You can execute SQL statements and scripts; edit and debug PL/SQL code; manipulate and export data. For more details, see the *Oracle SQL Developer User's Guide*.
- The `srvctl` utility facilitates your management of the Oracle Restart components and clustered environments. For more information, see the *Oracle Database Administrator's Guide*.
- ASMCMD is a command-line utility that you can use to manage Oracle ASM instances; disk groups; file access control for disk groups, files, and directories within disk groups; templates for disk groups; and volumes. For more information, see the *Oracle Automatic Storage Management Administrator's Guide*.

- Enterprise Manager (EM) provides a graphical interface to the most commonly used RMAN functionality. Wizards are employed for many of the RMAN operations through Enterprise Manager.
 - EM also provides links to the graphical interface of Oracle Secure Backup (OSB).
 - RMAN is integrated with OSB, which provides centralized tape backup management, protecting file system data and Oracle Database files.
- `obtool` is the command-line interface for OSB.
- Oracle Enterprise Manager Database Express (EM Express) is a web management product that consists of basic administration pages, such as the Database Home page. It supports configuration, storage, and performance-related tasks, but not backup and recovery operations. For more information, see *Oracle Database 2 Day DBA*.

Note: One of the tools will be used during the practices, but others may perform the same task.

Separation of DBA Duties

The **SYSBACKUP** administrative privilege:

- Includes permissions for backup and recovery (connecting to a closed database)
- Does not include data access privileges such as `SELECT ANY TABLE`
- Is granted to the **SYSBACKUP** user that is created during database installation
- Can be explicitly used in RMAN connections by a **SYSBACKUP** privileged user

```
$ rman target ''/ as sysbackup''  
connected to target database: ORCL (DBID=1297344416)
```

Note: Avoid the use of the **SYSDBA** privilege unless it is necessary.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Oracle Database 12c (and later versions) provide support for separation of database administration (DBA) duties for the Oracle database with task-specific and least-privileged administrative privileges that do not require the **SYSDBA** administrative privilege. The privilege to connect and execute commands in Recovery Manager (RMAN) is the **SYSBACKUP** privilege.

- Explicitly connect with the **SYSDBA** role:

```
rman target ''/ as sysbackup''
```

Note the single quotation marks within the double quotation marks.

- For backward compatibility, `rman target /` connects as **SYSDBA**.

Connecting to RMAN and a Target Database

```
. oraenv
orcl
$ rman target ''/ as sysbackup''

RMAN> BACKUP DATABASE;
Starting backup at 10-OCT-12
.
.
RMAN> LIST BACKUP;
BS Key  Type LV Size    Device Type Elapsed Time Completion Time
-----  --  --  --  -----
1       Full   1.06G   DISK          00:01:49   10-OCT-12
.
.
RMAN> DELETE OBSOLETE;
.
Do you really want to delete the above objects (enter YES or NO)? YES
deleted archived log
.
.
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In your Linux course environment, you have more than one local database. Use `. oraenv` to set your environment variables (as shown in the slide).

Invoke RMAN at the operating system command line and specify the appropriate options. Commonly used options are:

- **target**: The connect-string for the target database
- **catalog**: The connect-string for a recovery catalog
- **nocatalog**: Specifies there is no recovery catalog. This is the default.
- **cmdfile**: The name of an input command file
- **log**: The name of the output message log file

The RMAN invocation `rman target /` connects to the local database as the target. The example in the slide shows the default login with the SYSBACKUP privilege.

At the RMAN prompt, you can submit RMAN commands to manage your backup environment and create backups in many different ways, depending on your needs. Shown in the slide are commands to perform a database backup, to list your existing backups (`LIST BACKUP`) and to delete any obsolete backups (`DELETE OBSOLETE`).

Note: Refer to the *Oracle Database Backup and Recovery User's Guide* for more information about how to invoke RMAN. Refer to the *Oracle Database Backup and Recovery Reference* for the complete list of RMAN commands and their options.

Using SQL in RMAN

From the RMAN command line

- Execute SQL commands and PL/SQL procedures.
- Use the optional `SQL` prefix to avoid ambiguity.
- Use the `DESCRIBE` command to list the columns of a table or view. Syntax:

```
DESCRIBE (CATALOG) (schema.) table (@dblink);
```

```
RMAN> SELECT NAME, DBID, LOG_MODE
2> FROM V$DATABASE;

NAME          DBID LOG_MODE
-----
ORCL          1297344416 NOARCHIVELOG
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

- You can execute SQL commands and PL/SQL procedures from the RMAN command line. In versions earlier than Oracle Database 12.1, this required the `SQL` prefix and quotation marks.
- The `SQL` keyword is optional, but you should use it to eliminate ambiguity, especially for a few commands that exist in both RMAN and SQL and have different uses.
- The RMAN `DESCRIBE` command provides the functionality of the SQL*Plus `DESCRIBE` command. You can use the abbreviated version `DESC` or the spelled-out `DESCRIBE` to list the column definitions of a table or view. To access a table or view in another schema, you must have `SELECT` privileges on the object or connect in the `SYSDBA` mode.

Quick Start: A Problem-Solution Approach

Creating your test scenario:

1. Back up database in NOARCHIVELOG mode.
2. Create a test tablespace, user, and table.
3. Simulate failure.
4. Force instance recovery with SHUTDOWN ABORT.

Which situations can result in this situation?

And what must the database do for transactional consistency?



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

If an instance of an open database fails, either because of a SHUTDOWN ABORT statement or abnormal termination, the following situations can result:

- Data blocks committed by a transaction are not written to the data files and appear only in the online redo log. These changes must be reapplied to the database.
- The data files contain changes that had not been committed when the instance failed. These changes must be rolled back to ensure transactional consistency.

Instance recovery uses only online redo log files and current online data files to synchronize the data files and ensure that they are consistent.

Performing Restore and Recovery of a Database in NOARCHIVELOG Mode

If the database is in NOARCHIVELOG mode and if any data file is lost, perform the following tasks:

1. Shut down the instance if it is not already down.
2. **Restore** the entire database—including all data and control files—from the backup.
3. Open the database.
4. Inform users that they must re-enter all changes that were made since the last backup.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The loss of *any* data file from a database in NOARCHIVELOG mode requires complete restoration of the database, including control files and all data files.

With the database in NOARCHIVELOG mode, recovery is possible only up to the time of the last backup. So users must re-enter all changes made since that backup.

To perform this type of recovery:

1. Shut down the instance if it is not already down.
2. Click Perform Recovery on the Maintenance properties page.
3. Select Whole Database as the type of recovery.

If you have a database in NOARCHIVELOG mode that has an incremental backup strategy, RMAN first restores the most recent level 0 and then RMAN recovery applies the incremental backups.

Quiz

When the database is in NOARCHIVELOG mode, you must restore all data files before executing a recovery operation.

- a. True
- b. False



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Answer: a

Summary

In this lesson, you should have learned how to:

- Describe database architecture as it relates to backup and recovery
- Describe your ORCL instance in NOARCHIVELOG mode
- Distinguish Oracle tools for backup and recovery
- Perform basic backup and recovery (in NOARCHIVELOG mode)



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Practice Overview: Getting Started

These practices cover the following topics:

1. Backing up in NOARCHIVELOG mode
2. Creating a test case
3. Recovering in NOARCHIVELOG mode



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this practice you perform your first backup, then create a test case, simulate failure and recover from it. After you simulate the failure, you *must* complete the recovery practice.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and Error : You are not a Valid Partner use only

Configuring for Recoverability

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Configure and manage RMAN settings
- Configure the fast recovery area
- Configure the control file to ensure appropriate protection
- Configure the redo log files for recoverability
- Configure ARCHIVELOG mode and the archived redo log files for recoverability



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Types of RMAN Commands

RMAN commands are of the following types:

- Stand-alone command:
 - Is executed individually at the RMAN prompt
 - Cannot appear as subcommands within RUN
- Job command:
 - Must be within the braces of a RUN command
 - Is executed as a group

Some commands can be executed as both types.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can issue two basic types of RMAN commands: stand-alone and job commands.

Stand-alone commands are executed at the RMAN prompt and are generally self-contained.

Job commands are usually grouped and executed sequentially inside a command block. If any command within the block fails, RMAN ceases processing; no further commands within the block are executed. The effects of any already executed commands still remain, though; they are not undone in any way.

An example of a command that can be run only as a job command is ALLOCATE CHANNEL. The channel is allocated only for the execution of the job, so it cannot be issued as a stand-alone command. There are some commands that can be issued either at the prompt or within a RUN command block, such as BACKUP DATABASE. If you issue stand-alone commands, RMAN allocates any needed channels by using the automatic channel allocation feature.

You can execute stand-alone and job commands in interactive mode or batch mode.

Job Commands: Example

Job commands appear inside a RUN command block:

```
RMAN> RUN
2> {
3>   ALLOCATE CHANNEL c1 DEVICE TYPE DISK
4>     FORMAT "/disk2/%U";
5>   BACKUP AS BACKUPSET DATABASE;
6>   SQL 'alter system archive log current';
7> }
```

Execution of the entire block starts
when this line is entered.

Deallocated after the
RUN block completes



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Unlike stand-alone commands, job commands must appear within the braces of a RUN command. Commands placed inside a RUN block as shown in the slide are run as a single unit of commands. Any configurations made within the run block apply within the scope of the block and override any previously made settings. The following are examples of job commands, which must appear inside a RUN block:

- ALLOCATE CHANNEL
- SWITCH

RMAN executes the job commands inside a RUN command block sequentially. If any command within the block fails, RMAN ceases processing. No further commands within the block are executed. In effect, the RUN command defines a unit of command execution. When the last command within a RUN block completes, the Oracle database releases any server-side resources such as input/output (I/O) buffers or I/O slave processes allocated within the block.

Note: The SQL command on line 6 is just an example. It is NOT a required command for the backup operation.

Configuring Persistent Settings for RMAN

- RMAN is preset with default configuration settings.
- Use the CONFIGURE command to:
 - Configure automatic channels
 - Specify the backup retention policy
 - Specify the number of backup copies to be created
 - Set the default backup type to BACKUPSET or COPY
 - Limit the size of backup pieces
 - Exempt a tablespace from backup
 - Enable and disable backup optimization
 - Configure automatic backups of control files
 - Define the archive log deletion policy
 - Specify the parallelism for a device
 - Set the encryption and compression parameters to be used for backups



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

To simplify ongoing use of RMAN for backup and recovery, RMAN enables you to set several persistent configuration settings for each target database. These settings control many aspects of RMAN's behavior. You can save persistent configuration information such as channel parameters, parallelism, and the default device type in the RMAN repository. These configuration settings are always stored in the control file and in the recovery catalog database (if it exists).

These settings have default values, which allow you to use RMAN immediately. However, as you develop a more advanced backup and recovery strategy, you may have to change these settings to implement that strategy. You can use the CONFIGURE command to configure persistent settings for RMAN backup, restore, duplication, and maintenance jobs. These settings are in effect for any RMAN session until the configuration is cleared or changed.

Note: The configuration settings can be changed in an RMAN job (or session) just for the duration of the job (or session) with the SET command.

Enterprise Manager Note: The same is true for using RMAN via the Enterprise Manager interface. The backup settings provide the default settings for all backups taken. When creating a backup, some of these settings can be overridden for that specific backup.

Viewing Persistent Settings

To examine the persistent RMAN settings for a database:

- Use the RMAN SHOW ALL command to view all configuration settings
- Query the V\$RMAN_CONFIGURATION view to display configuration settings that have been explicitly set



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can view all RMAN persistent settings by connecting to the target and using the RMAN SHOW ALL command. You can query the V\$RMAN_CONFIGURATION view in the target database to display configuration settings that have been explicitly set.

Managing Persistent Settings

- Use multiple streams of data to and from a device:

```
RMAN> CONFIGURE DEVICE TYPE sbt PARALLELISM 3;
```

- Use the SHOW command to list current settings:

```
RMAN> SHOW CONTROLFILE AUTOBACKUP FORMAT;  
RMAN> SHOW EXCLUDE;  
RMAN> SHOW ALL;
```

- Use the CLEAR option of the CONFIGURE command to reset any persistent setting to its default value:

```
RMAN> CONFIGURE BACKUP OPTIMIZATION CLEAR;  
RMAN> CONFIGURE MAXSETSIZE CLEAR;  
RMAN> CONFIGURE DEFAULT DEVICE TYPE CLEAR;
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Parallelism is the number of streams of data that can be used to read from and write to the device. This effectively causes that number of channels to be allocated when the device is used by RMAN. For example, if a media manager has two tape drives available, parallelism 2 would allow both tape drives to be used simultaneously for BACKUP commands using that media manager. Parallelism for the disk device type is also useful, when you want to spread out a backup over multiple disks.

Specify the parallelism to be used on the device using the PARALLELISM clause, like this:

```
CONFIGURE DEVICE TYPE <device> PARALLELISM <n>
```

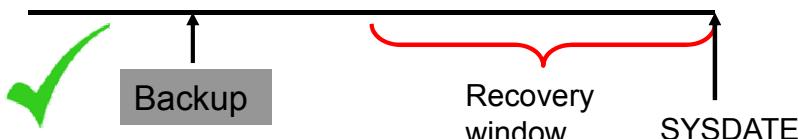
where *<n>* is the parallelism value.

Using the RMAN SHOW command, you can view the RMAN configuration settings. If SHOW ALL is executed when connected to a target database, only node-specific configurations and database configurations are displayed.

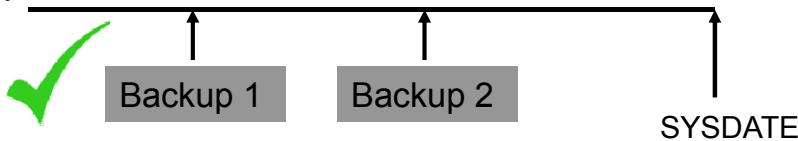
You can return to the default value for any CONFIGURE command by executing the same command with the CLEAR option.

Specifying a Retention Policy

- Retention policy: Describes which backups will be kept and for how long
- Two types of retention policies:
 - **Recovery window:** Establishes a period of time within which point-in-time recovery must be possible



- **Redundancy:** Establishes a fixed number of backups that must be kept



- Retention policies are mutually exclusive.

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

A *retention policy* describes which backups will be kept and for how long. You can set the value of the retention policy by using the RMAN CONFIGURE command or Enterprise Manager.

Recovery Window Retention Policy

The best practice is to establish a period of time during which it will be possible to discover logical errors and fix the affected objects by doing a point-in-time recovery to just before the error occurred. This period of time is called the *recovery window*. This policy is specified in number of days. For each data file, there must always exist at least one backup that satisfies the following condition:

```
SYSDATE - backup_checkpoint_time >= recovery_window
```

You can use the following command syntax to configure a recovery window retention policy:
RMAN> CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF <days> DAYS;
where <days> is the size of the recovery window.

If you are not using a recovery catalog, you should keep the recovery window time period less than or equal to the value of the `CONTROL_FILE_RECORD_KEEP_TIME` parameter to prevent the record of older backups from being overwritten in the control file. If you are using a recovery catalog, then make sure the value of `CONTROL_FILE_RECORD_KEEP_TIME` is greater than the time period between catalog resynchronizations. Resynchronizations happen when you:

- Create a backup. In this case, the synchronization is done implicitly.
- Execute the `RESYNC CATALOG` command.

Recovery catalogs are covered in more detail in the lesson titled “Using the RMAN Recovery Catalog.”

Redundancy Retention Policy

If you require a certain number of backups to be retained, you can set the retention policy on the basis of the redundancy option. This option requires that a specified number of backups be cataloged before any backup is identified as obsolete. The default retention policy has a redundancy of 1, which means that only one backup of a file must exist at any given time. A backup is deemed obsolete when a more recent version of the same file has been backed up.

You can use the following command to reconfigure a redundancy retention policy:

```
RMAN> CONFIGURE RETENTION POLICY TO REDUNDANCY <copies>;
```

where `<copies>` is the number of copies that are required for policy satisfaction.

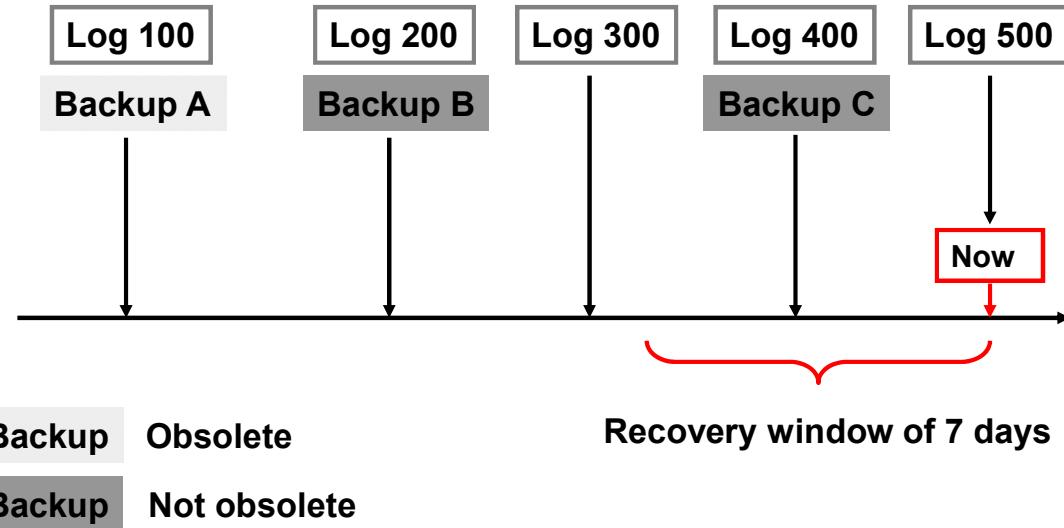
Disabling the Retention Policy

You may want to disable the retention policy totally. If you have a separate system, outside of RMAN, that backs up your disk backups to tape, you may want to do this. If you disable the retention policy, then RMAN never considers a backup obsolete. Because RMAN does not have to decide when to remove a backup from disk (because another utility is managing that), RMAN does not need to be configured for making that decision. In this case, records of each backup are maintained for as long as is specified by the `CONTROL_FILE_RECORD_KEEP_TIME` initialization parameter. Disable the retention policy by using this command:

```
RMAN> CONFIGURE RETENTION POLICY TO NONE;
```

Note: You can specify that a backup is an exception to the retention policy that you have defined by creating an archival backup.

Recovery Window Retention Policy: Example



Backup B and archive logs 201 through 500 are required to satisfy this retention policy.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The retention policy in the slide shows that it requires the ability to recover to any time within the last seven days. Some of the backups and logs are obsolete, because they are not needed to recover to a time within the seven-day window. This retention policy is configured thus:

```
RMAN> CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 7 DAYS;
```

Given the backups and archived log files available, the only data needed to recover to a point inside the recovery window is Backup B and logs 201 through 500. Note that Backup A is not needed because there is a later backup (B) that is still before the recovery window. Also, Backup C is not sufficient as the only backup to retain because it would not satisfy a need to recover to points in time at the beginning of the recovery window. The last backup that was taken before the beginning of the recovery window, including all logs since that backup, is what is necessary.

Quiz

RMAN configuration settings cannot be changed for a specific RMAN session.

- a. True
- b. False

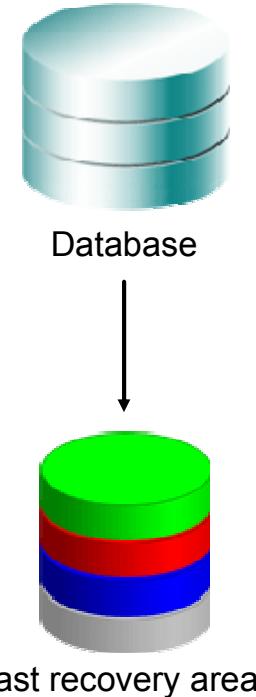


Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Answer: b

Using a Fast Recovery Area

- Permanent items:
 - Multiplexed copies of the current control file
 - Multiplexed copies of online redo logs
- Transient items:
 - Archived redo logs
 - Data file copies
 - Control file copies
 - Control file autobackups
 - Backup pieces
 - Flashback logs



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

A key component of the Oracle disk backup strategy is the fast recovery area (FRA), a storage location on a file system or Automatic Storage Management (ASM) disk group that organizes all recovery-related files and activities for an Oracle database. All files that are required to fully recover a database from media failure can reside in the fast recovery area, including control files, archived logs, data file copies, and RMAN backups.

What differentiates the FRA from simply keeping your backups on disk is the FRA's proactive space management. In addition to a location, the FRA is also assigned a quota, which represents the maximum amount of disk space that it can use at any time. For example, when new backups are created in the FRA and there is insufficient space to hold them, backups and archived logs that are not needed to satisfy RMAN configuration settings (such as retention policy and archive log deletion policy) are automatically deleted to free space. The alert log contains information indicating when disk space consumption is nearing its quota and there are no additional files that can be deleted. You can then take action to correct the situation by adding more disk space, backing up files to tape, or changing the retention policy.

The recovery-related files are of two types: permanent and transient. Permanent files are actively being used by the instance. Transient files are needed only in the event of some type of recovery operation.

Permanent Items

- **Control file:** Depending on the setting of several initialization parameters, a copy of the control file is created in the fast recovery area location when you create a new database or control file. For details, see the “Semantics” section of the CREATE CONTROLFILE command in the *Oracle Database SQL Language Reference*.
- **Multiplexed copies of online redo log files:** A mirrored copy from each redo log group can be here. When you create a database, you can specify the location of the online redo log files by using the LOGFILE clause. If you do not include that clause, the locations are set according to the values of the following initialization parameters:
 - **DB_CREATE_ONLINE_LOG_DEST_n:** If one or more of these variables is set, then these are the only locations used.
 - **DB_CREATE_FILE_DEST:** If this is set, this is the primary file location.
 - **DB_RECOVERY_FILE_DEST:** If this is set, in addition to DB_CREATE_FILE_DEST, this location is used as the mirror.

For more details on how these variables affect the location of the online redo logs, see the LOGFILE clause of the CREATE DATABASE statement in the *Oracle Database SQL Language Reference*.

Transient Items

- **Archived redo log files:** When the fast recovery area is configured, LOG_ARCHIVE_DEST_1 is automatically set to the fast recovery area location. The Archiver background process creates archived redo log files in the fast recovery area and in other configured LOG_ARCHIVE_DEST_n locations. If no LOG_ARCHIVE_DEST_n locations are defined, the default location for archived redo log files is in the fast recovery area.
- **Flashback logs:** Flashback logs are generated when Flashback Database is enabled.
- **Control file autobackups:** The default location for control file autobackups created by RMAN and autobackups generated by the Oracle database server is the fast recovery area.
- **Data file copies:** The BACKUP AS COPY command creates image data file copies in the fast recovery area.
- **RMAN files:** The fast recovery area is the default location that is used by RMAN for backups and restoration of the archive log content from tape for a recovery operation.

Note: If you need good performance for your FRA, consider creating it on its own physical disks and controllers.

Configuring the Fast Recovery Area

- Fast recovery area:
 - Strongly recommended for simplified backup storage management
 - Storage space (separate from working database files)
 - Location specified by the DB_RECOVERY_FILE_DEST parameter
 - Size specified by the DB_RECOVERY_FILE_DEST_SIZE parameter
One FRA can be used by multiple databases.
 - Large enough for backups, archived logs, flashback logs, multiplexed control files, and multiplexed redo logs
 - Automatically managed according to your backup retention and archived redo log file deletion policies
- Determine location, size, backup retention, and archived redo log deletion policies to configure.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The fast recovery area is a space that is set aside on the disk to contain archived logs, backups, flashback logs, multiplexed control files, and multiplexed redo logs. A fast recovery area simplifies backup storage management and is strongly recommended. You should place the fast recovery area on storage space that is separate from the location of your database data files and primary online log files and control file.

The amount of disk space to allocate for the fast recovery area depends on the size and activity levels of your database. As a general rule, the larger the fast recovery area, the more useful it is. Ideally, the fast recovery area should be large enough for copies of your data and control files and for flashback, online redo, and archived logs needed to recover the database with the backups kept based on the retention policy. (In short, the fast recovery area should be at least twice the size of the database so that it can hold one backup and several archived logs.)

Space management in the fast recovery area is governed by the backup retention and archived redo log deletion policies. A backup retention policy determines when files are obsolete, which means that they are no longer needed to meet your data recovery objectives. The Oracle database server automatically manages this storage by deleting files that are no longer needed.

The archived redo log deletion policy specifies when archived redo logs are eligible for deletion. This deletion policy applies to all archiving destinations, including the fast recovery area. Archived redo logs can be deleted automatically by the database server or as a result of user-initiated RMAN commands. Automatic deletion is only applicable for archived redo logs in the fast recovery area. The database server retains the archived redo logs in the fast recovery area as long as possible and automatically deletes eligible logs when additional disk space is required.

If the fast recovery area is not large enough to hold the flashback logs and files needed to satisfy the retention policy, flashback logs from the earliest SCNs may be deleted to make room for other files.

If you have a storage design where multiple databases share a FRA, then your only task is to ensure that adequate storage is allocated for the FRA. Each database maintains its own information under separate directories. (There are no additional DBA tasks.)

Sizing the Fast Recovery Area

- Control file backups and archived logs: Estimate size of archived logs generated between successive backups on the busiest days, and multiply total size by 2 to account for activity spikes
- Flashback logs: Redo rate x Flashback retention target time x 2
- Incremental backups: Estimated size
- On-disk image copy: Size of the database minus size of temporary files
- Backup sets: Size of backup minus the number of unused (skipped) blocks



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can use the fast recovery area to hold recovery-related files as previously described. The FRA is defined to a specific location (a file system directory or ASM group) and with an upper-limit space quota.

When there is space pressure in the FRA, “unneeded” files are automatically deleted. “Unneeded files” are files that are:

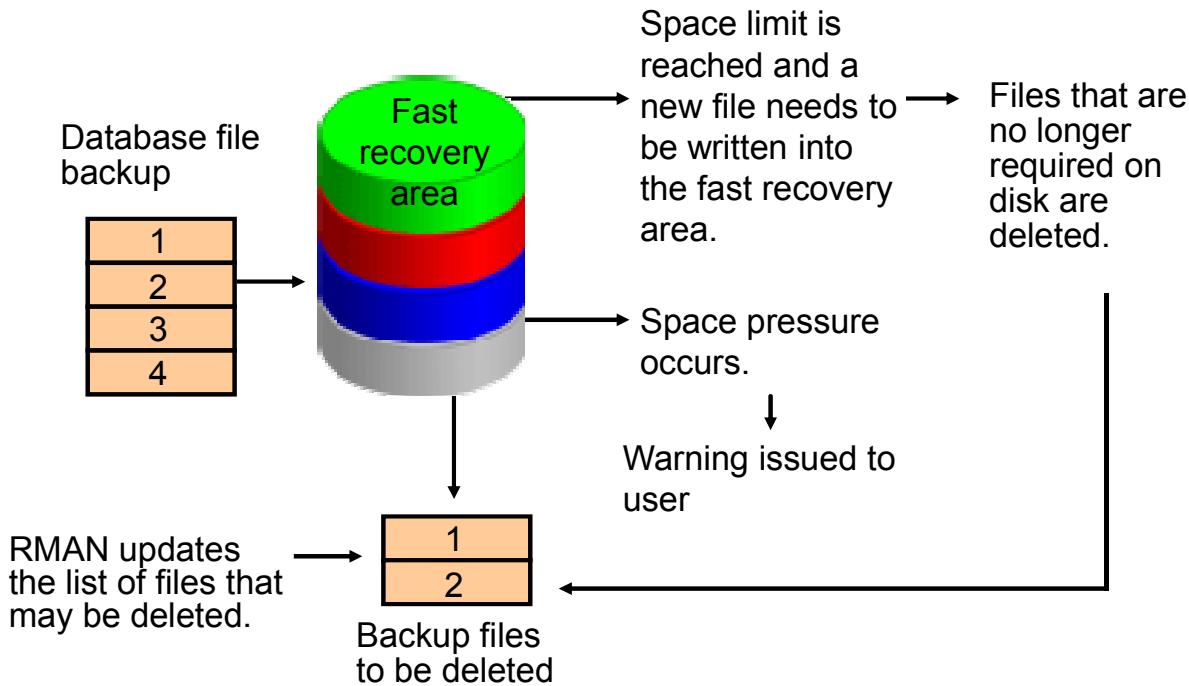
- Backed up to tape via RMAN
- Obsolete according to the RMAN retention policy

To calculate the initial size for the FRA, first determine which recovery-related files you want to keep. The following guidelines will assist you with determining the correct initial size:

- If you are going to keep only control file autobackups and archived logs, you can determine the initial FRA size by finding the total size of archived logs generated between successive backups on the busiest days. You only need enough space in the FRA to hold archived logs between two successive backups. Multiply this estimate by 2 to accommodate unexpected redo spikes. Control file autobackups are generally small.

- If you want to keep archived logs and flashback logs, multiply the archived log size by 2 to get an initial estimate. Flashback logs are generally created in proportion to archived redo logs generated during the same retention period. Multiply the redo rate by the Flashback retention target time by 2.
- The size of incremental backups depends on the amount of changes between backups. You can perform a test run of your incremental strategy to determine representative incremental sizes for a period of time, and then include those sizes in your calculation of the production FRA size.
- If you plan to keep an on-disk image copy backup in the FRA, add the size of the database less the size of temp files. RMAN does not back up temp files. An on-disk image copy backup allows for faster restore compared with restoring from tape. It can also be used as-is in place of the production storage data file.
- For the storage of backup sets, calculate the size of the backup (at a maximum, the size of the database minus the unused or skipped blocks).

Fast Recovery Area Space Management



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Each time RMAN creates a file in the fast recovery area, the list of files that are no longer required on disk is updated. Based on the value of `DB_RECOVERY_FILE_DEST_SIZE`, when the fast recovery area experiences space pressure or is low on free space because there are no files that can be deleted from the fast recovery area, you are warned of the danger of running out of space. The Oracle database server and RMAN continue to create files in the fast recovery area until 100% of the disk limit is reached.

When setting `DB_RECOVERY_FILE_DEST_SIZE`, you must allocate enough space to hold the recovery files, including backups that are waiting to be backed up to tape. Files that are obsolete or have been backed up to tape are likely candidates for deletion to provide free space. When a file is written into the fast recovery area and space is needed for that file, the Oracle database server deletes a file that is on the obsolete files list. When a file is written and deleted from the fast recovery area, notification is written into the alert log.

The backup retention policy and archived redo log deletion policy also affect space management in the fast recovery area.

Note: When the fast recovery area's used space is at 85%, a warning alert is issued, and when used space is at 97%, a critical alert is issued. These are internal settings and cannot be changed.

You can issue the following query to determine the action to take:

```
SQL> SELECT object_type, message_type, message_level,  
2 reason, suggested_action  
3 FROM dba_outstanding_alerts;
```

Your choice is to add additional disk space, back up files to a tertiary device, delete files from the fast recovery area using RMAN, or consider changing the RMAN retention policy.

Multiplexing Control Files

To protect against database failure, your database should have multiple copies of the control file.

	ASM Storage	File System Storage
Best Practice	One copy on each disk group (such as +DATA and +FRA)	At least two copies, each on separate disk (at least one on separate disk controller)
Steps to create additional control files	No additional control file copies required.	<ol style="list-style-type: none">1. Update the SPFILE with the ALTER SYSTEM SET control_files command.2. Shut down the database.3. Copy control file to a new location.4. Open the database and verify the addition of the new control file.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

A control file is a small binary file that describes the structure of the database. It must be available for writing by the Oracle server whenever the database is mounted or opened. Without this file, the database cannot be mounted, and recovery or re-creation of the control file is required. Your database should have a minimum of two control files on different storage devices to minimize the impact of a loss of one control file.

The loss of a single control file causes the instance to fail because all control files must be available at all times. However, recovery can be a simple matter of copying one of the other control files. The loss of all control files is slightly more difficult to recover from but is not usually catastrophic.

Control File Autobackups



```
RMAN> CONFIGURE CONTROLFILE AUTOBACKUP ON;
```

Backup Settings

Device Backup Set Policy

Backup Policy

Automatically backup the control file and server parameter file (SPFILE) with every backup and database structural change

Autobackup Disk Location

An existing directory or diskgroup name where the control file and server parameter file will be backed up. If you do not specify a location, the files will be backed up to the fast recovery area location.

Best Practice Tip: Oracle recommends that you enable control file autobackup.

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can use Oracle Enterprise Manager Cloud Control to specify the backup settings for an instance. From the Database Home page, navigate to Availability > Backup & Recovery > Backup Settings.

To easily recover from the loss of all control file copies, you should configure RMAN to take automatic backups of the control file. The automatic backup of the control file occurs independently of any backup of the current control file explicitly requested as part of your backup command. If you are using RMAN in NOCATALOG mode, it is highly recommended that you activate control file autobackup. Otherwise, if you lose your control file, your database may be unrecoverable.

To configure control file autobackup, modify the backup policy for your database by using Enterprise Manager or the `CONFIGURE CONTROLFILE AUTOBACKUP ON` command.

By default, control file backups are disabled. If you enable control file backups, RMAN automatically backs up the control file and the current server parameter file (if used to start up the database) under the following circumstances:

- At the end of a run script
- When a successful backup is recorded in the RMAN repository
- When a structural change of the database occurs (the Oracle kernel makes the backup)

The control file autobackup file name has a default format of %F for all device types, so that RMAN can infer the file location and restore it without a repository. This variable format translates into C-IIIIIIII-YYYYMMDD-QQ, where:

- IIIIIIII stands for the DBID
- YYYYMMDD is a time stamp of the day the backup is generated
- QQ is the hex sequence that starts with 00 and has a maximum of FF

You can change the default format by using the CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE *type* TO '*string*' command. The value of the string must contain the substitution variable %F and cannot contain other substitution variables. Example:

```
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT  
FOR DEVICE TYPE DISK TO '/u01/oradata/cf_ORCL_auto_%F';
```

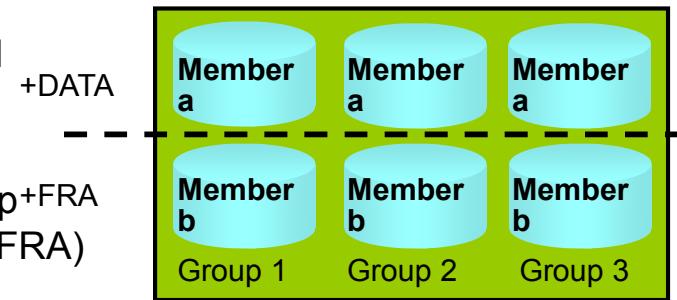
Control file backups are stored in the fast recovery area, unless otherwise specified.

With a control file backup, RMAN can recover the database even if the current control file, recovery catalog, and server parameter file are inaccessible. Because the path used to store the backup follows a well-known format, RMAN can search for and restore the server parameter file or control file from that backup.

Best Practice: Multiplexing Redo Log Files

Multiplex redo log groups to protect against media failure and loss of data. This increases database I/O. It is suggested that redo log groups have:

- At least two members (files) per group
- Each member:
 - On a separate disk and controller if using file system storage
 - In a separate disk group+FRA (such as +DATA and +FRA) if using ASM



Note: Multiplexing redo logs may impact overall database performance.

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Redo log groups are made up of one or more redo log files. Each log file in a group is a duplicate of the others. Oracle Corporation recommends that redo log groups have at least two files per group. If using file system storage, then each member should be distributed on separate disks or controllers so that no single equipment failure destroys an entire log group. If using ASM storage, then each member should be in a separate disk group, such as +DATA and +FRA.

The loss of an entire current log group is one of the most serious media failures because it can result in loss of data. The loss of a single member of a multiple-member log group is trivial and does not affect database operation (other than causing an alert to be published in the alert log).

Remember that multiplexing redo logs may heavily influence database performance because a commit cannot complete until the transaction information has been written to the logs. You should place your redo log files on your fastest disks served by your fastest controllers. If possible, do not place any other database files on the same disks as your redo log files (unless you are using ASM). Because only one group is written to at a given time, there is no performance impact in having members from several groups on the same disk.

Multiplexing the Redo Log

If Storage Type is File System, you are prompted to enter a File Name and File Directory.

Edit Redo Log Group: 1: Add Redo Log Member

Storage Type: Automatic Storage Management
DiskGroup: DATA
Template: <Default>
Alias Directory:
Alias Name:
Reuse File:

Edit Redo Log Group

Actions: Clear logfile
Group # 1
File size 51200 KB
Status ACTIVE
Redo Log Members

Select File Name

Object Name	File Directory
group_1.261.689304441	+DATA/orcl/onlinelog/
group_1.257.689304447	+FRA/orcl/onlinelog/

Add

Redo Log Groups

Search: Enter an object name to filter t
Object Name:
By default, the search returns all uppercase-sensitive match, double quote the
Selection Mode: Single
Edit View Delete Actions Clear logfile Go

Select	Group	Status	# of Members	Archived	Size (KB)	Sequence	First Change#
<input checked="" type="radio"/>	1	Active	2	No	51200	7	834285
<input type="radio"/>	2	Active	2	No	51200	8	849739
<input type="radio"/>	3	Current	2	No	51200	9	849745

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can multiplex your redo log by adding a member to an existing log group. To add a member to a redo log group, perform the following steps for each group you want to multiplex:

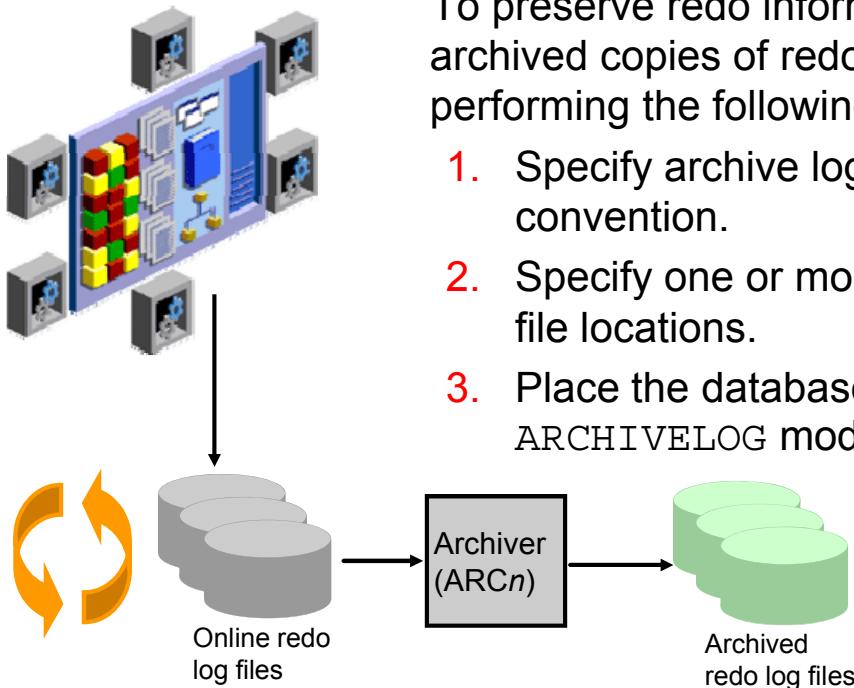
1. Select Enterprise Manager > Server > Redo Log Groups.
2. Select a group and click the Edit button, or click the group number link. The Edit Redo Log Group page appears.
3. In the Redo Log Members region, click Add. The Add Redo Log Member page appears.
4. Choose the appropriate Storage Type, and enter the required information. For ASM, choose the disk group and, if desired, specify template and alias information. For File System storage, enter the file name and the file directory. Click Continue.

An example showing the SQL syntax of adding a redo log member to redo log group 1 (using ASM) is shown here:

```
SQL> ALTER DATABASE ADD LOGFILE MEMBER '+DATA' TO GROUP 1;
```

When you add the redo log member to a group, the member's status is marked as INVALID (as can be seen in the V\$LOGFILE view). This is the expected state because the new member of the group has not yet been written to. When a log switch occurs and the group containing the new member becomes CURRENT, the member's status changes to null.

Creating Archived Redo Log Files



To preserve redo information, create archived copies of redo log files by performing the following steps:

1. Specify archive log file-naming convention.
2. Specify one or more archive log file locations.
3. Place the database in ARCHIVELOG mode.

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The instance treats the online redo log groups as a circular buffer in which to store transaction information, filling one group and then moving on to the next. After all groups have been written to, the instance begins overwriting information in the first log group.

To configure your database for maximum recoverability, you must instruct the database to make a copy of the online redo log group before allowing it to be overwritten. These copies are known as *archived logs*.

To facilitate the creation of archive log files:

1. Specify a naming convention for your archive logs
2. Specify a destination or destinations for storing your archive logs
3. Place the database in ARCHIVELOG mode

Note: Steps 1 and 2 are not necessary if you are using a fast recovery area.

The destination should exist before placing the database in ARCHIVELOG mode. When a directory is specified as a destination, there should be a slash at the end of the directory name.

Configuring ARCHIVELOG Mode

To place the database in ARCHIVELOG mode, perform the following steps:

- Using Enterprise Manager Cloud Control:
 1. Select the “ARCHIVELOG Mode” check box and click Apply. The database can be set to ARCHIVELOG mode only from the MOUNT state.
 2. Restart the database instance by clicking “Yes” when prompted.
- Using SQL commands:
 - Mount the database.
 - Issue the ALTER DATABASE ARCHIVELOG command.
 - Open the database.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Placing the database in ARCHIVELOG mode prevents redo logs from being overwritten until they have been archived.

In Enterprise Manager Cloud Control, navigate to Availability > Backup & Recovery > Recovery Settings and select the ARCHIVELOG Mode check box. The database must be restarted after making this change.

To issue the SQL command to put the database in ARCHIVELOG mode, the database must be in MOUNT mode. If the database is currently open, you must shut it down cleanly (not abort), and then mount it.

If you are using Oracle Restart (part of Oracle Grid Infrastructure), use the Server Control utility to manage your database instance.

With the database in NOARCHIVELOG mode (the default), recovery is possible only until the time of the last backup. All transactions made after that backup are lost.

In ARCHIVELOG mode, recovery is possible until the time of the last commit. Most production databases are operated in ARCHIVELOG mode.

Note: Back up your database after switching to ARCHIVELOG mode because your database is recoverable only from the first backup taken in that mode.

Quiz

Space management in the fast recovery area is governed by which of the following RMAN configuration settings?

- a. RETENTION POLICY
- b. BACKUP OPTIMIZATION
- c. ARCHIVELOG DELETION POLICY
- d. CONTROLFILE AUTOBACKUP



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Answer: a, c

Summary

In this lesson, you should have learned how to:

- Configure the fast recovery area appropriately for your recovery requirements
- Configure the control file to ensure appropriate protection
- Configure the redo log files for recoverability
- Configure ARCHIVELOG mode and the archived redo log files for recoverability



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Practice Overview: Configuring for Recoverability

- Practice 3-1 covers the following topics:
 - Configuring or confirming the default backup destination and size
 - Configuring ARCHIVELOG mode
 - Configuring the archive log file destination
 - Placing the database in ARCHIVELOG mode
- Practice 3-2 covers setting the date and time format for RMAN.
- Practice 3-3 covers the following topics:
 - Automatically backing up the control file and SPFILE
 - Ensuring that one redundant backup is kept



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Note: Completing these practices is a prerequisite for all other ones. If, at any point during the course, you and your instructor decide that it would be better to start with a new database, then you must repeat all practices in this lesson.

Practice Overview: Configuring for Recoverability

- Practice 3-4 covers the following topics:
 - Configuring control files
 - Using DBA tools: EM Express and SQL*Plus
- Practice 3-5 covers recovery settings in Cloud Control.
- Practice 3-6 covers configuring redo log files (in Cloud Control).

Note: Completing these practices is a prerequisite for all following ones.

Optional demo: *Configuring Basic Backup and Recovery Settings for a Database in Cloud Control 12c.*



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Using DBA tools includes:

- Enterprise Manager Database Express 12 (EM Express) to view the existing control files (as SYSDBA).
- SQL*Plus and an editor to update the CONTROL_FILES parameter in the initialization parameter file (as SYSDBA).

Using the RMAN Recovery Catalog



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Describe the use of the RMAN recovery catalog
- Create the RMAN recovery catalog
- Register a database in the RMAN recovery catalog



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

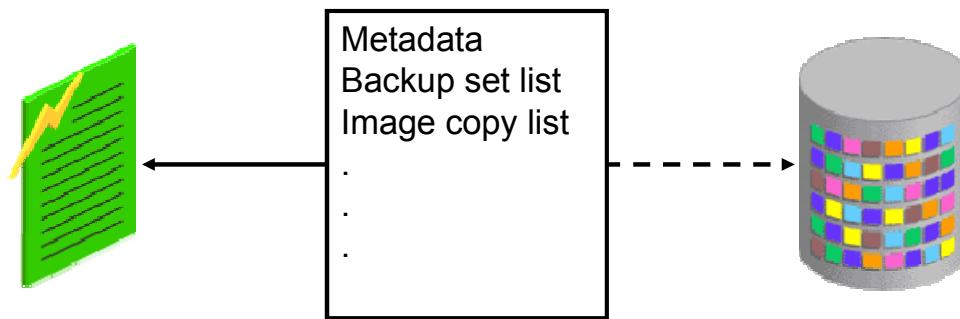
RMAN Repository Data Storage: Comparison of Options

Control file:

- Simpler administration
- Default

Recovery catalog:

- Replicates control file data
- Stores more backup history
- Services many targets
- Stores RMAN scripts
- Provides more protection options for metadata



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

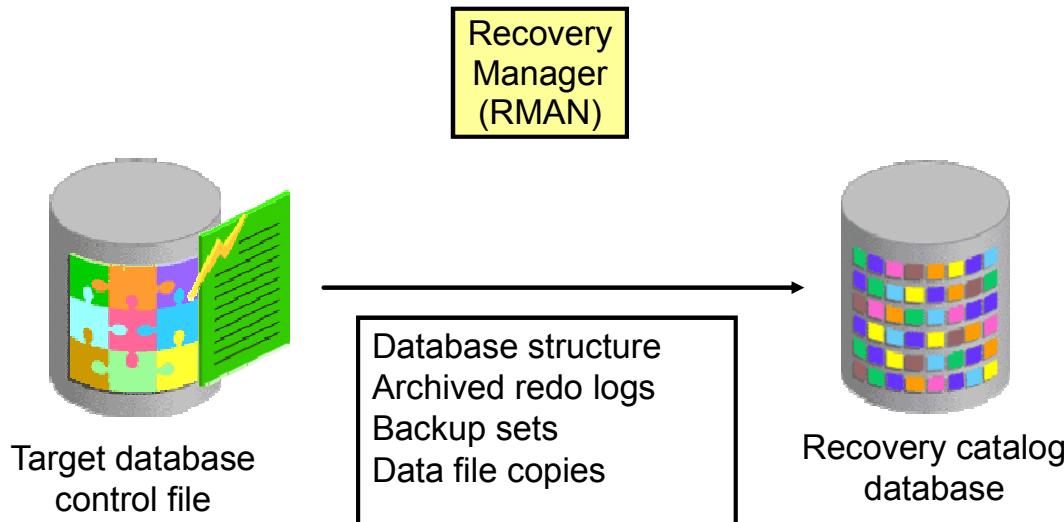
The RMAN repository data is always stored in the control file of the target database. But it can also, additionally, be stored in a separate database in a recovery catalog.

A recovery catalog preserves backup information in a separate database, which is useful in the event of a lost control file. This allows you to store a longer history of backups than what is possible with a control file-based repository. A single recovery catalog is able to store information for multiple target databases. The recovery catalog can also hold RMAN stored scripts, which are sequences of RMAN commands.

If you have very simple backup management requirements, Oracle recommends that you use the control file option rather than a recovery catalog. Having a recovery catalog means you need to manage and back up another database. Therefore, use a recovery catalog only if you can take advantage of the benefits it offers, such as longer backup retention.

A recovery catalog is required when you use RMAN in a Data Guard configuration.

Storing Information in the Recovery Catalog



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

RMAN propagates information about the database structure, archived redo logs, backup sets, and data file copies into the recovery catalog from the target database's control file after any operation that updates the repository, and also before certain operations.

Reasons to Use a Recovery Catalog

- Stores more historical information than the control file
- Enables you to use RMAN stored scripts
- Enables you to create customized reports for all registered targets
- Enables you to use the `KEEP FOREVER` clause of the `BACKUP` command
- Allows you to list the data files and tablespaces that are or were in the target database *at a given time*
- Makes it much easier to restore and recover following the loss of the control file because it preserves RMAN repository metadata



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Although you can use the control file as the sole repository for RMAN, it has finite space for records of backup activities. When you use a recovery catalog, you can store a much longer history of backups. This enables you to perform a recovery that goes back further in time than the history in the control file.

If you want to use RMAN stored scripts, you must use a recovery catalog.

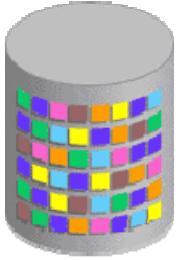
When you use a recovery catalog, the backup and recovery information for all registered targets is contained in one place allowing you to create customized reports by connecting as the recovery catalog owner and querying the various `RC_` views. If you do not use a recovery catalog, you must connect to each target database instance separately and query the `V$` views for the RMAN information in the target control file.

Note: Enterprise Manager Cloud Control also enables you to view backup information for multiple databases without the use of a recovery catalog.

You can use the `BACKUP . . . KEEP` command to create a backup that is retained for a different period of time from that specified by the configured retention policy. The `KEEP FOREVER` clause specifies that the backup or copy never expires and requires the use of a recovery catalog so that the backup records can be maintained indefinitely.

The `REPORT SCHEMA` command lists the tablespaces and data files in the target database. If you add the option of `AT [time|scn|logseq]`, you can see the information at some time in the past. You can use the `AT` option only if you are using a recovery catalog.

Creating the Recovery Catalog: Three Steps



Configure the recovery catalog database.



Create the recovery catalog owner.



Create the recovery catalog.

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

To create a recovery catalog, perform the following three steps:

1. Configure the database in which you want to store the recovery catalog.
2. Create the recovery catalog owner.
3. Create the recovery catalog.

Configuring the Recovery Catalog Database

- Allocate space for the recovery catalog. Consider:
 - Number of databases supported by the recovery catalog
 - Number of archived redo log files and backups recorded
 - Use of RMAN stored scripts
- Create a tablespace for the recovery catalog, which becomes the default tablespace for the recovery catalog owner.

```
SQL> CREATE TABLESPACE rcat_ts DATAFILE <data file name>
      SIZE 15M;
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Determine the database in which you will install the recovery catalog schema. Be sure to consider your backup and recovery procedures for this database.

The amount of space required by the recovery catalog schema depends on the number of databases monitored by the catalog. The space increases as the number of archived redo log files and backups for each database increases. If you use RMAN stored scripts, space must be allocated for those scripts. The sample space requirement is 15 MB for each database registered in the recovery catalog.

Creating the Recovery Catalog Owner

- Create the recovery catalog owner.
- Grant the RECOVERY_CATALOG_OWNER role.



```
SQL> CREATE USER rowner IDENTIFIED BY rpass  
  2  TEMPORARY TABLESPACE temp  
  3  DEFAULT TABLESPACE rcat_ts  
  4  QUOTA UNLIMITED ON rcat_ts;  
SQL> GRANT recovery_catalog_owner TO rowner;
```

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Create a user to serve as the recovery catalog owner. Set the default tablespace for this user to the tablespace you created for the recovery catalog. Be sure to provide UNLIMITED quota on this tablespace for the user. After you have created the user, grant the RECOVERY_CATALOG_OWNER role to the user.

The RECOVERY_CATALOG_OWNER role provides privileges for the owner of the recovery catalog. The role includes the following system privileges: ALTER SESSION, CREATE CLUSTER, CREATE DATABASE LINK, CREATE PROCEDURE, CREATE SEQUENCE, CREATE SESSION, CREATE SYNONYM, CREATE TABLE, CREATE TRIGGER, CREATE TYPE, and CREATE VIEW.

You can use SQL commands or Enterprise Manager to create the user and grant the role.

Creating the Recovery Catalog

- Connect to the recovery catalog database as the catalog owner:

```
$ rman  
RMAN> CONNECT CATALOG username/password@net_service_name
```

- Execute the CREATE CATALOG command:

```
RMAN> CREATE CATALOG;
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

After creating the catalog owner, use the RMAN CREATE CATALOG command to create the catalog tables in the default tablespace of the catalog owner.

Note: As with any database, if the ORACLE_SID environment variable is set to the SID for the recovery catalog database, there is no need to supply the service name in the CONNECT statement.

Managing Target Database Records in the Recovery Catalog

- Registering a target database in the recovery catalog
- Cataloging additional backup files
- Unregistering a target database from the recovery catalog



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Although most information is automatically propagated from the control file to the recovery catalog, there are a few operations you may need to perform to maintain target database records in the recovery catalog.

Registering a Database in the Recovery Catalog

- RMAN does the following when a database is registered:
 - Creates rows in the recovery catalog tables for the target database
 - Copies data from the target database control file to the recovery catalog tables
 - Synchronizes the recovery catalog with the control file
- Using the RMAN command line to register a database:

```
$ rman TARGET / CATALOG  
      username/password@net_service_name  
RMAN> REGISTER DATABASE;
```

- Using Enterprise Manager to register a database:
 1. Navigate to the Recovery Catalog Settings page.
 2. Add the recovery catalog to the configuration if not present.
 3. Specify the target database to use the recovery catalog.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

After creating the recovery catalog, you must register the target databases in the recovery catalog.

To register your target database by using the RMAN command line, perform the following steps:

1. Invoke RMAN and connect to the recovery catalog database and to the target database as shown in the following example:

```
% rman TARGET / CATALOG rman/rman@reccatdb
```
2. Ensure that the target database is mounted or open.
3. Issue the REGISTER command to register the target database in the recovery catalog:

```
RMAN> REGISTER DATABASE;
```

To register a database with a recovery catalog in Enterprise Manager, you must first add the recovery catalog to the Enterprise Manager configuration. Using Enterprise Manager on the target database, you select that recovery catalog to be the recovery catalog for the target database.

If you use RMAN to register the database, and do not perform the Enterprise Manager steps listed in the slide, then any backup and recovery operations performed using Enterprise Manager will not use the recovery catalog. So, if you plan to use Enterprise Manager, perform the registration steps described here even if you have previously executed the RMAN REGISTER DATABASE command.

Perform the following steps in Enterprise Manager Cloud Control to register your database:

1. From the Enterprise Manager Database home page, navigate to Availability > Backup & Recovery > Recovery Catalog Settings.
2. Click Add Recovery Catalog to specify the host, port, and SID of a database with an existing recovery catalog.
3. After you have defined the recovery catalog database, select “Use Recovery Catalog” on the Recovery Catalog Settings page to register the database in the recovery catalog database. When you click OK, the database is registered with the catalog.

Unregistering a Target Database from the Recovery Catalog

- This removes information about the target database from the recovery catalog.
- Use this when you no longer want the target database to be defined in the recovery catalog.

```
$ rman TARGET / CATALOG  
      username/password@net_service_name  
RMAN> UNREGISTER DATABASE;
```



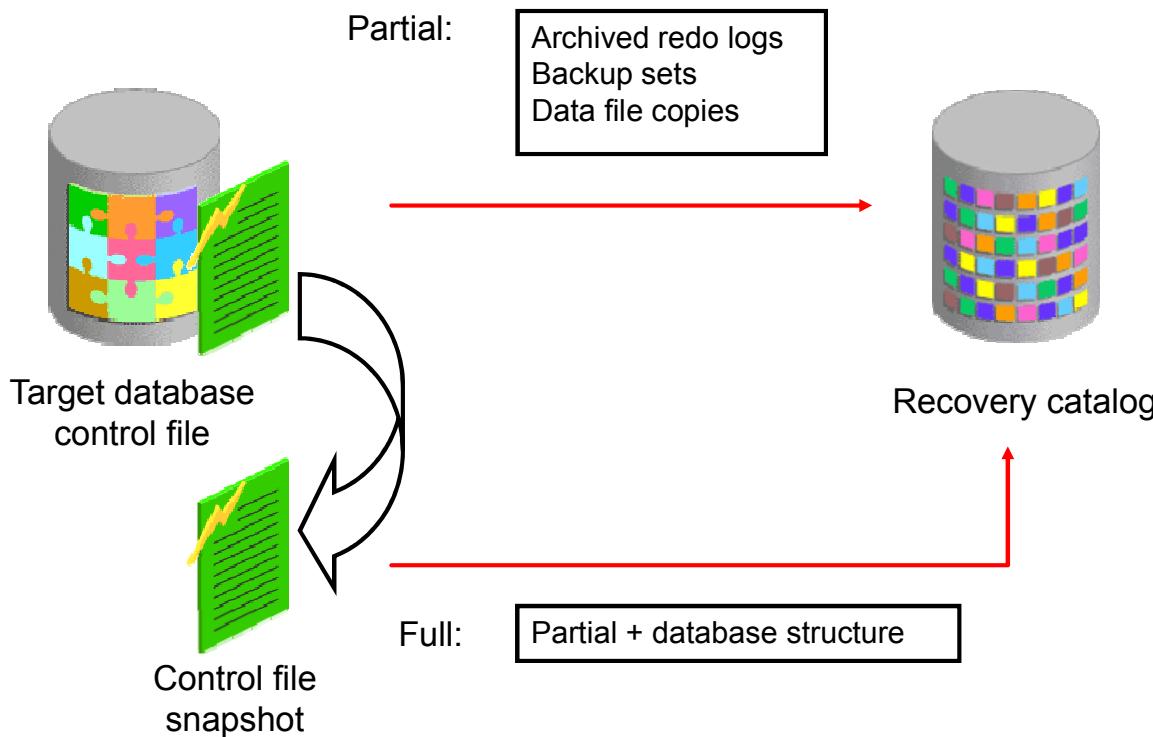
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

When you unregister a database from the recovery catalog, all RMAN repository records in the recovery catalog are lost. You can re-register the database. The recovery catalog records for that database are then based on the contents of the control file at the time of re-registration.

Typically, you would unregister a target database only if you no longer want to use the recovery catalog for that database or the database no longer exists.

Note: If you have used Enterprise Manager Cloud Control to register your database, you must use it again to unregister your database.

Recovery Catalog Resynchronization: Concepts



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

When RMAN performs a *resynchronization*, it compares the recovery catalog to either the current control file of the target database or a backup/standby control file and updates the recovery catalog with information that is missing or changed.

There are two types of resynchronization: partial and full.

- For **partial** resynchronization, RMAN compares the control file to the recovery catalog and updates the recovery catalog with any metadata concerning backups, archived redo logs, data file copies, and so on.
- For a **full** synchronization, RMAN first creates a control file snapshot, which is simply a temporary copy of the control file. It uses the snapshot to make the comparison to the recovery catalog. It compares and updates the same data as a partial resynchronization, but it also includes any database structure changes. For example, database schema changes or new tablespaces are included in a full resynchronization.

You can specify the location for the snapshot control file by using the `SNAPSHOT CONTROLFILE NAME` configuration setting. The default value for the snapshot control file name is platform-specific and depends on the Oracle home of each target database. In an Oracle RAC configuration, the snapshot control file needs to be globally available to all instances in the RAC configuration.

Refer to *Oracle Real Application Clusters Administration and Deployment Guide* for additional information on configuring the snapshot control file location in a RAC configuration.

If the only changes to the control file are those records that are governed by CONTROL_FILE_RECORD_KEEP_TIME, a partial resynchronization is done. Otherwise, a full resynchronization is done. A full resynchronization is also done when you issue the RESYNC CATALOG command.

Manually Resynchronizing the Recovery Catalog

Manually resynchronize the recovery catalog in the following situations:

- After the recovery catalog is unavailable for RMAN to automatically resynchronize it
- When you perform infrequent backups of your target database
- After making changes to the physical structure of the target database

```
RMAN> RESYNC CATALOG;
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Perform a manual resynchronization of the recovery catalog in the following situations:

- If the recovery catalog was unavailable when you issued RMAN commands that cause a partial resynchronization
- If you perform infrequent backups of your target database because the recovery catalog is not updated automatically when a redo log switch occurs or when a redo log is archived
- After making any change to the physical structure of the target database

Note: Refer to the *Backup and Recovery User's Guide* for detailed information about records that are updated during resynchronization.

Using RMAN Stored Scripts

Stored scripts are:

- An alternative to command files
- Available to any RMAN client that can connect to the target database and recovery catalog
- Of two types:
 - Local: Associated with the target database to which RMAN is connected when the script is created
 - Global: Can be executed against any database registered in the recovery catalog
- Created from a text file (additional option)

```
CREATE SCRIPT script_name
{ <RMAN commands> }
```

```
CREATE GLOBAL SCRIPT script_name
{ <RMAN commands> }
```

```
CREATE [GLOBAL] SCRIPT script_name FROM FILE 'file_name';
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can use RMAN stored scripts as an alternative to command files for managing frequently used sequences of RMAN commands. Unlike command files that are available only on the system on which they are stored, a stored script is always available to any RMAN client that can connect to the target database and recovery catalog.

Stored scripts can be defined as global or local. A local stored script is associated with the target database to which RMAN is connected when the script is created, and can be executed only when you are connected to that target database. A global stored script can be executed against any database registered in the recovery catalog, if the RMAN client is connected to the recovery catalog and a target database.

Creating RMAN Stored Scripts

Connect to the desired target database and the recovery catalog and execute the CREATE SCRIPT command to create a stored script.

Executing RMAN Stored Scripts

- Executing a script:

```
RUN { EXECUTE SCRIPT  
script_name  
; }
```

- Executing a global script:

```
RUN { EXECUTE GLOBAL SCRIPT  
script_name  
; }
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Connect to the target database and recovery catalog, and use the `EXECUTE SCRIPT` command to execute a stored script. Note that the `EXECUTE SCRIPT` command requires a `RUN` block. If an RMAN command in the script fails, subsequent RMAN commands in the script do not execute.

When you execute the script, it uses the automatic channels configured at the time. Use `ALLOCATE CHANNEL` commands in the script if you need to override the configured channels as shown in the following example:

```
RMAN> RUN  
{  
  ALLOCATE CHANNEL ch1 DEVICE TYPE DISK;  
  ALLOCATE CHANNEL ch2 DEVICE TYPE DISK;  
  ALLOCATE CHANNEL ch3 DEVICE TYPE DISK;  
  EXECUTE SCRIPT full_backup;  
}
```

Maintaining RMAN Stored Scripts

- Displaying a script:

```
PRINT [GLOBAL] SCRIPT script_name;
```

- Sending the contents of a script to a file:

```
PRINT [GLOBAL] SCRIPT script_name TO FILE 'file_name';
```

- Displaying the names of defined scripts:

```
LIST [GLOBAL] SCRIPT NAMES;
```

- Updating a script:

```
REPLACE [GLOBAL] SCRIPT script_name  
{ <RMAN commands> ; }
```

- Updating a script from a file:

```
REPLACE [GLOBAL] SCRIPT script_name FROM FILE  
'file_name';
```

- Deleting a script:

```
DELETE SCRIPT script_name;
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

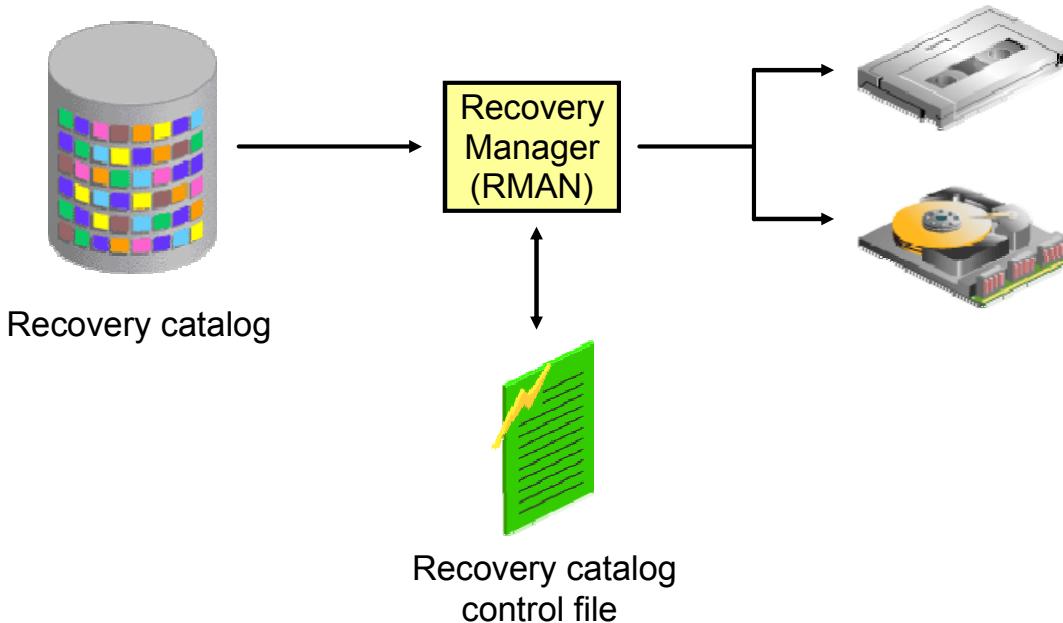
Connect to the target database and recovery catalog and use the `PRINT SCRIPT` command to display a stored script or write it out to a file.

Use the `LIST SCRIPT NAMES` command to display the names of scripts defined in the recovery catalog. This command displays the names of all stored scripts, both global and local, that can be executed for the target database to which you are currently connected.

Connect to the target database and recovery catalog and use the `REPLACE SCRIPT` command to update stored scripts. RMAN creates the script if it does not exist.

To delete a stored script from the recovery catalog, connect to the catalog and a target database, and use the `DELETE SCRIPT` command.

Backing Up the Recovery Catalog



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

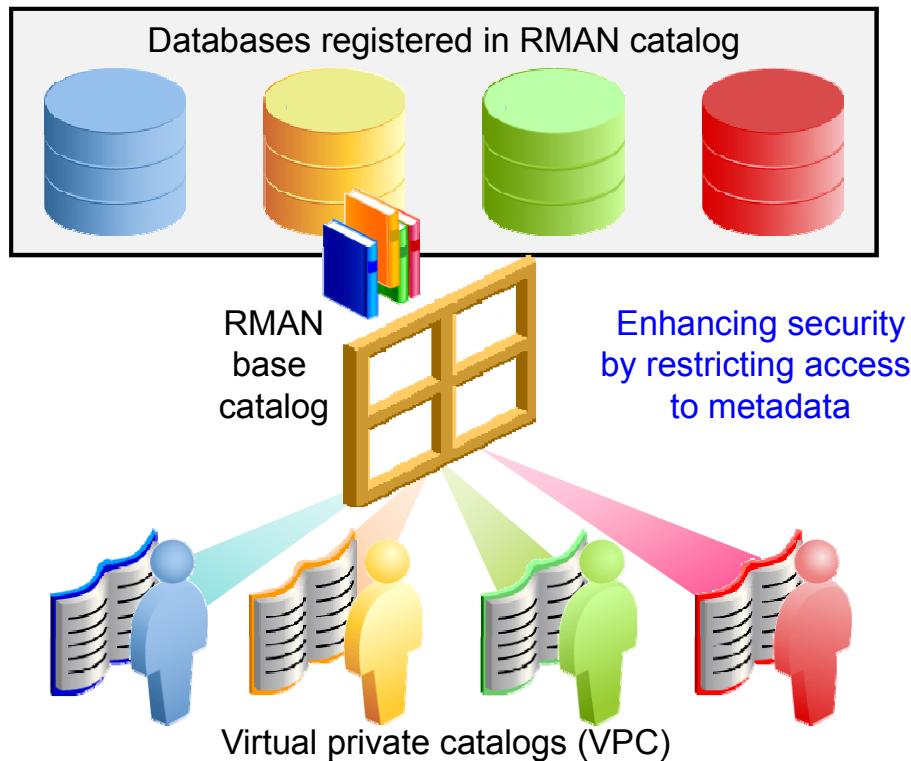
The recovery catalog is a schema in an Oracle database. The recovery catalog database should be backed up. Oracle recommends using RMAN to back it up. Never store a recovery catalog containing the RMAN repository for a database in the same database as the target database or on the same disks as the target database. A recovery catalog is effective only if it is separated from the data that it is designed to protect.

Configure control file autobackup so that the control file is backed up every time a backup is made of the recovery catalog. Any time you make a backup in the target database, back up the recovery catalog right afterward. This protects the record of the latest backup.

The following is a summary of recovery catalog configuration for backup and recovery:

- Configure ARCHIVELOG mode.
- Set the retention policy to a REDUNDANCY value greater than one.
- Back up the recovery catalog to disk and tape.
- To make the backups, use the BACKUP DATABASE PLUS ARCHIVELOG command.
- Use the control file (NOCATALOG), not another recovery catalog, as the RMAN repository.
- Configure control file autobackup to be ON.

Creating and Using Virtual Private Catalogs



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

This feature allows a consolidation of RMAN repositories and maintains a separation of responsibilities, which is a basic security requirement.

The RMAN catalog includes functionality to create virtual private RMAN catalogs (VPC) for groups of databases and users. In Oracle Database 12c Release 1 (12.1.0.2) the RMAN recovery catalog is created and managed by using Oracle Virtual Private Database (VPD), providing better performance and scalability when a large number of virtual private catalogs are created.

The catalog owner can grant access to a registered database and grant the REGISTER privilege to the virtual catalog owner. The virtual catalog owner can then connect to the catalog for a particular target or register a target database. After this configuration, the VPC owner uses the virtual private catalog just like a standard base catalog.

As the catalog owner, you can access all the registered database information in the catalog. You can list all databases registered with the SQL*Plus command:

```
SELECT DISTINCT db_name FROM DBINC;
```

As the virtual catalog owner, you can see only the databases to which you have been granted access.

Note: If a catalog owner has not been granted SYSDBA or SYSOPER on the target database, most RMAN operations cannot be performed.

Creating a Virtual Private Catalog (12.1.0.1)

1. Create the owner of the virtual private catalog (VPC) and grant privileges to the user.
 - A. Using SQL*Plus, connect to the recovery catalog database.
 - B. Create the virtual private catalog owner.
 - C. Grant the RECOVERY_CATALOG_OWNER role to the user.
 - D. Using RMAN, connect to the recovery catalog as the base recovery catalog owner.
 - E. Grant privileges (access to the metadata/ability to register new target databases) to the VPC owner.
2. Create the virtual private catalog.
 - A. Using RMAN, connect to the recovery catalog as the VPC owner.
 - B. Create the virtual private catalog by using the CREATE VIRTUAL CATALOG command.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The procedure to create a virtual private catalog varies depending on whether you are using Oracle Database 12c Release 1 version 12.1.0.1 or 12.1.0.2. The steps outlined on this page apply to version 12.1.0.1. Refer to the *Oracle Database Backup and Recovery User's Guide 12c Release 1 (12.1)* for additional information, including syntax examples.

Managing Virtual Private Catalogs

- If needed, revoke access to the metadata:

```
RMAN> REVOKE CATALOG FOR DATABASE prod1 FROM vpc1;
```

- To drop a virtual private catalog:

```
RMAN> DROP CATALOG;
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

To revoke access for a specific database, connect to the recovery catalog database as the recovery catalog owner:

```
RMAN> REVOKE CATALOG FOR DATABASE prod1 FROM vpc1;
```

To drop a virtual private catalog, connect to the recovery catalog database as the virtual private catalog owner:

```
RMAN> DROP CATALOG;
```

Creating a Virtual Private Catalog (12.1.0.2)

1. Create the owner of the virtual private catalog (VPC) and grant privileges to the user.
 - A. Using SQL*Plus, connect to the recovery catalog database.
 - B. Create the virtual private catalog owner.
 - C. Grant the CREATE SESSION privilege to the user.
 - D. Using RMAN, connect to the recovery catalog as the base recovery catalog owner.
 - E. Grant privileges (access to the metadata/ability to register new target databases) to the VPC owner.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In Oracle Database 12c Release 1 (12.1.0.2) virtual private catalogs are implemented through Oracle Virtual Private Database (VPD). Oracle Virtual Private Database (VPD) provides an ability to create security policies to control database access at the row and column level. Refer to the *Oracle Database Security Guide 12c Release 1 (12.1)* for detailed information about Oracle VPD.

The steps outlined on this page and the next apply to version 12.1.0.2. Refer to the *Oracle Database Backup and Recovery User's Guide 12c Release 1 (12.1)* for additional information, including syntax examples.

Creating a Virtual Private Catalog (12.1.0.2)

2. Register a database with a VPC and store backup metadata.
 - A. Using RMAN, connect to the recovery catalog database as the VPC owner and connect to the database that you want to register as TARGET.
 - B. Register the database with the VPC owner by using the REGISTER DATABASE command.
 - C. Use the BACKUP command to back up the database and store metadata related to the backup in the VPC.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Upgrading Virtual Private Catalogs for 12.1.0.2

1. Using SQL*Plus, connect to the recovery catalog database as the SYS user with the SYSDBA privilege.
2. Grant additional privileges to the RECOVERY_CATALOG_OWNER role by executing the \$ORACLE_HOME/rdbms/admin/dbmsrman.sys.sql script.
3. Using RMAN, connect to the base recovery catalog and upgrade the catalog by using the UPGRADE CATALOG command.
4. Using SQL*Plus, again connect to the recovery catalog database as the SYS user with the SYSDBA privilege.
5. Upgrade the VPC schemas to the VPD model by executing the \$ORACLE_HOME/rdbms/admin/dbmsrmanvpc.sql script.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

RMAN uses Oracle Virtual Private Database (VPD) to implement virtual private catalogs beginning with Oracle Database 12c Release 1 (12.1.0.2). If you created a recovery catalog and virtual private catalogs using an earlier release, you must upgrade to VPD as described in the slide. Sample syntax can be found in the Oracle Database Backup and Recovery User's Guide12c Release 1 (12.1).

Quiz

After a database is registered in a recovery catalog, RMAN no longer stores information in the database's control file.

- a. True
- b. False



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Answer: b

Quiz

Select the steps that are necessary to use a recovery catalog for a specific database.

- a. Create the recovery catalog schema in the specific database.
- b. Register the database in the recovery catalog.
- c. Copy the control file from the database to the recovery catalog database.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Answer: b

Summary

In this lesson, you should have learned how to:

- Describe the use of the RMAN recovery catalog
- Create the RMAN recovery catalog
- Register a database in the RMAN recovery catalog



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Practice Overview: Using the RMAN Catalog

- Practice 4-1 covers the following topics:
 - Creating the recovery catalog owner
 - Granting privileges to the recovery catalog owner
- Practice 4-2 covers creating the RMAN recovery catalog.
- Practice 4-3 covers registering the `orcl` database in the recovery catalog.
- Practice 4-4 covers configuring Enterprise Manager for the RMAN catalog.
- Practice 4-5 covers the following topics:
 - Configuring the recovery catalog for recovery
 - Backing up the RCAT database



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

1. In this practice, you first create a user and grant appropriate privileges.
2. Next, you create the RMAN recovery catalog in a database that has been precreated for you.
3. Then you register the `ORCL` database in the recovery catalog that you just created.
4. Finally, you configure ARCHIVELOG mode, set the retention policy for the recovery catalog, and back up your recovery catalog database by using the backup strategy of incremental backups applied to image copies. This provides a method of fast restore by switching to the image copy rather than copying the backups back to the original location.

Practice Overview: Preparing Your Training Environment

- Practice 4-6 covers preparing your training environment by taking a cold backup of the ORCL database.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You should prepare your training environment by taking a cold backup of the ORCL database that will enable you to restore the database if you are unable to complete the upcoming practices as described.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and Error : You are not a Valid Partner use only

Backup Strategies and Terminology

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Lesson Objectives

After completing this lesson, you should be able to:

- Describe Oracle backup solutions
- Describe RMAN backup types
- Describe, compare, and determine your backup strategy
- Schedule backups according to your strategy
- Perform whole backups

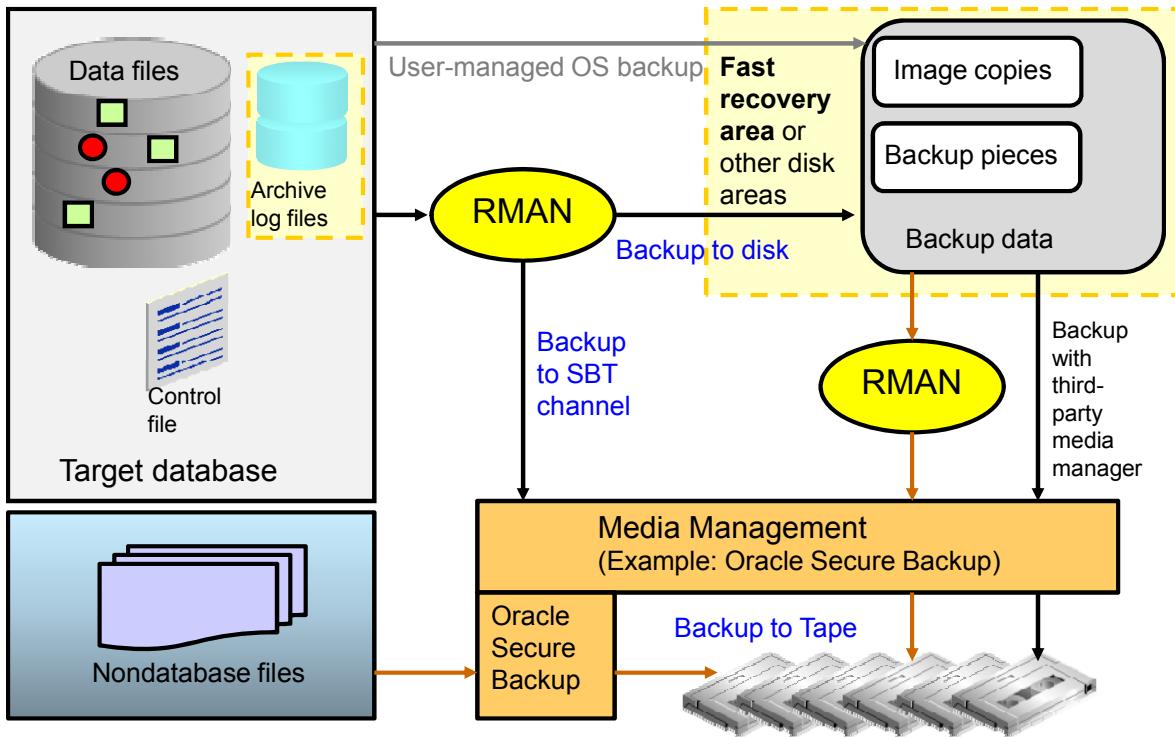


Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

This is the first lesson in the “Backup Unit,” which includes:

- **Lesson 5:** Backup Strategies and Terminology
- **Lesson 6:** Performing Backups
- **Lesson 7:** Improving Your Backups
- **Lesson 8:** Using RMAN-Encrypted Backups

Backup Solutions: Overview



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

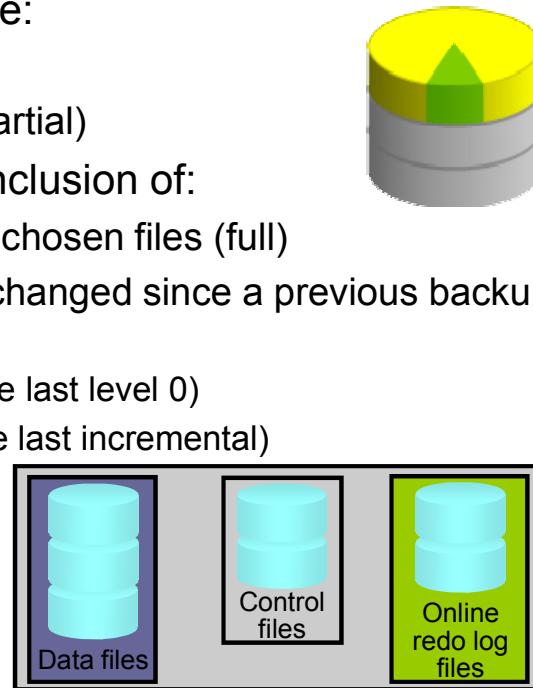
Recovery Manager (RMAN) is the recommended method of backing up your Oracle database. You can use it to back up to disk or to a System Backup to Tape (SBT) channel. Oracle recommends that disk backups are stored in the Fast Recovery Area (FRA).

Oracle Secure Backup complements existing functionality by adding backup to tape and backup of file system data. It interacts transparently with RMAN. Third-party media managers can also be used to back up to tape.

User-managed backups are non-RMAN backups, for example, using an OS utility. They are often based on scripts that a DBA must write. This option is being phased out because it is more labor intensive.

Backup Terminology

- *Backup strategy* may include:
 - Entire database (whole)
 - Portion of the database (partial)
- *Backup type* may indicate inclusion of:
 - All data blocks within your chosen files (full)
 - Only information that has changed since a previous backup (incremental)
 - Cumulative (changes since last level 0)
 - Differential (changes since last incremental)
- *Backup mode* may be:
 - Offline (consistent, cold)
 - Online (inconsistent, hot)



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

- **Whole database backup:** Includes all data files and at least one control file (Remember that all control files in a database are identical.)
- **Partial database backup:** May include zero or more tablespaces and zero or more data files; may or may not include a control file
- **Full backup:** Makes a copy of each data block that contains data and that is within the files being backed up
- **Incremental backup:** Makes a copy of all data blocks that have changed since a previous backup. The Oracle database supports two levels of incremental backup (0 and 1). A level 1 incremental backup can be one of two types: *cumulative* or *differential*. A cumulative backup backs up all changes since the last level 0 backup. A differential backup backs up all changes since the last incremental backup (which could be either a level 0 or level 1 backup). Change Tracking with RMAN supports incremental backups.
- **Offline backups** (also known as “cold” or *consistent* backup): Are taken while the database is not open. They are consistent because, at the time of the backup, the system change number (SCN) in data file headers matches the SCN in the control files.
- **Online backups** (also known as “hot” or *inconsistent* backup): Are taken while the database is open. They are inconsistent because, with the database open, there is no guarantee that the data files are synchronized with the control files.

Balancing Backup and Restore Requirements

Consideration	Performance Effect
Incremental Backup Strategy	<ul style="list-style-type: none"> Improves backup performance, with trade-off in recovery performance Block change tracking for fast incremental backups Cumulative and differential incremental backups “Incremental forever” requires an initial full backup.
Multiplexing	<ul style="list-style-type: none"> Back up files in parallel per channel to improve performance. Multiplexing level = min (FILESPERSET, MAXOPENFILES). Set MAXOPENFILES = 1 for SAME or ASM data files. Set # of RMAN channels = # of tape drives.
Hardware/Network/Storage	<ul style="list-style-type: none"> Assess host resources, production disk I/O, HBA/network, tape drive throughput. Minimum performing component of these will be a performance bottleneck.

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Incremental Backup Strategy

Incremental backups generally take up less space than a full backup and typically take less time to create. Recovery with incremental backups is generally faster than using redo log files to apply changes to a backup.

By incrementally updating backups, you can avoid the overhead of making full image copy backups of data files, while also minimizing time required for media recovery of your database.

The block change tracking feature for incremental backups can improve backup performance by recording changed chunks of blocks for each data file. If block change tracking is enabled on a primary or standby database, RMAN uses a block change tracking file to identify a chunk of changed blocks for incremental backups. By reading this small bitmap file, RMAN avoids having to scan every block in the data file that it is backing up.

In a *differential incremental backup*, all blocks changed after the most recent incremental level 1 or 0 backup are backed up. In a *cumulative incremental backup*, all blocks changed after the most recent level 0 incremental backup are backed up. When recovery time is more important than disk space, cumulative backups are preferable because fewer incremental backups need to be applied during recovery.

RMAN Multiplexing

An RMAN channel represents a single backup file stream. A channel can read multiple data files or archived logs into a multiplexed backup set, or can read one file at a time for an image copy backup. Increasing the number of RMAN channels increases backup parallelism, which may reduce the time required to create the backup.

RMAN multiplexing refers to the number of data files or archived logs that are read by one channel at any time. The default is min (FILESPERSET, MAXOPENFILES). FILESPERSET is specified in the BACKUP command and defaults to 64. MAXOPENFILES is specified in the CONFIGURE CHANNEL command and defaults to 8. Multiplexing therefore defaults to 8, meaning that a maximum of eight files will be read by one channel at any one time.

For Stripe and Mirror Everything (SAME) and ASM storage, MAXOPENFILES should be set to 1 because all files are striped appropriately across available disks and will be read efficiently by RMAN.

Do not use media management multiplexing (multiple channels per tape drive). RMAN backup pieces will not be efficiently restored due to the interleaving of pieces on the same tape volume, which may necessitate the forward and backward movement of the tape.

Comparing Backup Strategies

Strategy	Backup Factors	Recovery Factors
Option 1: Full and Incremental Backups	Fast incremental backups Save space with backup compression. Cost-effective tape storage	Full backup restored first, then incremental backups and archived logs Tape backups read sequentially
Option 2: Incrementally Updated Disk Backups	Incremental + roll forward to create up-to-date copy Requires 1x production storage for copy	Backups read via random access Restore-free recovery with SWITCH command.
Option 3: Offload Backups to Physical Standby Database	Above benefits + primary database free to handle more workloads Requires 1X production hardware and storage for standby database	Fast failover to standby database in the event of any failure Backups are last resort, in the event of double site failure.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

On the next few pages the backup strategy options presented in the table are discussed in detail.

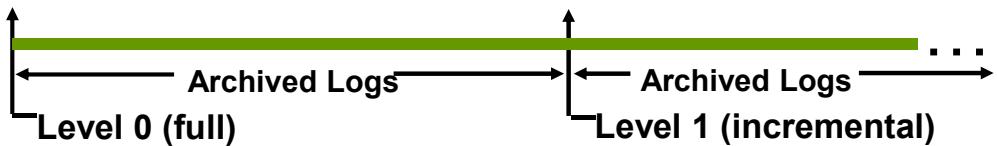
The options provide the following major advantages:

- **Option 1:** Space and cost effectiveness
- **Option 2:** Recovery effectiveness
- **Option 3:** Benefits of options 1 and 2, and the ability to offload production work

These examples should help you to develop your own backup and recovery strategy, which can be a combination of these options, tailored to your recovery requirements and available infrastructure (tapes, disks, and so on).

Option 1: Full and Incremental Backups

- Well-suited for:
 - Databases that can tolerate hours/days RTO
 - Environments where disk is premium
 - Low-medium change frequency between backups (<20%)
- Backup strategy:
 - Weekly level 0 and daily differential incremental backup sets to tape, with optional backup compression
 - Enable block change tracking so that only changed blocks are read and written during incremental backup.
 - Back up archived logs and retain on-disk, as needed.



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this option, you take weekly level 0 incremental backups to tape and daily differential incremental backups.

You can also take advantage of RMAN support for binary compression of backup sets. However, if your tape device performs its own compression, you should not use both RMAN backup set compression and the media manager vendor's compression. In most cases, the media manager compression provides better results.

When you enable block change tracking, RMAN uses a block change tracking file to identify changed blocks for incremental backups. Instead of scanning every block in the data file that is being backed up, RMAN reads the small bitmap file to determine which blocks have changed.

Option 2: Incrementally Updated Disk Backups

- Well-suited for:
 - Databases that can tolerate no more than a few hours RTO
 - Environments where disk can be allocated for 1x size of database or most critical tablespaces
- Backup strategy:
 - Initial image copy to FRA; daily incremental backups
 - New on-disk copy by using incrementals to roll forward copy
 - Full backup archived to tape as needed
 - Archived logs backed up and retained on-disk as needed
 - Fast recovery from disk or SWITCH to use image copies



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this option, you create incrementally updated backups. Create an image copy of each data file and then apply daily level 1 incremental backups to roll forward the image copy. With this technique you avoid the overhead of creating multiple full image copies of your data files; however, you can take advantage of the use of image copies during a recovery operation to minimize recovery time.

Option 3: Offloading Backups to Physical Standby Database in Data Guard Environment

- Well-suited for:
 - Databases that require no more than several minutes of recovery time in the event of any failure
 - Environments that can preferably allocate symmetric hardware and storage for physical standby database
 - Environments with tape infrastructure that can be shared between primary and standby database sites
- Backup strategy:
 - Full and incremental backups offloaded to physical standby database
 - Fast incremental backup on standby with Active Data Guard
 - Backups restored to primary or standby database
 - Backups taken at each database for optimal local protection



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In a Data Guard environment, you can offload incremental backups to a physical standby database. Incremental backups of a standby and primary database are interchangeable. You can apply an incremental backup of a standby database to a primary database, or apply an incremental backup of a primary database to a standby database.

Note: The Oracle Active Data Guard option is required for the use of block change tracking on the physical standby database.

Backing Up Read-Only Tablespaces

Considerations for backing up read-only tablespaces:

- Backup optimization causes RMAN to back up read-only tablespaces only when no backup exists that satisfies the retention policy.
- If you change the tablespace to read/write, back it up immediately.
- You can use the `SKIP READONLY` option of the RMAN `BACKUP` command to skip read-only tablespaces or data files.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Because read-only tablespaces are not being written to, there is no need to continually back them up as you do read/write tablespaces. You can use the `SKIP READONLY` option of the `BACKUP` command to let RMAN know to not back up read-only tablespaces.

Data Warehouse Backup and Recovery: Best Practices

- Exploit partitioning and read-only tablespaces:
 - Older partitions can be moved to read-only tablespaces.
 - Back up read-only tablespaces once and then periodically depending on tape retention policy.
- Divide full backup workload across multiple days.
- Leverage database and backup compression.
- Save time with tablespace-level backups.
 - Back up index tablespaces less frequently than data tablespaces.
 - Back up scarcely used tablespaces less frequently.
 - Reduce restore time for most critical tablespaces, by grouping them together in separate backups.
- Take incremental backup when NOLOGGING operations finish to ensure recoverability.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

By using a command similar to the following, you can divide the full backup workload across multiple days:

```
BACKUP DATABASE NOT BACKED UP SINCE 'SYSDATE-3' DURATION 06:00  
PARTIAL MINIMIZE TIME;
```

When you execute the command on the first day, a full backup begins. It executes for six hours. On the next day when the command executes again, the backup starts with the last file not backed up and resumes the full backup, again in a six-hour backup window. Over the course of a few days the entire database will be backed up.

Additional Backup Terminology

Backups may be stored as:

- Image copies
- Backup sets

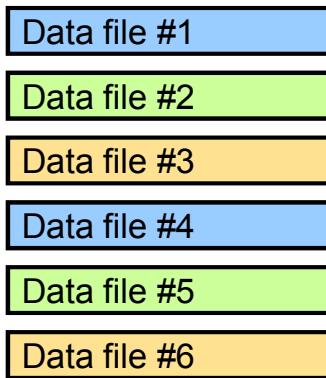
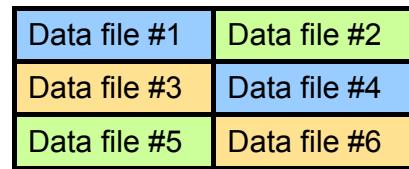


Image copies
(Duplicate data and log files in OS format)



Backup set
(Binary files in Oracle
proprietary format)

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

- **Image copies:** Are duplicates of data or archived log files (similar to simply copying the files by using operating system commands)
- **Backup sets:** Are collections of one or more binary files that contain one or more data files, control files, server parameter files, or archived log files. With backup sets, empty data blocks are not stored, thereby causing backup sets to use less space on the disk or tape. Backup sets can be compressed to further reduce the space requirements of the backup.

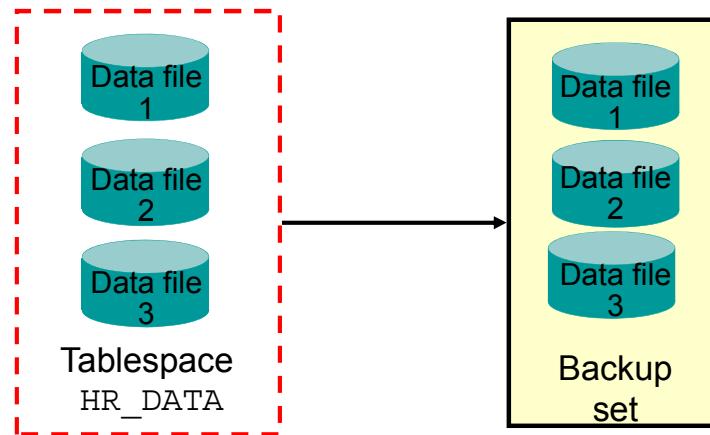
Image copies must be backed up to the disk. Backup sets can be sent to the disk or directly to the tape.

The advantage of creating a backup as an image copy is improved granularity of the restore operation. With an image copy, only the file or files need to be retrieved from your backup location. With backup sets, the entire backup set must be retrieved from your backup location before you extract the file or files that are needed.

The advantage of creating backups as backup sets is better space usage. In most databases, 20% or more of the data blocks are empty blocks. Image copies back up every data block, even if the data block is empty. Backup sets significantly reduce the space required by the backup. In most systems, the advantages of backup sets outweigh the advantages of image copies.

Creating Backup Sets

```
RMAN> BACKUP AS BACKUPSET  
2> FORMAT '/BACKUP/df_%d_%s_%p.bus'  
3> TABLESPACE hr_data;
```



ORACLE

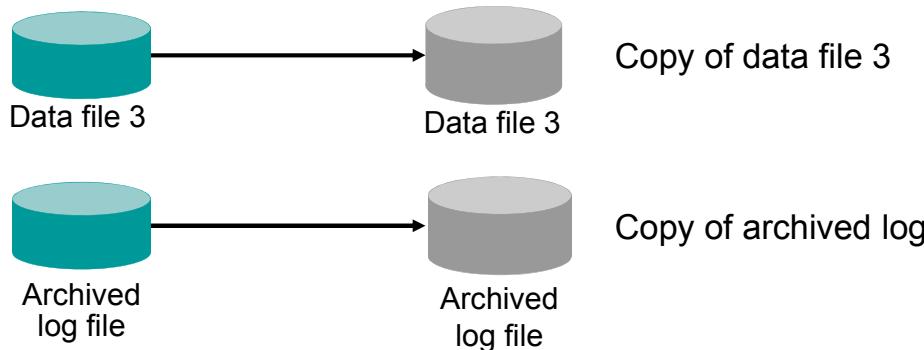
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

RMAN can store its backups in an RMAN-exclusive format called a backup set. A backup set is a collection of files called backup pieces, each of which may contain a backup of one or more database files.

Note: The `FORMAT` parameter specifies a pattern to use in creating a file name for the backup pieces created by this command. The `FORMAT` specification can also be provided through the `ALLOCATE CHANNEL` and `CONFIGURE` commands.

Creating Image Copies

```
RMAN> BACKUP AS COPY DATAFILE '/ORADATA/users_01_db01.dbf';
RMAN> BACKUP AS COPY ARCHIVELOG LIKE '/arch%';
```



ORACLE

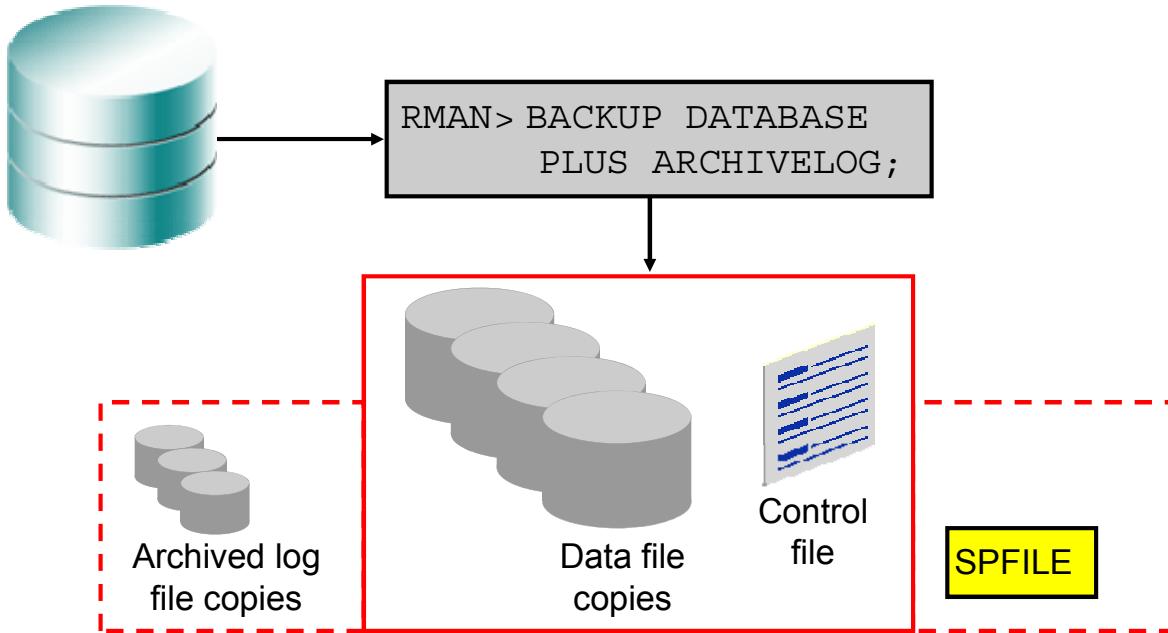
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

An image copy is a copy of a single data file, archived redo log, or control file. An image copy can be created with the `BACKUP AS COPY` command or with an operating system command. When you create the image copy with the RMAN `BACKUP AS COPY` command, the server session validates the blocks in the file and records the copy information in the control file.

An image copy has the following characteristics:

- An image copy can be written only to 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 by using the `SWITCH` command in RMAN, 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 database 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. By default, RMAN computes a checksum for each block backed up, and stores it with the backup. When the backup is restored, the checksum is verified.
- An image copy can be part of a full or incremental level 0 backup because a file copy always includes all blocks. You must use the level 0 option if the copy will be used in conjunction with an incremental backup set.

Creating a Whole Database Backup



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

A whole database backup can be either backup sets or image copies of the entire set of data files and must include the control file (included automatically). You can optionally add the server parameter file (SPFILE) and archived redo log files. Using Recovery Manager (RMAN) to make an image copy of all the database files simply requires mounting or opening the database, starting RMAN, and entering the BACKUP command shown in the slide.

Optionally, you can supply the `DELETE INPUT` option when backing up archive log files. That causes RMAN to remove the archive log files after backing them up. This is useful especially if you are not using a fast recovery area, which would perform space management for you, deleting files when space pressure grows. An example follows:

```
RMAN> BACKUP DATABASE PLUS ARCHIVELOG DELETE INPUT;
```

You must have issued the following `CONFIGURE` commands to make the backup as described previously:

- `CONFIGURE DEFAULT DEVICE TYPE TO disk;`
- `CONFIGURE DEVICE TYPE DISK BACKUP TYPE TO COPY;`
- `CONFIGURE CONTROLFILE AUTOBACKUP ON;`

To make a separate backup of SPFILE means to export it to a text file with the following SQL command:

- `CREATE PFILE FROM SPFILE;`

You can also create a backup (either a backup set or image copies) of previous image copies of all data files and control files in the database by using the following command:

```
RMAN> BACKUP COPY OF DATABASE;
```

By default, RMAN executes each BACKUP command serially. However, you can parallelize the copy operation by:

- Using the CONFIGURE DEVICE TYPE DISK PARALLELISM *n* command, where *n* is the desired degree of parallelism
- Allocating multiple channels
- Specifying one BACKUP AS COPY command and listing multiple files

Quiz

Oracle performs space management for the fast recovery area.

- a. True
- b. False



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Answer: a

Quiz

A full backup and a whole database backup are always identical.

- a. True
- b. False



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Answer: b

Quiz

Which of the following are true about RMAN image copies?

- a. Image copies do not contain empty blocks.
- b. Image copies are bit-by-bit copies of the data files.
- c. Image copies can be backed up to tape.
- d. Image copies can be backed up only to disk.
- e. Image copies can be processed as multisection backups.
- f. During restoration, data files must be extracted from image copies.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Answer: b, d, e

Summary

In this lesson, you should have learned how to:

- Describe Oracle backup solutions
- Describe RMAN backup types
- Describe, compare, and determine your backup strategy
- Schedule backups according to your strategy
- Perform whole backups



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Practice Overview: Developing a Backup Strategy

- Practice 5-1 covers designing appropriate backup strategy based on specified business requirements.
- Practice 5-2 covers creating a backup scheduled by using Enterprise Manager.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this practice, you develop backup strategies for different types of databases with different requirements and implement a backup schedule by using Enterprise Manager.

Case Study 1: How to Protect an OLTP Database

The first case is an online transaction processing (OLTP) database, handling a large number of transactions per day. The business requirements are no data loss, with minimal down time. The time to restore and recover must be less than an hour. The database is 300 GB. Several TB of disk space is available for backups. All the available disks have the same properties (size, I/O rate, and latency). Tape backup is available.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

ASSUMPTION: The full range of Oracle backup and recovery tools is available.

1. Question: What steps do you take to protect the database (for instance place the database in ARCHIVELOG mode)?
2. Question: How much disk space will you need?
3. Question: What is the retention policy?
4. Question: Will you use a fast recovery area?
5. Question: Will you use backup sets or image copies?
6. Question: Will you use full or incremental backups?
7. Question: What recovery method will you use?

Case Study 2: How to Protect a DSS Database

The database is a Decision Support System (DSS). Data is loaded via SQL*Loader files each night from several transaction databases. The database DSS keeps data for 10 years. The transaction databases keep only one year worth of data. The data is updated only in the transaction databases, and is replaced in the DSS database. Only new and updated records are transferred to the DSS database. The DSS database is 10 TB. Separate tablespaces are used to hold the data by year. There are approximately 200 tablespaces.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

ASSUMPTION: The full range of Oracle backup and recovery tools is available.

1. Question: What else do you need to know to design a backup strategy?
Examples: What is the cost, availability, and speed of disk storage?
2. Question: What steps do you take to protect the database?
3. Question: How much disk or tape space will you need?
4. Question: What is the retention policy?
5. Question: Will you use a fast recovery area?
6. Question: Will you use backup sets or image copies?
7. Question: Will you use full or incremental backups?
8. Question: What recovery method will you use?

Case Study 3: How to Protect the Recovery Catalog Database

The database is a recovery catalog, holding the RMAN catalog information for more than 20 databases in the company.

Backups and restore operations may be going on at any time.
The databases are mission critical.

1. Question: How are the databases recovered if the recovery catalog is unavailable?
2. Question: What is the retention policy?
3. Question: Will you use a fast recovery area?
4. Question: Will you use backup sets or image copies?
5. Question: Will you use full or incremental backups?
6. Question: What recovery method will you use?



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

ASSUMPTION: The full range of Oracle backup and recovery tools is available.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and Error : You are not a Valid Partner use only

6

Performing Backups

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Perform full and incremental backups
- Use Oracle-suggested backup strategy
- Report and manage backups
- Begin refining your basic backups:
 - Configure Block Change tracking
 - Perform an incremental level 1 backup
 - Recover an incremental level 0 backup with level 1 incremental



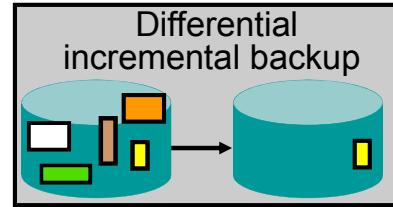
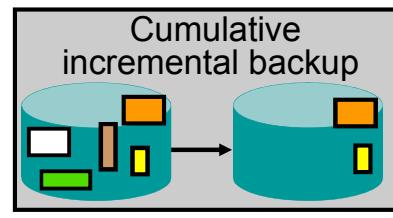
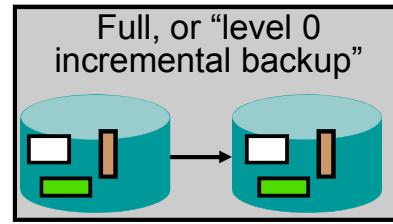
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

This is the second lesson in the “Backup Unit,” which includes:

- **Lesson 5:** Backup Strategies and Terminology
- **Lesson 6:** Performing Backups
- **Lesson 7:** Improving Your Backups
- **Lesson 8:** Using RMAN-Encrypted Backups

RMAN Backup Types

- A *full backup* contains all used data file blocks.
- A *level 0 incremental backup* is equivalent to a full backup that has been marked as level 0.
- A *cumulative level 1 incremental backup* contains only blocks modified since the last level 0 incremental backup.
- A *differential level 1 incremental backup* contains only blocks modified since the last incremental backup.



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Full Backups

A full backup is different from a whole database backup. A full data file backup is a backup that includes every used data block in the file. RMAN copies all blocks into the backup set or image copy, skipping only those data file blocks that are not part of an existing segment. For a full image copy, the entire file contents are reproduced exactly. A full backup cannot be part of an incremental backup strategy; it cannot be the parent for a subsequent incremental backup.

Incremental Backups

An incremental backup is either a level 0 backup, which includes every block in the data files except blocks that have never been used, or a level 1 backup, which includes only those blocks that have been changed since a previous backup was taken. A level 0 incremental backup is physically identical to a full backup. The only difference is that the level 0 backup (as well as an image copy) can be used as the base for a level 1 backup, but a full backup can never be used as the base for a level 1 backup.

Incremental backups are specified by using the `INCREMENTAL` keyword of the `BACKUP` command. You specify `INCREMENTAL LEVEL [0 | 1]`.

RMAN can create multilevel incremental backups as follows:

- **Differential:** Is the default type of incremental backup that backs up all blocks changed after the most recent incremental backup at either level 1 or level 0
- **Cumulative:** Backs up all blocks changed after the most recent backup at level 0

Examples

- To perform an incremental backup at level 0, use the following command:

```
    RMAN> BACKUP INCREMENTAL LEVEL 0 DATABASE;
```

- To perform a differential incremental backup, use the following command:

```
    RMAN> BACKUP INCREMENTAL LEVEL 1 DATABASE;
```

- To perform a cumulative incremental backup, use the following command:

```
    RMAN> BACKUP INCREMENTAL LEVEL 1 CUMULATIVE DATABASE;
```

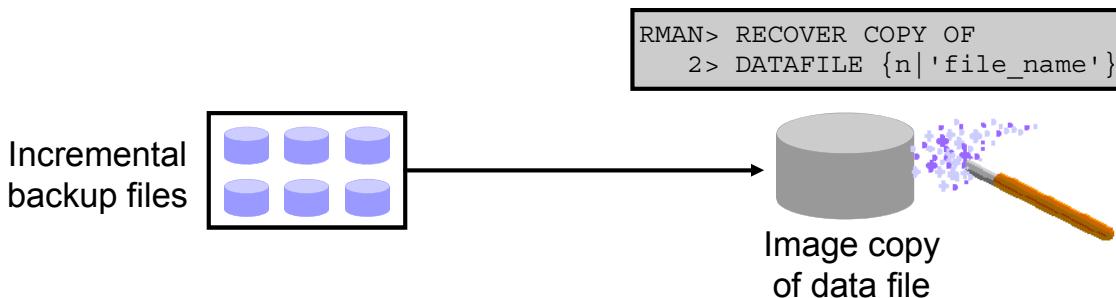
RMAN makes full backups by default if neither FULL nor INCREMENTAL is specified. Unused block compression causes never-written blocks to be skipped when backing up data files to backup sets, even for full backups.

A full backup has no effect on subsequent incremental backups, and is not considered part of any incremental backup strategy, although a full image copy backup can be incrementally updated by applying incremental backups with the RECOVER command.

Note: It is possible to perform any type of backup (full or incremental) of a database that is in NOARCHIVELOG mode—if, of course, the database is not open. Note also that recovery is limited to the time of the last backup. The database can be recovered to the last committed transaction only when the database is in ARCHIVELOG mode.

Incrementally Updated Backups

- Image copies are updated with all changes up to the incremental backup SCN.
- Incremental backup reduces the time required for media recovery.
- With incrementally updated backups, you can use the SWITCH command during the recovery operation.



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can use RMAN to apply incremental backups to data file image copies.

```
BACKUP AS COPY INCREMENTAL LEVEL 0 DATABASE;
```

With the use of incrementally updated backups, you can use the SWITCH command during the recovery operation.

RMAN can roll forward (recover) an image copy to the specified point in time by applying the incremental backups to the image copy. The image copy is updated with all changes up through the SCN at which the incremental backup was taken. RMAN uses the resulting updated data file in media recovery just as it would use a full image copy taken at that SCN, without the overhead of performing a full image copy of the database every day. The following are the benefits of applying incremental backups to data file image copies:

- You reduce the time required for media recovery (using archive logs) because you need to apply archive logs only since the last incremental backup.
- You do not need to perform a full image copy after the incremental restoration.

If the recovery process fails during the application of the incremental backup file, you simply restart the recovery process. RMAN automatically determines the required incremental backup files to apply, from before the image data file copy until the time at which you want to stop the recovery process. If there is more than one version of an image copy recorded in the RMAN catalog, RMAN automatically uses the latest version of the image copy. RMAN reports an error if it cannot merge an incremental backup file with an image copy.

Incrementally Updated Backups: Example

If you execute these commands daily:

```
RMAN> recover copy of database with tag 'daily_inc';
RMAN> backup incremental level 1 for recover of copy
2> with tag 'daily_inc' database;
```

This is the result:

	RECOVER	BACKUP
Day 1	Nothing	Create image copies
Day 2	Nothing	Create incremental level 1
Day 3 and onward	Recover copies based on incremental	Create incremental level 1



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

If you execute the commands shown in the slide daily, you get continuously updated image copies of all the database data files at any time.

The chart shows what happens for each run. Note that this algorithm requires some priming; the strategy does not come to fruition until after day 3.

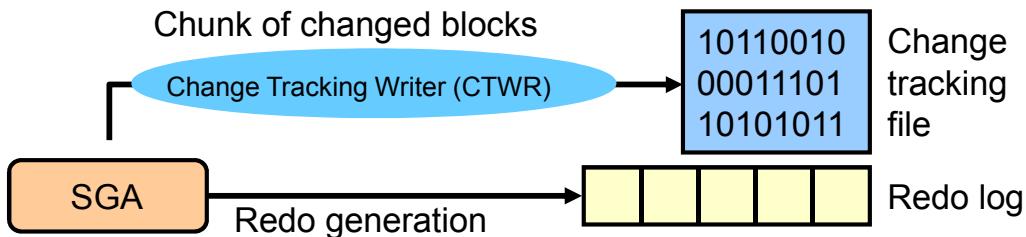
- **Day 1:** The RECOVER command does nothing. There are no image copies to recover. The BACKUP command creates the image copies.
- **Day 2:** The RECOVER command, again, does nothing. This is because there is no incremental backup yet. The BACKUP command creates the incremental backup, now that baseline image copies have been created on day 1.
- **Day 3:** The RECOVER command applies the changes from the incremental backup to the image copies. The BACKUP command takes another incremental backup, which will be used to recover the image copies on day 4. The cycle continues like this.

It is important to use tags when implementing this kind of backup strategy. They serve to link these particular incremental backups to the image copies that are made. Without the tag, the most recent, and possibly incorrect, incremental backup would be used to recover the image copies.

Fast Incremental Backup

Implemented by block change tracking, which:

- Maintains a record of block chunks that have changed since the last backup
- Writes this record to a file, as redo is generated
- Is automatically accessed when a backup is done, and can make the backup complete more quickly
- Is optimized for up to eight incremental backups
- Is recommended if the changes are less than 20%



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can perform fast incremental backup by enabling block change tracking (BCT). The BCT file can contain only eight bitmaps, so the backup cannot be optimized if there have been more than eight incremental backups since the parent-level backup that the new incremental will be based on. Consider the eight-bitmap limit when developing your incremental backup strategy. For example, if you make a level 0 database backup followed by seven differential incremental backups, then the block change tracking file now includes eight bitmaps. If you then make a cumulative level 1 incremental backup, RMAN cannot optimize the backup because the bitmap corresponding to the parent level 0 backup is overwritten with the bitmap that tracks the current changes.

When it is time to perform the incremental backup, RMAN can look at the block change tracking file, determine the modified block chunks, and back up only those blocks. It does not have to scan every block to see whether it has changed since the last backup. This can make incremental backup faster.

Block change tracking is recommended when your changes are 20% or less.

The maintenance of the tracking file is fully automatic and does not require your intervention. The minimum size for the block change tracking file is 10 MB, and any new space is allocated in 10 MB increments. The Oracle database server does not record block change information by default.

Maintaining Block Change Tracking File

- The DB_CREATE_FILE_DEST initialization parameter provides the default destination.
- Enable or disable with:

```
ALTER DATABASE
  {ENABLE|DISABLE} BLOCK CHANGE TRACKING
  [USING FILE '...']
```
- Rename block change tracking file with the ALTER DATABASE RENAME command (database must be in MOUNT state).



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can enable block change tracking in multiple tools.

For example, in Cloud Control, navigate to the database instance home page > Availability > Backup & Recovery > Backup Settings > Policy. You do not need to set the block change tracking file destination if the DB_CREATE_FILE_DEST initialization parameter is set. But you can specify the name of the block change tracking file, placing it in any location you choose.

You can also enable or disable this feature by using an ALTER DATABASE command. If the change tracking file is stored in the database area with your database files, it is deleted when you disable change tracking.

You can rename the block change tracking file by using the ALTER DATABASE RENAME command. Your database must be in the MOUNT state to rename the tracking file. The ALTER DATABASE RENAME FILE command updates the control file to refer to the new location. You can use the following syntax to change the location of the block change tracking file:

```
ALTER DATABASE RENAME FILE '...' TO '...';
```

Note: RMAN does not support backup and recovery of the block change tracking file. For this reason, you should not place it in the fast recovery area.

Monitoring Block Change Tracking

```
SQL> SELECT filename, status, bytes  
2   FROM v$block_change_tracking;
```

```
SQL> SELECT file#, avg(datafile_blocks),  
2           avg(blocks_read),  
3           avg(blocks_read/datafile_blocks)  
4             * 100 AS PCT_READ_FOR_BACKUP,  
5           avg(blocks)  
5   FROM v$backup_datafile  
6  WHERE used_change_tracking = 'YES'  
7  AND incremental_level > 0  
8  GROUP BY file#;
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

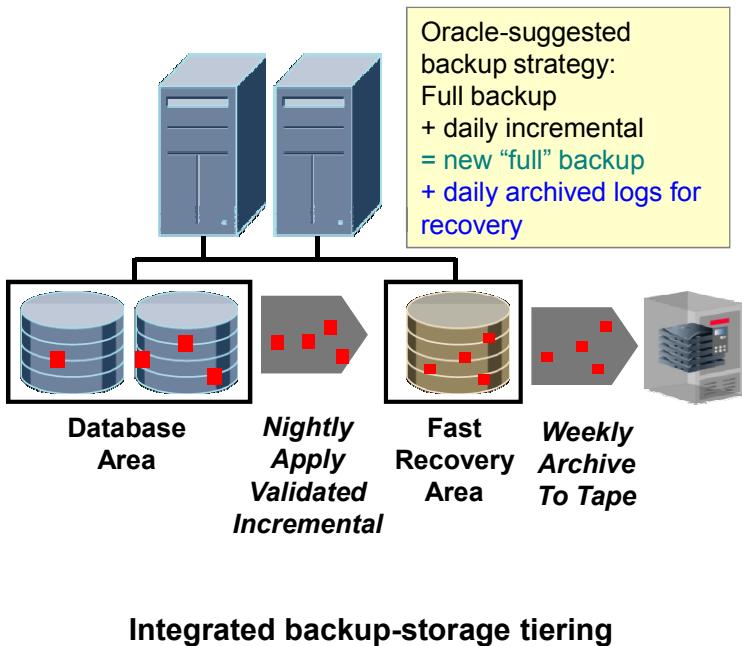
The output of the V\$BLOCK_CHANGE_TRACKING view shows where the block change tracking file is located, the status of block change tracking (ENABLED/DISABLED), and the size (in bytes) of the file.

The query on the V\$BACKUP_DATAFILE view shows how effective the block change tracking is in minimizing the incremental backup I/O (the PCT_READ_FOR_BACKUP column). A high value indicates that RMAN reads most blocks in the data file during an incremental backup. You can reduce this ratio by decreasing the time between the incremental backups.

A sample formatted output from the V\$BACKUP_DATAFILE query is shown in the following:

FILE#	BLOCKS_IN_FILE	BLOCKS_READ	PCT_READ_FOR_BACKUP	BLOCKS_BACKED_UP
1	56320	4480	7	462
2	3840	2688	70	2408
3	49920	16768	33	4457
4	640	64	10	1
5	19200	256	1	91

Automatic Disk-to-Disk Backup and Recovery



- Integrated disk-to-disk backup and recovery: Low-cost disks used for fast recovery area
- Fast incremental backups: Back up only changed blocks.
- Nightly incremental backup rolls forward recovery area backup: No need to do full backups

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

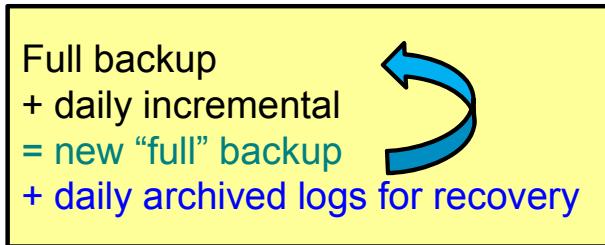
RMAN fully automates disk-based backup and recovery by using essentially a tiered storage configuration (as shown in the slide). You can use different types of storage for the database files and the fast recovery area (FRA).

With the **Oracle-suggested backup strategy**, you perform a full backup of your database (an image copy of each data file). Then set up automated nightly incremental backups, stored in the FRA. Only changed data blocks are backed up, which save considerable storage space. You can implement block change tracking to track the changed blocks more efficiently. The nightly incremental backups can be used to roll forward the database backup (by applying an incremental level 1 backup) and automatically create a full backup.

This procedure enables faster backups by propagating changes to the FRA. In addition, restores are faster because backup files are copied from the FRA or a copy of the file in the FRA. (If recovery is needed, then only the daily archived redo logs need to be applied to yesterday's full backup.)

Oracle-Suggested Backup

- Provides an out-of-the-box backup strategy based on the backup destination
- Sets up recovery window for backup management
- Schedules recurring and immediate backups:



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Enterprise Manager makes it easy for you to set up an Oracle-suggested backup strategy that protects your data and provides efficient recoverability. By default, there is a 24-hour recovery window from disk. The Oracle-suggested strategy uses the incremental backup and incrementally updated backup features, providing faster recoverability than is possible when applying database changes from the archived log files.

To establish an Oracle-suggested strategy, navigate to the database home page > Availability > Backup & Recovery > Schedule Backup. The Backup Strategies section enables you to select from the Oracle-suggested backup and Customized backup strategies. The Oracle-suggested strategy takes a full database copy as the first backup. Because it is a whole database backup, you might want to consider taking this at a period of least activity. After that, an incremental backup to disk is taken every day. Optionally, a weekly tape backup can be made, which backs up all recovery-related files.

Because these backups on disk are retained, you can always perform a full database recovery or a point-in-time recovery to any time within the past 24 hours, at the minimum. The default recovery time could reach back as far as 48 hours. This is because just before a backup is taken on a given day, the backup from the beginning of day n-1 still exists.

You can change this: depending on your organization's disk capacity, three (to seven) days are often used.

Reporting on Backups

RMAN commands:

- LIST: Displays information about backup sets, proxy copies, and image copies recorded in the repository
- REPORT: Produces a detailed analysis of the repository
- REPORT NEED BACKUP: Lists all data files that require a backup
- REPORT OBSOLETE: Identifies files that are no longer needed to satisfy backup retention policies

Enterprise Manager Cloud Control:

- Graphical, customizable interface



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

- Use the RMAN LIST command to display information about backup sets, proxy copies, and image copies recorded in the repository. There are a variety of ways in which you can list backup information.
- Use the RMAN REPORT command to analyze information in the RMAN repository in more detail.
- The REPORT NEED BACKUP command is used to identify all data files that need a backup. The report assumes that the most recent backup would be used in the event of a restore.
- With the REPORT OBSOLETE command, you can identify files that are no longer needed to satisfy backup retention policies. By default, the REPORT OBSOLETE command reports which files are obsolete under the currently configured retention policy. You can generate reports of files that are obsolete according to different retention policies by using REDUNDANCY or RECOVERY WINDOW retention policy options with the REPORT OBSOLETE command.

Using Dynamic Views

Query the following dynamic views in the target database to obtain information about your backups:

- V\$BACKUP_SET: Backup sets created
- V\$BACKUP_PIECE: Backup pieces that exist
- V\$DATAFILE_COPY: Copies of data files on disk
- V\$BACKUP_FILES: Information about all files created when creating backups



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

There are many views that provide backup-related information. The most commonly used ones are shown in the slide.

If you are using a recovery catalog, you can query corresponding views that contain the same information for each target database registered in the recovery catalog database. The corresponding views have the same name, except that the “V\$” is replaced with “RC_”. Also, they are in the schema owned by the recovery catalog owner. For example, the corresponding views in the recovery catalog, showing the information shown in the slide are RC_BACKUP_SET, RC_BACKUP_PIECE, RC_DATAFILE_COPY, and RC_BACKUP_FILES.

In order to query the RC_BACKUP_FILES view, you must first execute the following in the recovery catalog database:

```
SQL> CALL DBMS_RCMAN.SETDATABASE(null,null,null,<dbid>);
```

where <dbid> is the database ID of a target database.

Managing Backups: Cross-Checking and Deleting

Use the following RMAN commands to manage your backups:

- **CROSSCHECK:** Verifies the status of backups and copies recorded in the RMAN repository against media such as disk or tape
- **DELETE EXPIRED:** Removes only files whose status in the repository is EXPIRED
- **DELETE OBSOLETE:** Deletes backups that are no longer needed



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Use the CROSSCHECK command to ensure that data about backups in the recovery catalog or control file is synchronized with actual files on disk or in the media management catalog. The CROSSCHECK command operates only on files that are recorded in the RMAN repository.

The CROSSCHECK command checks only objects marked AVAILABLE or EXPIRED by examining the files on disk for DISK channels or by querying the media manager for sbt channels. The CROSSCHECK command updates the repository records to EXPIRED for any files that it is unable to find. It does not delete any files that it is unable to find.

The DELETE command can remove any file that the LIST and CROSSCHECK commands can operate on. For example, you can delete backup sets, archived redo logs, and data file copies. The DELETE command removes both the physical file and the catalog record for the file. The DELETE OBSOLETE command deletes backups that are no longer needed. It uses the same REDUNDANCY and RECOVERY WINDOW options as REPORT OBSOLETE.

If you delete backups without using RMAN, you can use the UNCATALOG command to remove the files from the recovery catalog, or you can use the CROSSCHECK and DELETE EXPIRED commands.

Refer to the *Oracle Database Backup and Recovery Reference* for detailed syntax information.

Quiz

Fast incremental backup enables RMAN to read-only blocks referenced in the block change tracking file during an incremental backup.

- a. True
- b. False



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Answer: a

Quiz

With the use of incrementally updated backups, you can use the SWITCH command during the recovery operation.

- a. True
- b. False



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Answer: a

Summary

In this lesson, you should have learned how to:

- Perform full and incremental backups
- Use Oracle-suggested backup strategy
- Report and manage backups
- Begin refining your basic backups:
 - Configure Block Change tracking
 - Perform incremental level 1 backup
 - Recover incremental level 0 backup with level 1 incremental



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Practice Overview: Creating Incremental Backups

- Practice 6-1 covers configuring block change tracking for fast incremental backups.
- Practice 6-2 covers the following topics:
 - Creating an incremental level 0 backup of the entire database
 - Creating an incremental level 1 backup
 - Recovering incremental level 0 backup with an incremental level 1 backup



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Improving Your Backups

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Lesson Objectives

After completing this lesson, you should be able to:

- Compress backups
- Use a media manager
- Create multi-section backups of very large files
- Create proxy copies
- Create duplexed backup sets
- Create archival backups
- Back up other files:
 - Back up a control file to trace
 - Back up archived redo log files
 - Catalog additional backup files
 - Back up ASM metadata



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

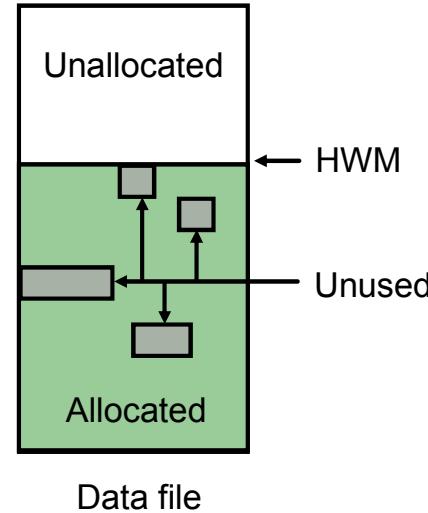
This is the third lesson in the “Backup Unit,” which includes:

- **Lesson 5:** Backup Strategies and Terminology
- **Lesson 6:** Performing Backups
- **Lesson 7:** Improving Your Backups
- **Lesson 8:** Using RMAN-Encrypted Backups

Saving Backup Space with Unused Block Compression

The following blocks may be skipped during certain types of backup operations:

- Unallocated blocks: These are above the data file's high-water mark (HWM).
- Unused blocks: These are blocks that have been allocated but no longer belong to a segment.



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

When certain types of backups occur, RMAN is able to skip some blocks. Unallocated blocks, which are above the high-water mark (HWM) may be skipped. Also, some allocated blocks that no longer belong to a segment (are not in use) may be skipped, provided the following are true:

- There are no guaranteed restore points defined.
- The data file contains data only for locally managed tablespaces.
- The data file is being backed up to a backup set as part of a full backup or a level 0 incremental.
- The backup is going to disk, or Oracle Secure Backup is the media manager.

Compressing Backups

RMAN can perform binary compression on any backup set that is generated.

- It can be performed in addition to unused block compression.
- Available compression algorithms are HIGH, MEDIUM, LOW, and BASIC.
- No extra steps are required by the DBA to restore a compressed backup.

```
CONFIGURE COMPRESSION ALGORITHM 'HIGH/MEDIUM/LOW/BASIC'
```

```
run {  
SET COMPRESSION ALGORITHM 'HIGH/MEDIUM/LOW/BASIC';  
.. }
```

```
BACKUP AS COMPRESSED BACKUPSET DATABASE PLUS ARCHIVELOG;
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

While unused block compression decreases the number of blocks that are written to the backup (and the backup time), binary compression can be used to algorithmically compact the data that is written. The available compression algorithms are HIGH, MEDIUM, LOW, and BASIC. If you specify it for a particular backup device, then use the COMPRESSED keyword after the BACKUP TYPE TO clause.

You do not have to perform any additional steps when restoring a compressed backup. Note, however, that compression and decompression operations require CPU resources. So both creating and restoring a compressed backup will probably take longer and require more system resources.

When choosing an algorithm, consider your disk space in addition to dynamic system resources such as CPU and memory.

You can configure compression per device type or individually for a backup set as shown in the slide.

Using RMAN Backup Compression

Compression Ratio or Level	Considerations	Requires Advanced Compression Option
LOW	Fastest. Best suited to address backup: CPU resources.	✓
MEDIUM	Fast. Good balance of CPU usage and compression ratio.	✓
HIGH	Best compression ratio at the expense of high CPU consumption. Best suited to address backup constraint: network.	✓
BASIC	Fair. Compression ratio similar to MEDIUM at expense of additional CPU usage. Compression ratio between MEDIUM and HIGH.	

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Binary compression of backup sets is supported with the algorithm settings as shown in the slide. All modes except BASIC require the Oracle Advanced Compression Database option.

Because the performance of the various compression levels depends on the nature of the data in the database, network configuration, system resources, and the type of your computer system and its capabilities, Oracle Corporation cannot document universally applicable performance statistics. To decide which level is best for you, consider how balanced your system is regarding bandwidth into the CPU, as well as the actual speed of the CPU. It is highly recommended that you run tests with the different compression levels on the data in your environment. Choosing a compression level based on your own environment, network traffic (workload), and dataset is the only way to ensure that the backup set compression level can satisfy your organization's performance requirements and any applicable service-level agreements.

The following level or compression ratios are available:

- **LOW:** This level is the fastest. It uses the least CPU, but provides less compression than MEDIUM.
- **MEDIUM:** This level provides a good balance of CPU usage and compression ratio.
- **HIGH:** This level provides the best compression ratio, but consumes the most CPU.
- **BASIC:** This offers a compression ratio comparable to MEDIUM, at the expense of additional CPU consumption.

Quiz

Binary compression is used to compact the data that is written to the backup file.

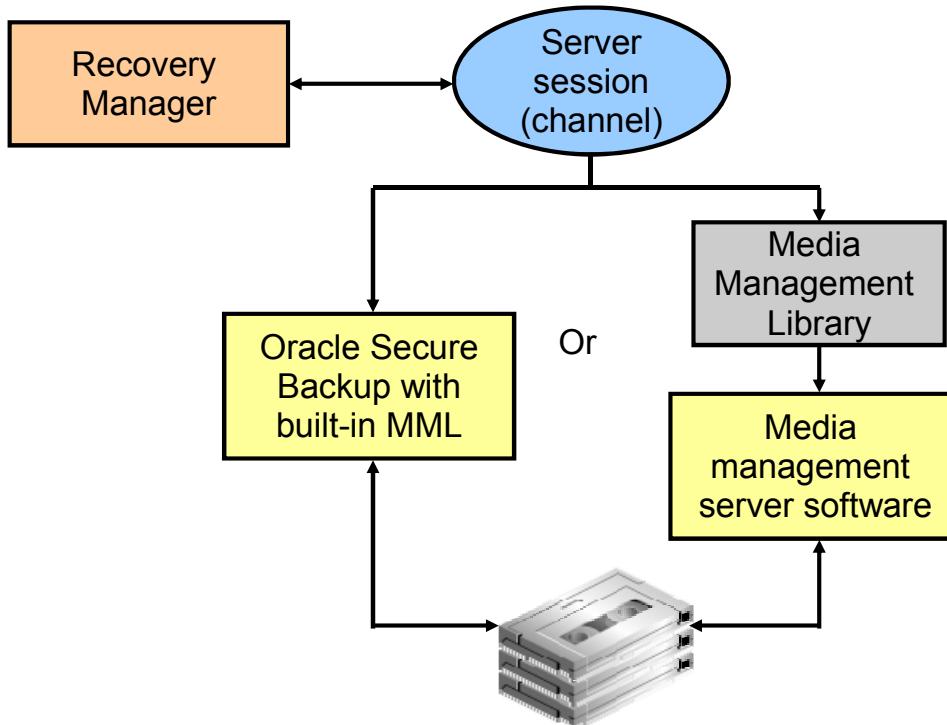
- a. True
- b. False



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Answer: a

Using a Media Manager



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

To use tape storage for your database backups, RMAN requires Oracle Secure Backup or 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. The Oracle database server calls Media Management Library (MML) software routines to back up and restore data files to and from media that is controlled by the media manager.

Note that the Oracle database server does not need to connect to the MML software when it backs up to disk.

Oracle Backup Solutions Program (BSP) provides a range of media management products that are compliant with Oracle's MML specification. Software that is compliant with the MML interface enables an Oracle database session to back up data to a media manager and request the media manager to restore backups. Check with your media vendor to determine whether it is a member of Oracle BSP.

Before you can begin using RMAN with a media manager, you must install the media manager software and make sure that RMAN can communicate with it. Instructions for this procedure should be available in the media manager vendor's software documentation.

Depending on the product that you are installing, perform the following basic steps:

1. Install and configure the media management software on the target host or production network. No RMAN integration is required at this stage.
2. Ensure that you can make non-RMAN backups of operating system files on the target database host. This step makes it easier to troubleshoot problems at a later time. Refer to your media management documentation to learn how to back up files to the media manager.
3. Obtain and install the third-party media management module for integration with the Oracle database. This module must contain the library loaded by the Oracle database server when accessing 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' for serial device  
    ALLOCATE CHANNEL ch1 DEVICE TYPE sbt;  
    BACKUP DATAFILE 3;  
}
```

When Recovery Manager executes this command, it sends the backup request to the Oracle database session performing the backup. The Oracle database 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 the names of the files on each tape. The media manager also handles restore operations. When you restore a file, the following steps occur:

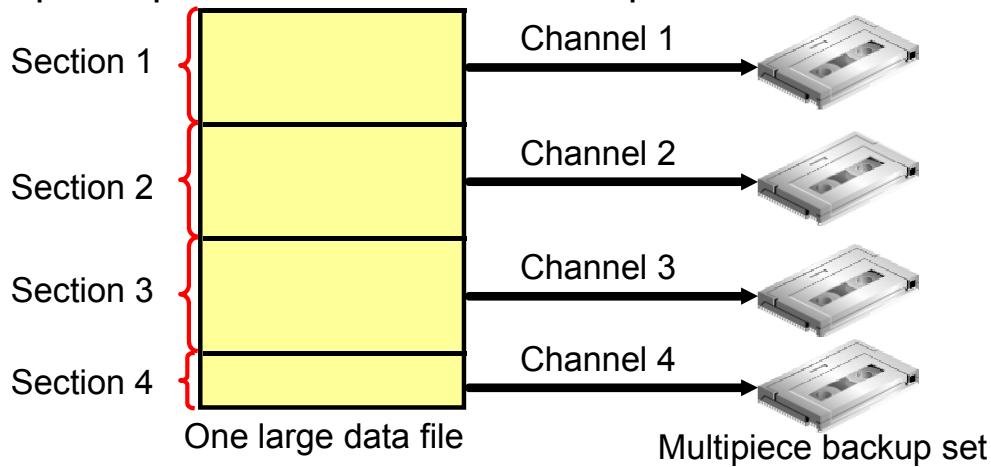
1. The Oracle database server requests the restoration of a particular file.
2. The media manager identifies the tape containing the file and reads the tape.
3. The media manager passes the information back to the Oracle database session.
4. The Oracle database server writes the file to disk.

RMAN also supports *proxy copy*, a feature that enables a media manager to manage completely the transfer of data between disk and backup media. RMAN supplies a list of files that need to be backed up or restored. The media manager then determines how and when to move the data.

Configuring Backup and Restore for Very Large Files

Multisection backups of a single file:

- Are created by RMAN, with your specified size value
- Are processed independently (serially or in parallel)
- Produce multipiece backup sets and image copies
- Improve performance of the backup



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Oracle data files can be up to 128 TB in size. Normally, the smallest unit of an RMAN backup is an entire file. This is not practical with such large files. RMAN can optionally break up large files into sections and back up and restore these sections independently. This feature is built into RMAN. You can use it by creating multisection backups, which break up the files generated for the backup set into separate files. This can be done for backup sets and image copies.

Each file section is a contiguous range of blocks of a file. Each file section can be processed independently, either serially or in parallel. Backing up a file into separate sections can improve the performance of the backup operation, and it also allows large file backups to be restarted.

A multisection backup job produces a multipiece backup set. Each piece contains one section of the file. All sections of a multisection backup, except perhaps for the last section, are of the same size. There are a maximum of 256 sections per file.

Backing Up and Restoring Very Large Files

Multisection backups of a single data file:

- Are created by RMAN, with your specified size value
- Are for backup sets and image copies
- Are for full and incremental backups

Benefits:

- Reduce image copy creation time
- Are processed independently (serially or in parallel)
- Benefit Exadata

Requirements and restrictions:

- COMPATIBLE=12.0
- Not for control or password files
- Not for applying a large value of parallelism



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

How Can This Help?

Multisection image copies reduce the image copy creation time for large data files, in particular benefiting Exadata environments. This can also reduce completion time for non-backup use cases. For example, copying a file as part of transportable tablespace procedure or creating a clone with active database duplication.

Each section can be processed independently, either serially or in parallel. Backing up a file into separate sections can improve the performance of the backup operation, and it also allows large file backups to be restarted.

Requirements and Restrictions for Multisection Backups

- To create multisection backups for image copies and incremental backups, the COMPATIBLE parameter must be 12.0 or higher.
- You can create multisection incremental backups only for data files, not for other database files, such as control files or password files.
- You should not apply large values of parallelism to back up a large file that resides on a small number of disks, because that would defeat the purpose of the parallel operation. Multiple simultaneous accesses to the same disk device would be competing with each other.

Creating RMAN Multisection Backups

RMAN command syntax:

```
BACKUP <options> SECTION SIZE <integer> [K | M | G]
```

```
VALIDATE DATAFILE <options> SECTION SIZE <integer> [K | M | G]
```

Example:

```
RMAN> BACKUP DATAFILE 5 SECTION SIZE = 25M TAG 'section25mb';
backing up blocks 1 through 3200
piece handle=/u01/.../o1_mf_nnndf_SECTION25MB_382dryt4_.bkp
tag=SECTION25MB comment=NONE
...
backing up blocks 9601 through 12800
piece handle=/u01/.../o1_mf_nnndf_SECTION25MB_382dsto8_.bkp
tag=SECTION25MB comment=NONE
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The BACKUP and VALIDATE DATAFILE commands accept the following option:

SECTION SIZE <integer> [K | M | G]

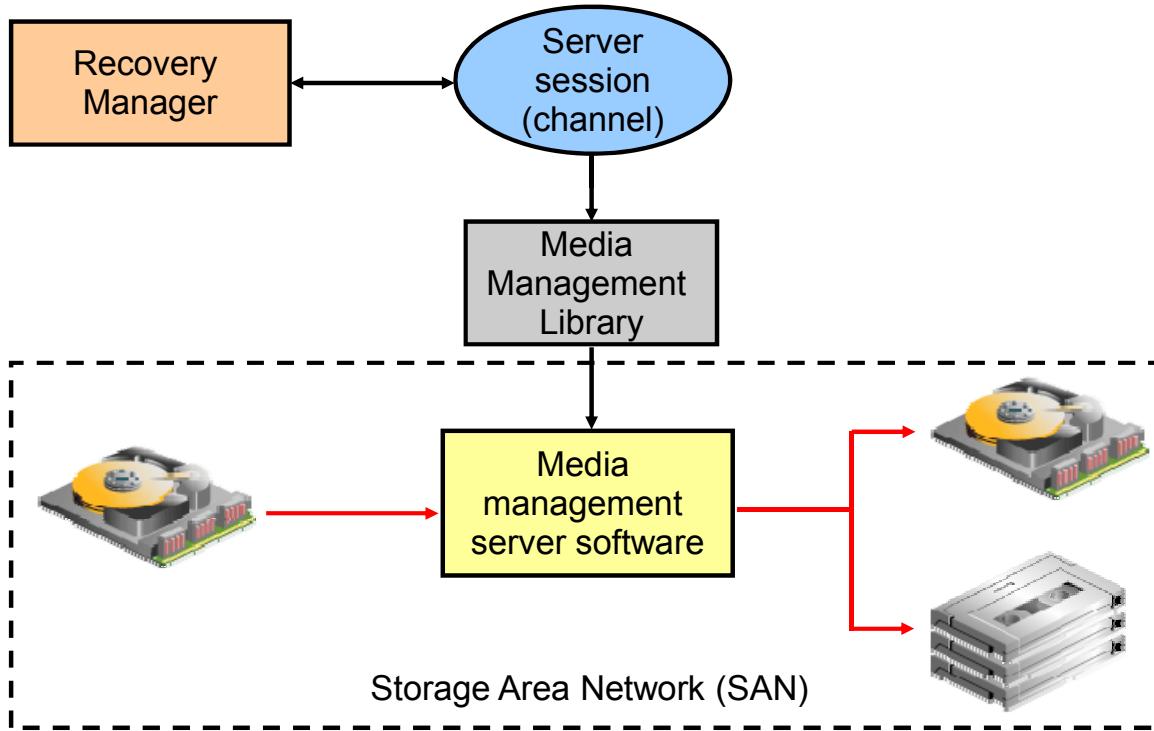
Use this to specify your planned size for each backup section. The option is both a backup command and backup spec-level option, so that you can apply different section sizes to different files in the same backup job.

In the example in the slide, a backup of data file 5 is being taken, and the section size is specified as 25 MB. The data file is 100 MB in size, so four sections are created. Note that, as indicated by the block ranges, block contiguity is maintained as they are written to the section files.

Viewing Metadata About Your Multisection Backup

- The V\$BACKUP_SET and RC_BACKUP_SET views have a MULTI_SECTION column, which indicates whether this is a multisection backup or not.
- The V\$BACKUP_DATAFILE and RC_BACKUP_DATAFILE views have a SECTION_SIZE column, which specifies the number of blocks in each section of a multisection backup. Zero means a whole-file backup.

Creating Proxy Copies



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Use the `PROXY` option of the RMAN `BACKUP` command to request an MML to perform the copy of the files.

Syntax:

```
BACKUP [AS BACKUPSET] ... PROXY [ONLY] DATABASE | TABLESPACE . . .
```

The `PROXY ONLY` option is useful for those media managers and storage networks where having the backup done by proxy may substantially reduce the storage net traffic.

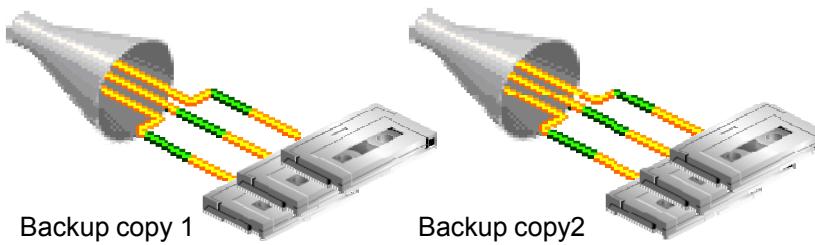
Some media management products can completely manage all data movement between Oracle data files and the backup devices. Some products that use high-speed connections between storage and media subsystems can reduce much of the backup load from the primary database server. This is beneficial in that the copying takes place across the SAN instead of the LAN. At that point, RMAN is no longer involved in the operation, except for communicating status across the LAN to and from the MML.

You can use the RMAN `LIST BACKUP` command to view information about the proxy copy. The `V$PROXY_DATAFILE` view contains descriptions of data file and control file proxy copy backups.

Creating Duplexed Backup Sets by Using BACKUP COPIES

Example of creating two copies of the backup set on tape:

```
RMAN> BACKUP AS BACKUPSET DEVICE TYPE sbt
2> COPIES 2
3> INCREMENTAL LEVEL 0
4> DATABASE;
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can use the BACKUP command with the COPIES option to override other COPIES or DUPLEX settings to create duplexed backup sets.

To duplex a backup with BACKUP COPIES, perform the following steps:

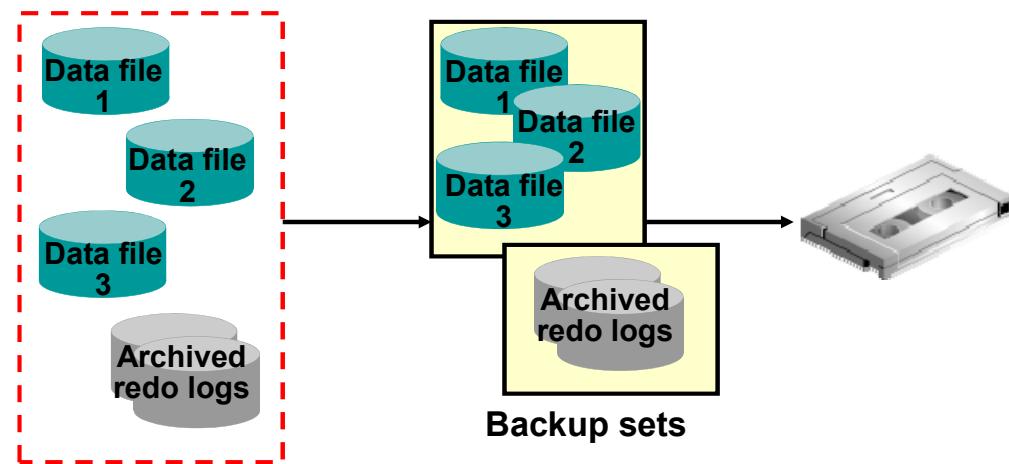
1. Specify the number of identical copies with the COPIES option of the BACKUP command.
2. Issue a LIST BACKUP command to verify your backup.

Duplexed backups require additional resources. For example, if a non-duplexed backup to tape uses three RMAN channels (each requiring one tape drive), then duplexing this backup (making two copies) requires six tape drives (as shown in the slide).

Underscore and number (_1, _2, and so on) are appended to the backup piece name to denote it as a duplexed copy.

Creating Backups of Backup Sets

```
RMAN> BACKUP DEVICE TYPE DISK AS BACKUPSET  
2> DATABASE PLUS ARCHIVELOG;  
RMAN> BACKUP DEVICE TYPE sbt BACKUPSET ALL;
```



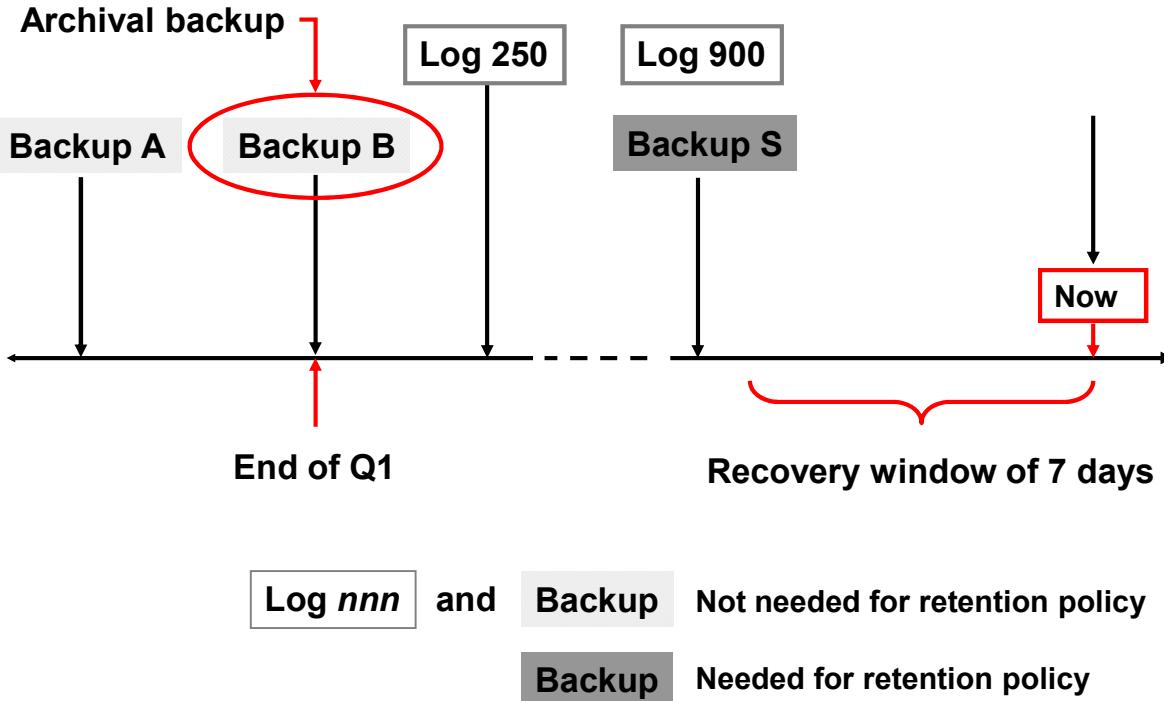
ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Use the RMAN BACKUP BACKUPSET command to back up previously created backup sets. Only backup sets that were created on device type DISK can be backed up using RMAN. The backup sets can be backed up to any available device type.

The BACKUP BACKUPSET command uses the default disk channel to copy backup sets from disk to disk. To back up from disk to tape, you must either configure or manually allocate a nondisk channel.

Archival Backups: Concepts



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

If you need to preserve an online backup for a specified amount of time, RMAN normally assumes you might want to perform point-in-time recovery for any time since that backup to the present. To satisfy this scenario, RMAN keeps the archived logs for that time period. However, you may have a requirement to simply keep the specific backup (and what is necessary to keep it consistent and recoverable) for a specified amount of time—for example, for two years. You do not intend to recover to a point in time since that backup, but you just want to be able to recover to the exact time of the backup, and no later. You also want to maintain a retention policy that keeps your backup area free of clutter, so making it reach back two years is not acceptable. This is a common need, when meeting business or legal requirements for data retention.

An archival backup solves this problem. If you mark a backup as an archival backup, that attribute overrides any configured retention policy for the purpose of this backup. You can retain archival backups such that they are either considered obsolete only after a specific time that you specify, or never considered obsolete. If you want to specify the latter, you need to use a recovery catalog.

The `KEEP` clause creates an archival backup that is a snapshot of the database at a point in time.

The only redo logs that are kept are those required to restore this backup to a consistent state. The `RESTORE POINT` clause issued after the backup is completed determines the number of redo logs that are kept (enough to restore the backup to the `RESTORE POINT` time).

An archival backup also guarantees that all of the files needed to restore the backup are included. RMAN includes the data files, SPFILE, archived log files (only those needed to recover an online backup), and the relevant autobackup files. All these files must go to the same media family (or group of tapes).

You can also specify a restore point to be created, which has the same SCN as the archival backup. That essentially gives a meaningful name to the point of time that the backup was made.

After an archival backup is created, it is retained for as long as specified. Even if you have a much smaller retention window and run the `DELETE OBSOLETE` command, the archival backup remains.

This backup is a snapshot of the database at a point in time, and can be used to restore the database to another host, for testing purposes, for example.

Note: Archival backups cannot be written to the fast recovery area. So if you have one, you must provide a `FORMAT` clause to specify a different location.

Creating Archival Backups with RMAN

- Specifying the KEEP clause when the database is online includes both data file and archive log backup sets:

```
KEEP {FOREVER | UNTIL TIME [=] 'date_string'}
NOKEEP
[RESTORE POINT rsname]
```

- List all restore points known to the RMAN repository:

```
LIST RESTORE POINT ALL;
```

- Display a specific restore point:

```
LIST RESTORE POINT 'rsname';
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Use the following syntax to create an archival backup using RMAN:

```
BACKUP ... KEEP {FOREVER|UNTIL TIME 'SYSDATE + <n>' } RESTORE POINT
<restore_point_name>
```

The UNTIL TIME clause enables you to specify when the archival backup is no longer immune to the retention policy. You can optionally specify FOREVER, meaning that the backup is an archival backup until you take some other action to change that.

Optionally, use the RESTORE POINT clause to specify the name of a restore point to be associated with this backup. The RESTORE POINT clause creates a “consistency” point in the control file. It assigns a name to a specific SCN. The SCN is captured just after the data file backup completes. The archival backup can be restored and recovered for this point in time, enabling the database to be opened. In contrast, the UNTIL TIME clause specifies the date until which the backup must be kept.

Managing Archival Database Backups

1 Archiving a database backup:

```
RMAN> CONNECT TARGET /
RMAN> CONNECT CATALOG rman/rman@catdb
RMAN> CHANGE BACKUP TAG 'consistent_db_bkup'
2> KEEP FOREVER;
```

2 Changing the status of a database copy:

```
RMAN> CHANGE COPY OF DATABASE CONTROLFILE NOKEEP;
```

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The CHANGE command changes the exemption status of a backup or copy in relation to the configured retention policy. For example, you can specify CHANGE . . . NOKEEP, to make a backup that is currently exempt from the retention policy eligible for the OBSOLETE status.

The first example changes a consistent backup into an archival backup, which you plan to store off-site. Because the database is consistent and, therefore, requires no recovery, you do not need to save archived redo logs with the backup.

The second example specifies that any long-term image copies of data files and control files should lose their exempt status and so become eligible to be obsolete according to the existing retention policy. This statement essentially removes the archival attribute from those backup files. If you do not specify a tag, as in this case, then the CHANGE execution applies to all backups of the type specified. You should specify a tag to change only the backup files you intend to change.

Note: The RESTORE POINT option is not valid with the CHANGE command, because there is no way to create the restore point for a time that has already passed (when the backup was created).

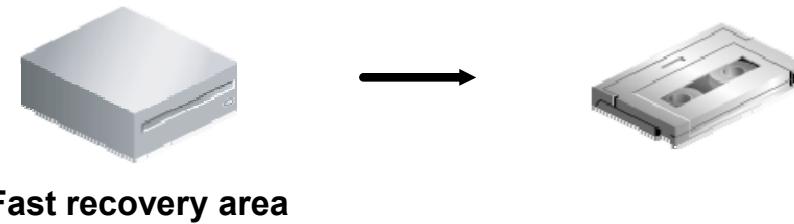
Backing Up Recovery Files

- Back up only the files in the fast recovery area:

```
RMAN> BACKUP RECOVERY AREA
```

- Back up all recovery files:

```
RMAN> BACKUP RECOVERY FILES
```



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

There are two ways to back up recovery data. The BACKUP RECOVERY AREA command backs up all files that are found in the current and any previous fast recovery areas. The BACKUP RECOVERY FILES command backs up all recovery files, even if they are not in the FRA. You gain added protection from loss by using the latter, which would back up, for example, any copies of control files or data files that are not in the fast recovery area.

By default, backup optimization is in effect for these two commands, even if you have disabled it using the CONFIGURE command. This means that the only recovery files that this command backs up are those that are not already backed up. You can force all files to be backed up by using the FORCE option.

You cannot specify DEVICE TYPE DISK for either of these commands.

Note: RMAN backs up only database files: data files, control files, SPFILEs, archive log files, and backups of these files. Placing an operating system file in the fast recovery area causes it to be included with a backup of the recovery area.

Backing Up the Control File to a Trace File

- A control file trace backup contains the SQL statement required to re-create the control files in the event that all control files are lost.
- It is recommended to do after each change in the physical structure of the database.
- Control file trace backups may be used to recover from loss of all control files.
- Choose your DBA tool: EM Express, Cloud Control, or command line.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Trace files are copies of the control files. A control file trace backup contains the SQL statement required to re-create the control files in the event that all control files are lost.

Although it is very unlikely that a properly configured database (with multiple copies of the control file placed on separate disks and separate controllers) would lose all control files at the same time, it is possible. Therefore, you should back up the control file to a trace file after each change to the physical structure of the database (adding tablespaces or data files).

You can perform the task in several DBA tools: EM Express, Cloud Control, or command line:

```
SQL> ALTER DATABASE BACKUP CONTROLFILE TO TRACE;
```

The trace backup is created in the location specified by the `DIAGNOSTIC_DEST` initialization parameter.

Cataloging Additional Backup Files

Using the CATALOG command:

- To catalog existing backup files that are no longer listed in the control file for RMAN restore operation
- To add the following file types to the recovery catalog:
 - CONTROLFILECOPY: Control file copies
 - DATAFILECOPY: Data file copies
 - BACKUPPIECE: Backup pieces
 - ARCHIVELOG: Archived redo log files
- With the START WITH option:

```
RMAN> CATALOG ARCHIVELOG '/disk1/arch_logs/archive1_731.log',
'/disk1/arch_logs/archive1_732.log';
RMAN> CATALOG START WITH '/tmp/arch_logs/';
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

If you have additional control file copies, data file copies, backup pieces, or archived redo log files on disk, you can catalog them in the recovery catalog by using the CATALOG command. If backups have aged out of the control file, you can catalog them so that RMAN can use them during a restore operation. Example to catalog all files in the currently enabled fast recovery area:

```
RMAN> CATALOG RECOVERY AREA NOPROMPT;
```

Use the START WITH option to catalog all files found in the specified directory tree. Provide a prefix that indicates the directory and possibly a file prefix to look for. You cannot use wildcards; this is only a prefix.

The second example in the slide catalogs all types of backup files that are found in the /tmp/arch_logs directory.

Suppose you want to be sure to catalog only those files in the /tmp directory whose file names start with the bset string. The following accomplishes that:

```
RMAN> CATALOG START WITH '/tmp/bset';
```

This command also catalogs any backup files that are found in directory trees that begin with /tmp/bset.

The CATALOG command can be used without being connected to a recovery catalog.

Backing Up ASM Disk Group Metadata

- Use the ASMCMD `md_backup` command to create a backup file containing metadata for one or more disk groups
- In the event of a loss of the ASM disk group, the backup file is used to reconstruct the disk group and its metadata.
- Without the metadata backup file, the disk group must be manually re-created in the event of a loss.
- Backing up all mounted disk groups:

```
ASMCMD> md_backup /backup/asm_metadata
```

- Backing up the DATA disk group:

```
ASMCMD> md_backup /backup/asm_metadata -G data
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can use the ASMCMD `md_backup` command to create a backup file of ASM disk group metadata. This backup file can be used to reconstruct the ASM disk group and its metadata if the disk group is lost. Without this metadata backup file, you must manually re-create the ASM disk group in the event of a loss of the disk group.

As shown in the first example in the slide, you can use the `md_backup` command to back up the metadata for all mounted groups. By using the `-G` option, you can name specific disk groups to be backed up.

If you do not specify a full path for the backup file, it is saved in the current working directory.

Quiz

Multisection backups can be taken of image copies and backup sets for full and incremental backups.

- a. True
- b. False



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Answer: a

Quiz

Backups should be taken only of data files. Backups of non-data files do not add to a backup and recovery strategy.

- a. True
- b. False



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Answer: b

Summary

In this lesson, you should have learned how to:

- Compress backups
- Use a media manager
- Create multi-section backups of very large files
- Create proxy copies
- Create duplexed backup sets
- Create archival backups
- Back up other files:
 - Back up a control file to trace
 - Back up archived redo log files
 - Catalog additional backup files
 - Back up ASM metadata



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Practice Overview: Backing Up Additional Files

- Practice 7-1 covers the following topics:
 - Backing up the control file
 - Backing up the archived redo log files
 - Backing up the control file to a trace file
- Optional practice 7-2 covers the following topics:
 - Creating an Archival Backup
 - Viewing current backups in Cloud Control



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this practice, you create backups of important database files that are not part of the default backup set.

8

Using RMAN-Encrypted Backups

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Describe and create RMAN-encrypted backups
- Distinguish and use the following modes:
 - Transparent-mode encryption
 - Password-mode encryption
 - Dual-mode encryption



ORACLE®

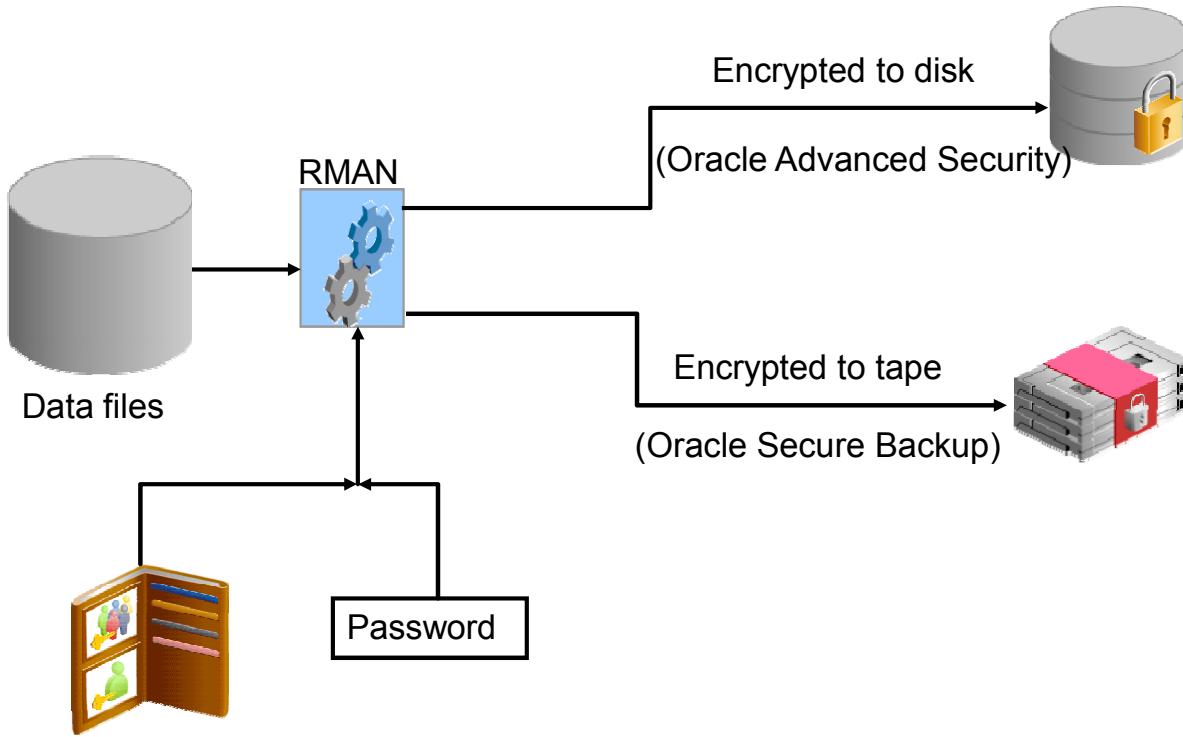
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

This is the last lesson in the “Backup Unit,” which includes:

- **Lesson 5:** Backup Strategies and Terminology
- **Lesson 6:** Performing Backups
- **Lesson 7:** Improving Your Backups
- **Lesson 8:** Using RMAN-Encrypted Backups

Note: More details about Transparent Data Encryption (TDE) are covered in the *Oracle Database 12c: Security* course and in the *Oracle Database Advanced Security Guide*.

RMAN-Encrypted Backups



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Recovery Manager (RMAN) can create encrypted backups to either tape or disk as long as the required key management infrastructure is available. RMAN encryption can use either a password-based key or a generated key held in the Oracle Wallet.

The data is encrypted by RMAN before it is transmitted to the disk or tape storage device, and no further encryption is performed.

RMAN backup encryption is available only in the Enterprise Edition of the database, and the COMPATIBLE parameter must be set to 10.2.0 or higher. Oracle Advanced Security is required for RMAN-encrypted backups.

If you are using Oracle Secure Backup (OSB) to provide encryption, then OSB manages the encryption keys. This is a different mechanism than the RMAN encryption.

Comparing OSB and RMAN Encryption

OSB Encryption		Via “Advanced Security Option”
For RMAN backup	For file-system data	Only for RMAN backup data
Global or host level	Global, host, backup, or volume level	Database or tablespace level
Data encryption on the client host		Data encryption within the database: no further encryption
OSB encryption keys: - Managed by OSB - Stored in host-specific encrypted key stores on administrative server		RMAN encryption keys: - Managed by database - Stored in database wallet
Encryption algorithms: AES128, AES192 (default), and AES256		Encryption algorithms up to 256-bit AES
Seamless decryption within domain		User-entered password for decryption

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Oracle Database backup encryption can be performed in one of two ways:

- OSB encryption:
 - For both: RMAN backup data (Oracle 9*i* and higher) and for file-system data
 - Oracle Secure Backup encrypts data on the client host. (Because there is no client software installation on NAS, NAS data cannot be encrypted by OSB.) While encryption occurs outside the database, the data is encrypted before transport over the network or before being written to a locally attached tape device. For decryption within the same domain, you do not have to provide a passphrase.
 - Embedded SSL technology provides secure transport of backup data and messages between two-way authenticated servers.
- RMAN encryption (available with “Advanced Security Option”):
 - RMAN can encrypt backups of an Oracle database on the database or tablespace level. RMAN encrypts the backup data within the database. This is generally faster than the OSB encryption. The RMAN encryption keys are stored in keystores and are managed by the database.
 - RMAN encrypted backups require the Advanced Security Option.
- When OSB encounters RMAN encrypted backups, it does not perform any additional encryption.

Creating RMAN-Encrypted Backups

RMAN supports three encryption modes:

- Transparent mode:
 - Uses a transparent data encryption (TDE) key
 - Requires that you first configure a keystore
- Password mode: Requires the use of the SET ENCRYPTION ON IDENTIFIED BY password ONLY command in your RMAN scripts
- Dual mode: Requires the use of the SET ENCRYPTION ON IDENTIFIED BY password command in your RMAN scripts



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

For improved security, RMAN backups created as backup sets can be encrypted. You can see this information in the ENCRYPTED column of the V\$BACKUP是一件 view.

Image copy backups cannot be encrypted.

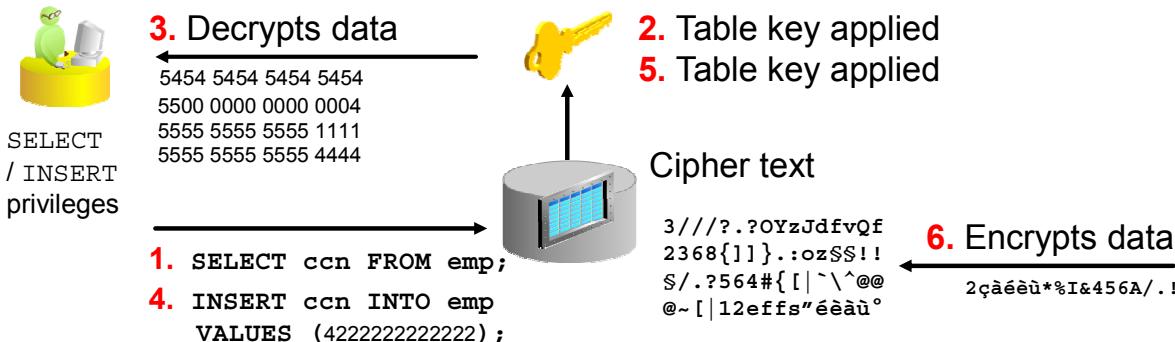
Encrypted backups are decrypted automatically during restore and recover operations, as long as the required decryption keys are available, by means of either a user-supplied password or the TDE key. The TDE key is stored in a keystore which is a password-protected container, in earlier releases known as a wallet.

RMAN supports three encryption modes:

- Transparent mode
- Password mode
- Dual mode

Additional information about each mode follows.

What Is TDE?



- Encrypts data in:
 - Data files (tablespaces, columns, indexes)
 - Redo log and archive log files
 - Memory (only for column encryption)
 - File backups
- Manages keys automatically
- Does not require changes to the application

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Transparent Data Encryption (TDE) is available with Oracle Advanced Security and provides easy-to-use protection for your data without requiring changes to your applications. TDE allows you to encrypt sensitive data in individual columns or entire tablespaces without having to manage encryption keys. TDE does not affect access controls, which are configured using database roles, secure application roles, system and object privileges, views, Virtual Private Database (VPD), Oracle Database Vault, and Oracle Label Security. Any application or user that previously had access to a table will still have access to an identical encrypted table.

TDE is designed to protect data in storage, but does not replace proper access control.

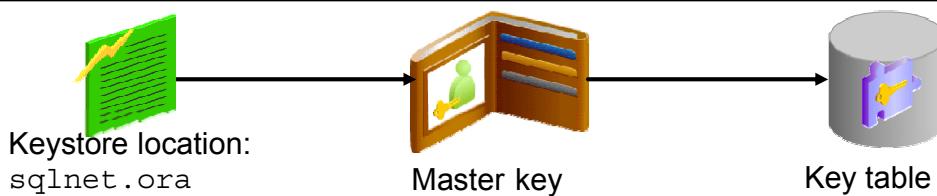
TDE is transparent to existing applications. Encryption and decryption occurs at different levels depending on whether it is tablespace or column level, but in either case, encrypted values are not displayed and are not handled by the application. For example, with TDE, applications designed to display a 16-digit credit card number do not have to be recoded to handle an encrypted string that may have many more characters.

TDE eliminates the ability of anyone who has direct access to the data files to gain access to the data by circumventing the database access control mechanisms. Even users with access to the data file at the operating system level cannot see the data unencrypted. TDE stores the master key outside the database in an external security module, thereby minimizing the possibility of both personally identifiable information (PII) and encryption keys being compromised. TDE decrypts the data only after database access mechanisms have been satisfied.

Using Transparent-Mode Encryption

1. Create a **directory** for the keystore.
2. Specify the keystore location in `sqlnet.ora`.

```
ENCRYPTION_WALLET_LOCATION =
(SOURCE =
(METHOD = FILE)
(METHOD_DATA =
(DIRECTORY =
/u01/app/oracle/admin/orcl/wallet)))
```



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

For the database to use TDE, a keystore must exist. TDE creates a key for each table that uses encrypted columns and each encrypted tablespace. The table key is stored in the data dictionary and the tablespace keys are stored in the tablespace data files. Both tablespace and table keys are encrypted with a master key. There is one master key for the database.

Use the following procedure to create a software keystore and a master key.

1. Create a directory to hold the keystore, which is accessible to the Oracle software owner.
2. Specify the location of the keystore file used to store the encryption master key by adding an entry in the `$ORACLE_HOME/network/admin/sqlnet.ora` file as shown in the example of the slide.

Note: The entry is sensitive to indents and spaces.

Using Transparent-Mode Encryption

3. Log in with the SYSKM (or SYSDBA) privilege.
4. Create the software keystore file.
5. Open the software keystore file.
6. Create the master encryption key.

```
SQL> CONNECT / AS SYSKM
SQL> ADMINISTER KEY MANAGEMENT CREATE KEYSTORE
      '/u01/app/oracle/admin/orcl/wallet'
      IDENTIFIED BY keystore_password;
```

```
SQL> ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN
      IDENTIFIED BY keystore_password;
```

```
SQL> ADMINISTER KEY MANAGEMENT SET KEY
      IDENTIFIED BY keystore_password
      WITH BACKUP USING 'for_12c' ;
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

3. Log in to the database instance as a user who is granted the ADMINISTER KEY MANAGEMENT or SYSKM privilege.
4. Create the software keystore file.
5. Open the software keystore file.
6. Create the master encryption key, which is stored encrypted in the keystore file. Use the WITH BACKUP clause to back up the keystore. You must use this option for password-based keystores. Optionally, you can use the USING clause to add a brief description of the backup. This identifier is appended to the named keystore file.

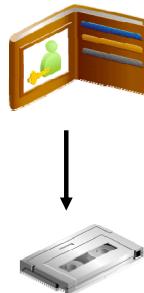
In a normal operation, you need to regenerate the master key only if it has been compromised. Changing the master periodically may be required by regulation. Regenerating the master key does not cause the data to be re-encrypted. The master key is used to encrypt table keys, used for column encryption, and tablespace keys. The table keys are used to encrypt column data. Tablespace keys are used to encrypt tablespace blocks. Changing the master key will cause the table and tablespace keys to be re-encrypted, which is a relatively quick operation, but the column data and the tablespace blocks are not re-encrypted.

All past master keys are held in the keystore, and the prior keys are available if the old data is recovered from a backup or if the database is recovered to a point in time before the key was regenerated.

Backing Up the Keystore

Backing up the current keystore:

```
SQL> ADMINISTER KEY MANAGEMENT BACKUP KEYSTORE  
IDENTIFIED BY keystore_password;
```



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The master keys are required to access encrypted data and you must protect these keys with backups. Because master keys reside in a keystore, the keystore should be periodically backed up in a secure location along with the database data files. You must back up a copy of the encryption keystore whenever a new master key is set.

If you lose the keystore that stores the master key, you can restore access to encrypted data by copying the backed-up version of the keystore to the appropriate location. If the restored keystore was archived after the last time the master key was reset, no additional action needs to be taken.

If the restored keystore does not contain the most recent master key, you can recover old data up to the point when the master key was reset by rolling back the state of the database to that point in time. All modifications to encrypted columns after the master key was reset are lost.

There can be two keystores present whenever the keystore is open: the password-based keystore identified with the .p12 extension, and an auto-login .cwallet.sso keystore. The auto-login keystore is changed every time the keystore is opened, so it is not useful to include it in backups. The password-based keystore holds current and past master keys. It must be included in backups.

There are separate and distinct keystores used for Recovery Manager (RMAN) and Oracle Secure Backup (OSB) encryption.

Configuring RMAN Encryption

1. Configure the RMAN encryption level (database, tablespace, or database excluding tablespaces):

```
RMAN> CONFIGURE ENCRYPTION FOR DATABASE ON
```

```
RMAN> CONFIGURE ENCRYPTION FOR TABLESPACE  
<tablespace_name> ON
```

2. Set the encryption algorithm, if needed:

```
RMAN> SET ENCRYPTION ALGORITHM 'algorithm name'
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

1. Configure the RMAN encryption level. The `CONFIGURE ENCRYPTION` command is used to specify encryption settings for the database or tablespaces within the database, which apply unless overridden, using the `SET` command. Options specified for an individual tablespace take precedence over options specified for the whole database.
2. Set an encryption algorithm, if needed. Query `V$RMAN_ENCRYPTION_ALGORITHMS` to obtain a list of encryption algorithms supported by RMAN. The default encryption algorithm is 128-bit AES.

Using Password-Mode Encryption

- Use case: Secure transport of backups to remote location
- Enable password mode encryption:
 - Only for the duration of an RMAN session

```
SET ENCRYPTION ON IDENTIFIED BY password ONLY
```

- Must provide password for backup creation
- Must provide password for backup restoration



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

When you use password encryption, you must provide a password when you create and restore encrypted backups. When you restore the password-encrypted backup, you must supply the same password that was used to create the backup. Password encryption is most appropriate for backups that will be restored at remote locations, but that must remain secure in transit.

Use the `SET ENCRYPTION ON IDENTIFIED BY password ONLY` command in your RMAN scripts to enable password encryption. Password encryption cannot be persistently configured.

The Enterprise Manager interface places the proper command in the RMAN backup scripts that it generates.

Note: For security reasons, it is not possible to permanently modify your existing backup environment so that RMAN backups are encrypted by using password mode. You can enable only password-encrypted backups for the duration of an RMAN session.

Using Dual-Mode Encryption

- Dual-mode encrypted backups can be restored transparently or by specifying a password.
- Enable password mode encryption in your RMAN session:

```
SET ENCRYPTION ON IDENTIFIED BY password
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Dual-mode encrypted backups can be restored transparently or by specifying a password. Dual-mode encrypted backups are useful when you create backups that are normally restored using the Oracle Encryption Wallet, but which occasionally need to be restored where the Oracle Encryption Wallet is not available.

To create dual-mode encrypted backup sets, specify the `SET ENCRYPTION ON IDENTIFIED BY password` command in your RMAN scripts.

RMAN-Encrypted Backups: Considerations

- Image copy backups cannot be encrypted.
- V\$RMAN_ENCRYPTION_ALGORITHMS contains the list of possible encryption algorithms.

```
RMAN> CONFIGURE ENCRYPTION ALGORITHM 'algorithmname'
```

```
RMAN> SET ENCRYPTION ALGORITHM 'algorithmname'
```

- One new encryption key is used for each new encrypted backup.
- You can increase disk performance by using multiple channels.
- You can change the master key any time without affecting your transparent encrypted backups.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

- Any RMAN backups created as backup sets can be encrypted. However, image copy backups cannot be encrypted.
- The V\$RMAN_ENCRYPTION_ALGORITHMS view contains a list of encryption algorithms supported by RMAN. If no encryption algorithm is specified, the default encryption algorithm is 128-bit AES. You can change the algorithm by using the commands shown in the slide.
- The Oracle Database server uses a new encryption key for every encrypted backup. The backup encryption key is then encrypted with either the password or the database master key, or both, depending on the chosen encryption mode. Individual backup encryption keys or passwords are never stored in clear text.
- Encryption can have a negative effect upon disk backup performance. Because encrypted backups use more CPU resource than nonencrypted backups, you can improve the performance of encrypted backups to disks by using more RMAN channels.

Restoring Encrypted Backups

- Before restoration, set the RMAN session to decrypt backups.
- Specify all required passwords with the SET DECRYPTION command when restoring from a set of backups that were created with different passwords.

```
SET DECRYPTION IDENTIFIED BY '<password_1>'  
{, '<password_2>', ..., '<password_n>' }
```

Notes

- If you lose the password for a password-encrypted backup, you cannot restore that backup.
- If you lose the keystore containing the key for a transparent encrypted backup, you cannot restore that backup.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Use the SET DECRYPTION command to specify one or more decryption passwords to be used when reading dual-mode or password-encrypted backups. When RMAN reads encrypted backup pieces, it tries each password in the list until it finds the correct one to decrypt that backup piece. An error is signaled if none of the specified keys are correct.

If you lose the password for a password-encrypted backup, you cannot restore that backup.

Because the Oracle key management infrastructure archives all previous master keys in the keystore (or wallet), changing or resetting the current database master key does not affect your ability to restore encrypted backups performed using an older master key. You may reset the database master key at any time, but RMAN will always be able to restore all encrypted backups that were ever created by this database.

If you lose the keystore containing the key for a transparent encrypted backup, you cannot restore that backup. Because the keystore contains all past backup encryption keys, a restored keystore can be used to restore past encrypted backups up to the backup time of the wallet. But encrypted backups made after the keystore backup will be not accessible.

Best Practice Tip: Back up the keystore frequently.

Quiz

To which of the following can you create RMAN-encrypted backups:

- a. Disk
- b. Tape by using a third-party media manager
- c. Tape by using Oracle Secure Backups



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Answer: a, c

Summary

After completing this lesson, you should be able to:

- Describe and create RMAN-encrypted backups
- Distinguish and use the following modes:
 - Transparent-mode encryption
 - Password-mode encryption
 - Dual-mode encryption



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Practice Overview: Using RMAN-Encrypted Backups

- Practice 8-1 covers encrypting a backup with RMAN.
- Practice 8-2 covers restoring an encrypted backup.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

This practice uses password encryption, which is an appropriate technique for backups that will be restored at a remote location.

The second part is an optional challenge practice (because, most likely, the class has not yet covered the restore and recover operations). The challenge should be attempted only if there is extra time available.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and Error : You are not a Valid Partner use only

9

Diagnosing Failures

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Detect and repair database corruption
- Use the Automatic Diagnostic Repository
- Analyze instance recovery with ADRCI
- Find and interpret message output and error stacks
- Use the Data Recovery Advisor
- Proactively check and handle block corruption

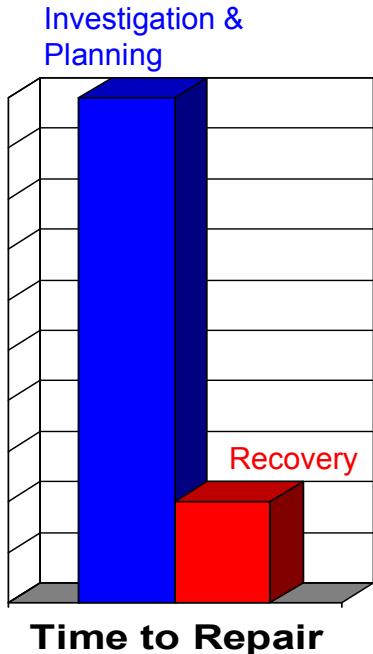


Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

This is the first lesson in the “Recovery Unit,” which includes:

- **Lesson 9:** Diagnosing Failures
- **Lesson 10:** Restore and Recovery Concepts
- **Lesson 11:** Performing Recovery I
- **Lesson 12:** Performing Recovery II

Reducing Problem Diagnosis Time



Oracle tools for data repair include:

- RMAN for physical media loss or corruptions
- Flashback for logical errors
- Data Guard for physical problems
- **Data Recovery Advisor** addresses:
 - Problem diagnosis (choosing the right solution can be error prone and time consuming)
 - Incorrect choices (errors more likely during emergencies)

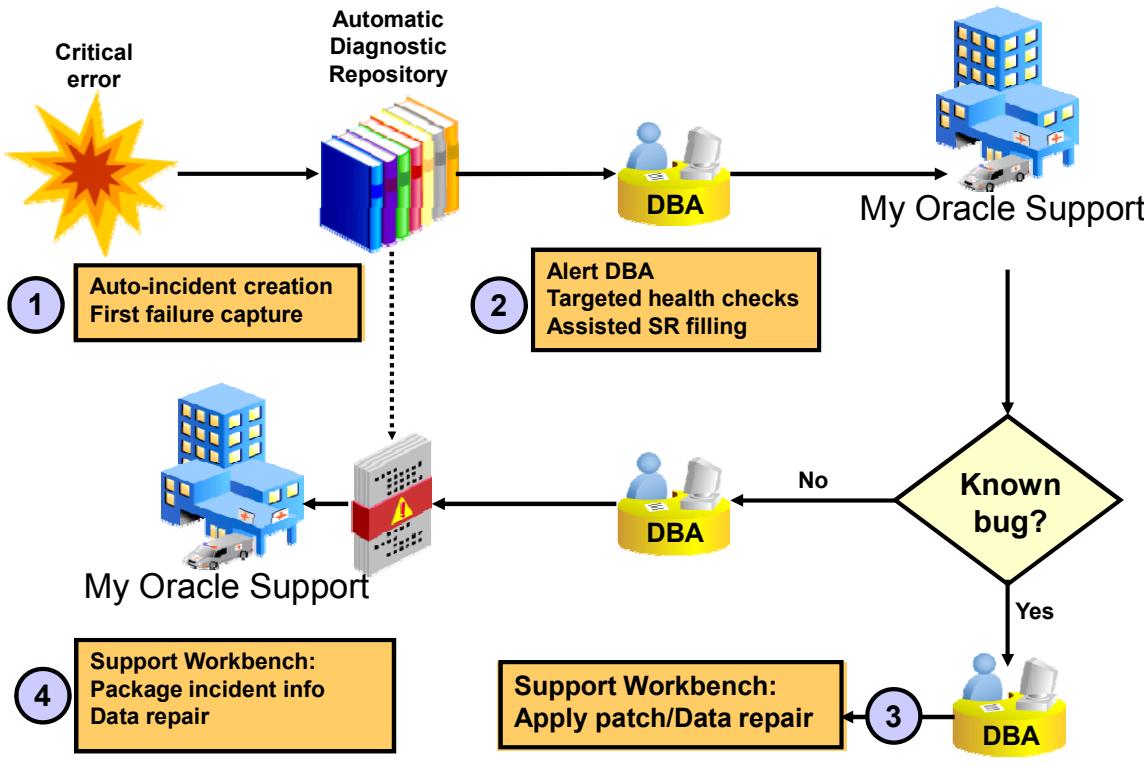


Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Research shows that database administrators spend a large percentage of “repair time” in investigating what, why, and how data has been compromised. They may need to analyze errors, alerts, and trace files. The Data Recovery Advisor provides intelligent database problem identification and may reduce overall system down time by eliminating or reducing the amount of time a database administrator spends researching a problem.

In addition, Oracle Data Recovery Advisor reduces uncertainty and confusion, which may often occur during an outage. The advisor uses a special repository, called Automatic Diagnostic Repository (ADR).

Automatic Diagnostic Workflow



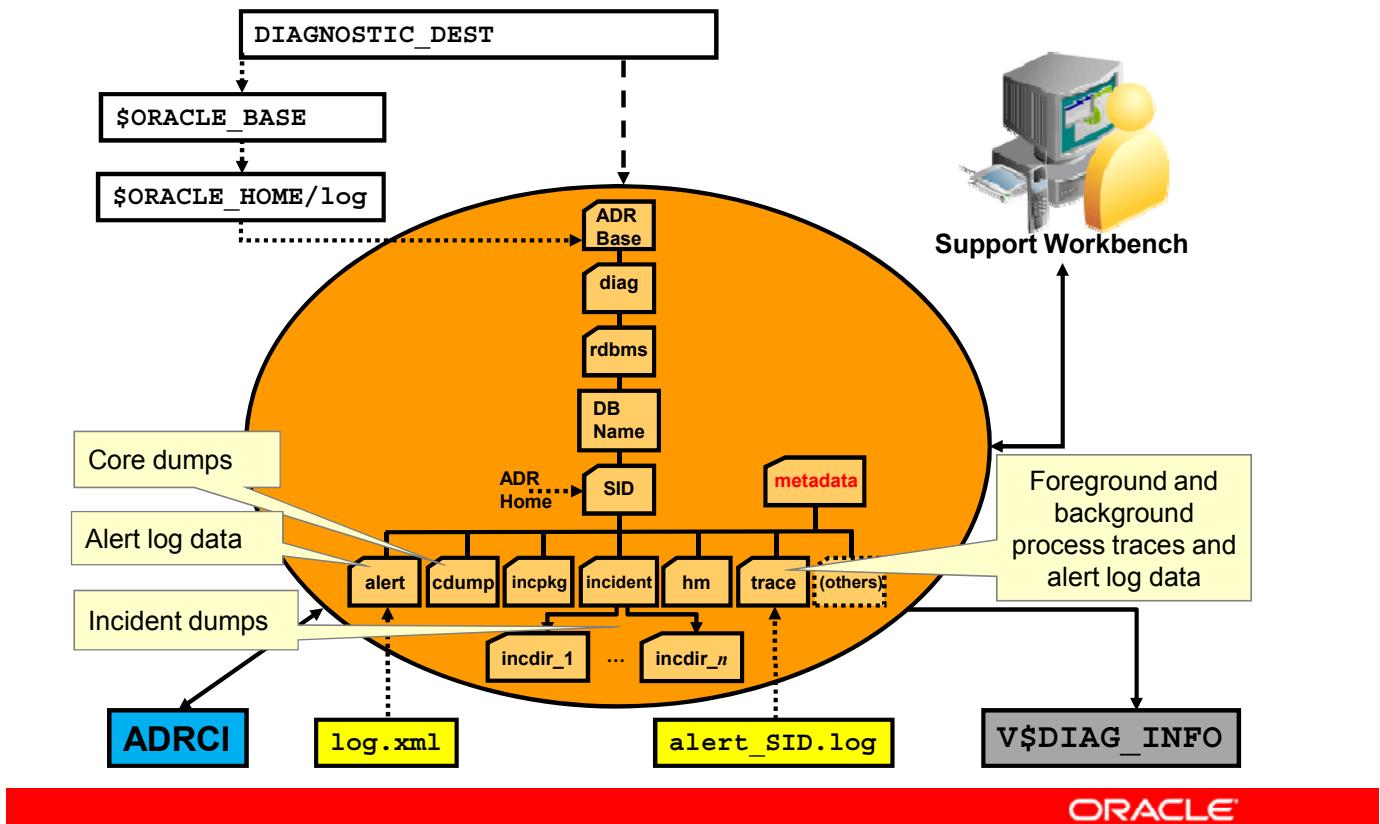
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

An always-on, in-memory tracing facility enables database components to capture diagnostic data upon first failure for critical errors. The ADR is automatically maintained to hold diagnostic information about critical error events. This information can be used by the Data Recovery Advisor and also to create incident packages for Oracle Support Services.

Here is a typical workflow for a diagnostic session including Oracle Support Services:

1. Incident causes an alert to be raised in Enterprise Manager (EM).
2. The DBA can view the alert via the Cloud Control Alert page.
3. The DBA can drill down to incident and problem details.
4. The DBA or Oracle Support Services can decide or ask for that information to be packaged and sent to Oracle Support Services via My Oracle Support. The DBA can add files to the data to be packaged automatically.

Automatic Diagnostic Repository



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The ADR is a file-based repository for database diagnostic data such as traces, incident dumps and packages, the alert log, Health Monitor reports, core dumps, and more. It has a unified directory structure across multiple instances and multiple products—stored outside of any database. It is, therefore, available for problem diagnosis when the database is down.

Beginning with Oracle Database 11g R1, the database, Automatic Storage Management (ASM), Cluster Ready Services (CRS), and other Oracle products or components store all diagnostic data in the ADR. Each instance of each product stores diagnostic data underneath its own ADR home directory. For example, in a Real Application Clusters environment with shared storage and ASM, each database instance and each ASM instance have a home directory within the ADR. ADR's unified directory structure, consistent diagnostic data formats across products and instances, and a unified set of tools enable customers and Oracle Support to correlate and analyze diagnostic data across multiple instances.

The ADR root directory is known as the ADR base. Its location is set by the `DIAGNOSTIC_DEST` initialization parameter. If this parameter is omitted or left null, the database sets `DIAGNOSTIC_DEST` upon startup as follows: If environment variable `ORACLE_BASE` is set, `DIAGNOSTIC_DEST` is set to `$ORACLE_BASE`. If environment variable `ORACLE_BASE` is not set, `DIAGNOSTIC_DEST` is set to `$ORACLE_HOME/log`.

ADR Command-Line Tool (ADRCI)

- ADRCI provides interaction with ADR from an operating system prompt.
- Using ADRCI, you can view diagnostic data within the ADR.

```
$ adrci
ADRCI: Release 12.1.0.1.0 - Production on Thu Nov 29 21:15:27 2012
Copyright (c) 1982, 2012, Oracle and/or its affiliates. All rights reserved.
ADR base = "/u01/app/oracle"          ADRCI> set editor gedit
ADRCI>                                     ADRCI> show alert
ADRCI> show incident
...
ADR Home = /u01/app/oracle/diag/rdbms/em12rep/em12rep:
*****
INCIDENT_ID PROBLEM_KEY      CREATE_TIME
-----
4985        ORA 4031          2012-11-21 00:57:43.823000 +00:00
5161        ORA 4031          2012-11-21 00:58:17.284000 +00:00
2 incident info records fetched
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

ADRCI is a command-line tool that is part of the database fault diagnosability infrastructure. ADRCI enables you to:

- View diagnostic data within the ADR
- Package incident and problem information into a ZIP file for transmission to Oracle Support

ADRCI can be used in interactive mode or within scripts. In addition, ADRCI can execute scripts of ADRCI commands in the same way that SQL*Plus executes scripts of SQL and PL/SQL commands. There is no need to log in to ADRCI, because the data in the ADR is not intended to be secure. ADR data is secured only by operating system permissions on the ADR directories.

The easiest way to package and otherwise manage diagnostic data is with the Support Workbench of Enterprise Manager (which assists with the resolution of database errors, as well as ASM errors).

ADRCI provides a command-line alternative to most of the functionality of Support Workbench, and adds capabilities such as listing and querying trace files. The example in the slide shows you an ADRCI session where you are listing all open incidents stored in ADR.

Note: For more information about ADRCI and the Support Workbench, refer to the *Oracle Database Utilities* guide.

V\$DIAG_INFO View

```
SQL> SELECT NAME, VALUE FROM V$DIAG_INFO;
```

NAME	VALUE
Diag Enabled	TRUE
ADR Base	/u01/app/oracle
ADR Home	/u01/app/oracle/diag/rdbms/orcl/orcl
Diag Trace	/u01/app/oracle/diag/rdbms/orcl/orcl/trace
Diag Alert	/u01/app/oracle/diag/rdbms/orcl/orcl/alert
Diag Incident	/u01/app/oracle/diag/rdbms/orcl/orcl/incident
Diag Cdmp	/u01/app/oracle/diag/rdbms/orcl/orcl/cdmp
Health Monitor	/u01/app/oracle/diag/rdbms/orcl/orcl/hm
Default Trace File	/u01/app/oracle/diag/.../trace/orcl_ora_11424.trc
Active Problem Count	3
Active Incident Count	8



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The V\$DIAG_INFO view lists all important ADR locations:

- **ADR Base:** Path of the ADR base
- **ADR Home:** Path of the ADR home for the current database instance
Note: This is a path name. There is no official environment variable called *ADR_HOME*.
- **Diag Trace:** Location of the text alert log and background/foreground process trace files
- **Diag Alert:** Location of an XML version of the alert log
- **Diag Incident:** Incident logs are written here.
- **Diag Cdmp:** Diagnostic core files are written to this directory.
- **Health Monitor:** Location of logs from Health Monitor runs
- **Default Trace File:** Path to the trace file for your session. SQL trace files are written here.

Interpreting RMAN Message Output

RMAN troubleshooting information can be found in:

- RMAN command output
- RMAN trace file
- Alert log
- Oracle server trace file
- `sbtio.log` file



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The RMAN command output contains actions that are relevant to the RMAN job as well as error messages that are generated by RMAN, the server, and the media vendor. RMAN error messages have an `RMAN-nnnn` prefix. The output is displayed to the terminal (standard output) but can be written to a file by defining the `LOG` option or by shell redirection.

The RMAN trace file contains the `DEBUG` output and is used only when the `TRACE` command option is used.

The alert log contains a chronological log of errors, nondefault initialization parameter settings, and administration operations. Because it records values for overwritten control file records, it can be useful for RMAN maintenance when operating without a recovery catalog.

The Oracle trace file contains detailed output that is generated by Oracle server processes. This file is created when an `ORA-600` or `ORA-3113` (following an `ORA-7445`) error message occurs, whenever RMAN cannot allocate a channel, and when the Media Management Library fails to load.

The `sbtio.log` file contains vendor-specific information that is written by the media management software and can be found in `USER_DUMP_DEST`. Note that this log does not contain Oracle server or RMAN errors.

DEBUG Option

- The DEBUG option is used to:
 - View the PL/SQL that is generated
 - Determine precisely where an RMAN command is hanging or faulting
- The DEBUG option is specified at the RMAN prompt or within a run block.
- The DEBUG option creates an enormous amount of output, so redirect the output to a trace file:

```
$ rman target / catalog rman/rman debug trace trace.log
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The DEBUG option displays all SQL statements that are executed during RMAN compilations and the results of these executions. Any information that is generated by the recovery catalog PL/SQL packages is also displayed. In the following example, the DEBUG output is written during the backup of data file 3, but not data file 4:

```
RMAN> run {
      debug on;
      allocate channel c1 type disk;
      backup datafile 3;
      debug off;
      backup datafile 4; }
```

Remember that the DEBUG output can be voluminous, so make sure that you have adequate disk space for the trace file. This simple backup session that does not generate any errors creates a trace file that is almost half a megabyte in size:

```
$ rman target / catalog rman/rman debug trace sample.log
RMAN> backup database;
RMAN> host "ls -l sample.log";
-rw-r--r--  1 user02    dba          576270 Apr  6 10:38 sample.log
host command complete
```

Interpreting RMAN Error Stacks

- Read the stack from bottom to top.
- Look for Additional information.
- RMAN-03009 identifies the failed command.

```
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03009: failure of backup command on c1 channel at
                  09/04/2012 13:18:19
ORA-19506: failed to create sequential file,
                  name="07d36ecp_1_1", parms=""
ORA-27007: failed to open file
SVR4 Error: 2: No such file or directory
Additional information: 7005
Additional information: 1
ORA-19511: Error from media manager layer,error text:
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Because of the amount of data that RMAN logs, you may find it difficult to identify the useful messages in the RMAN error stack. Note the following tips and suggestions:

- Because many of the messages in the error stack are not meaningful for troubleshooting, try to identify the one or two errors that are most important.
- Check for a line that says Additional information followed by an integer. This line indicates a media management error. The integer that follows refers to code that is explained in the text of the error message.
- Read the messages from bottom to top because this is the order in which RMAN issues the messages. The last one or two errors that are displayed in the stack are often informative.
- Look for the RMAN-03002 or RMAN-03009 message immediately following the banner. The RMAN-03009 is the same as RMAN-03002 but includes the channel ID. If the failure is related to an RMAN command, then these messages indicate which command failed. The syntax errors generate an RMAN-00558 error.

Data Recovery Advisor

- Fast detection, analysis, and repair of failures
- Minimizing disruptions for users
- Down-time and runtime failures
- User interfaces:
 - EM GUI interface (several paths)
 - RMAN command line
- Supported database configurations:
 - Single-instance
 - Not RAC
 - Supporting failover to standby, but not analysis and repair of standby databases



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The Data Recovery Advisor automatically gathers data failure information when an error is encountered. In addition, it can proactively check for failures. In this mode, it can potentially detect and analyze data failures before a database process discovers the corruption and signals an error. (Note that repairs are always under human control.)

Data failures can be very serious. For example, if your current log files are missing, you cannot start your database. Some data failures (such as block corruptions in data files) are not catastrophic, in that they do not take the database down or prevent you from starting the Oracle instance. The Data Recovery Advisor handles both cases: the one where you cannot start up the database (because some required database files are missing, inconsistent, or corrupted) and the one where file corruptions are discovered during run time.

User Interfaces

The Data Recovery Advisor is available from Enterprise Manager (EM) Cloud Control and in RMAN. When failures exist, there are several ways to access the Data Recovery Advisor. The following examples all begin on the Database Instance home page:

- Availability > Backup & Recovery > Perform Recovery > Advise and Recover
- Active Incidents link > on the Support Workbench “Problems” page: Checker Findings tabbed page > Launch Recovery Advisor
- Database Instance Health > click the specific link, for example, ORA 1578 in the Incidents section > Support Workbench, Problems Detail page > Data Recovery Advisor
- Database Instance Health > Related Links section: Support Workbench > Checker Findings tabbed page: Launch Recovery Advisor
- Related Link: Advisor Central > Advisors tabbed page: Data Recovery Advisor
- Related Link: Advisor Central > Checkers tabbed page: Details > Run Detail tabbed page: Launch Recovery Advisor

You can also use it via the RMAN command-line, for example:

```
rman target / nocatalog  
rman> list failure all;
```

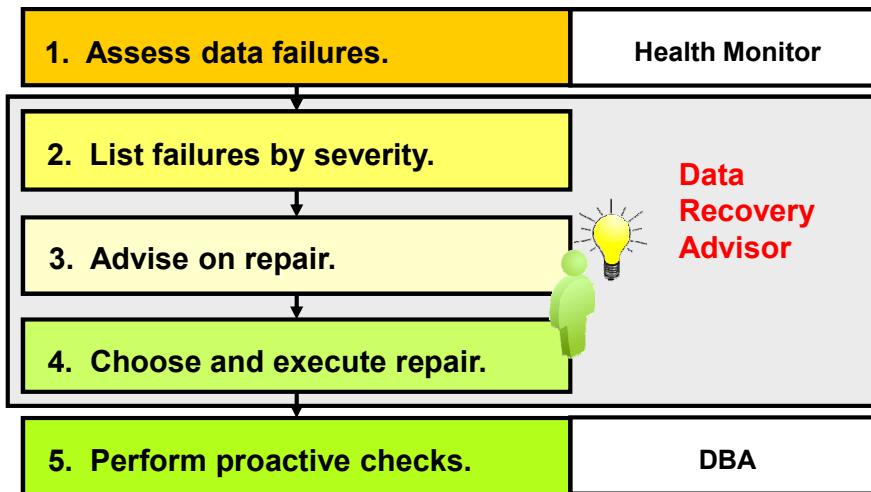
Supported Database Configurations

In the current release, the Data Recovery Advisor supports single-instance databases. Oracle Real Application Clusters (RAC) databases are not supported.

The Data Recovery Advisor cannot use blocks or files transferred from a standby database to repair failures on a primary database. Also, you cannot use the Data Recovery Advisor to diagnose and repair failures on a standby database. However, the Data Recovery Advisor does support failover to a standby database as a repair option (as mentioned earlier).

Data Recovery Advisor

Reducing down time by eliminating confusion:



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The automatic diagnostic workflow in Oracle Database 11g performs as follows. With the Data Recovery Advisor, you need to initiate only an advice and a repair.

1. The Health Monitor automatically executes checks and logs failures and their symptoms as “findings” into the ADR.
2. The Data Recovery Advisor consolidates findings into failures. It lists the results of previously executed assessments with failure severity (critical or high).
3. When you ask for repair advice on a failure, the Data Recovery Advisor maps failures to automatic and manual repair options, checks basic feasibility, and presents you with the repair advice.
4. You can choose to manually execute a repair or request the Data Recovery Advisor to do it for you.
5. In addition to the automatic, primarily “reactive” checks of the Health Monitor and Data Recovery Advisor, Oracle recommends to additionally use the VALIDATE command as a “proactive” check.

Data Failure: Examples

- Not accessible components, for example:
 - Missing data files at the OS level
 - Incorrect access permissions
 - Offline tablespace, and so on
- Physical corruptions, such as block checksum failures or invalid block header field values
- Logical corruptions, such as inconsistent dictionary, corrupt row piece, corrupt index entry, or corrupt transaction
- Inconsistencies, such as control file is older or newer than the data files and online redo logs
- I/O failures, such as a limit on the number of open files exceeded, channels inaccessible, network or I/O error



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Data failures are detected by checks, which are diagnostic procedures that assess the health of the database or its components. Each check can diagnose one or more failures, which are mapped to a repair.

Checks can be reactive or proactive. When an error occurs in the database, “reactive checks” are automatically executed. You can also initiate “proactive checks”, for example, by executing the `VALIDATE DATABASE` command.

In Cloud Control, navigate to the Perform Recovery page, if you find your database in a “down” or “mounted” state.

The Data Recovery Advisor can analyze failures and suggest repair options for issues, as outlined in the slide.

Data Recovery Advisor

RMAN Command-Line Interface

RMAN Command	Action
LIST FAILURE	Lists previously executed failure assessment
ADVISE FAILURE	Displays recommended repair option
REPAIR FAILURE	Repairs and closes failures (after ADVISE in the same RMAN session)
CHANGE FAILURE	Changes or closes one or more failures



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

If you suspect or know that a database failure has occurred, then use the LIST FAILURE command to obtain information about these failures. You can list all or a subset of failures and restrict output in various ways. Failures are uniquely identified by failure numbers. Note that these numbers are not consecutive, so gaps between failure numbers have no significance.

The ADVISE FAILURE command displays a recommended repair option for the specified failures. It prints a summary of the input failure and implicitly closes all open failures that are already fixed. The default behavior when no option is used is to advise on all the CRITICAL and HIGH priority failures that are recorded in ADR.

The REPAIR FAILURE command is used after an ADVISE FAILURE command **within the same RMAN session**. By default, the command uses the single, recommended repair option of the last ADVISE FAILURE execution in the current session. If none exists, the REPAIR FAILURE command initiates an implicit ADVISE FAILURE command. After completing the repair, the command closes the failure.

The CHANGE FAILURE command changes the failure priority or closes one or more failures. You can change a failure priority only for HIGH or LOW priorities. Open failures are closed implicitly when a failure is repaired. However, you can also explicitly close a failure.

Listing Data Failures

The RMAN LIST FAILURE command lists previously executed failure assessment:

- Including newly diagnosed failures
- Removing closed failures (by default)



Syntax:

```
LIST FAILURE
[ ALL | CRITICAL | HIGH | LOW | CLOSED |
  failnum[,failnum,...] ]
[ EXCLUDE FAILURE failnum[,failnum,...] ]
[ DETAIL ]
```

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The RMAN LIST FAILURE command lists failures. If the target instance uses a recovery catalog, it can be in STARTED mode, otherwise it must be in MOUNTED mode. The LIST FAILURE command does not initiate checks to diagnose new failures; rather, it lists the results of previously executed assessments. Repeatedly executing the LIST FAILURE command revalidates all existing failures. If the database diagnoses new ones (between command executions), they are displayed. If a user manually fixes failures, or if transient failures disappear, then the Data Recovery Advisor removes these failures from the LIST FAILURE output. The following is a description of the syntax:

- **failnum:** Number of the failure to display repair options for
- **ALL:** List failures of all priorities.
- **CRITICAL:** List failures of CRITICAL priority and OPEN status. These failures require immediate attention, because they make the whole database unavailable (for example, a missing control file).
- **HIGH:** List failures of HIGH priority and OPEN status. These failures make a database partly unavailable or unrecoverable; so they should be repaired quickly (for example, missing archived redo logs).
- **LOW:** List failures of LOW priority and OPEN status. Failures of a low priority can wait until more important failures are fixed.

- **CLOSED:** List only closed failures.
- **EXCLUDE FAILURE:** Exclude the specified list of failure numbers from the list.
- **DETAIL:** List failures by expanding the consolidated failure. For example, if there are multiple block corruptions in a file, the DETAIL option lists each one of them.

See the *Oracle Database Backup and Recovery Reference* for details of the command syntax.

Advising on Repair

The RMAN ADVISE FAILURE command:

- Displays a summary of input failure list
- Includes a warning, if new failures appeared in ADR
- Displays a manual checklist
- Lists a single recommended repair option
- Generates a repair script (for automatic or manual repair)

```
...
Repair script:
/u01/app/oracle/diag/rdbms/orcl/orcl/hm/reco_29791
28860.hm
RMAN>
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The RMAN ADVISE FAILURE command displays a recommended repair option for the specified failures. The ADVISE FAILURE command prints a summary of the input failure. The command implicitly closes all open failures that are already fixed.

The default behavior (when no option is used) is to advise on all the CRITICAL and HIGH priority failures that are recorded in ADR. If a new failure has been recorded in ADR since the last LIST FAILURE command, this command includes a WARNING before advising on all CRITICAL and HIGH failures.

Two general repair options are implemented: no-data-loss and data-loss repairs.

When the Data Recovery Advisor generates an automated repair option, it generates a script that shows you how RMAN plans to repair the failure. If you do not want the Data Recovery Advisor to automatically repair the failure, then you can use this script as a starting point for your manual repair. The operating system (OS) location of the script is printed at the end of the command output. You can examine this script, customize it (if needed), and also execute it manually if, for example, your audit trail requirements recommend such an action.

Syntax

ADVISE FAILURE

```
[ ALL | CRITICAL | HIGH | LOW | failnum[,failnum,...] ]
[ EXCLUDE FAILURE failnum [,failnum,...] ]
```

Executing Repairs

The RMAN REPAIR FAILURE command:

- Follows the ADVISE FAILURE command
- Repairs the specified failure
- Closes the repaired failure

Syntax:

```
REPAIR FAILURE  
[USING ADVISE OPTION integer]  
[ { NOPROMPT | PREVIEW}...]
```



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

This command should be used after an ADVISE FAILURE command in the same RMAN session. By default (with no option), the command uses the single, recommended repair option of the last ADVISE FAILURE execution in the current session. If none exists, the REPAIR FAILURE command initiates an implicit ADVISE FAILURE command.

With USING ADVISE OPTION *integer*, you specify your desired repair option by its option number (from the ADVISE FAILURE command); this is not the failure number.

By default, you are asked to confirm the command execution because you may be requesting substantial changes that take time to complete. During the execution of a repair, the output of the command indicates what phase of the repair is being executed.

After completing the repair, the command closes the failure.

You cannot run multiple concurrent repair sessions. However, concurrent REPAIR ... PREVIEW sessions are allowed.

- **PREVIEW means:** Do not execute the repairs; instead, display the previously generated RMAN script with all repair actions and comments.
- **NOPROMPT means:** Do not ask for confirmation.

Classifying (and Closing) Failures

The RMAN CHANGE FAILURE command:

- Changes the failure priority (except for CRITICAL)
- Closes one or more failures

Example:

```
RMAN> change failure 5 priority low;
List of Database Failures
=====
Failure ID Priority Status      Time Detected Summary
-----
5          HIGH    OPEN       20-DEC-06   one or more
           datafiles are missing
Do you really want to change the above failures (enter YES or
NO)? yes
changed 1 failures to LOW priority
```

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The CHANGE FAILURE command is used to change the failure priority or close one or more failures.

Syntax

```
CHANGE FAILURE
{ ALL | CRITICAL | HIGH | LOW | failnum[,failnum,...] }
[ EXCLUDE FAILURE failnum[,failnum,...] ]
{ PRIORITY {CRITICAL | HIGH | LOW} |
CLOSE } – Change status of the failure(s) to closed.
[ NOPROMPT ] – Do not ask user for a confirmation.
```

A failure priority can be changed only from HIGH to LOW and from LOW to HIGH. It is an error to change the priority level of CRITICAL. (One reason why you may want to change a failure from HIGH to LOW is to avoid seeing it on the default output list of the LIST FAILURE command. For example, if a block corruption has HIGH priority, you may want to temporarily change it to LOW if the block is in a little-used tablespace.)

Open failures are closed implicitly when a failure is repaired. However, you can also explicitly close a failure. This involves a re-evaluation of all other open failures, because some of them may become irrelevant as the result of the closure of the failure.

By default, the command asks the user to confirm a requested change.

Data Recovery Advisor Views

Querying V\$ views:

- V\$IR_FAILURE: List of all failures, including closed ones (result of the LIST FAILURE command)
- V\$IR_MANUAL_CHECKLIST: List of manual advice (result of the ADVISE FAILURE command)
- V\$IR_REPAIR: List of repairs (result of the ADVISE FAILURE command)
- V\$IR_FAILURE_SET: Cross-reference of failure and advice identifiers



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

See the *Oracle Database Reference* for details of the dynamic views that the Data Recovery Advisor uses.

Quiz

The Data Recovery Advisor handles both cases: when you cannot start up the database (because some required database files are missing, inconsistent, or corrupted) and when file corruptions are discovered during run time.

- a. True
- b. False



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Answer: a

Quiz

After executing the ADVISE FAILURE command, the repair is automatically executed. So, it is no longer under your control.

- a. True
- b. False



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Answer: b

Quiz

The ADR resides in the database. Therefore, an instance must be mounted for incident analysis.

- a. True
- b. False



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Answer: b

What Is Block Corruption?

- Whenever a block is read or written, a consistency check is performed.
 - Block version
 - DBA (data block address) value in cache as compared to the DBA value in the block buffer
 - Block-checksum, if enabled
- A corrupt block is identified as being one of the following:
 - Media corrupt
 - Logically (or software) corrupt



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

A corrupted data block is a block that is not in a recognized Oracle format, or whose contents are not internally consistent. Typically, corruptions are caused by faulty hardware or operating system problems. The Oracle database identifies corrupt blocks as either “logically corrupt” or “media corrupt.” If it is logically corrupt, there is an Oracle internal error. Logically corrupt blocks are marked corrupt by the Oracle database after it detects the inconsistency. If it is media corrupt, the block format is not correct; the information in the block does not make any sense after being read from disk.

As you just learned, a number of data failures and corruptions can be repaired with the Data Recovery Advisor. Now you learn about a manual way to diagnose and repair corruptions.

You can repair a media corrupt block by recovering the block or dropping the database object that contains the corrupt block, or both. If media corruption is due to faulty hardware, the problem will not be completely resolved until the hardware fault is corrected.

Block Corruption Symptoms: ORA-01578

The error ORA-01578: "ORACLE data block corrupted (file # %s, block # %s) " :

- Is generated when a corrupted data block is found
- Always returns the relative file number and block number
- Is returned to the session that issued the query being performed when the corruption was discovered
- Appears in the alert.log file



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Usually, the ORA-01578 error is the result of a hardware problem. If the ORA-01578 error is always returned with the same arguments, it is most likely a media corrupt block.

If the arguments change each time, there may be a hardware problem, and you should have the memory and page space checked, and the I/O subsystem checked for bad controllers.

Note: ORA-01578 returns the relative file number, but the accompanying ORA-01110 error displays the absolute file number.

How to Handle Corruption

- Check the alert log and operating system log file.
- Use available diagnostic tools to find out the type of corruption.
- Determine whether the error persists by running checks multiple times.

Include dependant objects,
for example, a table index

While DML is occurring

```
ANALYZE TABLE emp VALIDATE STRUCTURE CASCADE ONLINE;
```

- Recover data from the corrupted object if necessary.
- Resolve any hardware issues:
 - Memory boards
 - Disk controllers
 - Disks
- Recover or restore data from the corrupt object if necessary.

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Always try to find out whether the error is permanent. Run the ANALYZE command multiple times or, if possible, perform a shutdown and a startup and try again to perform the operation that failed earlier. Find out whether there are more corruptions. If you encounter one, there may be other corrupted blocks as well.

Hardware failures should be addressed immediately. When you encounter hardware problems, the vendor should be contacted and the machine should be checked and fixed before continuing. A full hardware diagnostics session should be run.

Many types of hardware failures are possible:

- Faulty I/O hardware or firmware
- Operating system I/O or caching problem
- Memory or paging problems
- Disk repair utilities

For more information, see the *Oracle Database Administrator's Guide*.

Setting Parameters to Detect Corruption

Name	Help	Revisions	Type	Comments	Type	Basic	Modified	Dynamic	Category
db_block_checking			FALSE	Prevent memory and data corruption					Diagnostics and Statistics
db_block_checksum			OFF LOW MEDIUM TRUE		String				Diagnostics and Statistics
db_block_size			MEDIUM		Integer	✓	✓		Memory
db_cache_advice			FULL		String				Memory
db_block_checksum			TYPICAL	Detect I/O storage, disk corruption					Diagnostics and Statistics
db_block_size			OFF FALSE TYPICAL TRUE		Integer	✓	✓		Memory
db_cache_advice			FULL		String		✓		Memory
db_cache_size			FULL		Big		✓		Memory
db_lost_write_protect			TYPICAL	Detect nonpersistent writes on physical standby					Database Identification
db_name			NONE TYPICAL FULL		String	✓	✓		Database Identification

- Cloud Control > Administration > Initialization Parameters
- EM Express > Configuration > Initialization Parameters



Tip: Test first, because these parameters have a performance impact.

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Recommended generic block-corruption parameters:

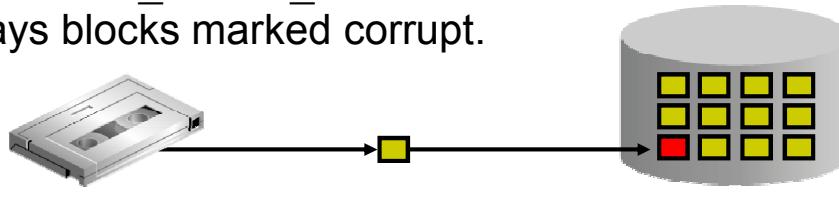
- DB_BLOCK_CHECKING, which initiates checking of database blocks. This check can often prevent memory and data corruption. (Default: FALSE, recommended: FULL or MEDIUM)
- DB_BLOCK_CHECKSUM, which initiates the calculation and storage of a checksum in the cache header of every data block when writing it to disk. Checksums assist in detecting corruption caused by underlying disks, storage systems, or I/O systems. (Default: TYPICAL, recommended: FULL)
- DB_LOST_WRITE_PROTECT, which initiates checking for “lost writes.” Data block lost writes occur on a physical standby database, when the I/O subsystem signals the completion of a block write, which has not yet been completely written in persistent storage. Of course, the write operation has been completed on the primary database. (Default: NONE, recommended: TYPICAL)

Note: Each of the parameters set to detect corruption impacts performance to varying degrees and thus requires testing before production use.

Block Media Recovery

Block media recovery:

- Lowers the mean time to recover (MTTR)
- Increases availability during media recovery
 - The data file remains online during recovery.
 - Only blocks being recovered are inaccessible.
- Is invoked using the RMAN **RECOVER...BLOCK** command
 - Restores blocks by using flashback logs and full or level 0 backups
 - Media recovery is performed using redo logs.
- The **V\$DATABASE_BLOCK_CORRUPTION** view displays blocks marked corrupt.



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In most cases, the database marks a block as media corrupt and then writes it to disk when the corruption is first encountered. No subsequent read of the block will be successful until the block is recovered. You can perform block recovery only on blocks that are marked corrupt or fail a corruption check. Block media recovery is performed using the RMAN RECOVER...BLOCK command. By default, RMAN searches the flashback logs for good copies of the blocks, and then searches for the blocks in full or level 0 incremental backups. When RMAN finds good copies, it restores them and performs media recovery on the blocks. Block media recovery can use only redo logs for media recovery, not incremental backups.

The V\$DATABASE_BLOCK_CORRUPTION view displays blocks marked corrupt by database components such as RMAN commands, ANALYZE, dbv, SQL queries, and so on. The following types of corruption result in rows added to this view:

- **Physical/Media corruption:** The database does not recognize the block: the checksum is invalid, the block contains all zeros, or the block header is fractured. Physical corruption checking is enabled by default.
- **Logical corruption:** The block has a valid checksum, the header and footer match, but the contents are inconsistent. Block media recovery cannot repair logical block corruption. Logical corruption checking is disabled by default. You can turn it on by specifying the CHECK LOGICAL option of the BACKUP, RESTORE, RECOVER, and VALIDATE commands.

Prerequisites for Block Media Recovery

- The target database must be in ARCHIVELOG mode.
- The backups of the data files containing the corrupt blocks must be full or level 0 backups.
 - Proxy copies must be restored to a non-default location before they can be used.
- RMAN can use only archived redo logs for the recovery.
- The corrupted data block can be restored from Flashback Logs if available.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The following prerequisites apply to the RECOVER . . . BLOCK command:

- The target database must run in ARCHIVELOG mode and be open or mounted with a current control file.
- The backups of the data files containing the corrupt blocks must be full or level 0 backups and not proxy copies. If only proxy copy backups exist, then you can restore them to a non-default location on disk, in which case RMAN considers them data file copies and searches them for blocks during block media recovery.
- RMAN can use only archived redo logs for the recovery. RMAN cannot use level 1 incremental backups. Block media recovery cannot survive a missing or inaccessible archived redo log, although it can sometimes survive missing redo records.
- Flashback Database must be enabled on the target database for RMAN to search the flashback logs for good copies of corrupt blocks. If flashback logging is enabled and contains older, uncorrupted versions of the corrupt blocks, then RMAN can use these blocks, possibly speeding up the recovery.

Recovering Individual Blocks

The RMAN RECOVER...BLOCK command:

- Identifies the backups containing the blocks to recover
- Reads the backups and accumulates requested blocks into in-memory buffers
- Manages the block media recovery session by reading the archive logs from backup if necessary

```
RECOVER DATAFILE 6 BLOCK 3; Recover a single block

RECOVER                               Recover multiple blocks
DATAFILE 2 BLOCK 43
DATAFILE 2 BLOCK 79
DATAFILE 6 BLOCK 183;                 in multiple data files

RECOVER CORRUPTION LIST;             Recover all blocks logged in
                                    V$DATABASE_BLOCK_CORRUPTION
```

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Before block recovery can take place, you must identify the corrupt blocks. Typically, block corruption is reported in the following locations:

- Results of the LIST FAILURE, VALIDATE, or BACKUP ... VALIDATE command
- The V\$DATABASE_BLOCK_CORRUPTION view
- Error messages in standard output
- The alert log and user trace files (identified in the V\$DIAG_INFO view)
- Results of the SQL ANALYZE TABLE and ANALYZE INDEX commands
- Results of the DBVERIFY utility

For example, you may discover the following messages in a user trace file:

```
ORA-01578: ORACLE data block corrupted (file # 7, block # 3)
ORA-01110: data file 7: '/oracle/oradata/orcl/tools01.dbf'
ORA-01578: ORACLE data block corrupted (file # 2, block # 235)
ORA-01110: data file 2: '/oracle/oradata/orcl/undotbs01.dbf'
```

After the blocks have been identified, run the RECOVER ... BLOCK command at the RMAN prompt, specifying the file and block numbers for the corrupted blocks.

```
RECOVER
DATAFILE 7 BLOCK 3
DATAFILE 2 BLOCK 235;
```

Best Practice: Proactive Checks

Invoking proactive health check of the database and its components:

- Health Monitor or RMAN VALIDATE DATABASE command
- Checking for logical and physical corruption
- Findings logged in the ADR



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

For very important databases, you may want to execute additional proactive checks (possibly daily during low peak interval periods). You can schedule periodic health checks through the Health Monitor or by using the RMAN VALIDATE command. In general, when a reactive check detects failure(s) in a database component, you may want to execute a more complete check of the affected component.

The RMAN VALIDATE DATABASE command is used to invoke health checks for the database and its components. It extends the existing VALIDATE BACKUPSET command. Any problem detected during validation is displayed to you. Problems initiate the execution of a failure assessment. If a failure is detected, it is logged into ADR as a finding. You can use the LIST FAILURE command to view all failures recorded in the repository.

The VALIDATE command supports validation of individual backup sets and data blocks. In a physical corruption, the database does not recognize the block at all. In a logical corruption, the contents of the block are logically inconsistent. By default, the VALIDATE command checks for physical corruption only. You can specify CHECK LOGICAL to check for logical corruption as well.

Block corruptions can be divided into interblock corruption and intrablock corruption. In intrablock corruption, the corruption occurs within the block itself and can be either physical or logical corruption. In interblock corruption, the corruption occurs between blocks and can be only logical corruption. The VALIDATE command checks for intrablock corruptions only.

Checking for Block Corruption

- The `VALIDATE` command:
 - Scans the specified files and verifies their contents
 - Confirms that the data files exist and are in the correct location
 - Checks for corrupt data blocks
 - Can check individual backup sets and data blocks
 - Skips never-used blocks
 - Displays failures and logs them in the ADR
- Specify the `CHECK LOGICAL` option to check for both physical and logical corruption.
- Query the `V$DATABASE_BLOCK_CORRUPTION` view to review output.



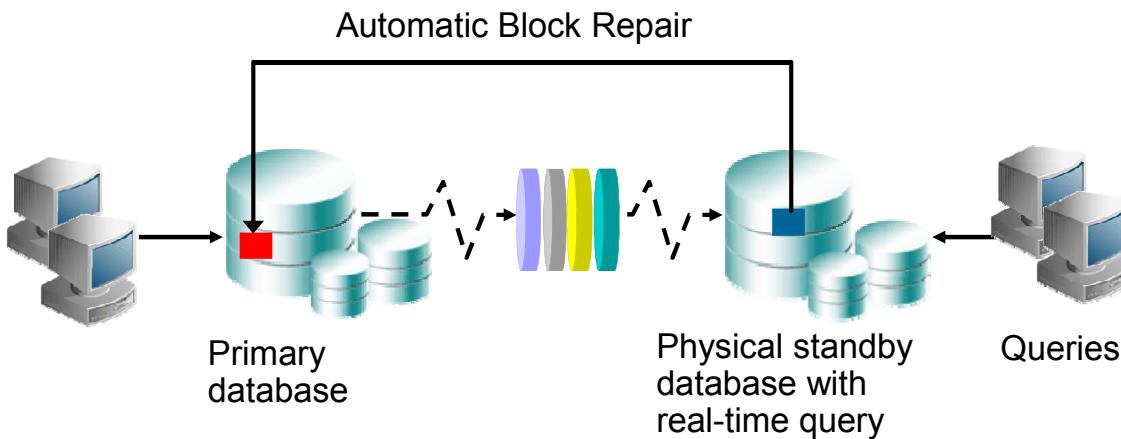
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The `VALIDATE` commands can be used to manually check for physical and logical corruptions in database files. You can also use the `BACKUP VALIDATE` command; however, only the `VALIDATE` command can be used to check individual backup sets and data blocks.

Refer to *Oracle Database Backup and Recovery Reference* for syntax detail.

Automatic Block Repair: Primary Database

- Corrupted blocks in the primary database are automatically repaired by using blocks from a physical standby database.
- Real-time query must be enabled on the physical standby database.



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The Automatic Block Repair feature enables block media recovery to use blocks from a physical standby database to perform the recovery without manual intervention. The physical standby database must have real-time query enabled to take advantage of this feature.

When a query is executed on a physical standby database configured with real-time query and a corrupted block is detected, a valid block is retrieved from the primary database.

When a corrupted block is detected during a recovery operation on the standby database, the recovery process will retrieve a valid block from the primary database.

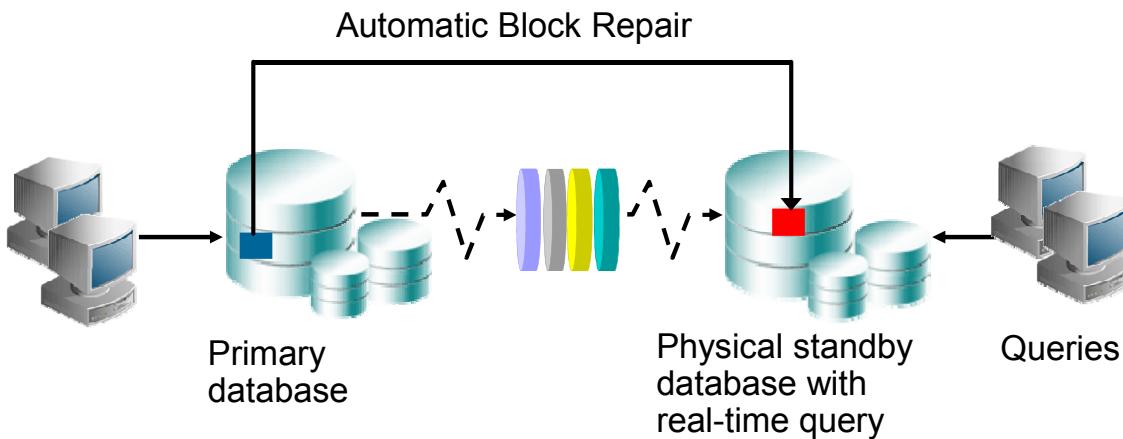
This feature reduces the amount of time that production data cannot be accessed due to block corruption, by automatically repairing the corruption as soon as it is detected. Block recovery time is reduced because up-to-date good blocks from a real-time, synchronized physical standby database are used rather than blocks from disk or tape backups, or flashback logs.

Note: Automatic Block Repair is applicable only for physical block corruption (when the checksum is invalid, the block contains all zeros, or the block header is fractured).

Real-time query is part of the Oracle Active Data Guard option.

Automatic Block Repair: Physical Standby Database

- Corrupted blocks in the physical standby database are automatically repaired by using blocks from the primary database.
- Real-time query must be enabled on the physical standby database.



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Automatic Block Repair also enables the automatic repair of corrupt blocks on the physical standby database. Block media recovery is performed by using blocks retrieved from the primary database. Real-time query must be enabled on the physical standby database to take advantage of this feature.

Summary

In this lesson, you should have learned how to:

- Detect and repair database corruption
 - View the Automatic Diagnostic Repository
 - Use the RMAN data repair commands to:
 - List failures
 - Receive a repair advice
 - Repair failures
 - Perform proactive failure checks
- Handle block corruption:
 - Verify block integrity in real time
 - Perform block media recovery
 - Proactively check and handle block corruption



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Practice Overview: Diagnosing Database Failure

- Practice 9-1 covers the following topics:
 - Discovering failures with the Data Recovery Advisor
 - Repairing failures
- Practice 9-2 covers performing and analyzing instance Recovery with ADRCI.
- Practice 9-3 covers discovering and repairing block corruption with the Data Recovery Advisor.

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

- The first practice intentionally introduces a database failure by deleting a practice data file from the file system. You use the Data Recovery Advisor to discover and repair the failure with a script generated by this advisor.
- The second one introduces you to the use of ADRCI.
- The third practice introduces block corruption. You use the Data Recovery Advisor to discover and repair the block corruption with a script generated by this advisor.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and Error : You are not a Valid Partner use only

10

Restore and Recovery Concepts

ORACLE®

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Understand how to employ the best Oracle Database recovery technology for your failure situation
- Describe instance or crash recovery
- Describe complete recovery
- Describe point-in-time recovery
- Describe recovery through RESETLOGS



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This is the second lesson in the “Recovery Unit,” which includes:

- **Lesson 9:** Diagnosing Failures
- **Lesson 10:** Restore and Recovery Concepts
- **Lesson 11:** Performing Recovery I
- **Lesson 12:** Performing Recovery II

Understanding File Loss

- File loss can be caused by:
 - User error
 - Application error
 - Media failure
- A *noncritical file* loss is one where the database can continue to function.
- The loss of a noncritical file can be addressed by:
 - Creating a new file
 - Rebuilding the file
 - Recovering the lost or damaged file



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Files can be lost or damaged due to:

- **User error:** An administrator may inadvertently delete or copy over a necessary operating system file.
- **Application error:** An application or script can also have a logic error in it, as it processes database files, resulting in a lost or damaged file.
- **Media failure:** A disk drive or controller may fail fully or partially, and introduce corruption into files, or even cause a total loss of files.

A noncritical file is one that the database and most applications can operate without. For example, if the database loses one multiplexed redo log file, there are still other redo log file copies that can be used to keep the database operating.

Although the loss of a noncritical file does not cause the database to crash, it can impair the functioning of the database, for example:

- The loss of an index tablespace can cause applications and queries to run much slower, or even make the application unusable, if the indexes were used to enforce constraints.
- The loss of an online redo log group, as long as it is not the current online log group, can cause database operations to be suspended until new log files are generated.

Data Repair Techniques

To respond to potential data loss:

- Physical failure (missing or corrupted data file):
 - Data Recovery Advisor
 - Data File Media Recovery
 - Block Recovery
- Logical failure (application or user error):
 - Logical Flashback Features
 - Oracle Flashback Database
 - Point-in-Time Recovery:
 - Database Point-in-Time Recovery (DBPITR)
 - Tablespace Point-in-Time Recovery (TSPITR)
 - Table Point-in-Time Recovery (TPITR)



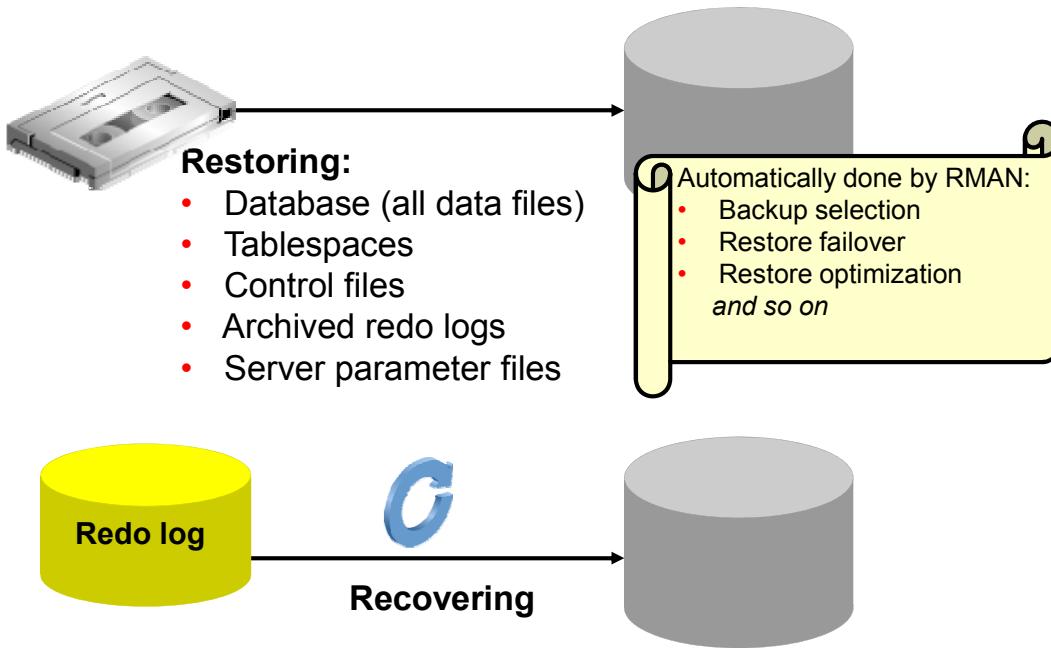
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

A brief summary of these data repair techniques:

- The Data Recovery Advisor can diagnose failures, advise you on how to respond to them, and repair the failures automatically.
- Data file media recovery is a form of media recovery that enables you to restore data file backups and apply archived redo logs or incremental backups to recover lost changes. You can either recover a whole database or a subset of the database. Data file media recovery is the most general-purpose form of recovery and can protect against both physical and logical failures.
- Block media recovery is a form of media recovery that enables you to recover individual blocks within a data file rather than the whole data file.
- Logical flashback features enable you to view or rewind individual database objects or transactions to a past time. These features do not require the use of RMAN.
- Oracle Flashback Database is a block-level recovery mechanism that is similar to media recovery, but is generally faster and does not require a backup to be restored. You can return your whole database to a previous state without restoring old copies of your data files from backup, if you have enabled flashback logging in advance. You must have a fast recovery area configured for logging for flashback database or guaranteed restore points.

- Point-in-time recovery (PITR) is a specialized form of recovery in which you recover to a noncurrent time, also known as incomplete recovery.
 - RMAN DBPITR restores the database from backups before the target time for recovery, then uses incremental backups and redo to roll the database forward to the target time.
 - RMAN TSPITR recovers a tablespace to a time earlier than the rest of the database.
 - RMAN TPITR recovers a table to an earlier point-in-time. Both TSPITR and TPITR use auxiliary database for the execution of these tasks.

Restoring and Recovering



ORACLE®

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The “recovery unit” includes two major types of activities: restoring and recovering.

- *Restoring* a file is the process of copying a backup into place to be used by the database. This is necessary if, for example, a file is damaged because the physical disk it is on fails. This is usually due to hardware problems, such as disk write errors, or controller failure. In that case, a backup of the file needs to be copied onto a new (or repaired) disk. The file types that can be restored are listed in the slide.
 - RMAN uses the records of available backup sets or image copies in the RMAN repository to select the best available backups. If two backups are from the same point in time, then RMAN prefers image copies over backup sets because RMAN can restore them more quickly (similar for disk versus tape).
 - RMAN automatically uses restore failover to skip corrupted or inaccessible backups and look for usable backups.
 - By default, RMAN skips restoring a data file if the file is present in the correct location and its header contains the expected information; and so on.
- *Recovering* the file entails applying redo such that the state of the file is brought forward in time, to whatever point you want. That point is usually as close to the time of failure as possible.

In the database industry, these two operations are often referred to, collectively, with the single term “recovery.”

Using RMAN RESTORE and RECOVER Commands

- RESTORE command: Restores database files from backup
- RECOVER command: Recovers restored files by applying changes recorded in incremental backups and redo log files

```
RMAN> ALTER TABLESPACE inv_tbs OFFLINE IMMEDIATE;
RMAN> RESTORE TABLESPACE inv_tbs;
RMAN> RECOVER TABLESPACE inv_tbs;
RMAN> ALTER TABLESPACE inv_tbs ONLINE;
```

- The Cloud Control Recovery Wizard creates and runs an RMAN script to perform the recovery.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Reconstructing the contents of an entire database or a part of it from a backup typically involves two phases: retrieving a copy of the data file from a backup, and reapplying changes to the file since the backup from the archived and online redo logs, to bring the database to the desired SCN (usually the most recent one).

- RESTORE {DATABASE | TABLESPACE name [,name]... | DATAFILE name [,name] }...

The RESTORE command retrieves the data file onto disk from a backup location on tape, disk, or other media, and makes it available to the database server. RMAN restores from backup any archived redo logs required during the recovery operation. If backups are stored on a media manager, channels must be configured or allocated for use in accessing backups stored there.

- RECOVER {DATABASE | TABLESPACE name [,name]... | DATAFILE name [,name] }...

The RECOVER command takes the restored copy of the data file and applies to it the changes recorded in the incremental backups and database's redo logs.

You can also perform complete or point-in-time recovery by using the Recovery Wizard. Navigate from the Cloud Control database home page: Availability > Backup & Recovery > Perform Recovery.

Note: An automated method of detecting the need for recovery, and carrying out that recovery makes use of the Data Recovery Advisor.

Instance Failure

Typical Causes	Possible Solutions
Power outage	Restart the instance by using the STARTUP command. Recovering from instance failure is automatic , including rolling forward changes in the redo logs and then rolling back any uncommitted transactions.
Hardware failure	Investigate the causes of failure by using the alert log, trace files, and Cloud Control.
Failure of one of the critical background processes	
Emergency shutdown procedures	

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.



Instance failure occurs when the database instance is shut down before synchronizing all database files. An instance failure can occur because of hardware or software failure or through the use of the emergency SHUTDOWN ABORT and STARTUP FORCE shutdown commands.

Administrator involvement in recovering from instance failure is rarely required if Oracle Restart is enabled and monitoring your database. Oracle Restart attempts to restart your database instance as soon as it fails. If manual intervention is required, then there may be a more serious problem that prevents the instance from restarting, such as a memory CPU failure.

Understanding Instance Recovery

Automatic instance or crash recovery:

- Is caused by attempts to open a database whose files are not synchronized on shutdown
- Uses information stored in redo log groups to synchronize files
- Involves two distinct operations:
 1. Rolling forward: Redo log changes (both committed and uncommitted) are applied to data files.
 2. Rolling back: Changes that are made but not committed are returned to their original state.

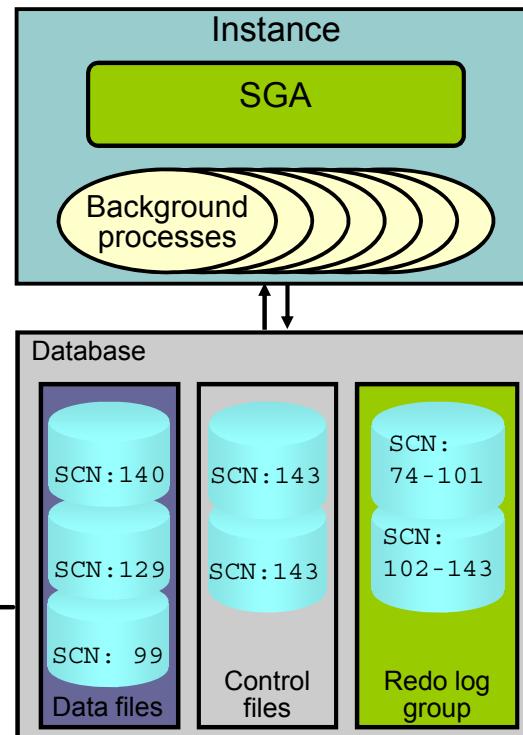


Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The Oracle database automatically recovers from instance failure. All that needs to happen is for the instance to be started normally. If Oracle Restart is enabled and configured to monitor this database, then this happens automatically. The instance mounts the control files and then attempts to open the data files. When it discovers that the data files have not been synchronized during shutdown, the instance uses information contained in the redo log groups to roll the data files forward to the time of shutdown. Then the database is opened and any uncommitted transactions are rolled back.

Phases of Instance Recovery

1. Startup instance (data files are out of sync)
2. Roll forward (redo)
3. Committed and uncommitted data in files
4. Database opened
5. Roll back (undo)
6. Committed data in files



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

For an instance to open a data file, the system change number (SCN) contained in the data file's header must match the current SCN that is stored in the database's control files.

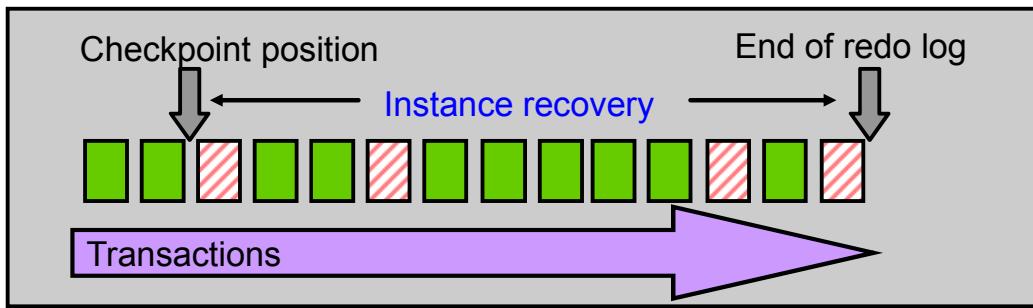
If the numbers do not match, the instance applies redo data from the online redo logs, sequentially "redoing" transactions until the data files are up to date. After all data files have been synchronized with the control files, the database is opened and users can log in.

When redo logs are applied, *all* transactions are applied to bring the database up to the state as of the time of failure. This usually includes transactions that are in progress but have not yet been committed. After the database has been opened, those uncommitted transactions are rolled back.

At the end of the rollback phase of instance recovery, the data files contain only committed data.

Tuning Instance Recovery

- During instance recovery, the transactions between the checkpoint position and the end of the redo log must be applied to data files.
- You tune instance recovery by controlling the difference between the checkpoint position and the end of the redo log.



ORACLE®

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Transaction information is recorded in the redo log groups before the instance returns commit complete for a transaction. The information in the redo log groups guarantees that the transaction can be recovered in case of a failure. The transaction information also needs to be written to the data file. The data file write usually happens at some time after the information is recorded in redo log groups because the data file write process is much slower than the redo writes. (Random writes for data files are slower than serial writes for redo log files.)

Every three seconds, the checkpoint process records information in the control file about the checkpoint position in the redo log. Therefore, the Oracle database knows that all redo log entries recorded before this point are not necessary for database recovery. In the graphic in the slide, the pink striped blocks have not yet been written to the disk.

The time required for instance recovery is the time required to bring data files from their last checkpoint to the latest SCN recorded in the control file. The administrator controls that time by setting an MTTR target (in seconds) and through the sizing of redo log groups. For example, for two redo groups, the distance between the checkpoint position and the end of the redo log group cannot be more than 90% of the smallest redo log group.

Reading the data blocks to which redo is applied of course consumes additional time.

Using the MTTR Advisor

- Specify the desired time in seconds.
- The default value is 0 (disabled).
- The maximum value is 3,600 seconds (one hour).

```
SQL> ALTER SYSTEM SET fast_start_mttr_target =  
      30 SCOPE=BOTH;
```

Seconds

ORACLE®

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The FAST_START_MTTR_TARGET initialization parameter simplifies the configuration of recovery time from instance or system failure. The MTTR Advisor converts the FAST_START_MTTR_TARGET value into several parameters to enable instance recovery in the desired time (or as close to it as possible).

Note: Explicitly setting the FAST_START_MTTR_TARGET parameter to 0 disables the MTTR advisor.

The FAST_START_MTTR_TARGET parameter must be set to a value that supports the service-level agreement for your system. A small value for the MTTR target increases I/O overhead because of additional data file writes (affecting the performance). However, if you set the MTTR target too large, the instance takes longer to recover after a crash.

Navigation tip: Cloud Control database home page: Availability > Backup & Recovery > Recovery Settings

Media Failure

Typical Causes	Possible Solutions
Failure of disk drive	<ol style="list-style-type: none">1. Restore the affected file from backup.
Failure of disk controller	<ol style="list-style-type: none">2. Inform the database about a new file location (if necessary).
Deletion or corruption of a file needed for database operation	<ol style="list-style-type: none">3. Recover the file by applying redo information (if necessary).



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

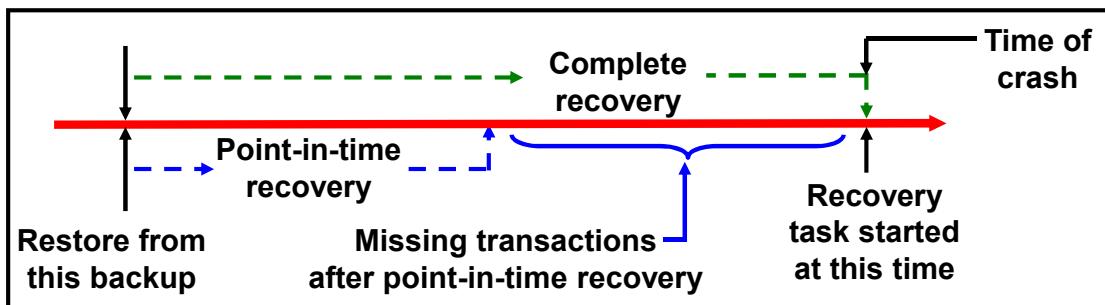
Oracle Corporation defines *media failure* as any failure that results in the loss or corruption of one or more database files (data, control, or redo log file).

Recovering from media failure requires that you restore and recover the missing files. To ensure that your database can be recovered from media failure, follow the best practices outlined on the next few pages.

Comparing Complete and Incomplete Recovery

Recovery can have two kinds of scope:

- Complete recovery: Brings the database or tablespace up to the present, including all committed data changes made to the point in time when the recovery was requested
- Incomplete or point-in-time recovery (PITR): Brings the database or tablespace up to a specified point in time in the past, before the recovery operation was requested



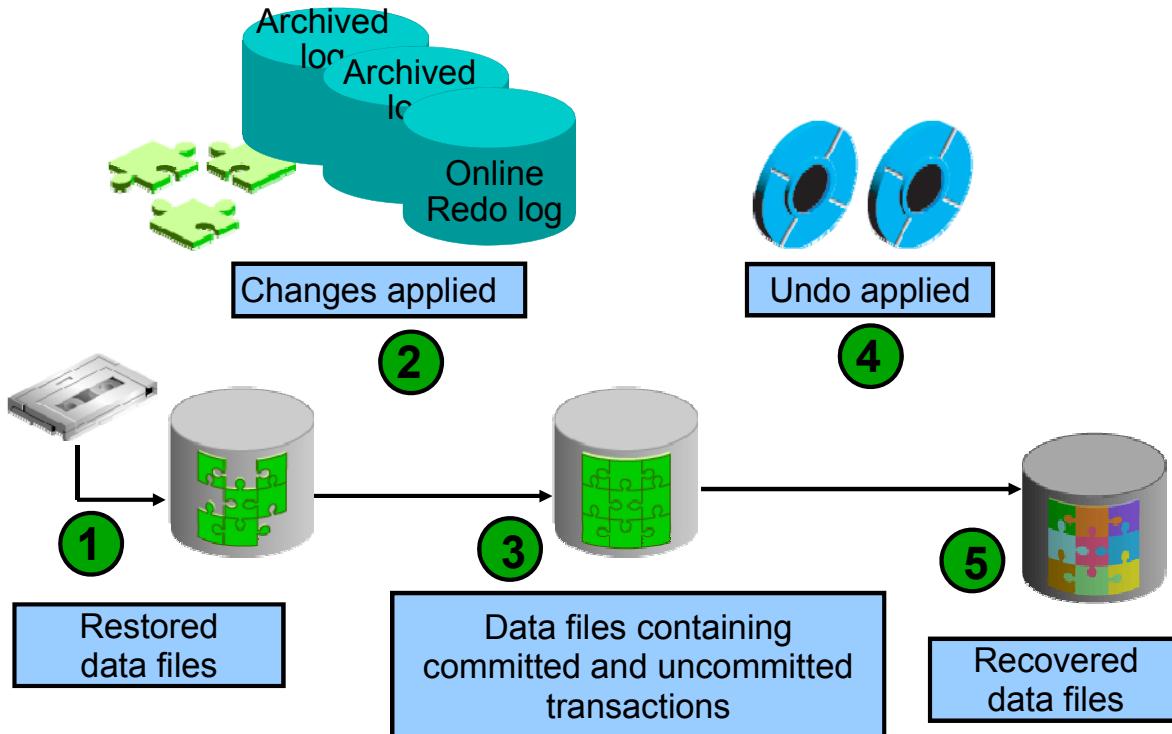
ORACLE®

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

When you perform complete recovery, you bring the database to the state where it is fully up-to-date, including all committed data modifications to the present time.

Incomplete recovery, however, brings the database or tablespace to some point in the past point-in-time. This is also known as “Point-in-Time Recovery (PITR).” It means there are missing transactions; any data modifications done between the recovery destination time and the present are lost. In many cases, this is the desirable goal because there may have been some changes made to the database that need to be undone. Recovering to a point in the past is a way to remove the unwanted changes.

Complete Recovery Process



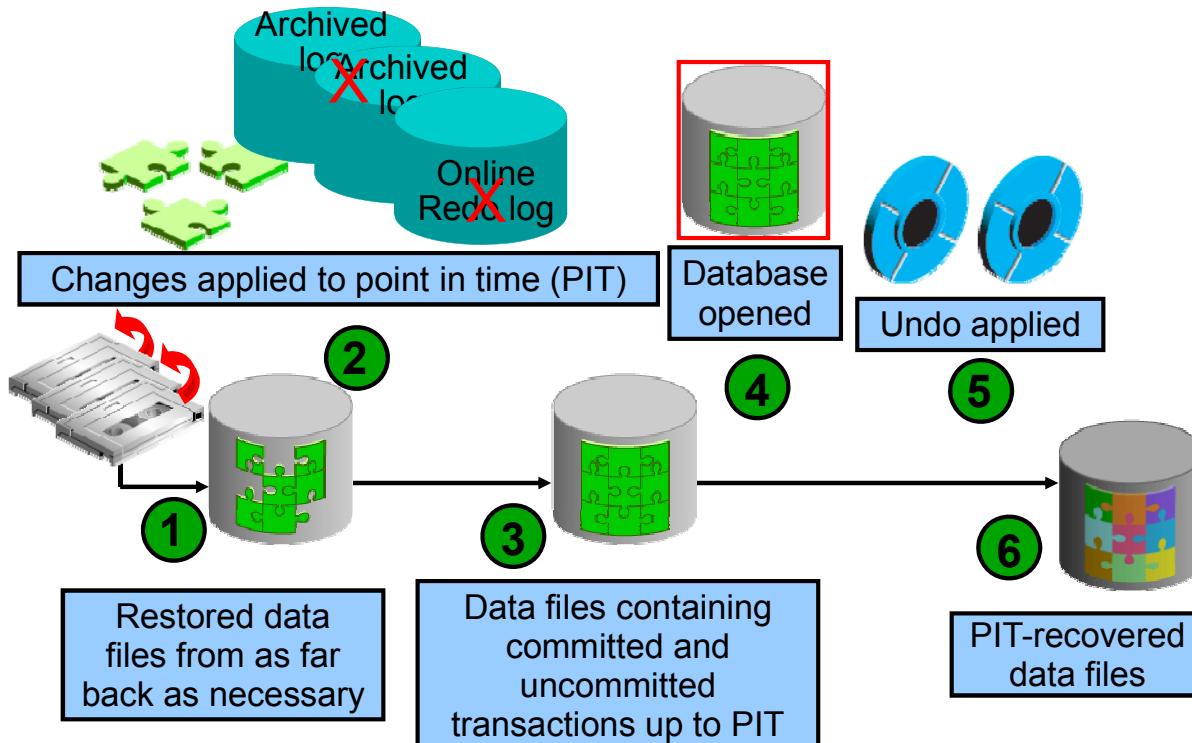
ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The following steps describe what takes place during complete recovery:

1. Damaged or missing files are restored from a backup.
2. Changes from incremental backups, archived redo log files, and online redo log files are applied as necessary. The redo log changes are applied to the data files until the current online log is reached and the most recent transactions have been re-entered. Undo blocks are generated during this entire process. This is referred to as rolling forward or cache recovery.
3. The restored data files may now contain committed and uncommitted changes.
4. The undo blocks are used to roll back any uncommitted changes. This is sometimes referred to as transaction recovery.
5. The data files are now in a recovered state and are consistent with the other data files in the database.

Point-in-Time Recovery Process



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Incomplete recovery, or database point-in-time recovery (DBPITR), uses a backup to produce a noncurrent version of the database. That is, you do not apply all of the redo records generated after the most recent backup. Perform this type of recovery only when absolutely necessary. To perform point-in-time recovery, you need:

- A valid offline or online backup of all the data files made before the recovery point
- All archived logs from the time of the backup until the specified time of recovery

The progression taken to perform a point-in-time recovery is listed as follows:

1. **Restore the data files from backup:** The backup that is used must be from before your target recovery point. This entails either copying files by using OS commands or by using the RMAN RESTORE command.
2. **Use the RECOVER command:** Apply redo from the archived redo log files, including as many as necessary to reach the restore point destination.
3. **State of over-recovery:** Now the data files contain some committed and some uncommitted transactions because the redo can contain uncommitted data.
4. **Use the ALTER DATABASE OPEN command:** The database is opened before undo is applied. This is to provide higher availability.

5. **Apply undo data:** While the redo was being applied, redo supporting the undo data files was also applied. So the undo is available to be applied to the data files in order to undo any uncommitted transactions. That is done next.
6. **Process complete:** The data files are now recovered to the point in time that you chose.

Oracle Flashback Database (covered in the lesson “Using Flashback Database”) is the most efficient alternative to DBPITR. Unlike the other flashback features, it operates at a physical level and reverts the current data files to their contents at a past time. The result is like the result of a DBPITR, including OPEN RESETLOGS, but Flashback Database is typically faster because it does not require you to restore data files and requires only limited application of redo compared to media recovery.

Recovery with RESETLOGS Option

- Issue: Missing archive logs for target recovery SCN
- Workflow:
 1. Restore backups.
 2. Recover as far forward as the unbroken series of archive logs allows.
 3. Open the database with the `RESETLOGS` option.
A new **database incarnation** is automatically created to avoid confusion when two different redo streams have the same SCNs, but occurred at different times.

Note: Changes after the last applied archive log **are lost**.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

PITR is the only option if you must perform a recovery and discover that you are missing an archived log containing transactions that occurred sometime between the time of the backup you are restoring from and the target recovery SCN. Without the missing log, you have no record of the updates to your data files during that period. Your only choice is to recover the database from the point in time of the restored backup, as far as the unbroken series of archived logs permits, and then open the database with the `RESETLOGS` option. All changes in or after the missing redo log file are lost.

A “current” database incarnation is created whenever you open the database with the `RESETLOGS` option. The incarnation from which the current one is branched is called “parent incarnation.” An incarnation number is used to uniquely tag and identify a stream of redo.

After complete recovery, you can resume normal operations without an `OPEN RESETLOGS`. After a DBPITR or recovery with a backup control file, however, you must open the database with the `RESETLOGS` option, thereby creating a new incarnation of the database. The database requires a new incarnation to avoid confusion when two different redo streams have the same SCNs, but occurred at different times. If you apply the wrong redo to your database, then you corrupt it.

The existence of multiple incarnations of a single database determines how RMAN treats backups that are not in the current incarnation path. Usually, the current database incarnation is the correct one to use.

Quiz

Which of the following Oracle technologies enable you to recover from logical errors in the database?

- a. Flashback technologies
- b. LogMiner
- c. RMAN TSPITR



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a, b, c

Quiz

During instance recovery, which of the following operations take place?

- a. Data files are restored from backups.
- b. Changes made by committed and uncommitted transactions are applied to the data files via redo log entries.
- c. Uncommitted transactions are rolled back.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: b, c

Quiz

To perform point-in-time recovery, all data files must be restored by using backups taken before the requested recovery point.

- a. True
- b. False



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Quiz

Oracle Flashback Database is the most efficient alternative to DBPITR because it does not require you to restore data files and requires only limited application of redo compared to media recovery.

- a. True
- b. False



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Summary

In this lesson, you should have learned how to:

- Determine the best Oracle Database recovery technology for your failure situation
- Describe instance/crash recovery
- Describe complete recovery
- Describe point-in-time recovery



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Practice Overview: Determining Recovery Procedures

Practice 10-1 is a case study. It covers determining the best approach for recovery based on a given set of circumstances.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In this practice, you will consider the circumstances of a failure and the backup setting to determine a strategy for restoration and recovery.

Case Study 1

Backups are taken during a nightly shutdown, with an incremental backup strategy. A level 1 backup is applied to the previous level 0 backup each night. The ARCHIVE LOG LIST command shows the following:

```
SQL> archive log list
Database log mode           No Archive Mode
Automatic archival          Disabled
Archive destination          USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence   61
Next log sequence to archive 63
Current log sequence         63
```

A disk containing the SYSAUX tablespace data files has crashed.

1. Is complete recovery possible? What are the steps?
2. If complete recovery is not possible, what are the steps to recover as much as possible? What data (transactions) is lost?



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 2

Database backups are taken nightly online, with an incremental backup strategy. A level 1 backup is applied to the previous level 0 backup each night. The ARCHIVE LOG LIST command shows the following:

```
SQL> archive log list
Database log mode          Archive Mode
Automatic archival         Enabled
Archive destination        USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence 61
Next log sequence to archive 63
Current log sequence       63
```

A data file that is part of the application tablespace, containing critical data has been lost.

Describe the steps to perform a complete recovery.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 3

The effects of a batch job that was incorrectly executed on the database last night at 8:00 p.m. have been removed by performing an incomplete recovery to 6:00 p.m. After the incomplete recovery, the database was reopened.

The checks that were performed following the recovery revealed that **some critical transactions performed prior to 7:15 p.m. are not in the database.**

1. What are valid options to recover these transactions?
2. Describe the requirements for recovering these transactions.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and Error : You are not a Valid Partner use only

11

Performing Recovery I

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Perform the appropriate type of restore and recovery operation based on the nature of your database failure
- Recover from media failures in data files
- Perform complete and incomplete or “point-in-time” recoveries



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

This is the third lesson in the “Recovery Unit,” which includes:

- **Lesson 9:** Diagnosing Failures
- **Lesson 10:** Restore and Recovery Concepts
- **Lesson 11:** Performing Recovery I
- **Lesson 12:** Performing Recovery II

Ensuring Backups Are Available

RMAN Command	Action
RESTORE PREVIEW	RMAN reports the backups and archived redo log files that RMAN uses to restore and recover the database to the specified time.
RESTORE VALIDATE	RMAN determines which backup sets, data file copies, and archived redo log files need to be restored, and then validates them.
RECOVER VALIDATE HEADER	Reports and validates the backups that RMAN could use to restore files needed for the recovery



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Use the following commands to ensure that all required backups are available and to determine whether you need to direct RMAN to use or avoid specific backups:

- **RESTORE PREVIEW:** Reports on the backups and archived redo log files that RMAN can use to restore and recover the database to the specified time. RMAN queries the metadata, but does not actually read the backup files. The output from this command is in the same format as the `LIST BACKUP` command output.
- **RESTORE VALIDATE:** Specifies that RMAN should decide which backup sets, data file copies, and archived redo log files need to be restored, and then validate them. No files are restored. For files on both disk and tape, RMAN reads all blocks in the backup piece or image copy. RMAN also validates off-site backups.
- **RECOVER VALIDATE HEADER:** Reports on and validates the backups that RMAN can use to restore files needed for recovery. When you execute the `RECOVER VALIDATE HEADER` command, RMAN performs the same operations as when you specify the `RESTORE PREVIEW` command. However, in addition to listing the files needed for the restoration and recovery, RMAN validates the backup file headers to determine whether the files on disk or in the media management catalog correspond to what is in the metadata in the RMAN repository.

RMAN enables you to validate container databases (CDBs) and pluggable databases (PDBs), which are covered in *Oracle Database 12c: Managing Multitenant Architecture* course.

Restoring in NOARCHIVELOG Mode

If the database is in NOARCHIVELOG mode and if any data file is lost, perform the following tasks:

1. Shut down the instance if it is not already down.
2. Restore the entire database, including all data and control files, from the backup.
3. Open the database.
4. Inform users that they must re-enter all changes that were made since the last backup.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The loss of *any* data file from a database in NOARCHIVELOG mode requires complete restoration of the database, including control files and all data files.

With the database in NOARCHIVELOG mode, recovery is possible only up to the time of the last backup. So users must re-enter all changes made since that backup.

To perform this type of recovery:

1. Shut down the instance if it is not already down.
2. Click Perform Recovery on the Maintenance properties page.
3. Select Whole Database as the type of recovery.

If you have a database in NOARCHIVELOG mode that has an incremental backup strategy, RMAN first restores the most recent level 0 and then RMAN recovery applies the incremental backups.

Recovery with Incremental Backups in NOARCHIVELOG Mode

Use incremental backups to perform limited recovery of a database in NOARCHIVELOG mode.

In SQL*Plus or RMAN:

```
STARTUP FORCE NOMOUNT;
RESTORE CONTROLFILE;
ALTER DATABASE MOUNT;
RESTORE DATABASE;
RECOVER DATABASE NOREDO;
ALTER DATABASE OPEN RESETLOGS;
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can perform limited recovery of a NOARCHIVELOG mode database by using incremental backups. The incremental backups must be consistent backups.

If you have taken incremental backups, RMAN uses your level 0 and level 1 backups to restore and recover the database.

You must specify the **NOREDO** option on the RECOVER DATABASE command if the online redo log files are lost or if the redo cannot be applied to the incremental backups. If you do not specify the **NOREDO** option, RMAN searches for the online redo log files after applying the incremental backups. If the online redo log files are not available, RMAN issues an error message.

If the current online redo log files contain all changes since the last incremental backup, you can issue the RECOVER DATABASE command without the **NOREDO** option and the changes will be applied.

Note: You need to restore the control file only if it is not current.

Performing Complete Recovery

Loss of a noncritical data file in ARCHIVELOG mode:

- If a data file is lost or corrupted, and if that file does not belong to the SYSTEM or UNDO tablespace, you restore and recover the missing data file while the database is **open**.
- Recovery is possible up to the time of the last commit and users are not required to re-enter any data.



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

With the database in ARCHIVELOG mode, the loss of any data file not belonging to the SYSTEM or UNDO tablespaces affects only the objects that are in the missing file. The rest of the database remains available for users to continue work.

To restore and recover the missing data file:

1. In Cloud Control, navigate to the Perform Recovery page.
2. Select Datafiles as the recovery type, and then select “Recover to current time.”
3. Add all data files that need recovery.
4. Determine whether you want to restore the files to the default location or (if a disk or controller is missing) to a new location.
5. Submit the RMAN job to restore and recover the missing files.

Because the database is in ARCHIVELOG mode, recovery is possible up to the time of the last commit and users are not required to re-enter any data.

Performing Complete Recovery

Loss of a critical data file in ARCHIVELOG mode:

1. The instance may or may not shut down automatically. If it does not, use SHUTDOWN ABORT to bring the instance down.
2. **Mount** the database.
3. Restore and recover the missing data file.
4. Open the database.



Perform this task in the hands-on practice.

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Data files belonging to the SYSTEM tablespace or containing UNDO data are considered system critical. A loss of one of these files requires the database to be restored from the MOUNT state (unlike other data files that may be restored with the database open).

To perform this recovery:

1. If the instance is not already shut down, shut it down.
2. Mount the database.
3. In Cloud Control, navigate to the Perform Recovery page.
4. Select Datafiles as the recovery type, and then select “Recover to current time.”
5. Add all data files that need recovery.
6. Determine whether you want to restore the files to the default location or (if a disk or controller is missing) to a new location.
7. Submit the RMAN job to restore and recover the missing files.
8. Open the database. Users are not required to re-enter data because the recovery is up to the time of the last commit.

Restoring ASM Disk Groups

- Use the ASMCMD `md_restore` command to restore ASM disk groups from a metadata backup file.
- In the event of a loss of the ASM disk group, the metadata backup file can be used to reconstruct the disk group and its metadata rather than re-creating the disk group manually.
- Restore options:
 - `full`: Creates a disk group and restores metadata
 - `nodg`: Restores metadata only
 - `newdg -o`: Creates a disk group with a different name than the original disk group and restores the metadata



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can use the ASMCMD `md_restore` command to re-create an ASM disk group from a previously created metadata backup file. The backup file is created by using the ASMCMD `md_backup` command. If you did not create an ASM disk group metadata backup file, you must manually re-create the ASM diskgroup in the event of a loss of the diskgroup.

Restoring ASM Disk Groups: Examples

- Restoring all disk groups in the metadata file:

```
ASMCMD> md_restore /backup/asm_metadata --full
```

- Restoring the metadata only for the DATA disk group:

```
ASMCMD> md_restore /backup/asm_metadata --nodg -G data
```

- Creating a SQL script to restore the DATA disk group:

```
ASMCMD> md_restore /backup/asm_metadata -full  
-S asmsql.sql -G data
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The examples in the slide illustrate many of the options for the ASMCMD `md_restore` command.

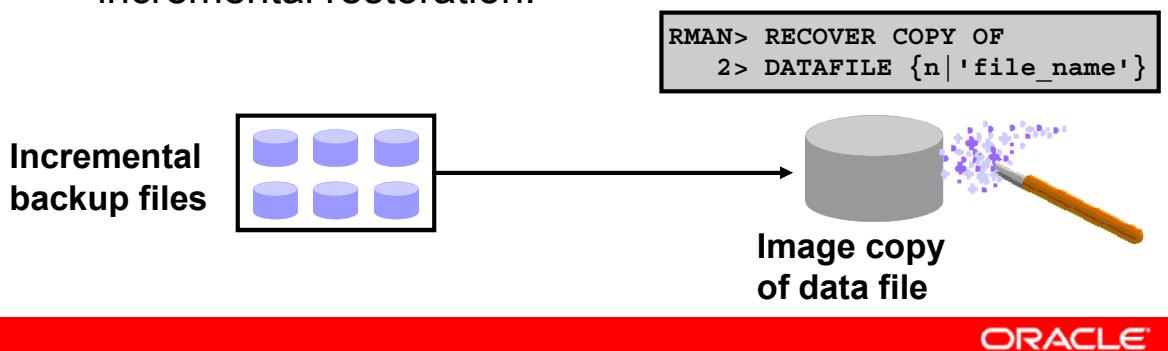
In the third example, a SQL script is created. The disk group is not re-created when the `-S` option is used.

Refer to the *Oracle Automatic Storage Management Administrator's Guide* for additional information about the `md_restore` command.

What You Already Know About Recovering Image Copies

RMAN can recover image copies by using incremental backups:

- Image copies are updated with all changes up to the incremental backup SCN.
- Incremental backup reduces the time required for media recovery.
- There is no need to perform an image copy after the incremental restoration.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can use RMAN to apply incremental backups to data file image copies. With this recovery method, you use RMAN to recover a copy of a data file—that is, you roll forward (recover) the image copy to the specified point in time by applying the incremental backups to the image copy. The image copy is updated with all changes up through the SCN at which the incremental backup was taken. RMAN uses the resulting updated data file in media recovery just as it would use a full image copy taken at that SCN, without the overhead of performing a full image copy of the database every day. The following are the benefits of applying incremental backups to data file image copies:

- You reduce the time required for media recovery (using archive logs) because you need to apply archive logs only since the last incremental backup.
- You do not need to perform a full image copy after the incremental restoration.

If the recovery process fails during the application of the incremental backup file, you simply restart the recovery process. RMAN automatically determines the required incremental backup files to apply, from before the image data file copy until the time at which you want to stop the recovery process. If there is more than one version of an image copy recorded in the RMAN catalog, RMAN automatically uses the latest version of the image copy. RMAN reports an error if it cannot merge an incremental backup file with an image copy.

Recovering Image Copies: Example

If you run these commands daily:

```
RMAN> recover copy of database with tag 'daily_inc';
RMAN> backup incremental level 1 for recover of copy
2> with tag 'daily_inc' database;
```

This is the result:

	RECOVER	BACKUP
Day 1	Nothing	Create image copies
Day 2	Nothing	Create incremental level 1
Day 3 and onward	Recover copies based on incremental	Create incremental level 1



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

If you run the commands shown in the slide daily, you get continuously updated image copies of all the database data files at any time.

The chart shows what happens for each run. Note that this algorithm requires some priming; the strategy does not come to fruition until after day 3.

- **Day 1:** The RECOVER command does nothing. There exist no image copies to recover yet. The BACKUP command creates the image copies.
- **Day 2:** The RECOVER command, again, does nothing. This is because there is no incremental backup yet. The BACKUP command creates the incremental backup, now that baseline image copies have been created on day 1.
- **Day 3:** The RECOVER command applies the changes from the incremental backup to the image copies. The BACKUP command takes another incremental backup, which will be used to recover the image copies on day 4. The cycle continues like this.

It is important to use tags when implementing this kind of backup strategy. They serve to link these particular incremental backups to the image copies that are made. Without the tag, the most recent, and possibly incorrect, incremental backup would be used to recover the image copies.

Performing a Fast Switch to Image Copies

Perform fast recovery by performing the following steps:

1. Take data files offline.
2. Use the SWITCH TO ... COPY command to switch to image copies.
3. Recover data files.
4. Bring data files online.

Now the data files are recovered and usable in their new location.

Optionally, do the following to put the files back into their original location:

5. Create an image copy of the data file in the original location.
6. Take data files offline.
7. SWITCH TO ... COPY
8. Recover data files.
9. Bring data files online.

```
SQL> SWITCH DATAFILE 'filename' TO COPY;
```

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can use image copies of data files for fast recovery by performing the following steps:

1. Take the data file offline.
2. Use the SWITCH TO ... COPY command to point to the image copy of the files.
3. Recover the data files.
4. Bring the data files online.

At this point, the database is usable, and the data files are recovered. But, if you want to put the data files back into their original location, proceed with the following steps:

5. Create an image copy of the data files in the original location by using the BACKUP AS COPY command.
6. Take the data files offline.
7. Switch to the copy you made in step 5 by using the SWITCH TO COPY command.
8. Recover the data files.
9. Bring the data files online.

You can recover data files, tablespaces, tempfiles, or the entire database with this command. The files being switched to must be image copies.

Using SET NEWNAME for Switching Files

- Use the SET NEWNAME command in a RUN block to restore to a nondefault location.

```
RUN
{
  ALLOCATE CHANNEL dev1 DEVICE TYPE DISK;
  ALLOCATE CHANNEL dev2 DEVICE TYPE sbt;
  SQL "ALTER TABLESPACE users OFFLINE IMMEDIATE";
  SET NEWNAME FOR DATAFILE '/disk1/oradata/prod/users01.dbf'
    TO '/disk2/users01.dbf';
  RESTORE TABLESPACE users;
  SWITCH DATAFILE ALL;
  RECOVER TABLESPACE users;
  SQL "ALTER TABLESPACE users ONLINE";
}
```

- Instead of individual names, specify a default name format for all files in a database or in a named tablespace.
- The default name is used for DUPLICATE, RESTORE, and SWITCH commands in the RUN block.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The SET NEWNAME command can be used only inside a RUN block. It prepares a name mapping for subsequent operations. In the example in the slide, the SET NEWNAME command defines the location where a restore operation of that data file will be written. When the RESTORE command executes, the users01.dbf data file is restored to /disk2/users01.dbf. It is written there, but the control file is still not pointing to that location. The SWITCH command causes the control file to be updated with the new location.

A more efficient way is to use the SET NEWNAME clause to specify the default name format for all data files in a named tablespace and all data files in the database (rather than setting file names individually, as in database versions prior to Oracle Database 11g Release 2).

The order of precedence for the SET NEWNAME command is as follows:

1. SET NEWNAME FOR DATAFILE and SET NEWNAME FOR TEMPFILE
2. SET NEWNAME FOR TABLESPACE
3. SET NEWNAME FOR DATABASE

Using Restore Points

A restore point provides a name to a point in time:

- Now:

```
SQL> CREATE RESTORE POINT before_mods;
```

- Some time in the past:

```
SQL> CREATE RESTORE POINT end_q1 AS OF SCN 100;
```

Timeline

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can give a name to a particular point in time, or an SCN number. This is useful for future reference, when performing point-in-time recovery or flashback operations.

- The first example in the slide creates a restore point that represents the present point in time. If you were about to apply an update of an application or data in the database, and you wanted to refer back to this state of the database, you could use the BEFORE_MODS restore point.
- The second example in the slide creates a restore point representing a past SCN, 100. This restore point can be used in the same ways as the previous one.

Normally, restore points are maintained in the database for at least as long as specified by the CONTROL_FILE_RECORD_KEEP_TIME initialization parameter. However, you can use the PRESERVE option when creating a restore point, which causes the restore point to be saved until you explicitly delete it.

You can see restore points in the V\$RESTORE_POINT view with name, SCN, time stamp, and other information.

Performing Point-in-Time Recovery

Perform server-managed point-in-time recovery by doing the following:

1. Determine the target point of the restore: SCN, time, restore point, or log sequence number.
2. Set the NLS environment variables appropriately.
3. Mount the database.
4. Prepare and execute a RUN block, using the SET UNTIL, RESTORE, and RECOVER commands.
5. Open the database in READ ONLY mode, and verify that the recovery point is correct.
6. Open the database by using RESETLOGS.



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can perform server-managed point-in-time recovery by using the following steps. The database must be in ARCHIVELOG mode.

1. Determine the restore target. This can be in terms of a date and time, an SCN, restore point, or log sequence number. For example, if you know that some bad transactions were submitted at 3:00 PM yesterday, then you can choose 2:59 PM yesterday as the target restore point time.
2. Set the National Language Support (NLS) OS environment variables, so that the time constants you provide to RMAN are formatted correctly. These are some example settings:

```
$ export NLS_LANG = american_america.us7ascii  
$ export NLS_DATE_FORMAT = "yyyy-mm-dd:hh24:mi:ss"
```

3. Mount the database. If it is open, you have to shut it down first, as in this example:

```
RMAN> shutdown immediate  
RMAN> startup mount
```

4. Create a RUN block and run it. The RECOVER and RESTORE commands should be in the same RUN block so that the UNTIL setting applies to both. For example, if you choose to recover to a particular SCN, the RESTORE command needs to know that value so it restores files from backups that are sufficiently old—that is, backups that are from before that SCN. Here is an example of a RUN block:

```
RUN
{
    SET UNTIL TIME '2012-08-14:21:59:00';
    RESTORE DATABASE;
    RECOVER DATABASE;
}
```

Note: If you do not want to rely on the NLS parameters, you can set the time explicitly, for example in a European format:

```
set until time
"to_date('14.08.2012 21:59:00', 'dd.mm.yyyy hh24:mi:ss')";
```

5. As soon as you open the database for read/write, you have committed to the restore you just performed. So, first, open the database READ ONLY, and view some data, to check whether the recovery did what you expected.

```
RMAN> SQL 'ALTER DATABASE OPEN READ ONLY';
```

6. If satisfied with the results of the recovery, open the database with the RESETLOGS option, as shown:

```
RMAN> ALTER DATABASE OPEN RESETLOGS;
```

Quiz

When the database is in ARCHIVELOG mode, the database can remain open while you recover from a loss to the SYSTEM tablespace.

- a. True
- b. False



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Answer: b

Summary

In this lesson, you should have learned how to:

- Perform the appropriate type of restore and recovery operation based on the nature of your database failure
- Recover from media failures in data files
- Perform complete and incomplete or “point-in-time” recoveries



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Practice Overview: Recovering from Media Failure

- Practice 11-1 covers performing **complete** recovery after loss of an essential data file.
- Practice 11-2 covers performing **incomplete** or point-in-time recovery.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In the first practice you will perform a complete recovery of the database after the loss of an essential data file. In this case Data Recovery Advisor cannot be used.

In the second practice you will discover a scenario that requires an incomplete recovery.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and Error : You are not a Valid Partner use only

12

Performing Recovery II

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Recover from the loss of the server parameter file
- Recover from control file and redo log file failures
- Re-create the password authentication file
- Recover index and read-only tablespaces
- Review the automatic recovery of the tempfile
- Describe the basic procedure of restoring the database to a new host
- Describe disaster recovery



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

This is the fourth lesson in the “Recovery Unit,” which includes:

- **Lesson 9:** Diagnosing Failures
- **Lesson 10:** Restore and Recovery Concepts
- **Lesson 11:** Performing Recovery I
- **Lesson 12:** Performing Recovery II

Recovery from Loss of Server Parameter File

The `FROM MEMORY` clause allows the creation of current systemwide parameter settings.

```
SQL> CREATE PFILE [= 'pfile_name']
      FROM { { SPFILE [= 'spfile_name'] } | MEMORY } ;
```

```
SQL> CREATE SPFILE [= 'spfile_name']
      FROM { { PFILE [= 'pfile_name'] } | MEMORY } ;
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

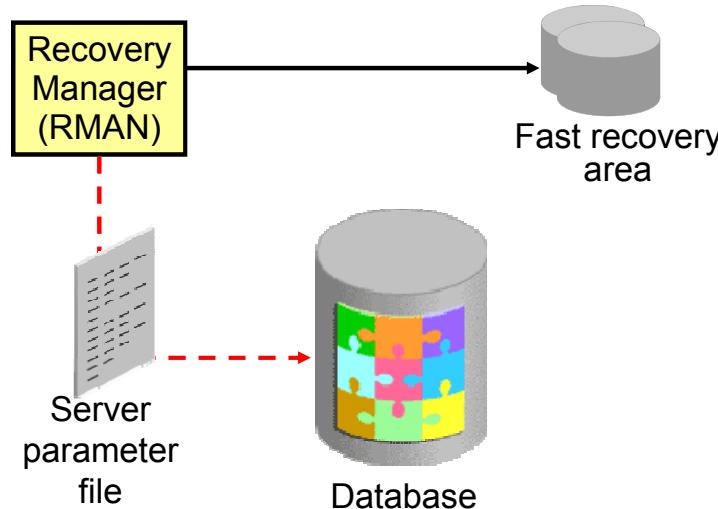
The easiest way to recover a server parameter file is to use the `FROM MEMORY` clause, which creates a text initialization parameter file (PFILE) or server parameter file (SPFILE) using the current systemwide parameter settings. In a RAC environment, the created file contains the parameter settings from each instance.

During instance startup, all parameter settings are logged to the `alert.log` file. Starting with Oracle Database 11g, the `alert.log` parameter dump text is written in valid parameter syntax. This facilitates cutting and pasting of parameters into a separate file, and then using it as a PFILE for a subsequent instance. The name of the PFILE or SPFILE is written to `alert.log` at instance startup time. In cases when an unknown client-side PFILE is used, the alert log indicates this as well.

To support this additional functionality, the `COMPATIBLE` initialization parameter must be set to 11.0.0.0 or higher.

Restoring the Server Parameter File from the Control File Autobackup

```
RMAN> STARTUP FORCE NOMOUNT;
RMAN> RESTORE SPFILE FROM AUTOBACKUP;
RMAN> STARTUP FORCE;
```



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

If you have lost the server parameter file and you cannot use the `FROM MEMORY` clause, then you can restore it from the autobackup. The procedure is similar to restoring the control file from autobackup. If the autobackup is not in the flash recovery area, set the DBID for your database. Issue the `RESTORE SPFILE FROM AUTOBACKUP` command.

If you are restoring the SPFILE to a non-default location, specify the command as follows:

```
RESTORE SPFILE TO <file_name> FROM AUTOBACKUP
```

If you are restoring the server parameter file from the fast recovery area, specify the command as follows:

```
RMAN> run {
2> restore spfile from autobackup
3> recovery area = '<flash recovery area destination>'
4> db_name = '<db_name>';
5> }
```

Loss of a Control File

If a control file is lost or corrupted, the instance normally aborts.

- If control files are stored in ASM disk groups, recovery options are as follows:
 - Perform guided recovery using Cloud Control.
 - Put the database in NOMOUNT mode and use an RMAN command to restore the control file from existing control file.

```
RMAN> restore controlfile from  
'+DATA/orcl/controlfile/current.260.695209463';
```

- If control files are stored as regular file system files, then:
 - Shut down the database
 - Copy the existing control file to replace the lost control file

Open the database after the control file is successfully restored.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The options for recovery from the loss of a control file depend on the storage configuration of the control files and on whether at least one control file remains or all have been lost.

If you are using ASM storage, and at least one control file copy remains, you can perform guided recovery by using Cloud Control or perform manual recovery using RMAN as follows:

1. Put the database in NOMOUNT mode.
2. Connect to RMAN and issue the RESTORE CONTROLFILE command to restore the control file from an existing control file, for example:
`restore controlfile from
'+DATA/orcl/controlfile/current.260.695209463';`
3. After the control file is successfully restored, open the database.

Note: You can also use the ASMCMD cp command to restore the control file.

If your control files are stored as regular file system files and at least one control file copy remains, then, while the database is down, you can just copy one of the remaining control files to the missing file's location. If the media failure is due to the loss of a disk drive or controller, copy one of the remaining control files to some other location and update the instance's parameter file to point to the new location. Alternatively, you can delete the reference to the missing control file from the initialization parameter file. Remember that Oracle recommends having at least two control files at all times.

Recovering from the Loss of All Control File Copies: Overview

	Current	Backup
Available	Restore backup control file, perform complete recovery, OPEN RESETLOGS.	Restore backup control file, perform complete recovery, OPEN RESETLOGS.
Unavailable	Re-create control file, OPEN RESETLOGS.	Restore backup control file, perform point-in-time recovery, OPEN RESETLOGS.

Online log status

Data file status



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Loss of all control files should never happen. **Prevention is better than recovery.** Even though you have copies of the control file stored in different locations, there is still the possibility that you will, at some point, have to recover from losing all those copies. If you have lost all copies of the current control file, and have a backup control file, your course of action depends on the status of the online log files and the data files. If you do not have a backup of the control file but have a text file created with the ALTER DATABASE BACKUP CONTROLFILE TO TRACE command, you may be able to use it to re-create the control file.

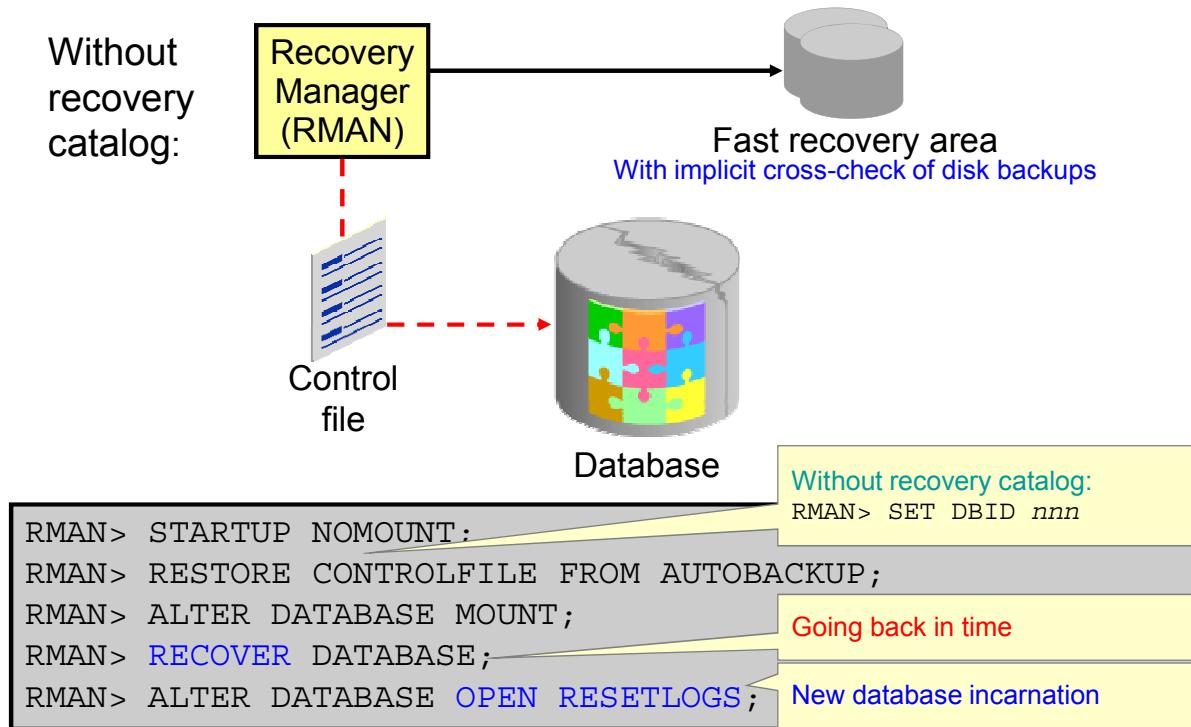
Online Logs Available

If the online logs are available and contain redo necessary for recovery, and the data files are current, then you can restore a backup control file, perform complete recovery, and open the database with the RESETLOGS option. You must specify the file names of the online redo logs during recovery. If the data files are not current, perform the same procedure.

Online Logs Not Available

If the online logs are not available, and the data files are current, then re-create the control file and open RESETLOGS. However, if the data files are not current, restore a backup control file, perform point-in-time recovery, and open RESETLOGS.

Restoring the Control File from Autobackup



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Oracle Corporation recommends that you should have AUTOBACKUP of the control file configured, so that you are able to quickly restore the control file if needed. The commands used for restoring your control file are the same, whether or not you are using a fast recovery area. However, if you are using a fast recovery area, RMAN implicitly cross-checks backups and image copies listed in the control file, and catalogs any files in the fast recovery area that are not recorded in the restored control file; thereby improving the usefulness of the restored control file in the restoration of the rest of your database.

Use the commands shown in the slide to recover from lost control files:

1. First, start the instance in `NOMOUNT` mode. It cannot be mounted because there is no control file.
2. Restore the control file from backup.
3. Now that there is a control file, you can mount the database.
4. You must recover the database, because you now have a backup control file that contains information about an older version of the database.
5. After recovering the database, you can open it. You must specify `RESETLOGS` because the new control file represents a different instantiation of the database.

Note: Tape backups are not automatically cross-checked after the restoration of a control file. After restoring the control file and mounting the database, you must cross-check the backups on tape.

Restoring the SPFILE and the Control File

With recovery catalog:

- Database in NOMOUNT state

```
RMAN> RESTORE CONTROLFILE;
```

Restore to all locations listed in the
CONTROL_FILES parameter.

```
RMAN> RESTORE CONTROLFILE... TO <destination>
```

Restore to non-default locations.

Loss of SPFILE and control file:

1. Set the DBID or use recovery catalog.
2. Restore the SPFILE from the autobackup.
3. Start the instance with the restored SPFILE.
4. Restore the control file from the autobackup.
5. Mount the database with the restored control file.
6. Restore and recover the database.
7. Open the database with the RESETLOGS option.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

If you have a recovery catalog, you do not have to set the DBID or use the control file autobackup to restore the control file. You can use the RESTORE CONTROLFILE command with no arguments:

```
RMAN> RESTORE CONTROLFILE;
```

The instance must be in the NOMOUNT state when you perform this operation, and RMAN must be connected to the recovery catalog. The restored control file is written to all locations listed in the CONTROL_FILES initialization parameter.

Use the RESTORE CONTROLFILE... TO <destination> command to restore the control file to a non-default location.

If you have also lost the SPFILE for the database and need to restore it from the autobackup, the procedure is similar to restoring the control file from autobackup. You must first set the DBID for your database, and then use the RESTORE SPFILE FROM AUTOBACKUP command.

After you have started the instance with the restored server parameter file, RMAN can restore the control file from the autobackup. After you restore and mount the control file, you have the backup information necessary to restore and recover the database.

After restoring the control files of your database from backup, you must perform complete media recovery and then open your database with the RESETLOGS option.

Quiz

Which of the following must you do to recover from the loss of one control file (when control files are multiplexed)?

- a. Shut down the database instance.
- b. Copy all data files from backups.
- c. Copy the control file from a backup.
- d. Copy one of the current control files to the lost location.



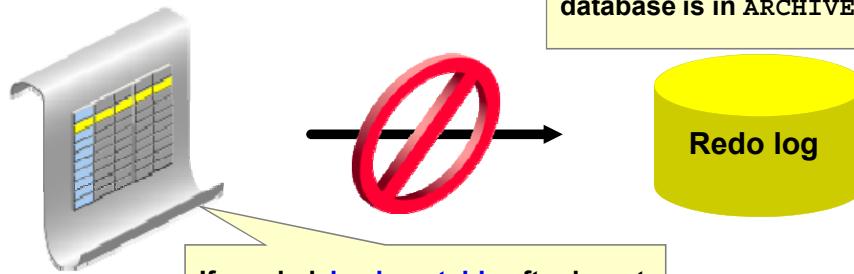
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Answer: a, d

Recovering NOLOGGING Database Objects

```
SQL> CREATE TABLE sales_copy NOLOGGING;
SQL> INSERT /*+ APPEND */ INTO sales_copy
2  SELECT * FROM sales_history;
```

Objects created with NOLOGGING cannot be recovered (even if the database is in ARCHIVELOG mode).



If needed, back up table after insert for recovery purposes.

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Take advantage of the efficiencies of the NOLOGGING attribute of tables and indexes if you can. When you create a table as NOLOGGING, minimal redo data is written to the redo stream to support the creation of the object. This is useful for making large inserts go faster.

In the example in the slide, the SALES_COPY table is created as a NOLOGGING table. As a result, when an insert is done with the APPEND hint, no redo is generated for that particular insert statement. As a result, **you cannot recover this transaction** on the SALES_HISTORY table. If that is a problem, it is important that you make a backup of whatever tables you populate in this way, right afterward. Then, you are able to go to the more recent backup of the table.

If you perform media recovery, and there are NOLOGGING objects involved, they will be marked logically corrupt during the recovery process. In this case, drop the NOLOGGING objects and re-create them.

Loss of a Redo Log File

If a member of a redo log file group is lost and if the group still has at least one member, note the following results:

- Normal operation of the instance is not affected.
- You receive a message in the alert log notifying you that a member cannot be found.
- You can restore the missing log file by dropping the lost redo log member and adding a new member.
- If the group with the missing log file has been archived, you can clear the log group to re-create the missing file.
- Immediately take a full database backup of the whole database.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Recovering from the loss of a single redo log group member should not affect the instance.

To perform this recovery:

1. Determine whether there is a missing log file by examining the alert log.
2. Restore the missing file by first dropping the lost redo log member:

```
ALTER DATABASE DROP LOGFILE MEMBER ''
```

Then, add a new member to replace the lost redo log member:

```
ALTER DATABASE ADD LOGFILE MEMBER '' TO GROUP n
```

Cloud Control can also be used to drop and re-create the log file member.

Note: If using Oracle Managed Files (OMF) for your redo log files and you use the preceding syntax to add a new redo log member to an existing group, that new redo log member file will not be an OMF file. If you want to ensure that the new redo log member is an OMF file, then the easiest recovery option would be to create a new redo log group and then drop the redo log group that had the missing redo log member.

3. If the media failure is due to the loss of a disk drive or controller, rename the missing file.

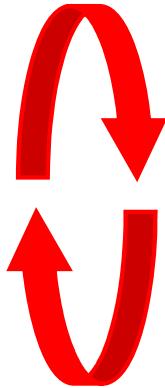
4. If the group has already been archived, or if you are in NOARCHIVELOG mode, you may choose to solve the problem by clearing the log group to re-create the missing file or files. Select the appropriate group and then select the Clear Logfile action. You can also clear the affected group manually with the following command:

```
ALTER DATABASE CLEAR LOGFILE GROUP #
```

Note: Cloud Control does not allow you to clear a log group that has not been archived.

Doing so breaks the chain of redo information. If you must clear an unarchived log group, you should *immediately* take a full backup of the whole database. Failure to do so may result in a loss of data if another failure occurs. To clear an unarchived log group, use the following command: ALTER DATABASE CLEAR UNARCHIVED LOGFILE GROUP #

Log Group Status: Review



A redo log group has a status of one of the following values at any given time:

- **CURRENT:** The LGWR process is currently writing redo data to it.
- **ACTIVE:** It is no longer being written to, but it is still required for instance recovery.
- **INACTIVE:** It is no longer being written to, and it is no longer required for instance recovery.

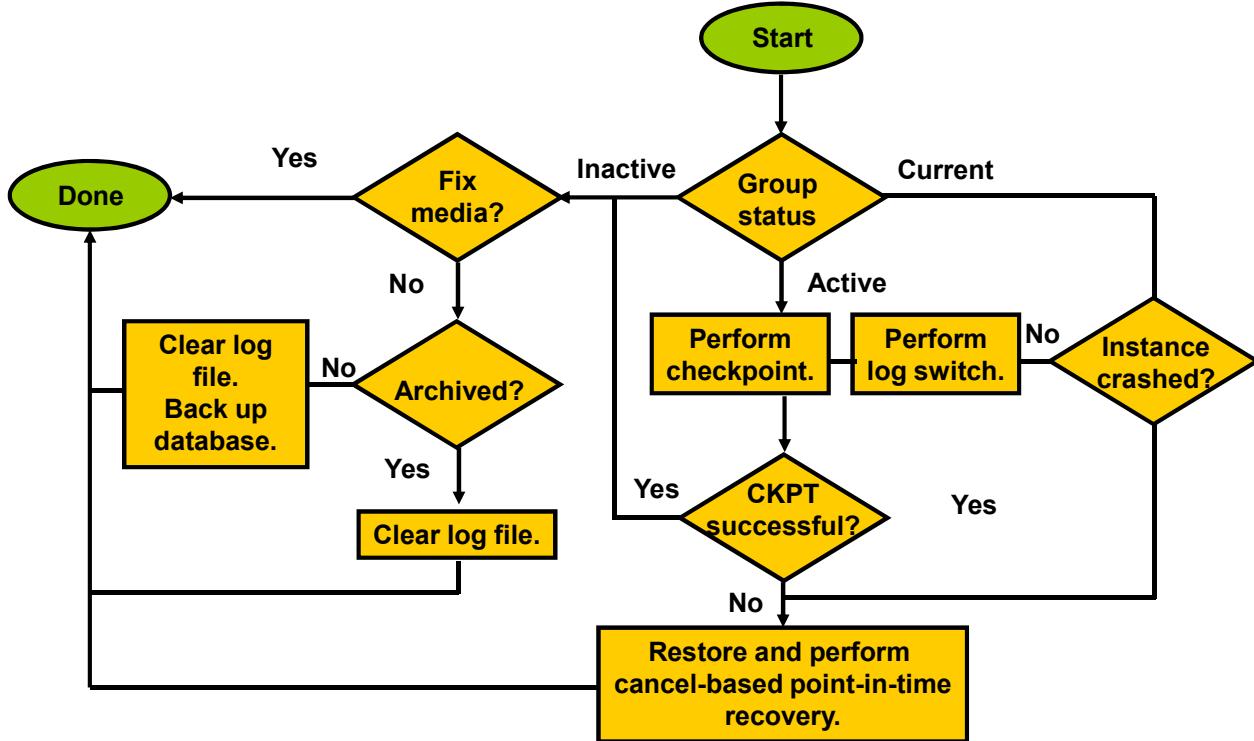
ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

To deal with the loss of redo log files, it is important to understand the possible states of redo log groups. Redo log groups cycle through three different states as part of the normal running of the Oracle database. They are, in order of the cycle:

- **CURRENT:** This state means that the redo log group is being written to by LGWR to record redo data for any transactions going on in the database. The log group remains in this state until there is a switch to another log group.
- **ACTIVE:** The redo log group still contains redo data that is required for instance recovery. This is the status during the time when a checkpoint has not yet executed that would write out to the data files all data changes that are represented in the redo log group.
- **INACTIVE:** The checkpoint discussed has indeed executed, meaning that the redo log group is no longer needed for instance recovery, and is free to become the next CURRENT log group.

Recovering from the Loss of a Redo Log Group



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

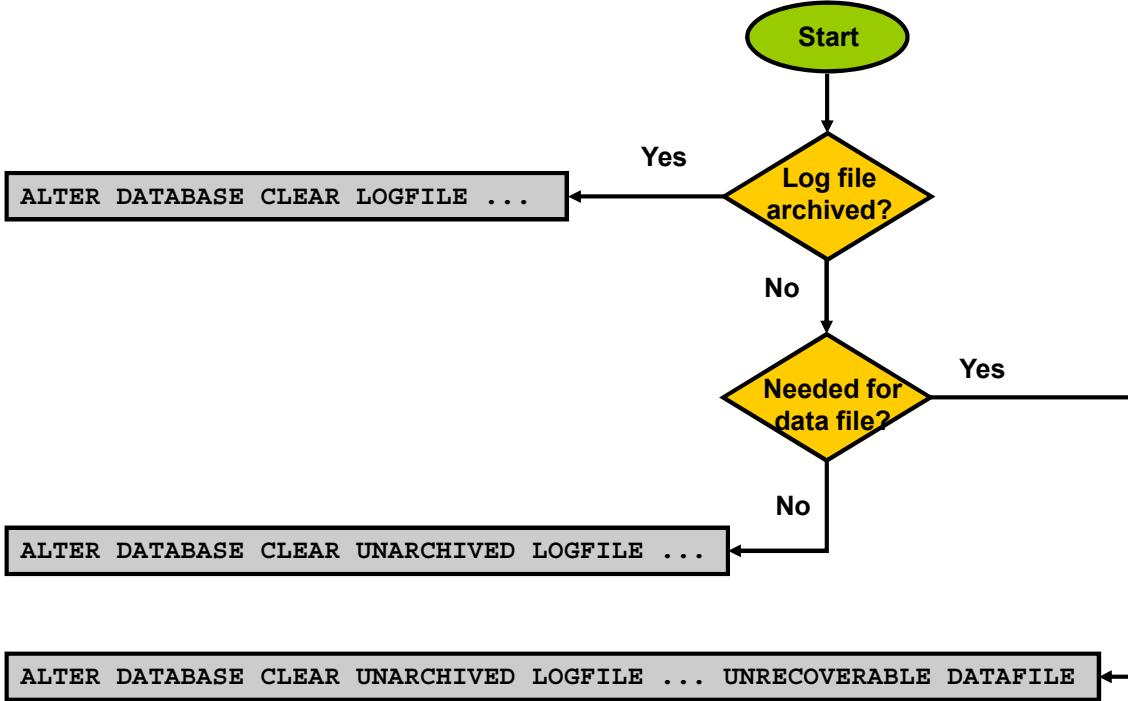
If you have lost an entire redo log group, then all copies of the log files for that group are unusable or gone.

The simplest case is where the redo log group is in the `INACTIVE` state. That means it is not currently being written to, and it is no longer needed for instance recovery. If the problem is temporary, or you are able to fix the media, then the database continues to run normally, and the group is reused when enough log switch events occur. Otherwise, if the media cannot be fixed, you can clear the log file. When you clear a log file, you are indicating that it can be reused.

If the redo log group in question is `ACTIVE`, then, even though it is not currently being written to, it is still needed for instance recovery. If you are able to perform a checkpoint, then the log file group is no longer needed for instance recovery, and you can proceed as if the group were in the inactive state.

If the log group is in the `CURRENT` state, then it is, or was, being actively written to at the time of the loss. You may even see the LGWR process fail in this case. If this happens, the instance crashes. Your only option at this point is to restore from backup, perform cancel-based point-in-time recovery, and then open the database with the `RESETLOGS` option.

Clearing a Log File



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Clear a log file using this command:

```
ALTER DATABASE CLEAR [UNARCHIVED] LOGFILE GROUP <n>
[UNRECOVERABLE DATAFILE]
```

When you clear a log file, you are indicating that it can be reused. If the log file has already been archived, the simplest form of the command can be used. Use the following query to determine which log groups have been archived:

```
SQL> SELECT GROUP#, STATUS, ARCHIVED FROM V$LOG;
```

For example, the following command clears redo log group 3, which has already been archived:

```
SQL> ALTER DATABASE CLEAR LOGFILE GROUP 3;
```

If the redo log group has not been archived, then you must specify the UNARCHIVED keyword. This forces you to acknowledge that it is possible that there are backups that rely on that redo log for recovery, and you have decided to forgo that recovery opportunity. This may be satisfactory for you, especially if you take another backup right after you correct the redo log group problem; you then no longer need that redo log file.

It is possible that the redo log is required to recover a data file that is currently offline.

Re-creating a Password Authentication File

```
SQL> grant sysdba to admin2;
grant sysdba to admin2
*
ERROR at line 1:
ORA-01994: GRANT failed: password file missing or disabled
```

To recover from the loss of a password file:

1. Re-create the password file by using `orapwd`.

```
$ orapwd file=$ORACLE_HOME/dbs/orapworcl password=ora entries=5
```

2. Add users to the password file and assign appropriate privileges to each user.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Use the `orapwd` password utility to create a password file. When you connect using the SYSDBA privilege, you are connecting as the SYS schema and not the schema associated with your username. For SYSOPER, you are connected to the PUBLIC schema. Access to the database using the password file is provided by GRANT commands issued by privileged users.

Typically, the password file is not included in backups because, in almost all situations, it can be easily re-created.

It is critically important to the security of your system that you protect your password file and the environment variables that identify the location of the password file. Any user with access to these could potentially compromise the security of the connection.

You should not remove or modify the password file if you have a database or instance mounted using `REMOTE_LOGIN_PASSWORDFILE=EXCLUSIVE` or `SHARED`. If you do, you will be unable to reconnect remotely using the password file.

Note: Passwords are case-sensitive, so you must take that into consideration when re-creating the password file. Also, if the original password file was created with the `IGNORECASE=Y` option, then it must be re-created with the same option.

Using a Password File

The following are the steps for re-creating the password file:

1. Create the password file by using the password utility orapwd.

```
orapwd file=filename password=password entries=max_users
```

where:

- **filename** is the name of the password file (mandatory).
- **password** is the password for SYS (optional). You are prompted for the password if you do not include the **password** argument.
- **max_users** is the maximum number of distinct users allowed to connect as SYSDBA or SYSOPER. If you exceed this number, you must create a new password file. It is safer to have a larger number. There are no spaces around the “equal to” (=) character.

Example: orapwd file=\$ORACLE_HOME/dbs/orapwU15 password=admin
entries=5

Other options (in release 12.1 and later) include:

- **ASM**: Set to **y**, creates an Oracle ASM password file. The default **n** creates a database password file.
- **FORMAT**: Set to **12** (the default), the password file is created in the release 12c format. This format supports the SYSBACKUP, SYSDG, and SYSKM system privileges. If set to **legacy**, the password file is in the legacy format, which is the format before Oracle Database 12c. This argument cannot be set to **legacy** when the **SYSBACKUP** or the **SYSDG** argument is specified.
- **SYSBACKUP**: **y** creates a **SYSBACKUP** entry in the password file. You are prompted for the password. The password is stored in the created password file.
- **SYSDG**: **y** creates a **SYSDG** entry in the password file. You are prompted for the password. The password is stored in the created password file.
- **SYSKM**: **y** creates a **SYSKM** entry in the password file. You are prompted for the password. The password is stored in the created password file.
- **DELETE**: **y** deletes the specified password file. If set to the default **n**, then the specified password file is created.
- **FORCE**: **y** permits overwriting an existing password file.

For a complete list of options, see the *Oracle Database Administrator's Guide*.

2. Connect to the database by using the password file created in step 1, and grant privileges as needed.

```
SQL> CONNECT sys/admin AS SYSDBA  
SQL> grant sysdba to admin2;
```

Password File Locations

UNIX: \$ORACLE_HOME/dbs

Windows: %ORACLE_HOME%\database

Maintaining the Password File

Delete the existing password file by using operating system commands, and create a new password file by using the password utility.

Recovering from a Lost Index Tablespace

- A tablespace that contains only indexes may be recovered without performing a RECOVER task.
- If a data file that belongs to an index-only tablespace is lost, it may be simpler to re-create the tablespace and re-create the indexes.
- Use options to reduce the time it takes to re-create the index:
 - PARALLEL
 - NOLOGGING

```
SQL> CREATE INDEX rname_idx
  2  ON hr.regions (region_name)
  3  PARALLEL 4;
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Indexes are computed objects, in that they do not provide any original data, and they are only a different representation of data that already exists. So, in most cases, indexes can be re-created easily. If you have a tablespace that contains only indexes, recovering from a loss of a data file belonging to that tablespace can be simplified.

When a data file like this is lost, you can perform the following steps:

1. Drop the data file.
2. Drop the tablespace.
3. Re-create the index tablespace.
4. Re-create the indexes that were in the tablespace.

When creating or re-creating an index, you can use the following keywords to reduce the creation time:

- **PARALLEL:** Enables multiple processes to work together simultaneously to create an index. NOPARALLEL is the default.
- **NOLOGGING:** Makes index creation faster because it creates a very minimal amount of redo log entries as a result of the creation process.

Use the DBMS_METADATA package to retrieve metadata from the data dictionary to re-create indexes.

Recovering a Read-Only Tablespace

Special user-managed backup and recovery considerations for a read-only tablespace:

- You do not have to put it in backup mode in order to make a copy of its data files.
- You do not have to take the tablespace or data file offline before making a copy of it.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Because read-only tablespaces are not being written to, there are special considerations to take into account, which can make the recovery process faster and more efficient. You do not have to put a read-only tablespace into backup mode or take it offline before copying it to the backup location. Simply copy it.

When restoring a read-only tablespace, take the tablespace offline, restore the data files belonging to the tablespace, and then bring the tablespace back online.

Consider the following scenario, where a read-only tablespace is changed to be read/write:

1. Make a backup of a read-only tablespace.
2. Make the tablespace read/write.
3. Recover the tablespace.

The backup you made in step 1 can still be used to recover this tablespace, even though, since the backup was made, the tablespace was made read/write, and has even possibly been written to. In this case, the tablespace requires recovery, after the files are stored from such a backup.

Automatic Tempfile Recovery

SQL statements that require temporary space to execute may fail if one of the tempfiles is missing.

```
SQL> select * from big_table order by
1,2,3,4,5,6,7,8,9,10,11,12,13;
select * from big_table order by
1,2,3,4,5,6,7,8,9,10,11,12,13
*
ERROR at line 1:
ORA-01565: error in identifying file
'/u01/app/oracle/oradata/orcl/temp01.dbf'
ORA-27037: unable to obtain file status
Linux Error: 2: No such file or directory
```

Good news:

- Automatic re-creation of temporary files at startup
- Manual re-creation also possible



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

If a temporary file (tempfile) belonging to the temporary tablespace is lost or damaged, the extents in that file will not be available. This problem may manifest itself as an error during the execution of SQL statements that require temporary space for sorting.

The SQL statement shown in the slide has a long list of columns to order by, which results in the need for temporary space. The missing file error is encountered when this statement requiring a sort is executed.

The Oracle database instance can start up with a missing temporary file. If any of the temporary files do not exist when the database instance is started, they are created automatically and the database opens normally. When this happens, a message like the following appears in the alert log during startup:

Re-creating tempfile /u01/app/oracle/oradata/orcl/temp01.dbf

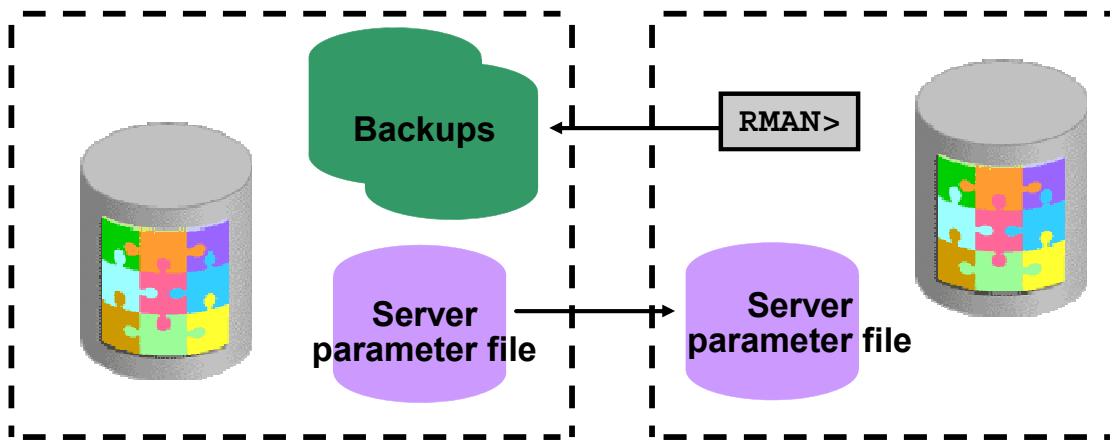
In the unlikely case that you decide a manual re-creation serves you better, use the following commands:

```
SQL> ALTER TABLESPACE temp ADD TEMPFILE
'/u01/app/oracle/oradata/orcl/temp02.dbf' SIZE 20M;
SQL> ALTER TABLESPACE temp DROP TEMPFILE
'/u01/app/oracle/oradata/orcl/temp01.dbf';
```

Restoring and Recovering the Database on a New Host

Use the procedure to:

- Perform test restores
- Move a production database to a new host



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Use the procedure described on the following pages to perform test restores. You can also use it to move a production database to a new host.

The database identifier (DBID) for the restored test database is the same as the DBID of the original database. If you are using a recovery catalog and connect to the test database and the recovery catalog database, the recovery catalog is updated with information about the test database. This can impact RMAN's ability to restore and recover the source database.

You should create a duplicate database by using the RMAN DUPLICATE command if your goal is to create a new copy of your target database for ongoing use on a new host. The duplicate database is assigned a new DBID that allows it to be registered in the same recovery catalog as the original target database. Refer to the lesson titled "Duplicating a Database" for detailed information about the DUPLICATE command.

Preparing to Restore the Database to a New Host

To prepare to restore a database, perform the following steps:

1. Record the database identifier (DBID) of your source database.
2. Copy the source database initialization parameter file to the new host.
3. Ensure that source backups, including the control file autobackup, are accessible on the restore host.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Perform the steps listed in the slide to prepare for the restore of the database to a new host.

Note: If you are performing a test restore, do not connect to the recovery catalog when restoring the data files. If you connect to the recovery catalog, RMAN records information about the restored data files in the recovery catalog and considers the restored database as the current target database. If your control file is not large enough to contain all of the RMAN repository data on the backups you need to restore and you must use a recovery catalog, then export the catalog and import it into a different schema or database. Use the copied recovery catalog for the test restore.

Restoring the Database to a New Host

Perform the following steps on the restore host to restore the database:

1. Configure the ORACLE_SID environment variable.
2. Start RMAN and connect to the target instance in NOCATALOG mode.
3. Set the database identifier (DBID).
4. Start the instance in NOMOUNT mode.
5. Restore the server parameter file from the backup sets.
6. Shut down the instance.
7. Edit the restored initialization parameter file.
8. Start the instance in NOMOUNT mode.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Perform the steps listed on this page and the next on the restore host to restore the database.

1. Configure the ORACLE_SID environment variable as shown in the following example:
\$ setenv ORACLE_SID orcl
 2. Start RMAN and connect to the target instance. Do not connect to the recovery catalog as shown in the following example:
\$ rman TARGET /
 3. Set the database identifier (DBID). You can find the DBID of your source database by querying the DBID column in V\$DATABASE.
RMAN> SET DBID 1090770270;
 4. Start the instance in NOMOUNT mode:
RMAN> STARTUP NOMOUNT
- You will receive an error similar to the following because the server parameter file has not been restored. RMAN uses a “dummy” parameter file to start the instance.
- ```
startup failed: ORA-01078: failure in processing system parameters
```

5. Restore the server parameter file from the backup sets and shut down the instance as shown in the example:

```
RESTORE SPFILE TO PFILE '?/oradata/test/inititorcl.ora' FROM
AUTOBACKUP;
```

6. Shut down the instance:

```
SHUTDOWN IMMEDIATE;
```

7. Edit the restored initialization parameter file to change any location-specific parameters, such as those ending in \_DEST, to reflect the new directory structure.

8. Start the instance in NOMOUNT mode using your edited text initialization parameter file.

```
RMAN> STARTUP NOMOUNT
> PFILE='?/oradata/test/inititorcl.ora';
```

## Restoring the Database to a New Host

9. Create a RUN block to:
  - Restore the control file
  - Mount the database
10. Create the RMAN recovery script to restore and recover the database.
11. Execute the RMAN script.
12. Open the database with the RESETLOGS option.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

9. Create a RUN block to restore the control file from an autobackup and mount the database as shown in the example:

```
RUN
{
 RESTORE CONTROLFILE FROM AUTOBACKUP ;
 ALTER DATABASE MOUNT ;
}
```

10. Query V\$DATAFILE on your new host to determine the database file names as recorded in the control file. Create the RMAN recovery script to restore and recover the database, including the following steps as appropriate:

- a. Use the SET NEWNAME command to specify the path on your new host for each of the data files that is restored to a different destination than on the original host.
- b. Use the SQL ALTER DATABASE RENAME FILE command to specify the path for the online redo log files.
- c. Include the SET UNTIL command to limit recovery to the end of the archived redo log files.
- d. Include the SWITCH command so that the control file recognizes the new path names as the correct names for the data files.

An example of a recovery script follows:

```
RUN
{
SET NEWNAME FOR DATAFILE 1 TO '?/oradata/test/system01.dbf';
SET NEWNAME FOR DATAFILE 2 TO '?/oradata/test/undotbs01.dbf';
SET NEWNAME FOR DATAFILE 3 TO '?/oradata/test/sysaux.dbf';
SET NEWNAME FOR DATAFILE 4 TO '?/oradata/test/users01.dbf';
SET NEWNAME FOR DATAFILE 5 TO '?/oradata/test/example01.dbf';
SQL "ALTER DATABASE RENAME FILE
'#/u01/app/oracle/oradata/orcl/redo01.log'
TO '?/oradata/test/redo01.log' ";
SQL "ALTER DATABASE RENAME FILE
'#/u01/app/oracle/oradata/orcl/redo02.log'
TO '?/oradata/test/redo02.log' ";
SQL "ALTER DATABASE RENAME FILE
'#/u01/app/oracle/oradata/orcl/redo03.log'
TO '?/oradata/test/redo03.log' ";
SET UNTIL SCN 4545727;
RESTORE DATABASE;
SWITCH DATAFILE ALL;
RECOVER DATABASE;
}
```

11. Execute the recovery script.

12. Open the database with the RESETLOGS option:

```
RMAN> ALTER DATABASE OPEN RESETLOGS;
```

After you have completed your test, you can shut down the test database instance and delete the test database with all its files.

## Performing Disaster Recovery

- Disaster implies the loss of the entire target database, the recovery catalog database, all current control files, all online redo log files, and all parameter files.
- Disaster recovery includes the restoration and recovery of the target database.
- Minimum required set of backups:
  - Backups of data files
  - Corresponding archived redo logs files
  - At least one control file autobackup



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Disaster recovery includes the restoration and recovery of the target database after the loss of the entire target database, all current control files, all online redo log files, all parameter files, and the recovery catalog database (if applicable).

To perform disaster recovery, the following backups are required as a minimum:

- Backups of data files
- Corresponding archived redo logs generated after the time of the backup
- At least one autobackup of the control file

**Note:** Refer to the *Oracle Data Guard Concepts and Administration* manual for information about how Oracle Data Guard can provide complete disaster protection.

## Performing Disaster Recovery

Basic procedure:

1. Restore an autobackup of the server parameter file.
2. Start the target database instance.
3. Restore the control file from autobackup.
4. Mount the database.
5. Restore the data files.
6. Recover the data files.
7. Open the database with the `RESETLOGS` option.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The basic procedure for performing disaster recovery is outlined in the slide. After you have mounted the database, follow the steps for performing recovery with a backup control file.

# Restoring Encrypted Backups

- Before restoration, set the RMAN session to decrypt backups.
- Specify all required passwords with the SET DECRYPTION command when restoring from a set of backups that were created with different passwords.

```
SET DECRYPTION IDENTIFIED BY '<password_1>'
{, '<password_2>', ..., '<password_n>' }
```

## Notes

- If you lose the password for a password-encrypted backup, you cannot restore that backup.
- If you lose the keystore containing the key for a transparent encrypted backup, you cannot restore that backup.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Use the SET DECRYPTION command to specify one or more decryption passwords to be used when reading dual-mode or password-encrypted backups. When RMAN reads encrypted backup pieces, it tries each password in the list until it finds the correct one to decrypt that backup piece. An error is signaled if none of the specified keys are correct.

If you lose the password for a password-encrypted backup, you cannot restore that backup.

Because the Oracle key management infrastructure archives all previous master keys in the keystore (or wallet), changing or resetting the current database master key does not affect your ability to restore encrypted backups performed using an older master key. You may reset the database master key at any time, but RMAN will always be able to restore all encrypted backups that were ever created by this database.

If you lose the keystore containing the key for a transparent encrypted backup, you cannot restore that backup. Because the keystore contains all past backup encryption keys, a restored keystore can be used to restore past encrypted backups up to the backup time of the wallet. But encrypted backups made after the keystore backup will be not accessible.

**Best Practice Tip:** Back up the keystore frequently.

## Quiz

Which of the following locations can be used to obtain a block during block media recovery?

- a. Flashback logs
- b. Full backups
- c. Incremental level 0 backups
- d. Incremental level 1 backups



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

**Answer: a, b, c**

## Summary

In this lesson, you should have learned how to:

- Recover from the loss of the server parameter file
- Recover from control file and redo log file failures
- Re-create the password authentication file
- Recover index and read-only tablespaces
- Review the automatic recovery of the tempfile
- Describe the basic procedure of restoring the database to a new host
- Describe disaster recovery



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

## Practice Overview: Performing Recoveries

- Practice 12-1 covers restoring a lost parameter file.
- Practice 12-2 covers restoring a single control file.
- Practice 12-3 covers restoring all copies of the control file.
- Practice 12-4 covers restoring the database password file.
- Practice 12-5 covers reviewing the automatic recovery when a tempfile is missing.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In these practices, you will restore a database after various losses. You can choose your own sequence of practices, but after you start one you must complete it.

## Practice Overview: Using RMAN Encryption

- Practice 12-6 covers the following topics:
  - Preparing the database for encryption
  - Creating a transparent encrypted backup
- Practice 12-7 covers recovering a lost data file by using an encrypted backup.
- Practice 12-8 covers recovering a lost encryption wallet.

**Note:** If you lose the wallet and do not have a backup of it, you will have to recover the database to a **point in time before the wallet was used**.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

- In the first practice, you create an encrypted backup that is protected against data breach if the backup media is lost. In this example, you will be using transparent encryption, which depends on an encryption wallet. If the encryption wallet is lost, the backup is not recoverable. To mitigate the loss of a wallet or to allow the backup to be recovered on a different machine, you can use password encryption instead of transparent encryption, or use both so that either the wallet or the password will allow the backup to be recovered.
- In the second practice, you will recover a lost data file by using an encrypted backup.
- In the third practice, you will recover a lost encryption wallet.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and Error : You are not a Valid Partner use only