Networking Overview
Objectives

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

• Explain solutions included with Oracle9i for managing complex networks
• Describe Oracle networking add-on solutions
Network Environment Challenges

- Configuring the network environment
- Maintaining the network
- Tuning, troubleshooting, and monitoring the network
- Implementing security in the network
- Integrating legacy systems
Simple Network: Two-Tier

- Network connects client and server
- Client and server speak the same "language" or protocol
Simple to Complex Network: N-Tier

- Client can be a thin client or a PC
- Middle tier can contain applications and services
- Server holds actual data
Oracle9i Networking Solutions

- Connectivity
- Directory Services
- Scalability
- Security
- Accessibility
Connectivity: Oracle Net Services

- Protocol independence
- Comprehensive platform support
- Integrated GUI administration tools
- Multiple configuration options
- Tracing and diagnostic toolset
- Basic security
Connectivity: Oracle Net Services

- Oracle Net Client
- Oracle Net Server

Any platform

Administration and configuration
Connectivity: Database Connectivity with IIOP and HTTP

Database connectivity can be achieved using the following additional protocols:

- Internet Inter-ORB Protocol (IIOP)
- Hypertext Transfer Protocol (HTTP)
Directory Naming

• Directory naming is the process of resolving a network alias using an LDAP-compliant directory server.
• Clients must be configured to use the LDAP compliant server.
Directory Services: Oracle Internet Directory

Oracle Internet Directory is Oracle’s LDAP compliant directory service. It provides the following features:

• Integrates tightly with Oracle9i
• Simplifies network administration
• Provides a secure and reliable directory structure
Scalability: Oracle Shared Server

- The Oracle Shared Server enables a large number of users to connect to a database simultaneously.
- Database resources are shared resulting in efficient memory and processing usage.
- Connections are routed via a dispatcher.
- Server processes are not dedicated to each client.
- Server processes serve client processes as needed.
Scalability: Connection Manager

Connection Manager offers:
- Multiplexing of connections
- Cross-protocol connectivity
- Network access control
Scalability: Connection Manager
Security: Advanced Security

- **Encryption**
  - Encodes between network nodes
  - DES, RSA, 3DES
- **Authentication**
  - Authenticates users through third-party services and Secure Sockets Layer (SSL)
    - Kerberos, Radius, CyberSafe
- **Data Integrity**
  - Ensures data integrity during transmission
    - MD5, SHA
Corporate earnings are up 45% this quarter

1. Oracle Advanced Security Installed

2. Encrypt

fdh37djf246gs’b[da,\ssk

Corporate earnings are up 45% this quarter

3. Oracle Advanced Security Installed

Client

Server
Security: Oracle Net Services and Firewalls

- Oracle works with key firewall vendors to provide firewall support
- Oracle Net Application Proxy Kit allows firewall vendors to provide connection support for Oracle environments
- Oracle Net Application Proxy is based on Connection Manager
- Oracle supports two categories of firewalls:
  - Proxy based firewalls
  - Stateful packet inspection firewalls
Accessibility: Heterogeneous Services

• Enables access of legacy data as if it resides in a single, local relational database
• Enables Oracle procedure calls to access non-Oracle systems, services, or APIs
Accessibility: External Procedures

• External procedures are functions written in a 3GL language that can be called from PL/SQL.
• Support of external procedures allows the developer more flexibility than SQL or PL/SQL provide.
• The Oracle listener can listen for external procedure calls.
• Connections to external procedure can be configured during or after server installation.
Welcome to the Oracle Net Configuration Assistant. This tool takes you through the following common configuration steps:

Choose the configuration you would like to do:

- Listener configuration
- Naming Methods configuration
- Local Net Service Name configuration
- Directory Usage Configuration
Welcome to the Oracle Net Manager! The Oracle Net Manager allows you to configure the following aspects of the network:

- Naming: Allows you to define simple names to identify the location of a service, such as a database. These simple names map to connect descriptors, which contain the network location and identification of the service.
- Naming Methods: Configure the different ways in which simple names are resolved into connect descriptors.
- Listeners: Create and configure listeners to receive client connections.
Summary

In this lesson, you should have learned how to:

• Explain Oracle’s solutions for managing complex networks:
  – Oracle Net Services
  – IIOP and HTTP Connectivity
  – Oracle Internet Directory
  – Oracle Shared Server
  – Connection Manager

• Describe Oracle’s add-on solutions:
  – Oracle Advanced Security
  – Heterogeneous Services
Oracle Net Architecture
Objectives

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

• Explain the key components of the Oracle Net stack communication architecture
• Explain Oracle Net’s role in client/server connections
• Describe how Web client connections are established through Oracle networking products
Oracle Net Connections

• Oracle Net is used to establish connections between applications on a network depending on the following:
  – The network configuration
  – The location of the nodes
  – The application
  – The network protocol

• The connections types can be:
  – Client/Server Application
  – Java Application
  – Web Client Application
Client-Server Application Connection: No Middle-Tier

Client

Client Application (uses OCI)

Two Task Common

Oracle Net Foundation Layer

Oracle Protocol Support

Network Protocol

Database Server

Oracle RDBMS (uses OPI)

Two Task Common

Oracle Net Foundation Layer

Oracle Protocol Support

Network Protocol
Web Client Application Connections

Web browsers can connect to an Oracle server in the following ways:

• Using a Web Server as a middle tier configured with:
  – JDBC Oracle Call Interface (OCI) driver
  – Thin JDBC driver

• Connecting directly to an Oracle server by using:
  – IIOP
  – HTTP
Web Client Application Connection: Web Server Middle-Tier

- Java application or applet
- Web browser
- HTTP
- User
- Application Web server
- Oracle Net
- Server
- Oracle server
Web Client Application Connection: Java Application Client

Application Web Server (client)

- Java application
- JDBC OCI Driver
- TTC
- Oracle Net Foundation Layer
- Oracle Protocol Stack
- Network Protocol

Database Server

- Oracle RDBMS
- TTC
- Oracle Net Foundation Layer
- Oracle Protocol Support
- Network Protocol
Web Client Application Connection: Java Applet Client

Application Web Server (client)

- Java Applet
- JDBC Thin driver
- JavaTTC
- JavaNet
- TCP/IP Network Protocol

Database Server

- Oracle RDBMS
- TTC
- Oracle Net Foundation Layer
- OPS
- TCP/IP Network Protocol
Web Client Application Connection: No Middle Tier

Client

Web browser

HTTP

Oracle
Server
supporting
HTTP and IIOP

IIOP

Web browser

Client
Web Client Application Connection: No Middle-Tier
Connectivity Concepts and Terminology

• Database services
• Service Name
  – A logical representation of a database
  – The way a database is presented to clients
• Connect Descriptor
  – Location of the database
  – Name of the database service
• Listener
  – Receives client connection requests
  – Hands requests to the database server
Connectivity Concepts and Terminology

• Service Registration
  – Database registers information with the listener
  – Service handlers available for each instance

• Service Handlers
  – Connection points
  – Dispatcher or dedicated server
Oracle Net Configuration Models

- Localized management
  - Local file on each computer in the network
- Centralized management
  - LDAP-compliant directory server
  - Oracle Names server
Oracle Net Configuration Files

- ldap.ora
- listener.ora
- names.ora
- sqlnet.ora
- tnsnames.ora
Summary

In this lesson, you should have learned how to:

- Explain the key components of the Oracle Net stack communication architecture
- Explain Oracle Net Services role in client server connections
- Describe how Web client connections are established through Oracle networking products
Basic Oracle Net Server
Side Configuration
Objectives

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

• Identify how the listener responds to incoming connections
• Describe Dynamic Service Registration
• Configure the listener by using Oracle Net Manager
• Control the listener by using the Listener Control Utility (**lsnrctl**) 
• Configure the listener for IIOP and HTTP connections
The Listener Process

Client

Listening Process

Server
Connection Methods

When a connection request is made by a client to a server, the listener performs one of the following:

- Spawns a server process and bequeaths (passes) the connection to it
- Hands off the connection to a dispatcher in an Oracle Shared Server configuration
- Redirects the connection to a dispatcher or server process
Spawn and Bequeath and Direct Hand-Off Connections

1. Client
2. Listener
3. Spawns server process
4. Client connects to listener
5. Listener forwards connection to spawned server process
Service Configuration and Registration

The listener can be configured in two ways:

- **Dynamic service registration**
  - Does not require configuration in `listener.ora` file
  - The listener relies on the PMON process

- **Static service configuration**
  - Used for Oracle8 and earlier releases
  - Requires `listener.ora` configuration
  - Required for Oracle Enterprise Manager and other services
Static Service Registration: The *listener.ora* File

When the Oracle software is installed, the *listener.ora* file is created for the starter database with the following default settings:

- **Listener name**: LISTENER
- **Port**: 1521
- **Protocols**: TCP/IP and IPC
- **SID name**: Default instance
- **Host name**: Default host name
Static Service Registration:
The listener.ora File

1. LISTENER =
2. (ADDRESS_LIST =
3.   (ADDRESS= (PROTOCOL=TCP)(Host=stc-sun02)(Port=1521)))
4. SID_LIST_LISTENER =
5.   (SID_LIST =
6.     (SID_DESC =
7.       (ORACLE_HOME= /home/oracle)
8.       (GLOBAL_DBNAME = ORCL.us.oracle.com)
9.       (SID_NAME = ORCL))))
Static Service Registration: Create a Listener

The Listeners folder allows you to configure one or more listeners in the LISTENER.ORA file.

A listener is configured to “listen on” one or more network protocols. Once started, the listener responds to connection requests on behalf of its registered database or non-database services.

To see if a listener has been created for this host: Double-click the Listeners folder. If no listeners exist, click “+” on the toolbar or choose Edit > Create.

See Also: “Local > Listeners” in the help contents.
Configure Services
Dynamic Service Registration: Configure Registration

To ensure that service registration is functional, the following initialization parameters must be configured:

• SERVICE_NAMES
• INSTANCE_NAME
Dynamic Service Registration:
Registering Information with the Listener

• By default, PMON registers with a local listener on the server on the default local address of TCP/IP, port 1521.

• PMON can register with a non default listener if:
  – LOCAL_LISTENER initialization parameter is defined
  – LISTENERS attribute of the DISPATCHERS initialization parameter is defined for Oracle Shared Server
Configure the Listener for Oracle9i JVM: IIOP and HTTP

- The listener can be configured to accept connections from clients using IIOP and HTTP.
- Use Static Listener Registration if the database is Oracle8i or an earlier version, even if Oracle9i listener is used.
- If both the listener and the database are Oracle9i, configuration occurs dynamically during service registration.
Listener Control Utility (LSNRCTL)

Listener Control Utility commands can be issued from the command-line or from the LSNRCTL prompt.

• UNIX command-line syntax:
  $ lsnrctl <command name>

• Prompt syntax:
  LSNRCTL> <command name>

• Control a non-default listener
  LSNRCTL> set current_listener listener02
LSNRCTL Commands

Use the following commands to control the listener:

• START [listener_name]
• STOP [listener_name]
LSNRCTL SET and SHOW Modifiers

• Change listener parameters with SET:

```
LSNRCTL> SET trc_level ADMIN
```

• Display the values of parameters with SHOW:

```
LSNRCTL> SHOW trc_directory
```
Summary

In this lesson, you should have learned how to:

- Configure the listener by using Oracle Net Manager
- Control the listener by using the Listener Control Utility (lsnrctl)
- Configure the listener for IIOP and HTTP connections
Practice 3 Overview

This practice covers the following topics:

• Configuring a non-default LISTENER
• Starting and stopping your listener
• Viewing the LISTENER log file
Naming Method Configuration
Objectives

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

• Describe the difference between host naming and local service name resolution

• Use Oracle Net Configuration Assistant to configure:
  – Host Naming method
  – Local naming method
  – Net service names

• Perform simple connection troubleshooting
Overview of Naming Methods

- Naming methods are used by a client application to resolve a connect identifier to a connect descriptor when attempting to connect to a database service.
- Oracle Net provides five naming methods:
  - Host naming
  - Local naming
  - Directory naming
  - Oracle Names
  - External naming
Host Naming

Clients can connect to a server using a host name under the following conditions:

• Connecting to an Oracle database service using Oracle Net Services Client software
• Client and server are connecting using TCP/IP protocol
• Host names are resolved through an IP address translation mechanism such as DNS or a local /etc/hosts file
• No advanced features such as Connection Manager or security options are used
Host Naming: Client Side

Client

TCP/IP

Server

... names.directory_path = (HOSTNAME)

sqlnet.ora

listener.ora
Host Naming: Server Side

Client

sqlnet.ora

TCP/IP

Server

SID_LIST_LISTENER =
(SID_LIST =
(SID_DESC =
 (GLOBAL_DBNAME = stc-sun02.us.oracle.com)
 (ORACLE_HOME = /u03/ora9i/rel12)
 (SID_NAME = TEST)
)
)

listener.ora
Host Naming Example

- **listener.ora file**

```
SID_LIST_LISTENER =
(SID_LIST =
(SID_DESC =
  (GLOBAL_DBNAME = stc-sun02.us.oracle.com)
  (ORACLE_HOME = /u03/ora9i/re112)
  (SID_NAME = TEST))
```

- **Connecting from the client**

```
sqlplus system/manger@stc-sun02.us.oracle.com
```
Naming Methods Configuration

Welcome to the Oracle Net Configuration Assistant. This tool takes you through the following common configuration steps:

Choose the configuration you would like to do:

- Listener configuration
- Naming Methods configuration
- Local Net Service Name configuration
- Directory Usage Configuration

Cancel  Help  Back  Next  Finish
Selecting the Host Naming Method

When connecting to a remote database or other service you specify a net service name. This name is resolved using one or more naming methods into the information necessary to locate and connect to the database or service.

Select the naming methods you want to use for resolving net service names and the order in which you want them used. Keep your configuration as simple as possible by configuring only the naming methods you need.

Recommended naming methods have been preselected for you.

Available Naming Met...
- Local
- Oracle Names
- Sun NIS
- DCE CDS

Selected Naming Meth...
- Host Name
Local Naming

Client
- tnsnames.ora
- sqlnet.ora

Server
- listener.ora
Selecting the Local Naming Method

When connecting to a remote database or other service you specify a net service name. This name is resolved using one or more naming methods into the information necessary to locate and connect to the database or service.

Select the naming methods you want to use for resolving net service names and the order in which you want them used. Keep your configuration as simple as possible by configuring only the naming methods you need.

Recommended naming methods have been preselected for you.

Available Naming Methods:
- Sun NIS
- DCE CDS
- Oracle Names

Selected Naming Methods:
- Local
- Host Name
Configuring Local Net Service Names

Welcome to the Oracle Net Configuration Assistant. This tool takes you through the following common configuration steps:

Choose the configuration you would like to do:

- Listener configuration
- Naming Methods configuration
- Local Net Service Name configuration
- Directory Usage Configuration
Working with Net Service Names

To access an Oracle database, or other service, across the network you use a net service name. The Oracle Net Configuration Assistant allows you to work with net service names resolved using local naming.

Select what you want to do:

- Add
- Reconfigure
- Delete
- Rename
- Test
Specify the Oracle Database Version

What version of Oracle database or service do you want to access?

- Oracle8i or later database or service
- Oracle8 release 8.0 or Oracle7 database or service
Database Service Name

For an Oracle8i or later database or service you must provide its service name. An Oracle8i or later database’s service name is normally its global database name.

Service Name: TEST
Network Protocol

To communicate with the database across a network, a network protocol is used. Select the protocol used for the database you want to access.

TCP
TCPS
IPC
Host Name and Listener Port

To communicate with the database using the TCP/IP protocol, the database computer’s host name is required. Enter the host name for the computer where the database is located.

Host name: stc-sun02.us.oracle.com

A TCP/IP port number is also required. In most cases the standard port number should be used.

- Use the standard port number of 1521
- Use another port number: 1521
Testing the Connection

You can verify that an Oracle database can be reached, using the information provided, by performing a connection test.

Would you like to test that a connection can be made to the database?

- No, do not test
- Yes, perform a test
Connection Test Result

Wait while the Oracle Net Configuration Assistant tries to connect to the database using the information you provided...

Details:
Connecting...Test successful.
Net Service Name

Choose a name for this net service name. The Oracle Net Configuration Assistant has defaulted the net service name to be the same as the service name you entered earlier, but you can change it to be any name you choose.

Net Service Name: my_service
Save the Net Service Name

Would you like to configure another net service name?

- No
- Yes
Generated Files:
      tnsnames.ora

# TNSNAMES.ORA Network Configuration
# File:/u03/ora9i/rel12/network/admin/tnsnames.ora
# Generated by Oracle configuration tools.
MY_SERVICE.US.ORACLE.COM =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS =
        (PROTOCOL = TCP)(HOST = stc-sun02.us.oracle.com)(PORT = 1521))
    )
  )
  (CONNECT_DATA =
    (SERVICE_NAME = TEST.us.oracle.com)
  )
)
Generated Files:
sqlnet.ora

# SQLNET.ORA Network Configuration File:
/u03/ora9i/rel12/network/admin/sqlnet.ora
# Generated by Oracle configuration tools.
NAMES.DEFAULT_DOMAIN = us.oracle.com
NAMES.DIRECTORY_PATH = (TNSNAMES, HOSTNAME)
SQLNET.EXPIRE_TIME=0

sqlplus system/manager@MY_SERVICE
(c) Copyright 2001 Oracle Corporation. All rights reserved.
Connected to:
Oracle9i Enterprise Edition Release 9.0.1.0.0 - Production
JServer Release 9.0.1.0.0 - Production
SQL>
## Troubleshooting the Client Side

The following error codes are related to problems on the client side:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORA-12154</td>
<td>&quot;TNS:could not resolve service name&quot;</td>
</tr>
<tr>
<td>ORA-12198</td>
<td>&quot;TNS:could not find path to destination&quot;</td>
</tr>
<tr>
<td>ORA-12203</td>
<td>&quot;TNS:unable to connect to destination&quot;</td>
</tr>
<tr>
<td>ORA-12533</td>
<td>&quot;TNS:illegal ADDRESS parameters&quot;</td>
</tr>
<tr>
<td>ORA-12541</td>
<td>&quot;TNS:no listener&quot;</td>
</tr>
</tbody>
</table>
Summary

In this lesson, you should have learned how to:

• Describe the difference between host naming and local service name resolution

• Use Oracle Net Configuration Assistant to configure:
  – Host naming method
  – Local naming method
  – Net service names

• Perform simple connection troubleshooting
Practice 4 Overview

This practice covers the following topics:

• Local Naming configuration
• Configuration of net service names
• Testing the configuration
5

Usage and Configuration of the Oracle Shared Server
Objectives

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

• Identify the components of the Oracle Shared Server
• Describe the Oracle Shared Server architecture
• Configure the Oracle Shared Server
• Identify and explain usefulness of related data dictionary views
Server Configurations

• Dedicated server process
• Shared server process
Dedicated Server Processes

Client

User process

Server process

Server

Instance

SGA

SMON
DBWn
PMON
CKPT
LGWR
ARCn
Benefits of Oracle Shared Server

- Reduces the number of processes against an instance
- Increases the number of possible users
- Achieves load balancing
- Reduces the number of idle server processes
- Reduces memory usage and system overhead
Using a Dedicated Server with Oracle Shared Server

TEST.world =
 (DESCRIPTION =
   (ADDRESS =
     (PROTOCOL = TCP)
     (HOST = stc-sun02)
     (PORT = 1521)
   )
)

(CONNECT_DATA=(SERVICE_NAME=TEST.us.oracle.com)
   (SERVER=DEDICATED)
)
)
Processing a Request

Shared server processes

Instance

SGA

Response queue for (D001)

Response queue for (D002)

Response queue for (D003)

Request Queue

Dispatcher D001

Dispatcher D002

Dispatcher D003

User process

User process

Database server client

Listener

1 2 3 4 5 6

1. Listener
2. User process
3. Shared server processes
4. Request Queue
5. Response queue for (D001)
6. Response queue for (D002)

Database server

Copyright © Oracle Corporation, 2001. All rights reserved.
The SGA and PGA

Dedicated Server: User session data is kept in the PGA

Oracle Shared Server: User session data is held in the SGA
Configuring Oracle Shared Server

• Required Initialization Parameters
  – DISPATCHERS
  – SHARED_SERVERS

• Optional Initialization Parameters
  – MAX_DISPATCHERS
  – MAX_SHARED_SERVERS
  – CIRCUITS
  – SHARED_SERVER_SESSIONS
DISPATCHERS

Specifies the number of dispatchers initially started for a given protocol

DISPATCHERS = "(PROTOCOL=TCP) (DISPATCHERS=2) \n(PROTOCOL=IPC) (DISPATCHERS=1)"

Dispatcher D001
TCP/IP

Dispatcher D002
TCP/IP

Dispatcher D003
IPC
MAX_DISPATCHERS

- Specifies the maximum number of dispatcher processes that can run simultaneously
- Issue the `ALTER SYSTEM` command to add more dispatchers than initially started

MAX_DISPATCHERS = 5

Dispatcher D001 TCP/IP
Dispatcher D002 TCP/IP
Dispatcher D003 IPC
Dispatcher D004
Dispatcher D005
SHARED_SERVERS

Specifies the number of server processes created when an instance is started up

\[
\text{SHARED\_SERVERS} = 6
\]
MAX_SHARED_SERVERS

- Specifies the maximum number of shared servers that can be started
- Allows shared servers to be allocated dynamically based on the length of the request queue

MAX_SHARED_SERVERS = 10
CIRCUITS

- Specifies the total number of virtual circuits that are available for inbound and outbound network sessions
- Contributes to total SGA size

CIRCUITS = 100
SHARED_SERVER_SESSIONS

- Specifies the total number of Oracle Shared Server user sessions to allow.
- Setting this parameter enables you to reserve user sessions for dedicated servers.

```
SHARED_SERVER_SESSIONS = 100
```
Related Parameters

Other initialization parameters affected by Oracle Shared Server that may require adjustment:

- LARGE_POOL_SIZE
- SESSIONS
Verifying Setup

• Verify that the dispatcher has registered with the listener when the instance was started by issuing:

   $ lsnrctl services

• Verify that you are connected using shared servers by making a single connection then query V$CIRCUIT view to show one entry per shared server connection.
Dynamic Views

- V$CIRCUIT
- V$SHARED_SERVER
- V$DISPATCHER
- V$SHARED_SERVER_MONITOR
- V$QUEUE
- V$SESSION
Summary

In this lesson, you should have learned how to:

• Identify the components of the Oracle Shared Server
• Describe the Oracle Shared Server architecture
• Configure the Oracle Shared Server
• Identify and explain usefulness of related data dictionary views
Practice 5 Overview

This practice covers the following topics:

• Configuring Oracle Shared Server
• Defining LOCAL_LISTENER for instance registration
• Using the Listener Control utility to verify services
• Verifying shared server configuration and performance using V$ views
• Verifying instance registration
Backup and Recovery Overview
Objectives

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

• Describe the basics of database backup, restore, and recovery
• List the types of failure that may occur in an Oracle environment
• Define a backup and recovery strategy
Backup and Recovery Issues

• Protect the database from numerous types of failures
• Increase Mean-Time-Between-Failures (MTBF)
• Decrease Mean-Time-To-Recover (MTTR)
• Minimize data loss
Categories of Failures

- Statement failure
- User process failure
- User error
- Network failure
- Instance failure
- Media failure
Causes of Statement Failures

• Logic error in an application
• Attempt to enter invalid data into the table
• Attempt an operation with insufficient privileges
• Attempt to create a table but exceed allotted quota limits
• Attempt an INSERT or UPDATE to a table, causing an extent to be allocated, but with insufficient free space available in the tablespace
Resolutions for Statement Failures

- Correct the logical flow of the program.
- Modify and reissue the SQL statement.
- Provide the necessary database privileges.
- Change the user’s quota limit by using the `ALTER USER` command.
- Add file space to the tablespace.
- Enable resumable space allocation.
Causes of User Process Failures

- The user performed an abnormal disconnect in the session.
- The user’s session was abnormally terminated.
- The user’s program raised an address exception, which terminated the session.
Resolution of User Process Failures

- The PMON process detects an abnormally terminated user process.
- PMON rolls back the transaction and releases any resources and locks being held by it.
Possible User Errors

SQL> DROP TABLE employees;

SQL> TRUNCATE TABLE employees;

SQL> DELETE FROM employees;
SQL> COMMIT;

SQL> UPDATE employees
2>   SET salary = salary * 1.5;
SQL> COMMIT;
Resolution of User Errors

- Train the database users.
- Recover from a valid backup.
- Import the table from an export file.
- Use LogMiner to determine the time of error.
- Recover with a point-in-time recovery.
- Use LogMiner to perform object-level recovery.
- Use FlashBack to view and repair historical data.
Causes of Instance Failure

Instance
  SGA
    Locks
    Data buffer
    Data dict. cache
  Shared pool
  Shared SQL and PL/SQL

User process
  Server process
    PGA

Parameter file
Password file

146 Datafile 1
146 Datafile 2
146 Datafile 3

Database

146 Control files
145 Redo log file 1
145 Redo log file 2
Recovery from Instance Failure

- No special recovery action is needed from DBA.
- Start the instance.
- Wait for the “database opened” notification.
- Notify users.
- Check the alert log to determine the reason for the failure.
Causes of Media Failures

- Head crash on a disk drive
- Physical problem in reading from or writing to database files
- File was accidentally erased
Resolutions for Media Failures

• The recovery strategy depends on which backup method was chosen and which files are affected.
• If available, apply archived redo log files to recover data committed since the last backup.
Defining a Backup and Recovery Strategy

- Business requirements
- Operational requirements
- Technical considerations
- Management concurrence
Business Requirements

• Mean time to recover
• Mean time between failure
• Evolutionary process
Operational Requirements

- 24-hour operations
- Testing and validating backups
- Database volatility
Technical Considerations

- Resources: hardware, software, manpower, and time
- Physical image copies of the operating system files
- Logical copies of the objects in the database
- Database configuration
- Transaction volume which affects desired frequency of backups
Disaster Recovery Issues

• How will your business be affected in the event of a major disaster?
  – Earthquake, flood, or fire
  – Complete loss of machine
  – Malfunction of storage hardware or software
  – Loss of key personnel, such as the database administrator

• Do you have a plan for testing your strategy periodically?
Summary

In this lesson, you should have learned how to:

• Evaluate potential failures in your environment
• Develop a strategy dictated by business, operational, and technical requirements
• Consider a test plan for a backup and recovery strategy
Instance and Media Recovery Structures
Objectives

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

• Describe the Oracle processes, memory structures, and files relating to recovery
• Identify the importance of checkpoints, redo log files, and archived log files
• Describe ways to tune instance recovery
Large Pool

- Can be configured as a separate memory area in the SGA to be used for:
  - Oracle backup and restore operations
  - I/O server processes
  - Session memory for the shared servers
- Is sized by the `LARGE_POOL_SIZE` parameter
Database Buffer Cache, DBWn, and Datafiles

Instance

SGA

Java pool
Large pool
Database buffer cache
Redo log buffer

Shared pool
Shared SQL and PL/SQL
Data dict. cache

SMON
PMON
CKPT
LGWR
ARCn

User process

Server process

PGA

Database

Datafile 1
Datafile 2
Datafile 3

Control file
Redo log file 1
Redo log file 2

Parameter file
Password file

Archived log files

Copyright © Oracle Corporation, 2001. All rights reserved.
Redo Log Buffer, LGWR, and Redo Log Files
Multiplexed Redo Log Files

- **Group 1**
  - Disk 1 (Member a)
    - log1a.rdo
  - Disk 2 (Member b)
    - log1b.rdo

- **Group 2**
  - log2a.rdo
  - log2b.rdo

- **Group 3**
  - log3a.rdo
  - log3b.rdo
Redo Log Files in Enterprise Manager
Checkpointing

- Checkpoints are used to determine where recovery should start.
- Checkpoint position – where recovery starts
- Checkpoint queue – link list of dirty blocks
Types of Checkpoints

- Full checkpoint
  - All dirty buffers are written
  - `SHUTDOWN NORMAL, IMMEDIATE, OR TRANSACTIONAL`
  - `ALTER SYSTEM CHECKPOINT`
- Incremental checkpoint (Fast-Start checkpoint)
  - Periodic writes
  - Only write the oldest blocks
- Partial checkpoint
  - Dirty buffers belonging to the tablespace
  - `ALTER TABLESPACE BEGIN BACKUP`
  - `ALTER TABLESPACE tablespace OFFLINE NORMAL`
CKPT Process

Instance

SGA
- Java pool
- Large pool
- Database buffer cache
- Redo log buffer
- Data dict. cache
- Shared pool
- Shared SQL and PL/SQL

Database
- Datafile 1
- Datafile 2
- Datafile 3
- Control file
- Redo log file 1
- Redo log file 2

Database buffers

Parameter file

Password file

User process

Server process

PGA

SMON
DBWN
PMON
CKPT
LGWR
ARCn

Archived log files

ARCn
Control Files in Enterprise Manager
ARCn Process and Archived Log Files

Instance

SGA
- Java pool
- Large pool
- Database buffer cache
- Redo log buffer
- Shared pool
- Shared SQL and PL/SQL
- Data dict. cache

Process

SMON
DBWR
PMON
CKPT
LGWR

ARC0

User process

Server process

PGA

Password file

Parameter file

Database

Datafile 1
Control file
Redo log file 1
Redo log file 2
Datafile 2
Datafile 3

PMON
CKPT
LGWR

ARC1

Archived log files dest 1

Archived log files dest 2
Database Synchronization

- All datafiles (except offline and read-only) must be synchronized for the database to open.
- Synchronization is based on the current checkpoint number.
- Applying changes recorded in the redo log files synchronizes datafiles.
- Redo log files are automatically requested by the Oracle server.
Phases for Instance Recovery

1. Datafiles out-of-synch
2. Roll forward (redo)
3. Committed and non-committed data in files
4. Roll back (undo)
5. Committed data in files
Tuning Crash and Instance Recovery Performance

- Tuning the duration of instance and crash recovery
- Tuning the phases of instance recovery
Tuning the Duration of Instance and Crash Recovery

Methods to keep the duration of instance and crash recovery within user-specified bounds:

• Set initialization parameters to influence the number of redo log records and data blocks involved in recovery.
• Size the redo log file to influence checkpointing frequency.
• Issue SQL statements to initiate checkpoints.
• Parallelize instance recovery operations.
## Initialization Parameters Influencing Checkpoints

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAST_START_MTTR_TARGET</td>
<td>Expected MTTR specified in seconds</td>
</tr>
<tr>
<td>LOG_CHECKPOINT_TIMEOUT</td>
<td>Amount of time that has passed since the incremental checkpoint at the position where the last write to the redo log occurred</td>
</tr>
<tr>
<td>LOG_CHECKPOINT_INTERVAL</td>
<td>Number of redo log file blocks that can exist between an incremental checkpoint and the last block written to the redo log</td>
</tr>
</tbody>
</table>
Controlling Instance Crash Recovery Time
V$INSTANCE_RECOVERY

- Used to monitor the mechanisms available to limit recovery I/O
- Statistics from this view to calculate which parameter has the greatest influence on checkpointing.
Tuning the Phases of Crash and Instance Recovery

• Tuning the roll forward phase
• Tuning the rollback phase
Tuning the Rolling Forward Phase

- Parallel block recovery
- `RECOVERY_PARALLELISM` specifies the number of concurrent recovery processes
Tuning the Rolling Back Phase

- Fast-start on-demand rollback
- Fast-start parallel rollback
Fast-Start On-Demand Rollback

Server process encountering data to be rolled back performs the following:
• Rolls back the block containing the required row
• Hands off further recovery, which may be in parallel, to SMON

Improved response
Fast-Start Parallel Rollback

undo segment

SMON

P000

P001

P002

P003

Tables
Controlling Fast-Start Parallel Rollback

**FAST_START_PARALLEL_ROLLBACK** parameter

<table>
<thead>
<tr>
<th>Value</th>
<th>Maximum Parallel Recovery Servers</th>
</tr>
</thead>
<tbody>
<tr>
<td>FALSE</td>
<td>None</td>
</tr>
<tr>
<td>LOW (default)</td>
<td>2 * CPU_COUNT</td>
</tr>
<tr>
<td>HIGH</td>
<td>4 * CPU_COUNT</td>
</tr>
</tbody>
</table>
Monitoring Parallel Rollback

- V$FAST_START_SERVERS
- V$FAST_START_TRANSACTIONS
Summary

In this lesson, you should have learned how to:

• Identify components of the instance and database that are significant to recovery
• Tune crash and instance recovery
Practice 7 Overview

This practice covers the following topics:

• Querying dynamic performance views to determine the current state and structure of the database
• Explaining the use of specific initialization parameters
• Mirroring of the control files and redo log files
Configuring the Database Archiving Mode
Objectives

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

• Describe the differences between ARCHIVELOG and NOARCHIVELOG modes
• Configure a database for ARCHIVELOG mode
• Enable automatic archiving
• Perform manual archiving of logs
• Configure multiple archive processes
• Configure multiple destinations, including remote destinations
Redo Log History

Online redo log files

LGWR

No redo history

Redo history

051
052
NOARCHIVELOG Mode

LGWR

Online redo log files

No redo history

Copyright © Oracle Corporation, 2001. All rights reserved.
ARCHIVELOG Mode

Online redo log files

Redo history

Archived log files

LGWR
Changing the Archiving Mode

1. SHUTDOWN NORMAL/IMMEDIATE/TRANSACTIONAL

2. STARTUP MOUNT

3. ALTER DATABASE ARCHIVELOG

4. ALTER DATABASE OPEN

5. Full database backup
Setting the Archiving Mode

In Archi... Oracle Corporation, 2001. All rights reserved.
Automatic and Manual Archiving

- **Automatic archiving**: `LOG_ARCHIVE_START=TRUE`

- **Manual archiving**: `LOG_ARCHIVE_START=FALSE`
Enabling Automatic Archiving

- Control instance crash recovery time
  - Desired mean time to recover: 10 Minutes

Media Recovery
- The database is currently in Archive Log mode.
  - Archive Log Mode
  - Automatic archival
    - Log Archive Filename Format: arch_%s.arc

It is recommended that archive log files be written to multiple locations spread across different disks.

Archive Log Destination(s)
- oracleed21oraDATAARCHIVE1 MANDATORY
- oracleed21oraDATAARCHIVE2 OPTIONAL

In Archive Log mode, hot backups and recoveries to the latest time is possible, but you must provide space for logs. If you change the mode to Archive Log mode, you should take a backup immediately. In No Archive Log mode, only cold backups can be taken, and data can be lost in cases of database corruption.
Specifying Multiple ARCn Processes

- The dynamic parameter LOG_ARCHIVE_MAX_PROCESSES controls the number of archive processes started at instance startup.
- A maximum of ten ARCn processes can be specified.
- The number of ARCn processes can be changed with ALTER SYSTEM.
Stop or Start Additional Archive Processes

```
LOG_ARCHIVE_MAX_PROCESSES=2

ALTER SYSTEM SET
LOG_ARCHIVE_MAX_PROCESSES = 3;
```
Enabling Automatic Archiving at Instance Startup

LOG_ARCHIVE_START=TRUE
LOG_ARCHIVE_MAX_PROCESSES=n
Enabling Automatic Archiving After Instance Startup

ALTER SYSTEM ARCHIVE LOG START;

Online redo logs

ARC0

051
Disabling Automatic Archiving

1. ALTER SYSTEM ARCHIVE LOG STOP;

2. LOG_ARCHIVE_START

Initialization parameters

Online redo logs

051 052

051 052
Manually Archiving Online Redo Log Files

ALTER SYSTEM ARCHIVE LOG CURRENT;

051 052 053
Online redo logs

Server Process

051 052
Manually Archive Redo Log Files

In Archive Log mode, hot backups and recovery to the latest time is possible, but you must provide space for logs. If you change the mode to Archive Log mode, you should take a backup immediately. In No Archive Log mode, only cold backups can be taken, and data can be lost in cases of database corruption.
Specifying the Archive Log Destination

- Use `LOG_ARCHIVE_DEST_n` to specify up to ten archival destinations.
- Use `LOG_ARCHIVE_FORMAT` to include the log sequence number and thread number as part of the filename.
Specifying Multiple Archive Log Destinations

Use `LOG_ARCHIVE_DEST_n` to specify up to ten archival destinations, which can be on a:

- Local disk
- Remote standby database

```
log_archive_dest_1 = "LOCATION=/archive1"
log_archive_dest_2 = "SERVICE=standby_db1"
```
**LOG_ARCHIVE_DEST_n Options**

- Set archive location as **MANDATORY** or **OPTIONAL**.
- Define time before retry in case of failures.

```plaintext
log_archive_dest_1="LOCATION=/archive/
                 MANDATORY REOPEN"
log_archive_dest_2="SERVICE=standby_db1
                    MANDATORY REOPEN=600"
log_archive_dest_3="LOCATION=/archive2/
                    OPTIONAL"
```
Specifying a Minimum Number of Local Destinations

- \texttt{LOG\_ARCHIVE\_MIN\_SUCCEED\_DEST} parameter

\begin{center}
\begin{tabular}{|l|}
\hline
\texttt{LOG\_ARCHIVE\_MIN\_SUCCEED\_DEST} = 2 \\
\hline
\end{tabular}
\end{center}

- An online redo log group can be reused only if:
  - Archiving has been done to all mandatory locations
  - The number of local locations archived is greater than or equal to the value of the \texttt{LOG\_ARCHIVE\_MIN\_SUCCEED\_DEST} parameter
Controlling Archiving to a Destination

• An archival destination can be disabled by using the dynamic initialization parameter
  LOG_ARCHIVE_DEST_STATE_n.

  $\text{LOG\_ARCHIVE\_DEST\_STATE\_2} = \text{DEFER}$

  $\text{ALTER SYSTEM SET log_archive_dest_state_3 = DEFER;}$

• Archiving to a destination can be enabled again.

  $\text{LOG\_ARCHIVE\_DEST\_STATE\_2} = \text{ENABLE}$

  $\text{ALTER SYSTEM SET log_archive_dest_state_3 = ENABLE;}$
Specifying the Filename Format

Online redo log files

Group 1
053
053
053

Group 2
052
052
052

Archived log file

/\ORADATA/archive/arch%\s.arc

ARC0

LOG_ARCHIVE_DEST_n

LOG_ARCHIVE_FORMAT
Obtaining Archive Log Information

Dynamic Views

V$ARCHIVED_LOG
V$ARCHIVE_DEST
V$LOG_HISTORY
V$DATABASE
V$ARCHIVE_PROCESSES

Command Line

ARCHIVE LOG LIST
Summary

In this lesson, you should have learned how to:

• Configure a database for ARCHIVELOG mode
• Enable automatic archiving
• Perform manual archiving of logs
• Configure multiple archive processes
• Configure multiple destinations
Practice 8 Overview

This practice covers the following topics:
- Enabling and disabling automatic archiving
- Configuring multiple archiver processes
- Configuring multiple archiving destinations
- Performing manual archiving of redo log files
Oracle Recovery Manager
Overview and Configuration
Objectives

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

- Identify the features and components of RMAN
- Describe the RMAN repository and control file usage
- Describe channel allocation
- Describe the Media Management Library interface
- Connect to RMAN without the recovery catalog
- Configure the RMAN environment
Recovery Manager Features

RMAN provides a flexible way to:

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

RMAN provides a flexible way to:

• Detect corrupted blocks during backup
• Increase performance through:
  – Automatic parallelization
  – Generation of less redo
  – Restricting I/O for backups
  – Tape streaming
• Manage backup and recovery tasks
Recovery Manager Components

Target database

Server session (channel)

Server session (channel)

Server session (channel)

Server session (default)

Recovery Manager (RMAN)

Enterprise Manager

Server session (rcvcat)

Recovery catalog DB

Enterprise Manager

Disk

Disk

MML
Using the Backup Management Wizards
RMAN Repository: Using the Control File

- RMAN repository is metadata about target database and backup and recovery operations.
- RMAN repository is always stored in the control file of the target database.
- \texttt{CONTROL\_FILE\_RECORD\_KEEP\_TIME} determines the minimum age in days of a record before it can be overwritten.
- The control file can grow in size.
Channel Allocation

- Target database
  - Server Session
  - Server Session
  - Channel (disk)
  - Channel (sbt)
- Disk
- RMAN
Automatic Channel Allocation

• Change the default device type:

```
RMAN> CONFIGURE DEFAULT DEVICE TYPE TO sbt;
```

• Configure parallelism for automatic channels:

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

• Configure automatic channel options:

```
RMAN> CONFIGURE CHANNEL DEVICE TYPE DISK 2> FORMAT = `/BACKUP/RMAN/%U';
RMAN> CONFIGURE CHANNEL DEVICE TYPE DISK 2> MAXPIECESIZE 2G;
```
Channel Allocation Using OEM
Manual Channel Allocation

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

```
RMAN> RUN {
    2> ALLOCATE CHANNEL c1 TYPE disk
    3> FORMAT = '/db01/BACKUP/usr0520.bak';
    4> BACKUP DATAFILE '/db01/ORADATA/users01.dbf';}
```
Types of Connections with RMAN

- Target database
- Recovery catalog database
- Auxiliary database
  - Standby database
  - Duplicate database
  - TSPITR instance
Connecting Without a Recovery Catalog

• Starting RMAN locally

UNIX: $ ORACLE_SID=DB01; export ORACLE_SID
     $ rman target / as sysdba

Windows NT:C:\> set ORACLE_SID=DB01
     C:\> rman target / as sysdba

• Starting RMAN remotely

rman target sys/target_pwd@DB01
Additional RMAN Command Line Arguments

- Writing RMAN output to a log file:
  
  ```
  $ rman target sys/oracle
     log $HOME/ORADATA/u03/rman.log append
  ```

- Executing a command file when RMAN is invoked:
  
  ```
  $ rman target sys/oracle
     log $HOME/ORADATA/u03/rman.log append
     @’$HOME/STUDENT/LABS/my_rman_script.rcv’
  ```
Recovery Manager Modes

• Interactive mode
  – Use it when doing analysis
  – Minimize regular usage
  – Avoid using with log option

• Batch mode
  – Meant for automated jobs
  – Minimize operator errors
  – Set the log file to obtain information
RMAN Commands

RMAN commands are of the following types:

• Stand-alone
  – Executed only at the RMAN prompt
  – Executed individually
  – Cannot appear as subcommands within `RUN`

• Job
  – Must be within the brackets of `RUN`
  – Executed as a group

• Stand-alone or job
RUN command:

```
RMAN> RUN {
    backup
    incremental level 0
    format '/u01/db01/backup/%d_%s_%p'
    fileperset 5
    (database include current controlfile);
    sql 'alter database archive log current';
}
```
RMAN Configuration Settings

- 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
  - Limit the size of backup sets
  - Exempt a tablespace from backup
  - Enable and disable backup optimization
The `CONFIGURE` Command

- Configure automatic channels:

  ```
  RMAN> CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT '/db01/BACKUP/%U';
  ```

- Implement retention policy by specifying a recovery window:

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

- Implement retention policy by specifying redundancy:

  ```
  RMAN> CONFIGURE RETENTION POLICY TO REDUNDANCY 2;
  ```
The `CONFIGURE` Command

- Configure duplexed backup sets:
  ```bash
  RMAN> CONFIGURE DATAFILE BACKUP COPIES FOR 2> DEVICE TYPE disk TO 2;
  ```

- Configure backup optimization:
  ```bash
  RMAN> CONFIGURE BACKUP OPTIMIZATION ON;
  ```

- Use the `CLEAR` option to return to the default value:
  ```bash
  RMAN> CONFIGURE RETENTION POLICY CLEAR;
  RMAN> CONFIGURE CHANNEL DEVICE TYPE sbt CLEAR;
  ```
The **SHOW** Command

- Displays persistent configuration settings
- Use the **SHOW** command to display:
  - Automatic channel configuration settings
  - Backup retention policy settings
  - Number of backup copies to be created
  - Backup set size limit
  - Tablespace excluded from backups
  - Backup optimization status
- Use **SHOW ALL** to display all settings:

```
RMAN> SHOW ALL;
```
**LIST Command Operations**

- Lists backup sets and copies of datafiles
- Lists backup sets and copies of any datafile for a specified tablespace
- Lists backup sets and copies containing archive logs for a specified range
The LIST Command

• List backups of all files in the database:

   RMAN> LIST BACKUP OF DATABASE;

• List all backup sets containing the users01.dbf datafile:

   RMAN> LIST BACKUP OF DATAFILE
       2> ”/db01/ORADATA/u03/users01.dbf”;

• List all copies of datafiles in the SYSTEM tablespace:

   RMAN> LIST COPY OF TABLESPACE ”SYSTEM”;

The **REPORT** Command

- Produces a detailed analysis of the repository
- Produces reports to answer:
  - Which files need a backup?
  - Which backups can be deleted?
  - Which files are unrecoverable?
The `REPORT NEED BACKUP` Command

- Lists all datafiles requiring a backup
- Assumes the most recent backup is used during a restore
- Provides three options:
  - Incremental
  - Days
  - Redundancy

Without options, takes into account the configured retention policy

```
REPORT NEED BACKUP incremental 3;
REPORT NEED BACKUP days 3;
REPORT NEED BACKUP redundancy 3;
```
Recovery Manager uses PL/SQL packages as its interface to:

- Target databases
- The recovery catalog
RMAN Usage Considerations

• Resources: Shared memory, more processes
• Privileges given to users
  – Database: SYSDBA
  – Operating System: Access to devices
• Remote operations
  – Set up the password file
  – Ensure that the password file is backed up
• Globalization environment variables
• Format used for the time parameters in RMAN commands
Summary

In this lesson, you should have learned how to:

• Configure the RMAN environment
• Use automatic channel allocation
• Manually allocate channels
• Connect to RMAN without the recovery catalog
• Retrieve information from the RMAN repository
Practice 9 Overview

This practice covers the following topics:

• Using Recovery Manager to connect to a target database in default NOCATALOG mode.
• Obtaining information from the target database control file.
• Configuring a retention policy
• Using the show command to display RMAN environment settings
User-Managed Backups
Objectives

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

• Describe user-managed backup and recovery operations
• Discuss backup issues associated with read-only tablespaces
• Perform closed database backups
• Perform open database backups
• Back up the control file
• Perform cleanup after a failed online backup
• Use the DBVERIFY utility to detect corruption
Terminology

• Whole database backup
  – Target database may be open or closed
  – Backup of all datafiles and the control file

• Partial database backups
  – Tablespace
  – Datafile
  – Control file

• Consistent backups

• Inconsistent backups
User-Managed Backup and Recovery

• Files are backed up with operating system commands
• Backups are restored with operating system commands
• Recovery is accomplished using SQL and SQL*Plus commands
Querying Views to Obtain Database File Information

V$DATAFILE
V$CONTROLFILE
V$LOGFILE
DBA_DATA_FILES
Obtaining Database File Information
Backup Methods

Closed database

NOARCHIVELOG mode

Closed or open database

ARCHIVELOG mode

Physical backup
Consistent Whole Database Backup
(Closed Database Backup)

- Datafiles
- Control files
- Redo log files
- Parameter file
- Password file

Online or offline storage
Advantages of Making Consistent Whole Database Backups

• Conceptually simple
• Easy to perform
• Require little operator interaction
Making a Consistent Whole Database Backup

1. Control files
2. SHUTDOWN IMMEDIATE
3. HOST cp <files> /backup/
4. STARTUP OPEN

Datafiles
Redo log files
Password file
Parameter file
Open Database Backup

- Datafiles
- Control files
- Password file
- Parameter file
- Archived redo log files
- Online redo log files
Advantages of Making Open Database Backups

• Maintains high database availability
• Can be done at a tablespace or datafile level
• Supports nonstop business operations
Open Database Backup Requirements

ARCHIVELOG mode

LGWR

Online redo log files

Archived redo log files
Open Database Backup Options

Tablespace backup

Tablespace USERS

users01.dbf  users02.dbf

Datafile backup

Tablespace USERS

users01.dbf  users02.dbf
Making a Backup of an Online Tablespace

SQL> ALTER TABLESPACE users
    2> BEGIN BACKUP;

SQL> !cp /.../users01.dbf /BACKUP/users01.dbf
Ending the Online Tablespace Backup

```
SQL> ALTER TABLESPACE users 
2> BEGIN BACKUP;

SQL> !cp /.../users01.dbf /BACKUP/users01.dbf

SQL> ALTER TABLESPACE users 
2> END BACKUP;
```
Backup Status Information

Dynamic views

V$BACKUP

V$DATAFILE_HEADER
Failure During Online Tablespace Backup

```
ALTER TABLESPACE users
BEGIN BACKUP;
```

cp /.../users01.dbf
/BACKUP/users01.dbf
Ending the Online Backup

- Query V$BACKUP to check backup status
- Issue the ALTER DATABASE command to change the status and unfreeze the header:

  ALTER DATABASE DATAFILE '<filename>' END BACKUP;

- Or use this command in Oracle9i:

  ALTER DATABASE END BACKUP;
Read-Only Tablespace Backup

1. `ALTER TABLESPACE query_data READ ONLY;`

2. File 1 backup copy

3. SCN 1

4. DBW0

SCN 1

Query_Data File 1

Users File 1

Users File 2

SCN 2
Read-Only Tablespace Backup Issues

- Only one backup is needed after altering the tablespace to read-only.
- Resume a normal backup schedule for that tablespace after making it read-write.
- The control file must correctly identify the tablespace in read-only mode; otherwise you must recover it.
## Backup Issues with Logging and Nologging Options

<table>
<thead>
<tr>
<th>Logging</th>
<th>Nologging</th>
</tr>
</thead>
<tbody>
<tr>
<td>All changes recorded to redo</td>
<td>Minimal redo recorded</td>
</tr>
<tr>
<td>Fully recoverable from last backup</td>
<td>Not recoverable from last backup</td>
</tr>
<tr>
<td>No additional backup</td>
<td>May require additional backup</td>
</tr>
</tbody>
</table>
Manual Control File Backups

• Creating a binary image
  
  ```sql
  ALTER DATABASE BACKUP CONTROLFILE TO 'controll.bkp';
  ```

• Creating a text trace file
  
  ```sql
  ALTER DATABASE BACKUP CONTROLFILE TO TRACE;
  ```
Backing Up the Initialization Parameter File

CREATE PFILE FROM SPFILE;

CREATE PFILE = `'/backup/init.ora'`
FROM SPFILE;
Verifying Backups Using the DBVERIFY Utility
DBVERIFY Command-Line Interface

- External command-line utility
- Used to ensure that a backup database or datafile is valid before a restore
- Can be a helpful diagnostic aid when data corruption problems are encountered

```bash
%dbv file=/ORADATA/u03/users01.dbf logfile=dbv.log
```
Summary

In this lesson, you should have learned how to:

- Determine which files require backup and when they should be backed up
- Make user-managed backups
- Backup the control file
- Backup the server initialization parameter file
- End an online backup that did not complete due to instance failure
- Use dynamic views to determine the status of backup operations
- Use DBVERIFY to verify the backup
Practice 10 Overview

This practice covers the following topics:

• Performing a full offline database backup
• Performing an online backup of a tablespace datafile
• Creating a trace file of the control file
RMAN Backups
Objectives

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

- Identify types of RMAN specific backups
- Use the RMAN \texttt{BACKUP} command to create backup sets
- Back up the control file
- Back up the archived redo log files
- Use the RMAN \texttt{COPY} command to create image copies
RMAN Backup Concepts

- Recovery Manager backup is a server-managed backup
- Recovery Manager uses Oracle server sessions for backup operations
- Can back up entire database, all datafiles in a tablespace, selected datafiles, control files, archived redo log files
- Closed database backup
  - Target database must be mounted (not open)
  - Includes datafiles, control files, archived redo log files
- Open database backup
  - Tablespaces should not be put in backup mode
  - Includes datafiles, control files, archived redo log files
Recovery Manager Backups

Image copy

- Datafile 3
- Control file
- Archived Log file

Backup set

- Datafile 1
- Datafile 2
- Datafile 3
- Control file

Copy of datafile 3
Copy of control file
Copy of archived log
Backup Sets

Backup set 1

Datafile 1

Datafile 2

Datafile 3

Control file

Backup set 2

Datafile 2

Datafile 3

Datafile 4

Backup set 3

Control file

Datafile 4
Characteristics of Backup Sets

- The `BACKUP` command creates backup sets.
- Backup sets usually contain more than one file.
- Backup sets can be written to a disk or tape.
- A restore operation is required to extract files from a backup set.
- Datafile backup sets can be incremental or full.
- Backup sets do not include never-used blocks.
Backup Piece

- A backup piece is a file in a backup set.
- A backup piece can contain blocks from more than one datafile.
Backup Piece Size

Backup piece size can be limited as follows:

```
RMAN> RUN {
  2> ALLOCATE CHANNEL t1 TYPE 'SBT'
  3> MAXPIECESIZE = 4G;
  4> BACKUP
  5>   FORMAT 'df_%t_%s_%p' FILESPERSET 3
  6>   (tablespace users); }
```
The **BACKUP** Command

```
RMAN> BACKUP
 2> FORMAT '/BACKUP/df_%d_%s_%p.bus'
 3> DATABASE filesperset = 2;
```
Creating a Backup Set
Multiplexed Backup Sets

Multiplex two or more datafiles into a backup set for tape streaming.

filesperset = 3

Datafile 1
Datafile 2
Datafile 3

Server process (channel)

Backup set
Datafile 1,2,3,1,2,3...

MML

Tape
Parallelization of Backup Sets

Allocate multiple channels, optionally specify filesperset, and include many files.

Backup Set 1
- Datafile 1
- Datafile 4
- Datafile 5

Backup Set 2
- Datafile 2
- Datafile 3
- Datafile 9

Backup Set 3
- Datafile 6
- Datafile 7
- Datafile 8

Set 1
Set 2
Set 3
Duplexed Backup Sets

[Diagram showing two sets of datafiles, one set labeled Backup 1 and the other labeled Backup 2.]
Backups of Backup Sets
Archived Redo Log File Backups

- Online redo log file switch is automatic.
- Archived log failover is performed.
Archived Redo Log Backup Sets

- Include only archived redo log files
- Are always full backups

```
RMAN> BACKUP
  2> FORMAT '/disk1/backup/ar_%t_%s_%p'
  3> ARCHIVELOG ALL DELETE ALL INPUT;
```
Backup Constraints

- The database must be mounted or open.
- Online redo log backups are not supported.
- Only “clean” backups are usable in NOARCHIVELOG mode.
- Only “current” datafile backups are usable in ARCHIVELOG mode.
Image Copies

Datafile 3 → Copy of datafile 3
Archived Log file → Copy of archived log
Characteristics of an Image Copy

• Can be written only to a disk
• Can be used for recovery immediately; does not need to be restored
• Is a physical copy of a single datafile, archived log, or control file
• Is most like an operating system backup (contains all blocks)
• Can be part of an incremental strategy
Creating an Image Copy
Image Copy Example

RMAN> COPY
2> DATAFILE '/ORADATA/users_01_db01.dbf' TO
3> '/BACKUP/users01.dbf' tag=DF3,
4> ARCHIVELOG 'arch_1060.arc' TO
5> 'arch_1060.bak';
The **COPY** Command

```sql
RMAN> COPY
  2> DATAFILE 3 TO '/BACKUP/file3.dbf',
  3> DATAFILE 1 TO '/BACKUP/file1.dbf';
```
Image Copy Parallelization

One COPY command with many channels

```
RMAN> CONFIGURE DEVICE TYPE disk parallelism 4;
  2> COPY       # 3 files copied in parallel
  3>     datafile 1 TO '/BACKUP/df1.dbf',
  4>     datafile 2 TO '/BACKUP/df2.dbf',
  5>     datafile 3 TO '/BACKUP/df3.dbf';
RMAN> COPY       # Second copy command
  2>     datafile 4 TO '/BACKUP/df4.dbf';
```
Copy the Whole Database

• Mount the database for a whole consistent backup.
• Use the REPORT SCHEMA command to list the files.
• Use the COPY command or make an image copy of each datafile.
• Use the LIST COPY command to verify the copies.
Making Incremental Backups

- **Full backups** contain all datafile blocks.
- **Differential incremental backups** contain only modified blocks from level \( n \) or lower.
- **Cumulative incremental backups** contain only modified blocks from level \( n-1 \) or lower.
Differential Incremental Backup Example

An \( n \) level backup of all blocks that have changed since the most recent backup at level \( n \) or lower.
Cumulative Incremental Backup Example

$n$ level backup which contains all blocks changed since the previous backup at a level $n - 1$ or lower.
Backup in NOARCHIVELOG Mode

1. Ensure sufficient space for the backup.
2. Shut down using the `NORMAL` or `IMMEDIATE` clause.
3. Mount the database.
4. Allocate multiple channels if not using automatic.
5. Run the `BACKUP` command.
6. Verify that the backup is finished and cataloged.
7. Open the database for normal use.

```
RMAN> BACKUP DATABASE FILESPERSET 3;
```
RMAN Control File Autobackups

- Use the `CONFIGURE CONTROLFILE AUTOBACKUP` command to enable
- When enabled, RMAN automatically performs a control file autobackup after `BACKUP` or `COPY` commands
- Backup is given a default name
Tags for Backups and Image Copies

Logical name assigned to a backup set or image copy

- month_full_backup
- week_full_backup
- Wednesday_1_backup
RMAN Dynamic Views

- V$ARCHIVED_LOG
- V$BACKUP_CORRUPTION
- V$COPY_CORRUPTION
- V$BACKUP_DATAFILE
- V$BACKUP_REDOLOG
- V$BACKUP_SET
- V$BACKUP PIECE
Monitoring RMAN Backups

- Correlate server sessions with channels with the `SET COMMAND ID` command.
- Query `V$PROCESS` and `V$SESSION` to determine which sessions correspond to which RMAN channels.
- Query `V$SESSION_LONGOPS` to monitor the progress of backups and copies.
- Use an operating system utility to monitor the process or threads.
Miscellaneous RMAN Issues

- Abnormal termination of a Recovery Manager job
- Detecting physical and logical block corruption
- Detecting a fractured block during open backups
Summary

In this lesson, you should have learned how to:

• Determine what type of RMAN backups should be taken
• Make backups with the RMAN COPY and BACKUP commands
• Back up the control file
• Back up the archived redo log files
This practice covers the following topics:

- Using Recovery Manager to back up one tablespace datafile and a controlfile
- Using Recovery Manager to back up archived log files
- Using the RMAN COPY command to create an image copy of a database file
User-Managed Complete Recovery
Objectives

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

- Describe media recovery
- Perform recovery in NOARCHIVELOG mode
- Perform complete recovery in ARCHIVELOG mode
- Restore datafiles to different locations
- Relocate and recover a tablespace by using archived redo log files
- Describe read-only tablespace recovery
Media Recovery

• Used to recover a lost or damaged current datafile or control file
• Requires explicit invocation
• Operates as follows:
  – Files are restored from backups
  – Redo data is applied to the restored files from archived redo log files and online redo logs
Recovery Steps

1. Restored datafiles
2. Redo applied
3. Database containing committed and uncommitted transactions
4. Undo applied
5. Recovered database
Restoration and Datafile Media Recovery with User-Managed Procedures

- Restore files using operating system commands
- Recover files using the SQL*Plus RECOVER command
ARCHIVELOG and NOARCHIVELOG Modes

- User process
- Server process
- PGA

Instance
- SGA
  - Java Pool
  - Large Pool
  - Data buffer
  - Redo log buffer
- Shared pool
  - Shared SQL and PL/SQL
  - Data dict. cache

SMON DBWn PMON CKPT LGWR ARCh

User process
- Server process
- PGA

Parameter File
- Password File

Database
- Datafile 1
- Datafile 2
- Datafile 3
- Control files
- Redo log file 1
- Redo log file 2

Copyright © Oracle Corporation, 2001. All rights reserved.
Recovery in NOARCHIVELOG Mode

• In NOARCHIVELOG mode, you must restore the following database files:
  – All datafiles
  – Control files

• You can also restore the following files:
  – Redo log files
  – Password file
  – Parameter file
Recovery in NOARCHIVELOG Mode

• Advantages
  – Easy to perform, with low risk of error
  – Recovery time is the time it takes to restore all files

• Disadvantages
  – Data is lost and must be reapplied manually
  – The entire database is restored to the point of the last whole closed backup
Recovery in NOARCHIVELOG Mode with Redo Log File Backups

Restore from the most recent backup
Recovery in NOARCHIVELOG Mode Without Redo Log File Backups

1. Shut down the instance.
2. Restore the datafiles and the control file from the most recent whole database backup.
3. Perform cancel-based recovery.
4. Open the database with the `RESETLOGS` option.
Recovery in ARCHIVELOG Mode

• Complete Recovery
  – Uses redo data or incremental backups
  – Updates the database to the most current point in time
  – Applies all redo changes

• Incomplete Recovery
  – Uses backup and redo logs to produce a noncurrent version of the database
Complete Recovery

- Make sure that datafiles for restore are offline.
- Restore only lost or damaged datafiles.
- Do not restore the control files, redo log files, password files, or parameter files.
- Recover the datafiles.
Complete Recovery in ARCHIVELOG Mode

• Advantages
  – Only need to restore lost files
  – Recovers all data to the time of failure
  – Recovery time is the time it takes to restore lost files and apply all archived log files

• Disadvantages
  – Must have all archived log files since the backup from which you are restoring
Determining Which Files Need Recovery

• View `V$RECOVER_FILE` to determine which datafiles need recovery.
• View `V$ARCHIVED_LOG` for a list of all archived redo log files for the database.
• View `V$RECOVERY_LOG` for a list of all archived redo log files required for recovery.
User-Managed Recovery Procedures:
RECOVER Command

• Recover a mounted database:

```
SQL> RECOVER DATABASE
```

Or

```
SQL> RECOVER DATAFILE
   2> '/ORADATA/u03/users01.dbf'
```

• Recover an open database:

```
SQL> RECOVER TABLESPACE users
```

Or

```
SQL> RECOVER DATAFILE
   2> '/ORADATA/u03/users01.dbf'
```
Using Archived Redo Log Files During Recovery

- To change archive location, use the `ALTER SYSTEM ARCHIVE LOG` command.
- To apply redo log files automatically:
  - Issue the `SET AUTORECOVERY ON` command before starting media recovery.
  - Enter `auto` when prompted for an archived log file.
  - Use the `RECOVER AUTOMATIC` command.
Restoring Datafiles to a New Location with User-Managed Procedures

• Use operating system commands to restore the datafile to the new location.
• Use the `ALTER DATABASE RENAME FILE` command to record the change in the control file.
Complete Recovery Methods

• Closed database recovery for:
  – System datafiles
  – Undo segment datafiles
  – Whole database
• Open database recovery, with database initially opened (for file loss)
• Open database recovery with database initially closed (for hardware failure)
• Datafile recovery with no datafile backup
Complete Recovery of a Closed Database

Closed database recovery is used for:

• System tablespace datafiles
• Rollback segment datafiles
• Whole database
Closed Database Recovery Example

1. Shut down the instance

2. Restore datafile 1 (Log Sequence 144)

3. Open the database
Open Database Recovery When the Database Is Initially Open

Use this method when:

- The database is currently open
- The database will remain open during the recovery
- The media failure does not affect the SYSTEM tablespace
Open Database Recovery Example

1. Take datafile 2 offline
2. Restore datafile 2 (Log Sequence 144)
3. Bring datafile 2 online
Open Database Recovery When the Database Is Initially Closed

Use this method when:

• The database is currently closed
• The database will be opened during recovery
• The media failure does not affect the SYSTEM tablespace
Open Database Recovery Example

1. Mount the database
2. Take datafile 2 offline
3. Open the database
4. Restore datafile 2
5. Bring datafile 2 online
Recovery of a Datafile Without a Backup

- Datafile is lost that was never backed up
- Cannot be used when it is a file from the SYSTEM tablespace
- Cannot be used if the control file has been recreated
Re-creating Lost Datafiles Without Backup

- Used when missing datafile cannot be restored because it had never been backed up.
- Description of missing datafile is still in data dictionary and control file.
- Re-create the datafile:
  
  ```sql
  SQL> ALTER DATABASE CREATE DATAFILE 'filename';
  ```

- Re-create the datafile with a different filename:
  
  ```sql
  SQL> ALTER DATABASE CREATE DATAFILE 'filename'
         AS 'new file name';
  ```
Open Database

1. Take the datafile or tablespace offline

2. Re-create the datafile

3. Archived log file

4. Bring the datafile or tablespace online

Recovered Database
Read-Only Tablespace Recovery

Case 1
- Read-Only

Case 2
- Read-Only
- Read-Write

Case 3
- Read-Write
- Read-Only

Backup 1
Backup 2
Recovery
Read-Only Tablespace Recovery Issues

Special considerations must be taken for read-only tablespaces when:

• Re-creating a control file
• Renaming datafiles
• Using a backup control file
Loss of Control Files

You may need to create control files if:

• All control files are lost because of a failure
• The name of a database needs to be changed
• The current settings in the control file need to be changed
Recovering Control Files

Methods to recover from loss of control file:

• Use the current control file
• Create a new control file
• Use a backup control file
Summary

In this lesson, you should have learned how to:

• Determine what type of recovery is required
• Determine which files need to be restored and recovered
• Recover a database in NOARCHIVELOG mode
• Recover a database in ARCHIVELOG mode
• Restore datafiles to different locations if the original location is unavailable
Practice 12 Overview

These practices cover the following topics:

- Performing complete database recovery with the database in NOARCHIVELOG mode
- Performing complete database recovery with the database in ARCHIVELOG mode
RMAN Complete Recovery
Objectives

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

• Describe the use of RMAN for restoration and recovery
• Perform complete recovery in ARCHIVELOG mode
• Restore datafiles to different locations
• Relocate and recover a tablespace by using archived redo log files
Restoration and Datafile Media Recovery Using RMAN

- Restore files from backup sets or image copies by using the RMAN RESTORE command
- Recover files by using the RMAN RECOVER command
Using RMAN to Recover a Database in ARCHIVELOG Mode

```
 rman target /
 RMAN> STARTUP MOUNT
 RMAN> RESTORE DATABASE;
 RMAN> RECOVER DATABASE;
 RMAN> ALTER DATABASE OPEN;
```
Using the Recovery Wizard
Using RMAN to Restore Datafiles to a New Location

• Use the `SET NEWNAME` command to restore the datafile to the new location.

  ```sql
  SET NEWNAME FOR DATAFILE 1 to
  '/<newdir>/system01.dbf';
  ```

• Use the `SWITCH` command to record the change in the control file.

  ```sql
  SWITCH DATAFILE ALL;
  ```
Restoring to a New Location
Using RMAN to Recover a Tablespace

Use the following RMAN commands to restore and recover a tablespace:

• RESTORE TABLESPACE
• RECOVER TABLESPACE

```sql
run{
    sql "alter tablespace users offline immediate";
    restore tablespace users;
    recover tablespace users;
    sql "alter tablespace users online";
}
```
Tablespace Recovery
Using RMAN to Relocate a Tablespace

- Use the SET NEWNAME command to restore the files.
- Use the SWITCH command to record the new names in the control file.
- Use the RECOVER TABLESPACE command to recover the datafiles of the tablespace.
Summary

In this lesson, you should have learned how to:

• Recover a database in ARCHIVELOG mode
• Restore datafiles to different locations if the original location is unavailable
Practices 13-1 and 13-2 Overview

These practices cover the following topics:

• Using RMAN to recover a tablesapce
• Using RMAN to recover relocated datafiles
User-Managed Incomplete Recovery
Objectives

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

- Describe the steps of incomplete recovery
- Perform an incomplete database recovery
- Identify the loss of current online redo log files
Incomplete Recovery Overview

Diagram showing the components of an Oracle database instance and related processes:

- **User process**
- **Server process**
- **SGA** (System Global Area):
  - Shared pool
  - Large pool
  - Buffer cache
  - Redo log buffer
- **Redo log**
- **Data dictionary cache**
- **Shared SQL and PL/SQL area**
- **PMON** (Background Process Monitor)
- **DBWR** (Database Writer)
- **SMON** (System Monitor)
- **LGWR** (Log Writer)
- **ARCn** (Archiver)

Database-related components:

- **Datafile 1**
- **Datafile 2**
- **Datafile 3**
- **Control files**
- **Redo log file 1**
- **Redo log file 2**

Additional files:

- **Parameter file**
- **Password file**
- ** Archived log files**
Situations Requiring Incomplete Recovery

• Complete recovery fails because an archived log is lost.
• All unarchived redo log files and a datafile are lost.
• User error
  – An important table was dropped.
  – Invalid data was committed in a table.
• Current control file is lost and a backup control file must be used to open the database.
Types of Incomplete Recovery

• There are three types of incomplete recovery:
  – Time-based recovery
  – Cancel-based recovery
  – Change-based recovery

• You may need to recover using a restored control file when:
  – Control files are lost
  – Performing incomplete recovery to a point when the database structure is different than the current
Incomplete Recovery Guidelines

- Follow all steps carefully.
- Take whole database backups before and after recovery.
- Always verify that the recovery was successful.
- Back up and remove archived logs.
Incomplete Recovery and the Alert Log

- Check the alert log before and after recovery
- Contains error information, hints, and SCNs
User-Managed Procedures for Incomplete Recovery

1. Shut down and back up the database.
2. Restore all datafiles. Do not restore the control file, redo logs, password file, or parameter file.
3. Mount the database.
4. Recover the datafiles to a point before the time of failure.
5. Open the database with `RESETLOGS`.
6. Perform a closed database backup.
RECOVER Command Overview

- Recover a database until cancel:
  ```sql
  RECOVER DATABASE until cancel
  ```

- Recover a database until time:
  ```sql
  RECOVER DATABASE
  until time '2001-03-04:14:22:03'
  ```

- Recover using backup control file:
  ```sql
  RECOVER DATABASE
  until time '2001-03-04:14:22:03'
  using backup controlfile
  ```
Time-Based Recovery Example

Scenario:
• The current time is 12:00 p.m. on March 9, 2001.
• The EMPLOYEES table has been dropped.
• The table was dropped at approximately 11:45 a.m.
• Database activity is minimal because most staff are currently in a meeting.
• The table must be recovered.
UNTIL TIME Recovery

1. Shut down and back up
2. Restore all datafiles
3. Mount the database
4. Open with Resetlogs
5. Back up the database
Cancel-Based Recovery Example

Scenario:
• The current time is 12:00 p.m. on March 9, 2001.
• The EMPLOYEES table was dropped while someone was trying to fix bad blocks.
• Log files exist on the same disk.
• The table was dropped at approximately 11:45 a.m.
• Staff are currently in a meeting.
Cancel-Based Recovery Example

Findings:
• Redo logs are not multiplexed.
• One of the online redo logs is missing.
• The missing redo log is not archived.
• The redo log contained information from 11:34 a.m.
• Twenty-six minutes of data will be lost.
• Users can recover their data.
Using a Backup Control File During Recovery

Scenario:

• The current time is 12:00 p.m. on March 9, 2001.
• The tablespace containing the EMPLOYEES table has been dropped.
• The error occurred around 11:45 a.m.
• Many employee records were updated this morning, but not since 11:00 a.m.
• Backups are taken every night.
Using a Backup Control File During Recovery

Findings:

- The backup from last night contains datafiles and control files required for recovery.
- The EMP_TS tablespace has one datafile.
- The current log sequence number is 61.
- You confirm that the tablespace was dropped at 11:44:54 a.m. on March 9, 2001.
- Datafile number 4 is offline.
Loss of Current Redo Log Files

If the database is closed:

• Attempt to open the database.
• Find the current log sequence number.
• Recover the database until cancel.
• Drop and re-create log files if necessary.
• Open the database using \texttt{RESETLOGS}.
• Perform a whole-database backup.
Summary

In this lesson, you should have learned how to:

- Perform incomplete database recovery
- Recover from the loss of current online redo log files
Practices 14-1 and 14-2 Overview

These practices cover the following topics:
- Recovery from user failure
- Recovery with lost archived redo log files
RMAN Incomplete Recovery
Objectives

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

• Perform an incomplete database recovery using UNTIL TIME

• Perform an incomplete database recovery using UNTIL SEQUENCE
Incomplete Recovery of a Database Using RMAN

1. Mount the database.
2. Allocate multiple channels for parallelization.
3. Restore all datafiles.
4. Recover the database by using UNTIL TIME, UNTIL SEQUENCE, or UNTIL SCN.
5. Open the database by using RESETLOGS.
6. Perform a whole database backup.
Specifying the Restore Time

If you press the finish button now, the database will be restored to the most recent time possible. If you want to restore to a point before, then enter a date and time below.

- **Until**
  - **Date:** November 1, 2001
  - **Time:** 10:58:11 AM EST
RMAN Incomplete Recovery

UNTIL TIME Example

RMAN> run {
  2> allocate channel c1 type DISK;
  3> allocate channel c2 type DISK;
  4> set until time = '2000-12-09:11:44:00';
  5> restore database;
  6> recover database;
  7> alter database open resetlogs; }
RMAN Incomplete Recovery
UNTIL SEQUENCE Example

RMAN> RUN {
  2> SET UNTIL SEQUENCE 120 THREAD 1;
  3> ALTER DATABASE MOUNT;
  4> RESTORE DATABASE;
  5> RECOVER DATABASE; # recovers through log 119
  6> ALTER DATABASE OPEN RESESTLOGS;
  7> }

Copyright © Oracle Corporation, 2001. All rights reserved.
Summary

In this lesson, you should have learned how to:

• Perform an incomplete database recovery using UNTIL TIME

• Perform an incomplete database recovery using UNTIL SEQUENCE
Practice 15 Overview

This practice covers recovery with lost archived redo log files.
RMAN Maintenance
Objectives

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

- Perform cross-checking of backups and copies
- Update the repository when backups have been deleted
- Change the availability status of backups and copies
- Make a backup or copy exempt from the retention policy
- Catalog backups made with operating system commands
Cross Checking Backups and Copies

Use the CROSSCHECK command to:

• Ensure repository information is synchronized with actual files
• Check the status of a backup or copy
• Update the repository when files have been deleted with operating system commands
The **CROSSCHECK** Command

- Cross-check all backups in the database:
  
  ```
  CROSSCHECK BACKUPSET OF DATABASE;
  ```

- Cross-check all copies in the database:
  
  ```
  CROSSCHECK COPY;
  ```
Deleting Backups and Copies

Use the DELETE command to:

• Delete physical backups and image copies
• Update repository status to DELETED
• Remove records from the recovery catalog
The **DELETE** Command

- Delete a specific backup set:
  ```sql
  DELETE BACKUPSET 102;
  ```

- Delete an expired backup without the confirmation:
  ```sql
  DELETE NOPROMPT EXPIRED BACKUP OF TABLESPACE users;
  ```

- Delete all backups, copies, and archived redo log files based on the configured retention policy:
  ```sql
  DELETE OBSOLETE;
  ```
Deleting Backups and Copies

Use the `BACKUP ... DELETE INPUT` command to:

- Delete input files upon successful creation of the backup set
- Delete archived redo log files, datafile copies, and backup sets
Changing the Availability of RMAN Backups and Copies

- Change the status of a backup or copy to Unavailable with the `CHANGE ... UNAVAILABLE` command.
- Return the status to Available with the `CHANGE ... AVAILABLE` command.
Changing the Availability Status

• Change the status of a specific datafile:

   CHANGE DATAFILECOPY '/DB01/BACKUP/users01.dbf' UNAVAILABLE;

• Change the status of a control file backup:

   CHANGE BACKUP OF CONTROLFILE UNAVAILABLE;
   CHANGE BACKUP OF CONTROLFILE AVAILABLE;

• Change the status of archived redo log files:

   CHANGE COPY OF ARCHIVELOG SEQUENCE BETWEEN 230 AND 240 UNAVAILABLE;
Exempting a Backup or Copy from the Retention Policy

- Use the `CHANGE ... KEEP` command to exempt a backup or copy from the retention policy.
- Use the `CHANGE ... NOKEEP` command to cancel the exemption.
The **CHANGE ... KEEP** Command

- Create a long-term backup:
  
  ```sql
  CHANGE BACKUPSET 123 KEEP FOREVER NOLOGS;
  ```

- Make a datafile exempt from the retention policy for 60 days:
  
  ```sql
  CHANGE DATAFILECOPY '/DB01/BACKUP/users01.dbf'
  KEEP UNTIL 'SYSDATE+60';
  ```
Cataloging Archived Redo Log Files and User-Managed Backups

You can use the `CATALOG` command to add information to the repository about:

- An operating system datafile copy
- An archived redo log copy
- A control file copy
The \texttt{CATALOG} Command

- Catalog a backup taken with an operating system command:
  \begin{verbatim}
  CATALOG DATAFILECOPY '/DB01/BACKUP/users01.dbf';
  \end{verbatim}

- Catalog archived redo log files:
  \begin{verbatim}
  CATALOG ARCHIVELOG
  '/ORADATA/ARCHIVE1/arch_12.arc',
  '/ORADATA/ARCHIVE1/arch_13.arc';
  \end{verbatim}
Uncataloging RMAN Records

Use the `CHANGE ... UNCATALOG` command to:

- Update the record in the repository to `DELETED` status
- Delete a specific backup or copy record from the recovery catalog
The `CHANGE ... UNCATALOG` Command

- Remove records for deleted archived redo log files:
  
  ```
  CHANGE ARCHIVELOG ... UNCATALOG;
  ```

- Remove records for a deleted datafile:
  
  ```
  CHANGE DATAFILECOPY '/DB01/BACKUP/users01.dbf' UNCATALOG;
  ```
In this lesson, you should have learned how to:

• Perform cross-checking of backups and copies
• Update the repository when backups have been deleted
• Change the availability status of backups and copies
• Make a backup or copy exempt from the retention policy
• Catalog backups made with operating system commands
Practice 16 Overview

This practice covers the following topics:

• Performing cross checking
• Cataloging files in the repository
Recovery Catalog Creation and Maintenance
Objectives

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

• Describe the contents of the recovery catalog
• List the RMAN features which require the recovery catalog
• Create the recovery catalog
• Maintain the recovery catalog by using RMAN commands
• Use RMAN to register, resynchronize, and reset a database
Objectives

- Query the recovery catalog to generate reports and lists
- Create, store, and run scripts
- Describe methods for backing up and recovering the recovery catalog
Overview

Target database

Restore/Recovery

Backup

Control file

Recovery Manager (RMAN)

Catalog Maintenance
Register
Resynchronize
Reset
Change/Delete/Catalog
Backup /Restore/Recover

Reporting REPORT LIST

Stored scripts

Catalog database

Enterprise Manager
Recovery Catalog Contents

The recovery catalog is an optional repository containing information on:

• Datafile and archived redo log file backup sets and backup pieces
• Datafile copies
• Archived redo log files
• The physical structure of the target database
Recovery Catalog Contents

The recovery catalog can also contain:

- Persistent RMAN configuration settings
- Stored job scripts
Benefits of Using a Recovery Catalog

The following features are available only when you use a recovery catalog:

- Metadata about multiple target databases in one catalog
- Metadata about multiple incarnations of a single target database
- Historical metadata
- Reporting on the target database at a noncurrent time
Create Recovery Catalog

1. Create tablespace
2. Create catalog owner
3. Grant privileges
4. Create catalog
5. Connect to target database
6. Register target database

Target database
Catalog database
Control file
Connecting Using a Recovery Catalog

- Example on UNIX

```
$ ORACLE_SID=db01; export ORACLE_SID
$ rman target /
RMAN> connect catalog rman_db01/rman_db01@catdb
```

- Example on Windows NT

```
C:\> set ORACLE_SID=db01
C:\> rman target /
RMAN> connect catalog rman_db01/rman_db01@catdb
```
Recovery Catalog Maintenance

- Target database
- Control file
- Backup
- Restore / Recovery
- Recovery Manager (RMAN)
- Reporting REPORT LIST
- Stored scripts
- Catalog Maintenance
  - Register
  - Resynchronize
  - Reset
  - Change/Delete/Catalog
  - Backup /Restore/Recover
- Catalog database

Enterprise Manager
Catalog Maintenance
Resynchronization of the Recovery Catalog

Resynchronization of the recovery catalog happens:

- Automatically with `BACKUP` and `COPY` commands
- Manually with `RESYNC CATALOG` command
Using **RESYNC CATALOG** for Resynchronization

Issue the **RESYNC CATALOG** command when you:

- Add or drop a tablespace
- Add or drop a datafile
- Relocate a database file

```
$ rman target / catalog rman/rman@catdb
RMAN> RESYNC CATALOG;
```
Resetting a Database Incarnation

- Use the `RESET DATABASE` command:
  - To direct RMAN to create a new database incarnation record
  - To distinguish between opening with `RESETLOGS` and an accidental restore operation of an old control file
- Open with `RESETLOGS` after `RESET DATABASE`
Viewing the Recovery Catalog

Data dictionary views:

- RC_DATABASE
- RC_DATAFILE
- RC_STORED_SCRIPT
- RC_STORED_SCRIPT_LINE
- RC_TABLESPACE
Script Examples

- **Use CREATE SCRIPT to store a script.**

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

- **Use EXECUTE SCRIPT to run a script.**

```sql
RMAN > run {execute script Level0Backup;}
```
Managing Scripts

- Use `REPLACE SCRIPT` to rewrite a script

```sql
RMAN> REPLACE SCRIPT Level0Backup {
    ...
    fileperset 3
    ...
}
```

- Use `DELETE SCRIPT` to remove a script

```sql
RMAN> DELETE SCRIPT Level0Backup;
```

- Use `PRINT SCRIPT` to display a script

```sql
RMAN> PRINT SCRIPT Level0Backup;
```
Backup of Recovery Catalog

• Whole database backup of the database containing the recovery catalog
• Tablespace backup of the tablespace containing the recovery catalog
• Export:
  – If catalog database is not very large, you can export the database at regular intervals.
  – If catalog database is large, export the schema containing the recovery catalog.
Recovering the Recovery Catalog

- Create a database from a previous backup of the recovery catalog database.
- Relocate the catalog into another database and import the data.
- Import the entire database from an export.
Summary

In this lesson, you should have learned that:

• Before using the recovery catalog, you must register the target database
• You should resynchronize the catalog frequently using the control file
• Scripts can be stored in the recovery catalog
Practice 17 Overview

This practice covers the following topics:

• Creating the recovery catalog
• Registering a target database with the recovery catalog
• Listing the incarnation of a target database
• Storing a script in the recovery catalog and executing it
Transporting Data Between Databases
Objectives

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

- Describe the uses of the Export and Import utilities
- Describe Export and Import concepts and structures
- Perform simple Export and Import operations
- List guidelines for using Export and Import
Oracle Export and Import Utilities
Oracle Export and Import Utility Overview

You can use these utilities to do the following:

- Archive historical data
- Save table definitions to protect them from user error failure
- Move data between machines and databases or between different versions of the Oracle server
- Transport tablespaces between databases
Methods of Invoking the Export and Import Utilities

- Command-line interface
- An interactive dialog
- Parameter files
- Oracle Enterprise Manager
## Export Modes

<table>
<thead>
<tr>
<th>Table Mode</th>
<th>User Mode</th>
<th>Tablespace Mode</th>
<th>Full Database Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table definitions</td>
<td>Tables definitions</td>
<td>Table definitions</td>
<td>Tables definitions</td>
</tr>
<tr>
<td>Table data (all or</td>
<td>Tables data</td>
<td>Tables data</td>
<td>Tables data</td>
</tr>
<tr>
<td>selected rows)</td>
<td>Owner’s grants</td>
<td>Owner’s grants</td>
<td>Grants</td>
</tr>
<tr>
<td>Owner’s table</td>
<td>Owner’s indexes</td>
<td>Indexes</td>
<td>Indexes</td>
</tr>
<tr>
<td>indexes</td>
<td>Tables constraints</td>
<td>Table constraints</td>
<td>Tables constraints</td>
</tr>
<tr>
<td>Table constraints</td>
<td></td>
<td>Triggers</td>
<td></td>
</tr>
</tbody>
</table>
Invoking Export

• Syntax:

```
exp keyword = value, value2, ... ,valuen
```

• Examples:

```
exp hr/hr TABLES=employees,departments
    rows=y file=exp1.dmp

exp system/manager OWNER=hr direct=y
    file=expdat.dmp

exp '\username/password AS SYSDBA\'
    TRANSPORT_TABLESPACE=y
    TABLESPACES=ts_emp log=ts_emp.log
```
Using the Export Wizard
Direct-Path Export Concepts

- **Export**
  - Generate SQL commands
  - Dump file

- **Oracle Server**
  - Two-Task common (TTC)
  - SQL command processing
  - Buffer cache manager

- **Direct Path**
  - Write blocks
  - Read blocks

- **Conventional Path**
Direct-Path Export Features

- The type of Export is indicated on the screen output, export dump file, and the log file.
- Data is already in the format that Export expects, avoiding unnecessary data conversion.
- Uses an optimized SQL `SELECT` statement.
Direct-Path Export Restrictions

• The direct-path option cannot be invoked interactively.
• Client-side and server-side character sets must be the same.
• The buffer parameter has no affect.
• You cannot use the direct-path option to export rows containing LOB, BFILE, REF, or object types.
Specifying Direct-Path Export

- As command line argument to the Export command:
  ```
  exp userid=hr/hr full=y direct=y
  ```

- As a keyword in a parameter file:
  ```
  exp parfile=<Parameter file>
  ```

Parameter file
```
.....(Other Parameters)
DIRECT = Y
.....(Other Parameters)
```
Uses of the Import Utility for Recovery

- Create table definitions
- Extract data from a valid Export file
- Import from a complete or cumulative Export file
- Recover from user-error failures
## Import Modes

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>Import specified tables into a schema.</td>
</tr>
<tr>
<td>User</td>
<td>Import all objects that belong to a schema</td>
</tr>
<tr>
<td>Tablespace</td>
<td>Import all definitions of the objects contained in the tablespace</td>
</tr>
<tr>
<td>Full Database</td>
<td>Import all objects from the export file</td>
</tr>
</tbody>
</table>
Invoking Import

• Syntax:

imp keyword = value or keyword = value,
value2, ... value n

• Examples:

imp hr/hr TABLES=employees,departments
rows=y file=exp1.dmp

imp system/manager FROMUSER=hr file=exp2.dmp

imp \'username/password AS SYSDBA\'
TRANSPORT_TABLESPACE=y
TABLESPACES=ts_employees
Using the Import Wizard

![Import Wizard screenshot]

**Import File**

Specify the full path and name for the import file(s) on the database server machine.

- **File Name**
  - `/oracle/ed21/oradata/u01/EXPDAT.DMP`

Select from the following import choices:

- **Manually specify what you want to import.**
  - You need to know the contents of the import file(s) to correctly use this option.
- **Read import file(s) and select what objects you want to import.**
  - An Enterprise Manager job is submitted to read the import file(s). The contents of the file(s) are then displayed graphically and you can then easily select the objects you want to import.

[Next button]
Invoking Import as SYSDBA

• You need to invoke Import as SYSDBA under the following conditions:
  – At the request of Oracle technical support
  – When importing a transportable tablespace set

• To invoke Import as SYSDBA:

  imp \'username/password AS SYSDBA\'
Import Process Sequence

1. New tables are created
2. Data is imported
3. Indexes are built
4. Triggers are imported
5. Integrity constraints are enabled on the new tables
6. Any bitmap, functional, and/or domain indexes are built
Globalization Support Considerations

- The Export file identifies the character encoding scheme used for the character data in the file.
- The Import utility translates data to the character set of its host system.
- A multibyte character set Export file must be imported into a system that has the same characteristics.
Summary

In this lesson, you should have learned how to:

- Describe the uses of Export and Import
- Describe Export and Import concepts and structures
- Perform simple Export and Import operations
- List guidelines for using Export and Import
Practice 18 Overview

This practice covers the following topics:
• Using the Export utility
• Using the Import utility
Workshop
Objectives

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

• Document a database configuration by using the Database Configuration Worksheet
• Configure an Oracle9i database to support stated business requirements
• Recover a failed database while minimizing down time and data loss
Objectives

- Enable and use trace output for troubleshooting
- Identify and troubleshoot:
  - Listener problems
  - Client configuration issues
Workshop Methodology

- Group-oriented and interactive
- Intensive hands-on diagnosis and problem resolution
- Variety of failure scenarios
- Recovery solutions
- Variety of configuration errors
- Develop troubleshooting skills
Workshop Approach

• Physical investigation:
  – Use views and tools to derive information
  – View trace files and log files
  – View command output and log files
  – Use views and tools to confirm proper database configuration

• Database configuration:
  – Archiving is enabled
  – Control files and log files are mirrored
  – Control file is backed up
Business Requirements

• Twenty-four hour availability
• Peak usage varies across all time zones
• Daily backups are required
• Complete database recovery is required
Resolving a Database Failure

• Phase I: Diagnose the problem
• Phase II: Restore appropriate files
• Phase III: Recover the database
• Phase IV: Back up the database
Resolving a Network Failure

- Use OS utilities like ping and telnet to test network connectivity
- Use Oracle Net utilities like tnsping to test service connectivity
- Check log files initially to diagnose problems
- Use trace files to get detailed information
- Use tracing sparingly and trim the trace files regularly because of their potential for rapid growth
Enable Tracing

- Oracle Net Manager
- Edit sqlnet.ora file

```
TRACE_DIRECTORY_CLIENT = /u01/user01/NETWORK/LOG
NAMES.DEFAULT_DOMAIN = us.oracle.com
TRACE_UNIQUE_CLIENT = on
TRACE_FILE_CLIENT = client.trc
TRACE_LEVEL_CLIENT = SUPPORT
NAMES.DIRECTORY_PATH= (TNSNAMES)
```
Using Trace Files

Trace files will give you a better understanding of:

- The flow of packets between network nodes
- Which component of Oracle Net is failing
- Pertinent error codes
Summary

• Instructor-facilitated workshop
• Group-oriented
• Hands-on approach
• Simulated “real-world” environment
• Minimize down time and data loss
• Use tools and diagnostics to determine the type of failure
Loading Data into a Database
Objectives

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

• Demonstrate usage of direct-load insert operations
• Describe the usage of SQL*Loader
• Perform basic SQL*Loader operations
• List guidelines for using SQL*Loader and direct-load INSERT
Data Loading Methods

Other applications

SQL*Loader

Oracle database

Export

Import

Direct-load INSERT

Oracle database
Direct-Load **INSERT**

Direct-load **INSERT** can be performed in the following ways:

- Normal (serially) or in parallel
- Into partitioned tables, nonpartitioned tables, or single partitions of a table
- With or without logging of redo data
Serial Direct-Load INSERT

```
INSERT /*+ APPEND */ INTO emp
NOLOGGING
SELECT * FROM t_employees;
COMMIT;
```
Parallel Direct-Load INSERT

```
ALTER SESSION ENABLE PARALLEL DML;
INSERT /*+PARALLEL(hr.employees,2) */
INTO hr.employees NOLOGGING
SELECT * FROM hr.old_employees;
```
Conventional and Direct Path Loads

- **Conventional**
- **Direct path**

**Table**

- Space used only by conventional load

**Instance**

- SGA
- Shared pool

**Data save**

**High-water mark**
### Comparing Direct and Conventional Path Loads

<table>
<thead>
<tr>
<th>Conventional Load</th>
<th>Direct Path Load</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Uses</strong> \texttt{COMMIT}s to make changes permanent</td>
<td>Uses data saves</td>
</tr>
<tr>
<td>Redo log entries always generated</td>
<td>Generates redo only under specific conditions</td>
</tr>
<tr>
<td>Enforces all constraints</td>
<td>Enforces only primary key, unique, and \texttt{NOT NULL}</td>
</tr>
<tr>
<td>\texttt{INSERT} triggers fire</td>
<td>\texttt{INSERT} triggers do not fire</td>
</tr>
<tr>
<td>Can load into clustered tables</td>
<td>Cannot load into clustered tables</td>
</tr>
<tr>
<td>Other users can make changes to tables</td>
<td>Other users cannot make changes to tables</td>
</tr>
</tbody>
</table>
Parallel Direct-Path Load

Table

load1.dat
load1.ctl

load2.dat
load2.ctl

load3.dat
load3.ctl

SQL*Loader

Temporary segments

High-water mark

Copyright © Oracle Corporation, 2001. All rights reserved.
Using SQL*Loader

```bash
$sqlldr hr/hr \
> control=case1.ctl \n> log=case1.log direct=Y
```
SQL*Loader Control File

The loader control file tells SQL*Loader:

- Where to find the load data
- The data format
- Configuration details
  - Memory management
  - Record rejection
  - Interrupted load handling details
- How to manipulate the data
Control File Syntax Considerations

- The syntax is free-format
- Syntax is case insensitive
- Comments extend from the two hyphens (--) that mark the beginning of the comment to the end of the line
- The CONSTANT keyword is reserved
Input Data and Datafiles

• SQL*Loader reads data from one or more files specified in the control file
• From SQL*Loader’s perspective, the data in the datafile is organized as records
• A datafile can be in one of three formats:
  – Fixed-record format
  – Variable-record format
  – Stream-record format
Logical Records

SQL*Loader can be instructed to follow one of the following two logical record-forming strategies:

• Combine a fixed number of physical records to form each logical record
• Combine physical records into logical records while a certain condition is true
Data Conversion

During a conventional path load, data fields in the datafile are converted into columns in the database in two steps:

• The field specifications in the control file are used to interpret the format of the datafile and convert it to a SQL INSERT statement using that data.

• The Oracle database server accepts the data and executes the INSERT statement to store the data in the database.
Discarded or Rejected Records

- **Bad file**
  - SQL*Loader rejects records when the input format is invalid
  - If the Oracle database finds that the row is invalid, the record is rejected and SQL*Loader puts it in the bad file

- **Discard file**
  - This can be used only if it has been enabled
  - This file contains records that were filtered out because they did not match any record-selection criteria specified in the control file
Log File Contents

• Header Information
• Global Information
• Table Information
• Datafile Information
• Table Load Information
• Summary Statistics
• Additional statistics for direct path loads and multithreading Information
SQL*Loader Guidelines

- Use a parameter file to specify commonly used command line options
- Place data within the control file only for a small, one-time load
- Improve performance by:
  - Allocating sufficient space
  - Sorting the data on the largest index
  - Specifying different files for temporary segments for parallel loads
Summary

In this lesson, you should have learned how to:

• Describe the usage of SQL*Loader
• Perform basic SQL*Loader operations
• Demonstrate proficiency using direct-load INSERT operations
• List guidelines for using SQL*Loader and direct-load INSERT
Practice 19 Overview

This practice covers the following topics:
• Using SQL*Loader to restore data
  – Using a control file
  – Using a datafile
• Using direct-load insert to load data