Course Description

Overview
In this class, experienced DBAs will learn the fundamentals of backup and recovery, managing Oracle Net and configuring the shared server architecture. Students will learn guidelines and techniques for performing backup and recovery in an Oracle environment. Students will perform backup and recovery operations using ARCHIVELOG and NOARCHIVELOG mode. Administration issues of managing the internal components of an Oracle server that are related to backup and recovery are also covered. Students will perform complete and incomplete recovery using real life scenarios. This class will cover advanced diagnostic techniques for solving recovery issues. Recovery manager will be covered for performing backup and recovery. Over 40% of class time is devoted to hands-on labs with real life scenarios.

Prerequisites
Students must understand the fundamentals of Oracle9i database administration. At least six months experience with managing an Oracle9i server is recommended.

Objectives
Upon completion of this course, students will understand:
♦ The Oracle9i architecture in relation to backup and recovery
♦ Guidelines, good practices and techniques for performing backup and recovery
♦ Backup and recovery in NOARCHIVELOG mode
♦ Backup and recovery in ARCHIVELOG mode
♦ Management issues for redo logs and archiving
♦ Point-in-time, cancel-based and SCN-based recovery
♦ Fundamentals of Recovery Manager
♦ Fundamental backup and recovery concepts
♦ New Oracle9i features related to backup and recovery
♦ How to configure and manage the shared server architecture
♦ Fundamentals of Oracle Net Services

Duration
Five days
## Overview of Backup and Recovery

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter Overview</td>
<td>1-1</td>
</tr>
<tr>
<td>Factors Impacting Backup and Recovery</td>
<td>1-3</td>
</tr>
<tr>
<td>Understanding why Systems Fail</td>
<td>1-4</td>
</tr>
<tr>
<td>Types of Oracle Failures</td>
<td>1-5</td>
</tr>
<tr>
<td>Scenario Testing</td>
<td>1-6</td>
</tr>
<tr>
<td>Understanding your Hardware</td>
<td>1-7</td>
</tr>
<tr>
<td>Backing up Different Environments</td>
<td>1-8</td>
</tr>
<tr>
<td>Factors Impacting Strategies</td>
<td>1-9</td>
</tr>
<tr>
<td>Oracle Backup Options</td>
<td>1-11</td>
</tr>
<tr>
<td>Types of Oracle Recovery</td>
<td>1-12</td>
</tr>
<tr>
<td>Database Files</td>
<td>1-13</td>
</tr>
<tr>
<td>Views used in Backup and Recovery</td>
<td>1-14</td>
</tr>
<tr>
<td>Summary</td>
<td>1-15</td>
</tr>
<tr>
<td>Exercises</td>
<td>1-16</td>
</tr>
</tbody>
</table>

## Managing Instance Recovery

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter Overview</td>
<td>2-1</td>
</tr>
<tr>
<td>The LRU and Dirty Lists</td>
<td>2-3</td>
</tr>
<tr>
<td>Server Processes and the Buffer Cache</td>
<td>2-4</td>
</tr>
<tr>
<td>DBWR and the Buffer Cache</td>
<td>2-5</td>
</tr>
<tr>
<td>Checkpoints</td>
<td>2-6</td>
</tr>
<tr>
<td>Checkpoint Events</td>
<td>2-7</td>
</tr>
<tr>
<td>LOG CHECKPOINT Parameters</td>
<td>2-8</td>
</tr>
<tr>
<td>Using LOG CHECKPOINT Parameters</td>
<td>2-9</td>
</tr>
<tr>
<td>MTTR and Checkpoint intervals</td>
<td>2-10</td>
</tr>
<tr>
<td>Oracle8i Methods for Controlling Checkpointing</td>
<td>2-11</td>
</tr>
<tr>
<td>Methods for Controlling Checkpointing in Oracle9i</td>
<td>2-12</td>
</tr>
<tr>
<td>V$INSTANCE_RECOVERY View</td>
<td>2-13</td>
</tr>
<tr>
<td>Minimal I/O Recovery</td>
<td>2-14</td>
</tr>
<tr>
<td>Setting different levels of Checkpointing</td>
<td>2-15</td>
</tr>
<tr>
<td>Checkpoint and File Queues</td>
<td>2-16</td>
</tr>
<tr>
<td>Checkpoint Queue</td>
<td>2-17</td>
</tr>
<tr>
<td>Instance Recovery</td>
<td>2-18</td>
</tr>
<tr>
<td>Rolling Forward and Backward</td>
<td>2-19</td>
</tr>
<tr>
<td>FAST_START_PARALLEL_ROLLBACK</td>
<td>2-20</td>
</tr>
<tr>
<td>Controlling DBWR Activity</td>
<td>2-21</td>
</tr>
<tr>
<td>Multiple DBWR IO Processes</td>
<td>2-22</td>
</tr>
</tbody>
</table>
Oracle9i DBA Fundamentals II

Fundamentals of Oracle Backups

Chapter Overview .......................................................... 3-1
Good Design Fundamentals and Guidelines ................................... 3-3
Additional Guidelines ............................................................ 3-4
What Needs to Be Backed Up? .............................................. 3-5
Different Types of Backups .................................................. 3-6
Logical and Physical Backups .............................................. 3-7
Operating System Backups .................................................. 3-8
Cold Backups ...................................................................... 3-9
Hot Backups ....................................................................... 3-10
Logical Backups ................................................................. 3-11
Performing a Full Cold Backup .......................................... 3-12
Automating Backup and Recovery Tasks ......................... 3-13
Basic Unix commands ....................................................... 3-14
Summary .......................................................................... 3-15
Exercises ........................................................................... 3-16

Recovery in Noachivelog Mode (without RMAN)

Chapter Overview .......................................................... 4-1
Different Recovery Environments ....................................... 4-3
The Alert Log File .............................................................. 4-4
Oracle Trace Files .............................................................. 4-5
Recovery in NOARCHIVELOG Mode ............................. 4-6
Access the Situation Before Performing Recovery .............. 4-7
Recovery from Media Failure ........................................... 4-8
Restore a Database to a New Location .............................. 4-9
Datafile Loss in NOARCHIVELOG Mode ...................... 4-10
Advantages and Disadvantages of NOARCHIVELOG Mode ...... 4-11
Managing Control Files .................................................... 4-12
When to Backup the Control File .................................. 4-13
The CREATE CONTROLFILE Command ...................... 4-14
Creating a CONTROLFILE Script in Unix .................. 4-15
Recovery using a Backup Control File ......................... 4-16
The Create Control File Command ................................ 4-17
Managing Multiple Control Files .................................. 4-18
Summary .......................................................................... 4-19
Exercises ........................................................................... 4-20
Redo Log Files and Recovery (without RMAN) 5-1

Chapter Overview ................................................................. 5-1
The Role of the Archiver Process.............................................. 5-3
Setting up Archiving ............................................................. 5-4
Setting LOG_ARCHIVE_FORMAT ............................................ 5-5
Archive Destinations ............................................................. 5-6
Version 7 Style Archive Destinations ..................................... 5-7
Converting the ArchiveLog Mode .......................................... 5-7
Converting the ArchiveLog Mode .......................................... 5-8
Mandatory or Optional Destinations ...................................... 5-9
Managing Archive Destinations ............................................. 5-10
Dynamic Archive Parameters .............................................. 5-11
Archive Information ............................................................. 5-12
Managing Online Redo Log Files ........................................ 5-13
Getting Information on Redo Logs ....................................... 5-14
Redo Log Failure Scenarios ................................................ 5-15
Recovering Redo Log Files ................................................ 5-16
Recovery After Losing an Online Redo Log File .................... 5-17
Replacing a Redo Log Member .............................................. 5-18
Determine Invalid Members in a Group .............................. 5-19
Clearing a Redo Log Group ................................................ 5-20
Clearing an Active Redo Log Group ...................................... 5-21
Accessing Redo Log Information .......................................... 5-22
Summary .............................................................................. 5-23
Exercises ............................................................................. 5-24

Recovery in Archivelog Mode (without RMAN) 6-1

Chapter Overview ................................................................. 6-1
Recovery in ARCHIVELOG Mode .......................................... 6-3
Recovery Information ........................................................... 6-4
Performing Media Recovery in Parallel ............................... 6-5
Media Recovery ................................................................. 6-6
Closed Database Recovery .................................................. 6-7
Issues of Backing up Online Redo Log Files ....................... 6-8
The RECOVER Command .................................................... 6-9
Recovery Options ............................................................... 6-10
Recovery Examples ............................................................ 6-11
ALTER DATABASE Examples ................................................. 6-12
ALTER DATABASE Examples (Continued) ............................ 6-13
Database Recovery Guidelines ............................................. 6-14
Recovering a SYSTEM or Rollback Segment Data File ........... 6-15
Recovering a Data File ....................................................... 6-16
Recovering a Data File ....................................................... 6-17
Recovering a Data File With No Backup .............................. 6-18
# Diagnostics and Troubleshooting

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter Overview</td>
<td>8-1</td>
</tr>
<tr>
<td>Detecting Software Corrupt Blocks</td>
<td>8-3</td>
</tr>
<tr>
<td>Review: DB_BLOCK_CHECKSUM Parameter</td>
<td>8-4</td>
</tr>
<tr>
<td>Review: DB_BLOCK_CHECKING Parameter</td>
<td>8-5</td>
</tr>
<tr>
<td>DBVERIFY</td>
<td>8-6</td>
</tr>
<tr>
<td>DBVERIFY Parameters</td>
<td>8-7</td>
</tr>
<tr>
<td>DBVERIFY Output</td>
<td>8-8</td>
</tr>
<tr>
<td>DBVERIFY Example</td>
<td>8-9</td>
</tr>
<tr>
<td>The DBMS_REPAIR Package</td>
<td>8-10</td>
</tr>
<tr>
<td>DBMS_REPAIR Restrictions</td>
<td>8-11</td>
</tr>
<tr>
<td>DBMS_REPAIR Constants</td>
<td>8-12</td>
</tr>
<tr>
<td>DBMS_REPAIR Exceptions</td>
<td>8-13</td>
</tr>
<tr>
<td>DBMS_REPAIR.ADMIN_TABLES Procedure</td>
<td>8-14</td>
</tr>
<tr>
<td>DBMS_REPAIR.CHECK_OBJECT Procedure</td>
<td>8-15</td>
</tr>
<tr>
<td>DBMS_REPAIR.FIX_CORRUPT_BLOCKS Procedure</td>
<td>8-16</td>
</tr>
<tr>
<td>DBMS_REPAIR.SKIP_CORRUPT_BLOCKS Procedure</td>
<td>8-17</td>
</tr>
<tr>
<td>Issues of Skipping Corrupt Blocks</td>
<td>8-18</td>
</tr>
<tr>
<td>DBMS_REPAIR.DUMP_ORPHAN_KEYS Procedure</td>
<td>8-19</td>
</tr>
<tr>
<td>DBMS_REPAIR.DUMP_ORPHAN_KEYS Procedure (2)</td>
<td>8-20</td>
</tr>
<tr>
<td>DBMS_REPAIR.DUMP_ORPHAN_KEYS Procedure (3)</td>
<td>8-21</td>
</tr>
<tr>
<td>DBMS_REPAIR.REBUILD_FREELISTS Procedure</td>
<td>8-22</td>
</tr>
<tr>
<td>The Log Miner Utility</td>
<td>8-23</td>
</tr>
<tr>
<td>LogMiner Packages</td>
<td>8-24</td>
</tr>
<tr>
<td>V$LOGMNR_CONTENTS</td>
<td>8-25</td>
</tr>
<tr>
<td>Creating a LogMiner Data Dictionary File</td>
<td>8-26</td>
</tr>
<tr>
<td>Using LogMiner to Analyze Log Files</td>
<td>8-27</td>
</tr>
<tr>
<td>Analyzing the Log Files with DBMS_LOGMNR</td>
<td>8-28</td>
</tr>
<tr>
<td>START_LOGMNR and END_LOGMNR</td>
<td>8-29</td>
</tr>
<tr>
<td>ANALYZE VALIDATE STRUCTURE Command</td>
<td>8-30</td>
</tr>
<tr>
<td>Summary</td>
<td>8-31</td>
</tr>
<tr>
<td>Exercises</td>
<td>8-32</td>
</tr>
</tbody>
</table>

# Recovery Manager Overview

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter Overview</td>
<td>9-1</td>
</tr>
<tr>
<td>Issues with Managing Backups</td>
<td>9-1</td>
</tr>
<tr>
<td>Issues with Managing Backups</td>
<td>9-3</td>
</tr>
<tr>
<td>What is Recovery Manager?</td>
<td>9-4</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>RMAN Interfaces</td>
<td>9-5</td>
</tr>
<tr>
<td>Working with Recovery Manager</td>
<td>9-6</td>
</tr>
<tr>
<td>Recovery Manager Components</td>
<td>9-7</td>
</tr>
<tr>
<td>Recovery Manager Components (example)</td>
<td>9-8</td>
</tr>
<tr>
<td>RMAN Compatibility</td>
<td>9-9</td>
</tr>
<tr>
<td>Using Recovery Manager</td>
<td>9-10</td>
</tr>
<tr>
<td>More on Recovery Manager (continued)</td>
<td>9-11</td>
</tr>
<tr>
<td>Using RMAN with Multiple Platforms</td>
<td>9-12</td>
</tr>
<tr>
<td>Advantages of RMAN</td>
<td>9-13</td>
</tr>
<tr>
<td>Advantages of RMAN (2)</td>
<td>9-14</td>
</tr>
<tr>
<td>Disadvantages of RMAN</td>
<td>9-15</td>
</tr>
<tr>
<td>Oracle9i RMAN Enhancements</td>
<td>9-16</td>
</tr>
<tr>
<td>Improved Error Output</td>
<td>9-17</td>
</tr>
<tr>
<td>Recovery Manager Terminology</td>
<td>9-18</td>
</tr>
<tr>
<td>Recovery Manager Terminology (2)</td>
<td>9-19</td>
</tr>
<tr>
<td>Review Questions</td>
<td>9-20</td>
</tr>
<tr>
<td>Summary</td>
<td>9-21</td>
</tr>
</tbody>
</table>

---

## Recovery Catalog

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter Overview</td>
<td>10-1</td>
</tr>
<tr>
<td>Recovery Manager without the Recovery Catalog</td>
<td>10-3</td>
</tr>
<tr>
<td>Recovery Manager without the Recovery Catalog (cont.)</td>
<td>10-4</td>
</tr>
<tr>
<td>Contents of the Recovery Catalog</td>
<td>10-5</td>
</tr>
<tr>
<td>Contents of the Recovery Catalog (cont.)</td>
<td>10-6</td>
</tr>
<tr>
<td>Catalog Considerations</td>
<td>10-7</td>
</tr>
<tr>
<td>Advantages of the Recovery Catalog</td>
<td>10-8</td>
</tr>
<tr>
<td>Disadvantages of the Recovery Catalog</td>
<td>10-9</td>
</tr>
<tr>
<td>Creating a Catalog Database</td>
<td>10-10</td>
</tr>
<tr>
<td>Creating a Catalog Database (example)</td>
<td>10-11</td>
</tr>
<tr>
<td>Connecting to a Catalog Database</td>
<td>10-12</td>
</tr>
<tr>
<td>Connecting without a Recovery Catalog</td>
<td>10-13</td>
</tr>
<tr>
<td>Connect to a Target Database with a Recovery Catalog</td>
<td>10-14</td>
</tr>
<tr>
<td>Connecting to an Auxiliary Instance</td>
<td>10-15</td>
</tr>
<tr>
<td>Connect Using a Script:</td>
<td>10-16</td>
</tr>
<tr>
<td>Register the Target Database with the Recovery Catalog</td>
<td>10-17</td>
</tr>
<tr>
<td>Unregister the Target Database with the Recovery Catalog</td>
<td>10-18</td>
</tr>
<tr>
<td>Unregister the Target Database with the Recovery Catalog</td>
<td>10-19</td>
</tr>
<tr>
<td>Resynching the Catalog and Target Databases</td>
<td>10-20</td>
</tr>
<tr>
<td>Review Questions</td>
<td>10-21</td>
</tr>
<tr>
<td>Summary</td>
<td>10-22</td>
</tr>
<tr>
<td>Summary</td>
<td>10-23</td>
</tr>
<tr>
<td>Exercises</td>
<td>10-24</td>
</tr>
</tbody>
</table>
Backup and Recovery Workshop Part II  

Chapter Overview ....................................................... 14-1  
Backup Archived Redo Logs ......................................... 14-3  
Backups of Archived Logs Needing Backup .................. 14-4  
The Backup Archivelog ... Delete Input command .......... 14-5  
Backup Archived Redo Logs (example) ......................... 14-6  
Backup Archived Redo Logs (example) ......................... 14-7  
Managing Space When Restoring Archived Logs .......... 14-8  
Backing up Archived Redo Logs Enhancements ............. 14-9  
Exercises 6-1 .............................................................. 14-10  
Restore Archived Redo Logs ........................................ 14-11  
Restore Archived Redo Logs ........................................ 14-12  
Archive Log Features .................................................. 14-13  
Exercises 6-2 .............................................................. 14-14  
Catalog Operating System Backups ......................... 14-15  
Catalog Operating System Backups ......................... 14-16  
Catalog Operating System Backups ......................... 14-17  
Exercize 6-3 .............................................................. 14-18  
Image Copies ............................................................. 14-19  
Image Copies (example) .............................................. 14-20  
Image Copies (example) .............................................. 14-21  
Image Copies (example) .............................................. 14-23  
Image Copies (example) .............................................. 14-24  
Exercises 6-4 .............................................................. 14-25  
Incomplete Recovery Using Until Time .................... 14-26  
Incomplete Recovery Using Until Time .................... 14-27  
Incomplete Recovery Using Until Time .................... 14-28  
Incomplete Recovery Using Until Time .................... 14-29  
Incomplete Recovery Using SCN ............................... 14-30  
Review Questions ...................................................... 14-34  
Summary ................................................................. 14-35  
Exercises 6-5 .............................................................. 14-36  

Introduction to Oracle Net Services  

Chapter Overview ....................................................... 15-1  
What is Oracle Net Services? ........................................ 15-3  
Oracle Net Services Fundamentals ............................. 15-4  
Oracle Net Services Fundamental (2) ......................... 15-5  
Oracle Net Services Terminology ............................... 15-6  
Oracle Net Services Terminology (2) ......................... 15-7
Chapter Overview

This chapter will provide DBAs with an overview of backup and recovery in an Oracle environment. Backup and recovery strategies as well as the environments that backup and recovery have to be performed in will be covered.

At the end of this chapter, students will understand:

- Fundamentals of backup and recovery strategies
- Different ways systems can fail
- How Oracle configurations have different backup issues
- Hardware options
- The different types of backups
- 24x7 environments
- Oracle backup options for performing backup and recovery
Factors Impacting Backup and Recovery

When developing a backup strategy, organizations must weigh different issues:

- Business issues.
- Impact of data loss or down time.
- Hardware options.
- Support costs.
- Management perspective.

As companies move towards Internet solutions and global markets, high availability is becoming more important than ever before.

Companies need to weigh organizational needs against administration flexibility.

- Organization issues include understanding the importance and issues of running 24x7:
  - Costs and ramifications of downtime.
- Administration flexibility includes understanding:
  - How often cold (full) backups need to be performed.
  - Occasional software updates (patches) to Oracle and operating system software.
  - Importance for reorganizing data segments for optimizing storage and minimizing fragmentation and reducing chaining/migration.
  - Running hot backups to minimize downtime.
  - The cost-based optimizer requires statistics be updated periodically.
  - Creating and rebuilding of index segments.
  - Executing large batch applications.
Understanding why Systems Fail

DBAs need to understand all the different ways a database can fail and the time and resources required to restore the system.

Some of the main reasons a database can fail include:

- Hardware failure.
- Software failure.
- Network failure.
- Application failure.
- Loss of power.
- DBA decisions.
- User actions.
- Natural disasters.
- Man-made disasters.
- Loss of personnel.
- Accidental loss of data.

Notes

Oracle databases can fail for a number of different reasons. A DBA must prepare and test for different type of failures. Failing to prepare usually means preparing to fail for backup and recovery scenarios.
Overview of Backup and Recovery

Types of Oracle Failures

DBAs also need to understand different ways failure can occur inside of a database:

- Instance crash.
- Unusable indexes.
- Loss of data integrity (loss of data or data modifications which corrupt data integrity).
- Broken procedures or functions.
- SQL.
- User actions.
- Process failure.
- Instance.
- Media failure.
- Security violations can bring down a system.
- Exceeding Oracle limits (extents, processes, sessions logical reads, etc.).
- Network failure.

N-tier architectures have created additional points of failure for database applications. DBAs also have to consider disaster and recovery scenarios for application servers.
Scenario Testing

A DBA needs to develop a recovery plan, then perform scenario testing of all possible methods of failure and evaluate the people, resources, time, and steps necessary to correct the problem.

It is important that DBAs have confidence in all backup and recovery scenarios that they may be involved in.

The recovery plan needs to be evaluated periodically and updated.

It is important everyone understand his or her role in a recovery scenario.

An emergency escalation list should be available to everyone involved in the recovery team.

Notes

Any questions about backup and recovery need to be addressed during scenario testing. This is where a DBA needs to develop skill and expertise.
Understanding your Hardware

A DBA needs to understand the hardware configuration running the Oracle Server:

- Number and types (hot swappable, SCSI, …) of disks and controllers.
- Amount of disk space being used and the amount of free space.
- Different hardware components and the vendors who supply the hardware (include time to replace hardware versus keeping spare parts on-site).
- Is loaner hardware an option?
- Physical memory.
- Is RAID available and what type of RAID is available?
- Is it a clustered environment?
- Number of CPUs.

DBAs are always trying to weigh the pros and cons of buying fewer large disks instead of a larger number of small disks.

- Large disks can offer more options for laying out RAID for striping and mirroring.
- For most applications, it is better to have a larger number of small disks. More disks offer better flexibility in laying out and distributing data.

Periodic capacity planning is important to remain proactive and to prepare for future recovery issues.
Backing up Different Environments

Each type of database environment has its own backup and recovery issues:

- OLTP.
- Hybrid.
- DSS.
- Client-server.
- Production versus test versus development.
- N-tier architecture.
  - Application or web servers, connection pooling, messaging systems.

Each Oracle configuration has its own backup and recovery issues:

- Dedicated server.
- Shared server architecture.
- Real Application Clusters (RAC).
- Replication.
- N-tier architecture.
- Data Guard.
- JServer.
- Catalog database (Recovery Manager).
- Oracle Net Services.
- Third party software solutions from CA (Brightstor), Quest, BMC, Veritas, etc.
Factors Impacting Strategies

DBAs have to weigh all the different factors before determining a backup and recovery strategy. The weight of importance is determined by business needs versus technical needs.

- Decrease Mean-Time-To-Recover (MTTR).
- Increase Mean-Time-Between-Failures Recovery (MTBF).
- Business costs of downtime and loss of data.
- Hardware resources and costs.
- Financial costs of minimizing failures.

Maintenance operations can impact availability. A DBA must be proactive in maintenance tasks to reduce the risks of large down times.

DBAs are constantly balancing performance versus increasing database availability.
Managing Recovery Scenarios

While managing the issues of recovering a database that is down, it is important that a DBA remain calm and correctly identify the problem and the action necessary to repair the system.

- The DBA will, hopefully, identify the scenario as one he or she has prepared for. The DBA can, then, follow the defined steps necessary to restore the database.
- Often in a recovery scenario, the first step a DBA takes is the wrong one, and the DBA can turn a small recovery scenario into a larger one.
- Think through the problem and walk through the solution before taking any action.

After a recovery scenario, access what caused the problem and determine what can be done to reduce the probability of the failure occurring again. Also, evaluate what can be done to improve the responsiveness and reduce the down time.
Oracle Backup Options

Oracle offers a number of different options for performing backup and recovery.

Some of these backup and recovery options include:

- Archivelog versus noarchivelog mode.
- Recovery Manager.
- Third party software.
- Hardware protection and redundancy (RAID).
- Hot versus cold backups.
- Complete and incomplete backups.
- Physical backups versus logical backups.
- Replication.
- Data Guard.
- Oracle Enterprise Manager (OEM) can be used to perform RMAN backups.
- Operating system backups.
Types of Oracle Recovery

Oracle supports different types of recovery:

- Block.
- Thread.
- Media.
- Complete.
- Incomplete.

A DBA must provide enough resources for trace files and alert logs.

A DBA must also prepare for periodic upgrades, reorganizations, and migrations.

- Practice upgrades, reorganizations, and migrations on a small scale before performing the operation on a larger system.
- Get buglists and additional FYI information from Oracle support or Metalink before running an upgrade or migration.

Before performing an upgrade, a DBA needs to perform scenario testing for a downgrade.
Overview of Backup and Recovery

Database Files

DBAs need to consider all the physical files that make up an Oracle database into the backup and recovery equation.

Oracle database files that need to be backed up include:

- Server parameter file and initialization parameter files.
- Control files.
- Database files.
- Redo log files.
- Archived redo log files.
- Administration scripts, logs and documentation.

Using the Optimal Flexible Architecture (OFA) can provide good guidelines to follow for laying out files to protect against recovery scenarios.
Views used in Backup and Recovery

V$ (dynamic) views are based on the X$ tables. The X$ objects are dynamic memory structures containing information on the instance. The following lists common views used for backup and recovery.

<table>
<thead>
<tr>
<th>Instance</th>
<th>Session</th>
</tr>
</thead>
<tbody>
<tr>
<td>v$instance</td>
<td>v$session</td>
</tr>
<tr>
<td>v$option</td>
<td>v$process</td>
</tr>
<tr>
<td>v$parameter</td>
<td>user session info.</td>
</tr>
<tr>
<td>v$bg_process</td>
<td>list background processes</td>
</tr>
<tr>
<td>v$process</td>
<td>data on active processes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Recovery</th>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>v$rollname</td>
<td>v$database</td>
</tr>
<tr>
<td>v$rollstat</td>
<td>v$datafile</td>
</tr>
<tr>
<td>v$undostat</td>
<td>v$controlfile</td>
</tr>
<tr>
<td>v$instance_recovery</td>
<td>v$log</td>
</tr>
<tr>
<td>v$recovery_log</td>
<td>v$logfile</td>
</tr>
<tr>
<td>v$recovery_progress</td>
<td>v$log_history</td>
</tr>
<tr>
<td>v$recovery_status</td>
<td>Recovery stats for rman</td>
</tr>
<tr>
<td>v$recovery_file_status</td>
<td>Info on datafiles</td>
</tr>
<tr>
<td>v$archived_log</td>
<td>v$sga</td>
</tr>
<tr>
<td>v$archive_dest</td>
<td>Arch log destinations</td>
</tr>
<tr>
<td>v$log_history</td>
<td>Log history info</td>
</tr>
<tr>
<td>v$archive_processes</td>
<td>State on arch processes</td>
</tr>
<tr>
<td>v$fast_start_transactions</td>
<td>Info on transactions</td>
</tr>
<tr>
<td>v$fast_start_servers</td>
<td>Parallel recovery slaves</td>
</tr>
</tbody>
</table>
Summary

A DBA needs to develop a recovery plan, then perform scenario testing of all possible methods of failure and evaluate the people, resources, time, and steps necessary to correct the problem.

While managing the issues of recovering a database that is down, it is important that a DBA remain calm and correctly identify the problem and the action necessary to repair the system.

Oracle offers a number of different options for performing backup and recovery. Some of these options include:

- Archivelog versus noarchivelog mode
- Recovery Manager
- 3rd party software
- Hardware protection and redundancy
- Hot versus cold backups
- Complete and incomplete backups
- Physical backups versus logical backups
- Replication
- Enterprise Backup Utility (EBU)
- Oracle Enterprise Manager (OEM)

DBAs have to weigh all the different factors before determining a backup and recovery strategy:

- Decrease Mean-Time-To-Recover
- Increase Mean-Time-Between-Failures Recovery (MTBF)
- Business costs of downtime and loss of data
- Hardware resources and costs
- Financial costs of minimizing failures
Exercises

A DBA needs to understand their environment before performing backup and recovery. Verify that you understand your class environment. Use scripts for all tasks you anticipate performing again. These scripts can then be used with any database you work with.

Information sources include:
V$DATAFILE – names of all the datafiles
V$LOG and V$LOGFILE – names and status of online redo log files
V$CONTROLFILE – names of all control files
V$DATABASE – name of database
V$PROCESS – location of processes still connected to system before shutting down

At the end of this lab you should write down information for all of the following.

1. List the values of important operating system variables.

<table>
<thead>
<tr>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORACLE_HOME</td>
</tr>
<tr>
<td>ORACLE_SID</td>
</tr>
<tr>
<td>PATH</td>
</tr>
</tbody>
</table>

2. List the values of your Oracle database file names and locations.

<table>
<thead>
<tr>
<th>File Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>data files</td>
</tr>
<tr>
<td>redo log files</td>
</tr>
<tr>
<td>control files</td>
</tr>
</tbody>
</table>
3. List the following values for your operating system and hardware.

<table>
<thead>
<tr>
<th>Operating system version</th>
<th>All available free disk space</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>All available filesystems</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Layout all of your administration directories</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Additional hardware configurations</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4. List the values of your current archiving environment.

<table>
<thead>
<tr>
<th>Archive mode</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Important archive parameters</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Archive destinations</th>
</tr>
</thead>
</table>

5. Run the following script to verify the log mode.

```sql
SQL> SELECT dbid, name, log_mode FROM v$database;
```

6. Run the following script to display the Global database name.

```sql
SQL> SELECT property_value FROM database_properties
WHERE property_name = 'GLOBAL_DB_NAME';
```
7. Run the following script on Unix to list the background processes.

```sql
SQL> SELECT s.program
       FROM v$session s, v$process p
       WHERE s.paddr = p.addr
       AND s.username is NULL;
```

If working on Windows, Oracle runs as a process and the background process run as threads. Run this script on Windows. Focus on processes that contain a valid memory address.

```sql
SQL> SELECT name, paddr, description
       FROM v$bgprocess
       ORDER BY paddr DESC;
```

List of users connected to the database. Log in as SCOTT and SYSTEM and then run the query below in the SYSTEM session.

```sql
SQL> SELECT s.username
       FROM v$session s, v$process p
       WHERE s.paddr = p.addr AND s.username IS NOT NULL;
```
Chapter Overview

At the end of this chapter the student will have had gained hands on experience creating a stored script and performing various backup and recovery scenarios.

At the end of this chapter, students will understand the following:

- Creating a Stored Script
- Backing up a Datafile
- Backing up a Tablespace with NOCATALOG
- Views to Determine if Recovery is Needed
- Recovering a Tablespace that is Accessible
- Recovering a Tablespace that is Inaccessible
Create a Stored Script

Stored scripts are a combination of SQL and RMAN commands.

- RMAN allows frequently used scripts to be stored in the recovery catalog and executed in the run command.
- Storing scripts in the recovery catalog has the advantage of making them available to any DBA with proper authority no matter where the DBA or the target database is located.

To store a script use the create script ‘script_name’ command:

```
RMAN> create script 'full_db' {
allocate channel ch1 type disk;
backup database;
}
RMAN-03022: compiling command: create script
RMAN-03023: executing command: create script
RMAN-08085: created script full_db
```

A script can be edited and then replaced in the recovery catalog:

```
RMAN> replace script 'full_db' {
allocate channel ch1 type disk;
backup database;
backup archivelog all;
}
```

A script can be deleted:

```
RMAN> delete script 'full_db';
```

A script can be printed to a file:

```
RMAN> print script 'full_db';
```

A script is executed in the run command:

```
RMAN> run {execute script 'full_db'}
```
Create a Stored Script

Recovery Manager can be used to create backup scripts:

```
RMAN> create script FullFinBackup {
    allocate channel c1 type disk;
    backup
    incremental level 0
    format '/disk1/fin/backup/%d_%s_%p'
    fileperset 4
    (database include current controlfile);
    sql 'alter database archive log current';
    release channel c1;}
```

To store a script, you must be connected to the recovery catalog. If you are not connected to a catalog, you will receive the following error:

```
RMAN-03002: failure during compilation of command
RMAN-03013: command type: create script
RMAN-06002: command not allowed when not connected to a recovery catalog
```

Notes

The RUN command can be used to execute host commands, RMAN scripts, and execute SQL commands.

UNIX Example:

```
RMAN > run {execute script FullFinBackup;}
RMAN > run { host "df -k"; }
RMAN > run { sql "alter system switch logfile";}
```
Exercises 13-1

1. Connect to the target database with the no catalog option.
2. Create a script named full_db to perform a full database backup.
3. What were the results of creating the script?
4. Connect to the target database with the catalog option.
5. Create a script named full_db to perform a full database backup.
6. Alter the script to include the archived redo logs.
7. Print the script to verify the script was changed.
Backup a Datafile

Datafiles may also be backed up with RMAN with the database either open (inconsistent) or closed (consistent).

- Closed means in the mounted state but not open.
- Do not issue the SQL command ALTER DATABASE BEGIN BACKUP before starting an online datafile backup.

To make a consistent backup of a datafile follow these steps:

- Perform a consistent shutdown of the target database (or make sure that the last shutdown was consistent i.e. not a crash or SHUTDOWN ABORT).
- If the option is available here, leave the database in the mounted state.

1. Connect to the target database with the nocatalog option.

   If desired, a log file can be specified with the startup command (log_file_spec).

   ```
   C:\>rman target /@orcl2 nocatalog
   Recovery Manager: Release 9.2.0.1.0 - Production
   RMAN-06005: connected to target database: ORCL2
   (DBID=446765398)
   RMAN-06009: using target database controlfile instead of recovery catalog
   ```

2. If the database is shut down, mount, but do not open, the database.

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

3. Issue the run command with the channel allocation and backup datafile file_number commands within the curly brackets. This particular example backs up datafile 1 and datafile 2.
Backup a Datafile (example)

Example:

RMAN> run {
    allocate channel ch1 type disk;
    backup(datafile 1,2);
}
RMAN-03022: compiling command: allocate
RMAN-03023: executing command: allocate
RMAN-08030: allocated channel: ch1
RMAN-08500: channel ch1: sid=14 devtype=DISK

RMAN-03022: compiling command: backup
RMAN-03023: executing command: backup
RMAN-08008: channel ch1: starting full datafile backupset
RMAN-08502: set_count=6 set_stamp=466304536
creation_time=03-JUL-02
RMAN-08010: channel ch1: specifying datafile(s) in backupset
RMAN-08522: input datafile fno=00001
name=C:\ORACLE\ORADATA\ORCL2\SYSTEM01.DBF
RMAN-08011: including current controlfile in backupset
RMAN-08522: input datafile fno=00002
name=C:\ORACLE\ORADATA\ORCL2\RBS01.DBF
RMAN-08013: channel ch1: piece 1 created
RMAN-08503: piece
handle=C:\ORACLE\ORA81\DATABASE\06DSMFGO_1_1 comment=NONE
RMAN-08525: backup set complete, elapsed time: 00:01:06
RMAN-08031: released channel: ch1

If the option FORMAT is not specified in the BACKUP command RMAN will automatically assign the %U substitution variable to guarantee a unique name.

4. The list command may be used to verify the backup.

RMAN> LIST BACKUPSET OF DATAFILE 1, 2;

Notes

Datafile image copies may be backed up in a similar fashion using the datafilecopy ‘file_spec’ option of the backup command instead of the datafile file_number option.
Datafile copies cannot have incremental backups applied to them.
Exercises 13-2

1. Connect to the target database with the nocatalog option.
2. Place the database into a mount state.
3. Backup datafiles 1 and 2.
4. Use the list command to find the valid backups of datafiles 1 and 2.
Making Backups of Tablespaces

Tablespaces may be backed up with RMAN with the database either open (inconsistent) or closed (consistent).

- Closed means in the mounted state but not open.
- IMPORTANT! Do not issue the SQL command ALTER DATABASE BEGIN BACKUP before starting an online tablespace backup.

To make a consistent backup of a tablespace follow these steps:

- Perform a consistent shutdown of the target database (or make sure that the last shutdown was consistent i.e. not a crash or SHUTDOWN ABORT).
- If the option is available here, leave the database in the mounted state.

1. Connect to the target database with the nocatalog option.
   - If desired a log file can be specified with the startup command (log_file_spec).

```
C:\>rman target /@orcl2 nocatalog
Recovery Manager: Release 9.2.0.1.0 - Production
RMAN-06005: connected to target database: ORCL2
(DBID=446765398)
RMAN-06009: using target database controlfile instead of recovery catalog
```

2. If the database is shut down, mount, but do not open, the database.

```
SQL> alter database mount;
```
Making Backups of Tablespaces (2)

3. Issue the run command with the channel allocation and backup tablespace commands within the curly brackets.

```
RMAN> run {
    allocate channel ch1 type disk;
    backup
    tablespace users;
}
RMAN-03022: compiling command: allocate
RMAN-03023: executing command: allocate
RMAN-08030: allocated channel: ch1
RMAN-08500: channel ch1: sid=14 devtype=DISK
RMAN-03022: compiling command: backup
RMAN-03023: executing command: backup
RMAN-08008: channel ch1: starting full datafile backupset
RMAN-08502: set_count=7 set_stamp=466378086
  creation_time=03-JUL-02
RMAN-08010: channel ch1: specifying datafile(s) in backupset
RMAN-08522: input datafile fno=00003
  name=C:\ORACLE\ORADATA\ORCL2\USERS01.DBF
RMAN-08013: channel ch1: piece 1 created
RMAN-08503: piece
  handle=C:\ORACLE\ORA81\DATABASE\07DSONB6_1_1 comment=None
RMAN-08525: backup set complete, elapsed time: 00:00:03
RMAN-08031: released channel: ch1
```

4. The list command may be used to verify the backup.

```
RMAN> LIST BACKUP OF TABLESPACE USERS;
```
Exercises 13-3

1. Connect to the target database with the nocatalog option.

2. Place the database into a mount state.

3. Backup the USERS tablespace.

4. Use the list command to find the valid backups of the users tablespace.
Views to Determine if Recovery is Needed

V$DATAFILE_HEADER will read the header block of each datafile and report the status of each datafile.

This view will not show report an error if there is a corrupt block within a datafile.

```
SQL> SELECT FILE#, STATUS, ERROR, RECOVER, TABLESPACE_NAME, NAME FROM V$DATAFILE_HEADER;
```

V$DATAFILE and V$TABLESPACE can be used to determine the tablespace name for the datafiles needing recovery.

```
SQL> SELECT r.FILE# AS df#, d.NAME AS df_name, t.NAME AS tbsp_name, d.STATUS, r.ERROR, r.CHANGE#, r.time FROM V$RECOVER_FILE r, V$DATAFILE d, V$TABLESPACE t WHERE t.TS# = d.TS# AND d.FILES# = r.FILE#;
```

Notes

In addition to the previously mentioned views, it is essential to review the database's alert log. This will provide essential information as to when and where the errors first started.

If Oracle Support is needed for assistance in a recovery situation, it is essential that Oracle Support receive a copy of the alert log and any associated trace files as soon as an issue is opened. Having this information ready will greatly reduce the time needed for Oracle Support to assist in resolving the recovery issue.
Recover a Tablespace that is Accessible

This procedure allows you to restore and recover the USERS tablespace while the database is closed.

All datafiles associated with the USERS tablespace will be restored automatically with this procedure.

After the datafiles are restored, the recovery process will apply redo to bring the backed datafiles to the current status with the rest of the database.

1. Connect to RMAN with the NOCATALOG option

C:\> rman target /@orcl2 nocatalog

Recovery Manager: Release 9.2.0.1.0 - Production

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

2. Place the database into a MOUNT state. If the status of the database is OPEN issue a shutdown and then mount the database.

SQL> startup mount;

Using RMAN recover and restore the tablespace users.

RMAN> run {
2> allocate channel ch2 type disk;
3> restore tablespace users;
4> recover tablespace users;
5> }

RMAN-03022: compiling command: allocate
RMAN-03023: executing command: allocate
RMAN-03023: executing command: IRESTORE
RMAN-08016: channel ch2: starting datafile backupset restore
RMAN-08502: set_count=7 set_stamp=466378086
creation_time=03-JUL-02
Recover a Tablespace that is Accessible

RMAN-08089: channel ch2: specifying datafile(s) to restore from backup set
RMAN-08523: restoring datafile 00003 to C:\ORACLE\ORADATA\ORCL2\USERS01.DBF
RMAN-08023: channel ch2: restored backup piece 1
RMAN-08511: piece handle=C:\ORACLE\ORA81\DATABASE\07DSOBN6_1_1 tag=null params=NULL
RMAN-08024: channel ch2: restore complete
RMAN-03022: compiling command: recover
RMAN-03022: compiling command: recover(1)
RMAN-03022: compiling command: recover(2)
RMAN-03022: compiling command: recover(3)
RMAN-03023: executing command: recover(3)
RMAN-08054: starting media recovery
RMAN-08055: media recovery complete
RMAN-03022: compiling command: recover(4)
RMAN-08031: released channel: ch2

Alter the database open to verify that the restore and recovery was successful.

SQL> alter database open;
Database altered.

1. Check the status of the USERS tablespace.

SQL> select tablespace_name, status from dba_tablespaces;

<table>
<thead>
<tr>
<th>TABLESPACE_NAME</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTEM</td>
<td>ONLINE</td>
</tr>
<tr>
<td>RBS</td>
<td>ONLINE</td>
</tr>
<tr>
<td>USERS</td>
<td>ONLINE</td>
</tr>
<tr>
<td>TEMP</td>
<td>ONLINE</td>
</tr>
<tr>
<td>TOOLS</td>
<td>ONLINE</td>
</tr>
<tr>
<td>INDX</td>
<td>ONLINE</td>
</tr>
<tr>
<td>DRSYS</td>
<td>ONLINE</td>
</tr>
<tr>
<td>RCVCAT</td>
<td>ONLINE</td>
</tr>
</tbody>
</table>

8 rows selected.

Changing the status of the database can be done through SQL*Plus or from within RMAN.
Exercise 13-4

1. Connect to RMAN with the NOCATALOG option

2. Place the database into a MOUNT state. If the status of the database is OPEN, issue a shutdown and then mount the database.

3. Using RMAN recover and restore the tablespace users.

4. Alter the database open to verify that the restore and recovery was successful.

5. Check the status of the USERS tablespace.
Recover an Inaccessible Tablespace

This scenario allows for recovery of a tablespace that is no longer accessible while the database is open.

An inaccessible tablespace could occur if one of the disks of the underlying datafiles fails.

SET NEWNAME is used if the datafile cannot be restored to the default location.

- If SET NEWNAME is used then the SWITCH command must be used to change the location of the datafile in the target databases control file. The SWITCH command is equivalent to the ALTER DATABASE RENAME DATAFILE statement that can be manually issued during a user managed recovery.

Connect to the target database with the NOCATALOG option.

C:\>rman target /@orcl2 nocatalog

Recovery Manager: Release 9.2.0.1.0 - Production

RMAN-06005: connected to target database: ORCL2
(DBID=446765398)
RMAN-06009: using target database controlfile instead of recovery catalog

Use SQL Plus verify the database is open.

SQL> select status from v$thread;

STATUS
-----
OPEN
Recover an Inaccessible Tablespace

Create a run block that accomplishes the following:

- Offline the USERS tablespace
- Changes the name of the underlying USERS datafile
- Restores the USERS tablespace
- Updates the control file with the datafiles new location
- Recover the USERS tablespace
- Online the USERS tablespace

RMAN> run {
2> SQL 'ALTER TABLESPACE users OFFLINE IMMEDIATE';
3> allocate channel ch1 type disk;
4> set newname for datafile
   'C:\oracle\oradata\orcl2\users01.dbf' to 'C:\oracle\oradata\orcl2\rman\users01.dbf';
5> restore tablespace users;
6> switch datafile all;
7> recover tablespace users;
8> SQL 'ALTER TABLESPACE users ONLINE';
9> }

RMAN-03022: compiling command: sql
RMAN-06162: sql statement: ALTER TABLESPACE users OFFLINE IMMEDIATE
RMAN-03023: executing command: sql
RMAN-03022: compiling command: allocate
RMAN-03023: executing command: allocate
...
RMAN-08089: channel ch1: specifying datafile(s) to restore from backup set
RMAN-08523: restoring datafile 00003 to
C:\ORACLE\ORADATA\ORCL2\RMAN\USERS01.DBF
RMAN-08023: channel ch1: restored backup piece 1
RMAN-08511: piece
handle=C:\ORACLE\ORA81\DATABASE\07DSONB6_1_1 tag=null
params=N
ULL
RMAN-08024: channel ch1: restore complete
Recover an Inaccessible Tablespace

Output continued from previous page.

RMAN-03022: compiling command: switch
RMAN-03023: executing command: switch
RMAN-08015: datafile 3 switched to datafile copy
RMAN-08507: input datafilecopy recid=26 stamp=466383974
filename=C:\ORACLE\ORADATA\ORCL2\RMAN\USERS01.DBF
RMAN-03022: compiling command: recover
RMAN-03022: compiling command: recover(1)
RMAN-03022: compiling command: recover(2)
RMAN-03022: compiling command: recover(3)
RMAN-03023: executing command: recover(3)
RMAN-08054: starting media recovery
RMAN-08055: media recovery complete
RMAN-03022: compiling command: recover(4)
RMAN-03022: compiling command: sql
RMAN-06162: sql statement: ALTER TABLESPACE users ONLINE
RMAN-03023: executing command: sql
RMAN-08031: released channel: ch1

Notes

It is important that the tablespaces is altered ONLINE to as the last step of the recovery. The altering of the tablespace is not done automatically by RMAN. If this is not done manually or in the RUN block, the data in the tablespace will not be accessible.
Review Questions

1. When restoring a tablespaces datafiles to a new location using RMAN, which command is required?
   A. RECOVER DATABASE
   B. ALTER DATABASE RENAME DATAFILE
   C. SET NEWNAME
   D. RECOVER FILESPERSET

2. When restoring a tablespace all datafiles associated with the tablespace will be restored.
   A. True
   B. False

3. V$DATAFILE_HEADER will report all of the following except:
   A. Error
   B. Recover
   C. Status
   D. DB_NAME

4. Stored scripts are saved in the target databases control file.
   A. True
   B. False

5. A consistent backup of the database can be made if SHUTDOWN ABORT was used the last time the database was closed.
   A. True
   B. False
Summary

Creating and storing a script within the RMAN catalog
Performing a backup of a datafile
Backing up a tablespace with NOCATALOG
Which views are used to determine if recovery is needed
Recovering a tablespace that is accessible
Performing a recovery of a tablespace that is inaccessible
Exercises 13-5

1. Connect to the target database with the NOCATALOG option.
2. Use SQL Plus verify the database is open.
3. Offline the USERS tablespace.
4. Change the name of the underlying USERS datafile.
5. Restore the USERS tablespace.
6. Update the control file with the datafiles new location.
7. Recover the USERS tablespace
8. Online the USERS tablespace
Chapter Overview

Oracle databases are supporting larger and larger systems. DBAs need to determine the architecture that will support user scalability. The two options are the dedicated and shared server architectures. This chapter will cover the shared server architecture and do a compare and contrast with the dedicated server architecture.

At the end of this chapter, students will understand:

- The role of the server process in the Oracle architecture
- Differences between a dedicated and a shared server architecture
- Features of the shared server architecture
- Configuring the shared server architecture
The Shared Server Architecture

In a dedicated server environment, each user process has a dedicated server process to handle all of their activity.

- Server processes are often idling a large percentage of time in a dedicated server environment.
- Each server process requires operating system memory, CPU and additional resources to run. This is not scalable to large environments.
- A dedicated server environment is good for large data intensive work. Large batch users should remain in a dedicated server environment.
- The User Global Area (UGA) is stored in the Program Global Area of dedicated server processes.

In a shared server environment, user processes share server processes.

- The objective of the shared server environment is to be able to scale to a large number of concurrent users.

  Server processes are more efficient since they can support multiple user processes. For example, 100 shared server processes may be able to support 1000 user processes. A dedicated server environment would require 1000 server processes to support 1000 user processes.

  This allows an shared server environment to use less server processes, therefore requiring less system resources. This is important to Internet databases.

- The User Global Area (UGA) is stored in the shared pool area of the System Global Area (SGA). The UGA stores session data for user and cursor information.

- The large pool area can be configured to hold the UGA for the shared server.

Notes

The shared server architecture supports connect manager and connection pooling features in Oracle Net Services. The TCP/IP protocol must be used for dispatchers when setting up MTS on Windows NT/2000/XP.
Server Processes Review

The server process runs on the same machine as the Oracle instance. When a record is inserted, modified, or deleted, the server process will copy the before-image data to the undo segment. The server process environment has been configured three different ways:

- A two-task (dedicated server) environment executes a server process for every user process. The user and server processes usually run on separate platforms.
- A shared server (multithreaded server) runs shared server processes. Each server process can support multiple user processes.
- A single-task environment combines the user and server process into a single application to conserve resources. The user and server process must run on the same platform.

*Users processes are applications requesting services from the instance. Examples include:*  

- Oracle Forms, SQL*Plus, Server Manager (obsolete in Oracle 9i), Enterprise Manager, C Programs, etc.

Notes

A server process starts up to handle and manage requests from user processes. Server processes:

- Parse and execute SQL commands for the user process.
- Get the data the user process has requested:
  - The server process will check to see if the data already exists in the data block buffer cache
  - If the data is not in the data block buffer cache, the server process will read the data from disk and load it into the data buffer cache.
- Contain user session data: profile limits, roles, optimizer goals, PL/SQL bindings, and NLS settings.
- Return the data back to the user process.
Program Global Areas

The program global area (PGA) is a section of non-shared memory allocated for a server or background process. This section focuses on the PGA in the server process.

- This memory is allocated for a server process when a user connects to an instance.

Where the information is stored depends on whether a dedicated or shared server environment is running.

The PGA contains stack space to hold variables, arrays, and additional program information for the session. This memory is fixed in size and depends on the operating system. Information stored in the PGA includes:

- Cursor state information.
- Stack space.

Each server process has a PGA. Only the server process can write to the PGA.

Notes

The PGA is also referred to as the process global area.

Init.ora parameters OPEN_LINKS, LOG_FILES, and DB_FILES have an impact on the PGA memory size.
User Global Area

The program global area has a User Global Area (UGA).

The UGA stores session specific information and the cursor state. The SQL work memory areas are defined with the session information.

A dedicated server configuration stores the UGA in the dedicated server process memory.

A shared server (MTS) configuration stores the UGA in the system global area.

In the shared pool area or in the large pool area. If the large pool is configured, the UGA is stored in the large pool.

Notes

The next paragraph contains a little more detailed information than you may care for at this time, but we wanted to see if you were occasionally looking at the notes. The next paragraph is a good dinner party topic, as a way of breaking the ice, if you don’t know a lot of people there.

The UGA has two components a Fixed UGA and a variable UGA. The fixed UGA stores information about open cursors, roles, events, database links, etc. The Variable UGA contains the private SQL and PL/SQL areas.

If you’re still reading these notes, and someone at the party is actually able to answer your question, you may need an additional statement. Tell them, “It is important to limit the amount of SGA memory for each UGA in a multi-threaded server (MTS).”

If they answer, “Isn’t it funny how some DBAs don’t know that they should do this with the profile resource limit PRIVATE_SGA,” then tell them, “Oh, I see they have some new refreshments!” Say, “Good bye,” and walk away quickly. You can take our Oracle9i Server Performance tuning class and then look them up at the next party.
Dedicated Server Architecture (Review)

With the dedicated server architecture, Oracle will start up a server process to handle communication for each user process.

- If there are 100 users connected to the system there will be 100 server processes running on the platform where the Oracle instance is running.
- This is excellent for an individual user process because they have a dedicated server process to handle all database requests.
- Often user processes sit idle. While the user process sits idle, the server process is still taking up resources on the platform where the Oracle instance is running.
- Numerous studies have shown that over 70% of the time dedicated server processes sit idle.
- This architecture is not scalable for a large number of users. It is also not good for an Internet database than can have tremendous spikes in users.
- This architecture wastes system resources when user processes sit idle.
What is the Shared Server Architecture?

The shared server architecture allows multiple user processes to share a server process.

- This eliminates the overhead of a dedicated server environment which requires a “dedicated” server process for each user process.

Individual dedicated server processes can still connect to the Oracle instance in a shared server environment.

- System processes like starting up and shutting down an instance require a dedicated server process.
- Large batch processes run better in a dedicated server environment.

The shared server environment requires:

- Oracle net services (or SQL*NET V2).
- A listener process is required for connecting shared server processes.
- One or more dispatcher processes.
- One or more shared server processes.
The multithreaded server (MTS) has been renamed to the Shared Server Architecture in Oracle9i.

New features in the Oracle9i Shared Server include:

- MTS parameters have been deprecated.
- New V$SHARED_SERVER_MONITOR data dictionary view.
- Redesign of connection protocols
- Enhancements to the dispatcher.
- Oracle9i OEM Performance Manager can help monitor listeners, dispatchers and shared server processes.

The shared server environment can now support the DBMS_LDAP package. This was a restriction in Oracle8i.

Notes

Oracle Network services is used to setup networking. Oracle networking is a large subject and cannot be covered in the Database Administration course. Oracle Network services is covered in the Oracle9i Networking course.

Historically, tnsnames has been used to setup Oracle networking in most configurations. Tnsnames uses the listener.ora and tnsnames.ora configuration files.

The 9i Application Server Release 2 will start becoming a more popular method of configuring Oracle networking in Internet environments. The Oracle Internet Directory (OID) supports centralized networking instead of having to put the network configuration files on each client (tnsnames method).

Oracle names is no longer supported in Oracle9i. However there is an Oracle Names proxy that can be setup using LDAP to make it easier for Oracle Names administrators to make the transition to the Oracle Internet Directory.
The Shared Server Architecture

The shared server architecture is scalable to support a large number of concurrent users.
Dispatcher Processes

At least one dispatcher process must be defined and started.

- At least one separate dispatcher process is required for each network protocol that is configured.

The listener process handles any user processes requesting a connection.

- The listener process determines if a user process can use a shared server process.
- If the user process is allowed to connect to a shared server process, the listener process will provide the address of a dispatcher process.
- The listener will pick a dispatcher with the lightest load.
- The user process will then connect directly to the dispatcher. A user process will keep the same dispatcher throughout the session.
- User processes connect through net services in a shared server environment.
The Listener Process

A listener must be initiated before the instance starts.

- Dispatchers need to register with the listener during instance startup.
- The listener control command SERVICES can be used to determine if the dispatcher is registered with the listener.

The listener alias in the tnsnames.ora file must contain an appropriate definition. This definition specifies the listener address defined in the initialization file.

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

Verify a dispatcher is registered with a listener when the instance is started.

```
D:> lsnrctl services
```

Verify the instance is properly using shared servers.

- Start up the instance.
- Connect using a single session.
- Query the data dictionary view V$CIRCUIT and there should be an entry for a shared server connection.
Dispatcher Syntax

The dispatcher clause is an ASCII string.

```
DISPATCHERS = 'dispatcher_clause'
```

The dispatcher clause can define a protocol, address or description.

```
(PROTOCOL = protocol) | (ADDRESS = address) | (DESCRIPTION = description )
```

A number of dispatcher options are available.

```
(DISPATCHERS = integer | SESSIONS = integer | CONNECTIONS = integer | TICKS = seconds |
POOL = {1 | ON | YES | TRUE | BOTH |
{(IN | OUT} = ticks) | 0 | OFF | NO | FALSE | ticks} |
MULTIPLEX = {1 | ON | YES | TRUE | 0 | OFF | NO | FALSE | BOTH | IN | OUT} |
LISTENER = tnsname |
SERVICE = service |
INDEX = integer)
```

Dispatcher clause definitions:

- **PROTOCOL** (PRO or PROT) – Defines the network protocol.
- **ADDRESS** (ADD or ADDR) - Defines the network protocol address.
- **DESCRIPTION** (DES or DESC) - Defines the network description and the protocol address.
Dispatcher Option Definitions

Dispatcher options clause definitions.

- **DISPATCHERS (DIS or DISP)** - Initial number of dispatchers to start.
- **SESSIONS (SES or SESS)** – Sets the maximum number of network sessions for each dispatcher.
- **CONNECTIONS (CON or CONN)** – Defines the maximum number of network connections for each dispatcher.
- **TICKS (TIC or TICK)** – The number of timeout seconds for a network tick.
- **POOL (POO)** - Allows connection pooling. The integer defines the number of seconds for a network tick.
- **ON, YES, TRUE, and BOTH** defines Connection Pooling for both incoming and outgoing network connections.
- **IN** defines Connection Pooling for incoming network connections.
- **OUT** defines Connection Pooling for outgoing network.
- **NO, OFF, and FALSE** disables Connection Pooling for both incoming and outgoing network connections (default).
- **MULTIPLEX (MUL or MULT)** – Defines the Oracle Connection Manager for multiplexing.
- **1, ON, YES, TRUE, and BOTH** defines multiplexing for incoming and outgoing network connections.
- **IN** defines multiplexing for incoming network connections.
- **OUT** defines multiplexing for outgoing network connections.
- **0, NO, OFF, and FALSE** disables multiplexing for both incoming and outgoing network connections (default).
- **LISTENER (LIS, LIST)** – Defines the network name of an address. An address can be defined for Oracle Net listeners the dispatchers will register with. The LISTENER definition takes precedence over the LOCAL_LISTENER and REMOTE_LISTENER parameters.
- **SERVICE (SER, SERV)** - Defines the service names the dispatchers register with the listeners.
- **INDEX** – Use with the ALTER SYSTEM SET DISPATCHERS statement to specify which dispatcher is being defined.
Setting up the Shared Server

Oracle Net Services must be configured for the shared server.

- User processes must connect using Oracle Net Services (or Net8) even if they are on the same host as the Oracle instance.

The following files must be configured to run the shared server environment:

- Listener.ora.
- Tnsnames.ora.

**LOCAL_LISTENER**

- This parameter defines the service name for a listener that dispatchers use to register their services.
  - The dispatchers and instance register their information with the listener. The listener then uses this information to connect user processes with dispatchers and with dedicated server processes.
  - If the LOCAL_LISTENER parameter is not defined, the default is to use TCP/IP port 1521 and the ORACLE_SID value.

Setting LOCAL_LISTENER initialization parameter:

```
LOCAL_LISTENER = mylist1
```

- TNSNAMES.ORA entry:

```
mylist1 =
(DESCRIPTION =
  (ADDRESS =
     (PROTOCOL = TCP)
     (HOST = myhost)
     (PORT = 1720)
  )
  (CONNECT_DATA=(SID=ORC9)))
```
Configuring the Shared Server Architecture

Multiple shared server processes can be defined for an instance.

- The initial number of shared server processes to start when an instance is brought up can be defined (SHARED_SERVERS). The SHARED_SERVERS parameter is dynamic.
- The maximum number of shared server processes can also be defined (MAX_SHARED_SERVERS). The MAX_SHARED_SERVERS parameter is static.

Multiple dispatchers can be defined for an instance.

- The initial number of dispatchers to start when an instance is brought up can be defined (DISPATCHERS). The DISPATCHERS parameter is dynamic.
- The maximum number of dispatchers can also be defined (MAX_DISPATCHERS). The MAX_DISPATCHERS parameter is static.

The maximum number of circuits is defined with the CIRCUITS static parameter.

The SHARED_SERVER_SESSIONS dynamic parameter defines the maximum number of shared server user sessions that can be started. This needs to be set to reserve sessions for dedicated servers.

The PGA aggregate target area cannot be used with the shared server architecture. With the shared server architecture the SQL work areas must be defined manually.
Configuring the Shared Server Architecture (2)

Be default the User Global Area (user session information) is stored in the shared pool area. Set the LARGE_POOL_SIZE parameter to have user session information stored in the large pool memory area. This will reduce contention for the shared pool area.

This architecture allows a small number of shared server processes to handle a much larger number of user processes. The shared server architecture offers user scalability.

If a user process requests a dedicated server process, the listener will start a dedicated process to establish a dedicated server connection.

Multiplexing uses Connection Manager (CMAN) to maximize throughput.

▶ CMAN establishes a shared connection to the dispatcher. CMAN then uses the shared connection for multiple users.
Connection Pooling

Connection pooling can make sure all the resources are fully utilized in a system with a large number of users. Connection pooling will maximize the number of physical connections to a shared server.

- If an existing session is idle another session can use the connection. The idle session can reconnect when another session becomes idle.
- This configuration is advantageous when sessions can have high idle times.
- Connection pooling shares dispatcher connections for multiple user processes.
- Multiple connections can be set for a dispatcher in an MTS configuration.
- Once the maximum number of connections for a dispatcher is reached, a user process will search for an idle connection.
- An idle connection will become temporarily disconnected.
- A disconnected session will search for an idle connection so it can continue its processing.
- Connection pooling is ideal for applications that are highly interactive and contain high idle times.
Setting up Connection Pooling

Connection pooling (CP) is set up with the DISPATCHERS parameter.

Connection pooling attributes:

<table>
<thead>
<tr>
<th>View</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>POOL (POO)</td>
<td>Enables connection pooling by setting the value to “ON”, “YES”, “TRUE” or “BOTH”. These values set CP for incoming and outgoing connections. “IN” sets CP for incoming connections. “OUT” sets CP for outgoing connections. “NO”, “OFF”, “FALSE” disable CP for incoming and outgoing connections. If a timeout number is set, CP is enabled for incoming and outgoing network connections. Examples of setting a timeout: (IN=20), (OUT=10), ((IN=20)(OUT=10)). If no timeout number is set the default is used. If the timeout number is set to 0 then the default will be used. CP is disabled by default.</td>
</tr>
<tr>
<td>CONNECTION (CON or CONN)</td>
<td>Defines the maximum number of connections per dispatcher. SESSIONS defines the maximum number of user processes that can work with a dispatcher. CONNECTION defines the maximum number of connections that can be established through a dispatcher. The number of SESSIONS – CONNECTIONS defines the number of sessions that can be waiting to process a request.</td>
</tr>
<tr>
<td>TICKS (TIC or TICK)</td>
<td>Defines the number of 10-second ticks. If TICKS is not set, the default of waiting for a connection is usually 100 seconds or 10 ticks. Most platforms have a tick set to 10 seconds.</td>
</tr>
</tbody>
</table>

Example:

```
DISPATCHERS="(PRO=TCP)(CONN=15)(DIS=2)(POOL=ON)(TICK=2)(SESS=30)"
```
Dispatcher Definitions

The DISPATCHERS initialization parameter is used to define dispatcher configurations such as:

- Number of dispatchers.
- Connection pooling.
- Multiplexing.
- Host and port addresses.

Syntax: Use single or double quotes.

```
DISPATCHERS = "(parameter=value)"
```

Example: Define the host system the dispatchers will listen on. Oracle Net Services will dynamically set the port if one is not specified.

```
DISPATCHERS = "(ADDRESS=(PROTOCOL=TCP)(HOST=200.39.23.12))(DISPATCHERS=3)"
```

Example: Define the host system and the port the dispatchers will listen on.

```
DISPATCHERS = "(ADDRESS=(PROTOCOL=TCP)(HOST=200.39.23.12)(PORT=5005))(DISPATCHERS=3)"
```

Example: Set connection pooling.

```
DISPATCHERS = "(PROTOCOL=TCP)(POOL=ON) (TICK=2)"
```

Example: Set multiplexing.

```
DISPATCHERS = "(PROTOCOL=TCP)(MULTIPLEX=ON)"
```
Client Configurations

A client can be defined to use a specific dispatcher.

```
ORCL =
   (DESCRIPTION =
      (ADDRESS_LIST =
         (ADDRESS = (PROTOCOL = TCP)(HOST = Glaptop)(PORT = 1521)) )
   (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl) )

INST1_HTTP =
   (DESCRIPTION =
      (ADDRESS_LIST =
         (ADDRESS = (PROTOCOL = TCP)(HOST = Glaptop)(PORT = 1521)) )
   (CONNECT_DATA =
      (SERVER = SHARED)
      (SERVICE_NAME = MODOSE)
      (PRESENTATION = http://HRService) )
```

The sqlnet.ora can be used to configure the dispatcher to use. The parameter USE_DEDICATED_SERVER=ON will override the connect descriptor SERVER = DEDICATED in the CONNECT_DATA section.
Processing Requests and Results

Each client is connected to the instance with a virtual circuit. A virtual circuit is a section of memory used by the dispatcher to handle requests and responses by a user process.

- The virtual circuits maintain permanent connections.

**Processing requests:**

- Virtual circuits are stored in a shared memory area called the request queue.
- There is a single request queue for all dispatchers. All dispatchers share this common request queue.
- The request queue resides in the System Global Area.
- Shared server processes monitor the request queue to see if there are additional requests to process.

**Processing results:**

- A shared server process will handle multiple virtual circuits in the request queue on a first-in-first-out (FIFO) basis.
- The shared server process will place the results in a dispatcher response queue.
- Dispatcher response queues are stored in the System Global Area.
- Each dispatcher has its own response queue.
- The dispatcher will process the results and return them to the user process.
Deprecated Parameters

The MTS parameters have been deprecated in Oracle9i.

<table>
<thead>
<tr>
<th>Oracle8i MTS Parameters</th>
<th>Oracle9i Shared Server Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>mts_circuits</td>
<td>circuits</td>
</tr>
<tr>
<td>mts_dispatchers</td>
<td>dispatchers</td>
</tr>
<tr>
<td>mts_max_dispatchers</td>
<td>max_dispatchers</td>
</tr>
<tr>
<td>mts_servers</td>
<td>shared_servers</td>
</tr>
<tr>
<td>mts_max_servers</td>
<td>max_shared_servers</td>
</tr>
<tr>
<td>mts_sessions</td>
<td>shared_server_sessions</td>
</tr>
<tr>
<td>mts_listener_address</td>
<td>shared_memory_address</td>
</tr>
</tbody>
</table>

The data dictionary view V$SHARED_SERVER_MONITOR is used to monitor the shared server.

<table>
<thead>
<tr>
<th>Columns</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAXIMUM_CONNECTIONS</td>
<td>The largest number of virtual circuits that have been used at a point in time.</td>
</tr>
<tr>
<td>MAXIMUM_SESSIONS</td>
<td>The largest number of shared server sessions that have been used at a point in time. Increase this value when it gets close to the setting for SHARED_SERVER_SESSIONS.</td>
</tr>
<tr>
<td>SERVERS_STARTED</td>
<td>The largest number of shared server processes that have been used at a point in time.</td>
</tr>
<tr>
<td>SERVERS_TERMINATED</td>
<td>The total number of shared servers stopped since the instance started.</td>
</tr>
<tr>
<td>SERVERS_HIGHWATER</td>
<td>The largest number of shared server sessions that have been used at a point in time.</td>
</tr>
</tbody>
</table>
Shared Server Parameters

Important parameters to define for a shared server configuration.

- **DISPATCHERS** - The number of dispatcher processes created at startup.
- **MAX_DISPATCHERS** - The maximum number of dispatcher processes.
- **SHARED_SERVERS** - The number of shared server processes created at startup.
- **MAX_SHARED_SERVERS** - The maximum number of shared server processes.
- **CIRCUITS** - The number of virtual circuits for network sessions.
- **SHARED_SERVER__SESSIONS** - Defines the maximum number of shared server user sessions. This allows user sessions to be reserved for dedicated servers.
- **LARGE_POOL_SIZE** - Defines the size of the large pool heap area.
- **SESSIONS** - Defines the total number of sessions for an instance.

**Shared Server Parameters:**

<table>
<thead>
<tr>
<th>Required</th>
<th>Optional</th>
<th>Additional</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISPATCHERS</td>
<td>MAX_DISPATCHERS, SHARED_SERVERS</td>
<td>SESSIONS, LARGE_POOL_SIZE</td>
</tr>
<tr>
<td></td>
<td>MAX_SHARED_SERVERS, CIRCUITS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SHARED_SERVER__SESSIONS</td>
<td></td>
</tr>
</tbody>
</table>

**Notes**
Oracle will use default values for optional parameters if they are not defined.
Data Dictionary Views

The following data dictionary views contain information on the shared server architecture.

<table>
<thead>
<tr>
<th>View</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>V$CIRCUIT</td>
<td>Displays information on virtual circuits.</td>
</tr>
<tr>
<td>V$DISPATCHER</td>
<td>Displays information on dispatcher processes.</td>
</tr>
<tr>
<td>V$DISPATCHER_RATE</td>
<td>Current, average and maximum statistics.</td>
</tr>
<tr>
<td>V$QUEUE</td>
<td>Process, wait and total time for message queues.</td>
</tr>
<tr>
<td>V$SGA</td>
<td>SGA size information.</td>
</tr>
<tr>
<td>V$SGASTAT</td>
<td>SGA information on shared, large and Java pools.</td>
</tr>
<tr>
<td>V$SHARED_POOL_RESERVED</td>
<td>Statistics for reserved pool.</td>
</tr>
<tr>
<td>V$SHARED_SERVER</td>
<td>Information on shared server processes.</td>
</tr>
<tr>
<td>V$SHARED_SERVER_MONITOR</td>
<td>Displays usage information on sessions.</td>
</tr>
</tbody>
</table>
Review Questions

1. What is a virtual circuit?
   A. A pulse set through the network to verify the shared server is up and running.
   B. A section of memory used by the dispatcher to handle requests and responses by a user process.
   C. A simulated connect to test troubleshooting.
   D. A method of simulating a connection for benchmarking.

2. Which is NOT a benefit of running the shared server?
   A. Reduces system memory and resources for running the instance.
   B. Cuts down on the number of server processes running for an instance.
   C. Performs load balancing across hardware servers.
   D. Cuts down on idle server processes.

3. Where are two places the user session data can be stored for a shared server? (Choose two)
   A. PGA
   B. Java pool area
   C. Large pool area
   D. Database buffer cache
   E. Shared pool area

4. Which statement is true about connection pooling?
   A. Connection pooling starts up multiple request queues to maximize dispatcher throughput.
   B. Connection pooling allows network connections to be shared to a dispatcher.
   C. Connection pooling starts up a pool of dispatchers to maximize client connections.
   D. Connection pooling starts up a pool of server processes.

5. What is the purpose of the LOCAL_LISTENER parameter?
   A. Defines the name of the local listener.
   B. Defines a priority listener name in case of failover.
   C. Specifies the local listener should have priority over the remote listener.
   D. Defines a service name for listeners so dispatchers can register their services.
Summary

In this chapter we covered:

- Reviewing the PGA and UGA areas.
- The shared server architecture.
- How to configure a shared server platform.
- Compared the differences between a dedicated and shared server architecture.
- Outlined old MTS parameters.
Exercises

These exercises will focus on configuring the Shared Server platform.

1. Setup these parameters in your init.ora or SPFILE file.

   | LOCAL_LISTENER=temp |
   | DISPATCHERS="(PROTOCOL=TCP) (DISPATCHERS=2)"
   | SHARED_SERVERS=2 |
   | MAX_DISPATCHERS=2 |
   | MAX_SHARED_SERVERS=4 |
   | CIRCUITS=10 |
   | SHARED_SERVER_SESSIONS=2 |

2. Setup a listener in your listener.ora file with this configuration

   TEMP =
   (DESCRIPTION_LIST =
     (DESCRIPTION =
       (ADDRESS_LIST =
         (ADDRESS = (PROTOCOL = TCP)(HOST = YourMachine)(PORT = 1524))
       )
     )
   )

   SID_LIST_TEMP =
   (SID_LIST =
     (SID_DESC =
       (SID_NAME = YourORACLE_SID)
       (ORACLE_HOME = YourORACLEHOME)
       (GLOBAL_DBNAME=YourORACLESID.DOMAIN.COM)
     )
   )

3. Shutdown and Startup your Oracle Instance, with the new initialization parameters.
4. Stop and start your new listener temp.

5. Check the configuration of your listener

```bash
LSNRCTL> services temp
```

Observe the configuration regarding the dispatchers. Should look something similar to the below section:

```
Instance "TEMP", status READY, has 3 handler(s) for this service...
  Handler(s):
    "DEDICATED" established:0 refused:0 state:ready
    LOCAL SERVER
    "D001" established:0 refused:0 current:0 max:972 state:ready
    DISPATCHER <machine: YourMachine, pid: 14948>
      (ADDRESS=(PROTOCOL=tcp)(HOST=YourMachine)(PORT=33739))
    "D000" established:0 refused:0 current:0 max:972 state:ready
    DISPATCHER <machine: pblqrdb01, pid: 14946>
      (ADDRESS=(PROTOCOL=tcp)(HOST=YourMachine)(PORT=33737))
```

6. Add the following section to your tnsnames.ora file

```sql
TEMP =
  (DESCRIPTION=
    (ADDRESS=(PROTOCOL=TCP) (HOST=YourMachine) (PORT=1524) )
    (CONNECT_DATA =
      (SERVICE_NAME=YourServiceName)
    )
  )
```

©2004 ITCourseware, LLC

The Shared Server Architecture 18-29
7. From another SQLPLUS window, make a connection to your database

```
C:\> sqlplus system/password@temp;
```

Leave this window with sqlplus connected and open one more window and connect to the database in a similar fashion.

Now make sure that these connections were established using the dispatcher to validate the shared server configuration.

```
SQL> select dispatcher,circuit,server,status from v$circuit
```

Observe the output, should be something similar to:

<table>
<thead>
<tr>
<th>DISPATCHER</th>
<th>CIRCUIT</th>
<th>SERVER</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>000000039242F410</td>
<td>0000000390214268</td>
<td>00</td>
<td>NORMAL</td>
</tr>
<tr>
<td>000000039242F890</td>
<td>0000000390214460</td>
<td>00</td>
<td>NORMAL</td>
</tr>
</tbody>
</table>

This validates that the shared servers and virtual circuits are working.

8. With the 2 SQLPLUS windows still open, trace the relationship between the client process and the dispatcher. This is a very important task in an enterprise environment, when troubleshooting performance issues.

```
SQL> SELECT d.name Dispatcher,
             substr(s.username,1,10) UserN,
             s.sid Osid, substr(p.username,1,10) os_user,
             substr(s.program,1,20) Appl
       FROM  v$dispatcher d, v$circuit c,
             v$session s,  v$process p
       WHERE d.paddr = c.dispatcher (+)
            AND c.saddr = s.saddr (+)
            AND s.paddr = p.addr (+)
       ORDER BY d.name, s.username;
```

You should see an output similar to this:

<table>
<thead>
<tr>
<th>DISP</th>
<th>USERN</th>
<th>OSID</th>
<th>OS_USER</th>
<th>APPL</th>
</tr>
</thead>
<tbody>
<tr>
<td>D000</td>
<td>SYSTEM</td>
<td>9</td>
<td>oracle</td>
<td>sqlplus@pblqrdb01 (T</td>
</tr>
<tr>
<td>D001</td>
<td>SYSTEM</td>
<td>11</td>
<td>oracle</td>
<td>sqlplus@pblqrdb01 (T</td>
</tr>
</tbody>
</table>
9. In a Shared Server environment if the number of dispatchers and shared servers are not configured appropriate to the processing demand it could result in a overall poor system performance.

To determine the average response time for a dispatcher, you can run this query and determine if you need to increase the number of dispatchers.

```
SQL> SELECT d.name Dispatcher,
       DECODE (SUM (totalq), 0, 'No Responses',
               ((SUM (wait) / SUM (totalq)),3)) MilliSeconds
       FROM v$queue q, v$dispatcher d
       WHERE q.type = 'DISPATCHER'
       AND q.paddr = d.paddr
       GROUP BY name;
```

Output in Milliseconds would be something like this, which would potentially increase with the server load.

```
DISP  MILLISECONDS
----  -------------------
D000  .008
D001  0
```

To appropriately determine the number of dispatchers for your system, use this formula:

Max Dispatchers = Maximum Number of concurrent sessions

__________________________________
Connections per Dispatcher

What is the maximum number of connections that a dispatcher can serve?

It's an operating system specific and can be found by using the services command in the lsnrctl utility for that listener.

Determine the load on the shared servers.

```
SQL> SELECT name "NAME",
       paddr,requests,
       round(busy/(busy + idle) * 100,3) "%TIME BUSY",status
       FROM v$shared_server;
```
Output should be something like this:

```
<table>
<thead>
<tr>
<th>NAME</th>
<th>PADDR</th>
<th>REQUESTS</th>
<th>%TIME BUSY</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>----</td>
<td>-------------------</td>
<td>----------</td>
<td>------------</td>
<td>----------</td>
</tr>
<tr>
<td>S000</td>
<td>000000039242EB10</td>
<td>475</td>
<td>.042</td>
<td>EXEC</td>
</tr>
<tr>
<td>S001</td>
<td>000000039242EF90</td>
<td>0</td>
<td>0</td>
<td>WAIT (COMMON)</td>
</tr>
</tbody>
</table>
```

10. With the 2 windows of sqlplus still open try to start another sqlplus connection to the database using your listener.

```
Sqlplus system/password@temp
```

What happened? Why?

Hint: What is your max_dispatchers set to?

11. In a shared server architecture the user process information is stored in the User Global Area as opposed to the Process Global Area. UGA being a part of shared pool, its very important to keep a track of memory usage per session to appropriately size the SGA. You can determine the UGA needs per session, at any given time and monitor the maximum memory ever needed by the process.

```
SQL> SELECT s.sid, substr(n.name, 1, 25) name, 
       round((value/1024), 2) sess_mem_in_KB 
FROM v$sesstat s, v$statname n, v$session ses 
WHERE s.statistic# = n.statistic# 
  AND n.name in ('session uga memory', 
                 'session uga memory max') 
  AND s.sid = ses.sid 
  AND ses.username is not null 
ORDER BY 1;
```

You should get an output similar to this:

```
<table>
<thead>
<tr>
<th>SID</th>
<th>NAME</th>
<th>SESS_MEM_IN_KB</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>session uga memory</td>
<td>203.73</td>
</tr>
<tr>
<td>8</td>
<td>session uga memory max</td>
<td>267.76</td>
</tr>
</tbody>
</table>
```
12. Connecting to a Shared Server with a dedicated connection

Insert this section into your tnsnames.ora file

```sql
TEMPE_DED =
    (DESCRIPTION=
        (ADDRESS=(PROTOCOL=TCP) (HOST=YourMachine) (PORT=1524) )
    (CONNECT_DATA =
        (SERVICE_NAME=YourServiceName)
        (SERVER=DEDICATED)
    )
)
```

Connect to the database using the service name TEMP_DED. Yea!

```
C:\> sqlplus system/password@temp_ded
```

From another window connect to the lsnrctl utility and check the listener.

```
LSNRCTL> status temp
```

Look for something like

```
"DEDICATED" established:1 refused:0 state:ready
```