

# Strategies for Oracle Database Backup and Recovery: Case Studies

Mingguang Xu

Office of Institutional Research  
University of Georgia  
[www.oir.uga.edu/oirpres.html](http://www.oir.uga.edu/oirpres.html)

# Oracle Files

Oracle requires the following files for its operation

1. Datafiles
2. Redo log files: online and archived redo logs
3. Control file: holds the information of the physical structure of the database, current database state (SCN) and the backup taken by RMAN
4. Server parameter file: holds the initiation parameters
5. Password file: holds information of DBA users
6. Network files: tnsnames.ora, listener.ora

# Oracle Instance

- An instance is made up of Oracle processes and associated memory structure.
- Created at database startup.
- An instance can be started up in various modes:
  1. Startup nomount: read parameter file
  2. Startup mount: read control files
  3. Startup: open data files

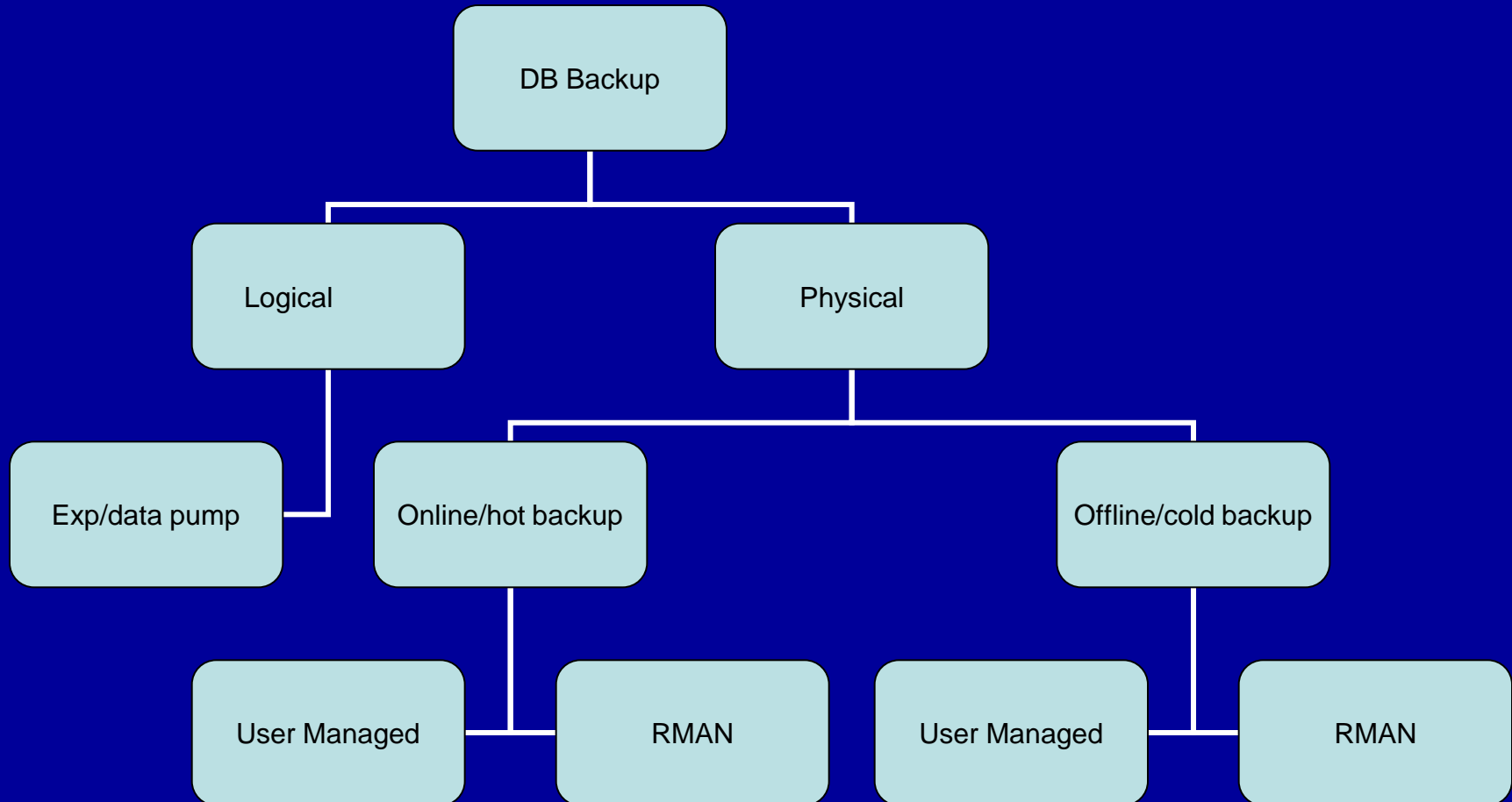
```
SQL> select status from v$instance;
```

```
STATUS
```

```
-----
```

```
OPEN
```

# Backup Overviews



# Database Mode Determines Backup Strategy

---

- If database in NOARCHIVELOG mode, then only cold backup is valid.
- If database in ARCHIVELOG mode, then the database can be backed up either online or offline/cold

# NOARCHIVELOG Mode

- User managed: shutdown database, and copy the backup files by using OS tools.
- Server managed: RMAN.
- Not acceptable in a 24/7 production system.

# ARCHIVELOG Mode

User managed: can be online or offline

If conducting online backup:

1. *ALTER TABLESPACE BEGIN BACKUP / ALTER DATABASE BEGIN BACKUP*

2. Copy database files

3. *ALTER TABLESPACE ... END BACKUP / ALTER DATABASE END BACKUP*

Server managed/RMAN: can be online or offline

Oracle database server reads the datafiles, not an operating system utility. The server reads each block and determines whether the block is fractured. If the block is fractured, then Oracle re-reads the block until it gets a consistent picture of the data.

# Backup Strategy at OIR

- Platform: SUSE linux
- Oracle version: 10g, release 2
- Database available 24/7
- Database in ARCHIVELOG mode
- Database size is now less than 100g
- No RAC
- SAN is the primary storage

# Backup Strategy at OIR -- *Continued*

- Physical backup with RMAN
- Online/hot backup
- Weekly full database with spfile and control file automated backup
- Backup archived logs
  
- Use EXP utility for logical backup
  
- Move the backup files off the disk to other permanent storage media

# Why Use RMAN

1. RMAN is a database backup utility that comes with the Oracle database, at no extra cost
2. RMAN is aware of the internal structure of Oracle datafiles and controlfiles, and knows how to take consistent copies of data blocks even as they are being written to
3. For online backup, It does not require the database in backup mode. Therefore RMAN does not cause a massive increase in generated redo
4. Backs up only those blocks that have held or currently hold data. RMAN backups of datafiles are generally smaller than the datafiles themselves. In contrast, OS copies of datafiles have the same size as the original datafiles
5. Can make incremental backup
6. Possible to recover individual blocks in case of block corruption of datafiles.

# Catalog or Nocatalog - a Big Decision

- RMAN can be run in two modes - catalog and nocatalog
- In the former, backup information and RMAN scripts are stored in another database known as the RMAN catalog.
- In the latter, RMAN stores backup information in the target database controlfile. Catalog mode is more flexible, but requires the maintenance of a separate database on another machine. Nocatalog mode has the advantage of not needing a separate database, but places more responsibility on the controlfile.
- OIR uses nocatalog mode, as this is a perfectly valid choice for sites with a small number of databases.

# Start RMAN

RMAN can be invoked from the command line on the database host machine

```
oracle@oirrep:~> $ORACLE_HOME/bin/rman target /
```

```
Recovery Manager: Release 10.2.0.1.0 - Production on Tue Sep 19 11:02:10  
2006
```

```
Copyright (c) 1982, 2005, Oracle. All rights reserved.
```

```
connected to target database: OIR10GR2 (DBID=3090918307)
```

```
RMAN>
```

# RMAN Configuration

- RMAN can be configured through various persistent parameters. Note that persistent parameters can be configured only for Oracle versions 9i and better. The current configuration can be seen via the "show all" command:

```
RMAN> show all;
```

```
RMAN configuration parameters are:
```

```
CONFIGURE RETENTION POLICY TO REDUNDANCY 2;
```

```
CONFIGURE BACKUP OPTIMIZATION OFF; # default
```

```
CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default
```

```
CONFIGURE CONTROLFILE AUTOBACKUP ON;
```

```
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE  
TYPE DISK TO
```

```
RMAN>
```

# RMAN Configurations - *continued*

- **Retention Policy:** This instructs RMAN on the backups that are eligible for deletion. For example: A retention policy with redundancy 2 would mean that two backups - the latest and the one prior to that - should be retained. All other backups are candidates for deletion. Retention policy can also be configured based on time - check the docs for details on this option.
- **Default Device Type:** This can be "disk" or "sbt" (system backup to tape). We will backup to disk and then have our OS backup utility copy the completed backup, and other supporting files, to a permanent storage.
- **Controlfile Autobackup:** This can be set to "on" or "off". When set to "on", RMAN takes a backup of the controlfile AND server parameter file each time a backup is performed. "off" is the default.
- **Controlfile Autobackup Format:** This tells RMAN where the controlfile backup is to be stored. The "%F" in the file name instructs RMAN to append the database identifier and backup timestamp to the backup filename. The database identifier, or DBID, is a unique integer identifier for the database.

# RMAN Configurations - *continued*

- **Parallelism:** This tells RMAN how many server processes you want dedicated to performing the backups.
- **Device Type Format:** This specifies the location and name of the backup files. We need to specify the format for each channel. The "%U" ensures that Oracle appends a unique identifier to the backup file name. The MAXPIECESIZE attribute sets a maximum file size for each file in the backup set.
- Any of the above parameters can be changed using the commands displayed by the "show all" command.

# Use RMAN in a Script

```
#!/bin/bash
export CLASSPATH=Put your Oracle Classpath here
export ORACLE_HOME=Put your Oracle Home here
export PATH=$PATH:$ORACLE_HOME/bin
export DATABASE_NAME=you DB name
export SYS_PASSWORD=password for SYS
export BACKUP_DIR=/opt/oracle/backup/rman
export BACKUP_DAY=`date +%m%d%Y`

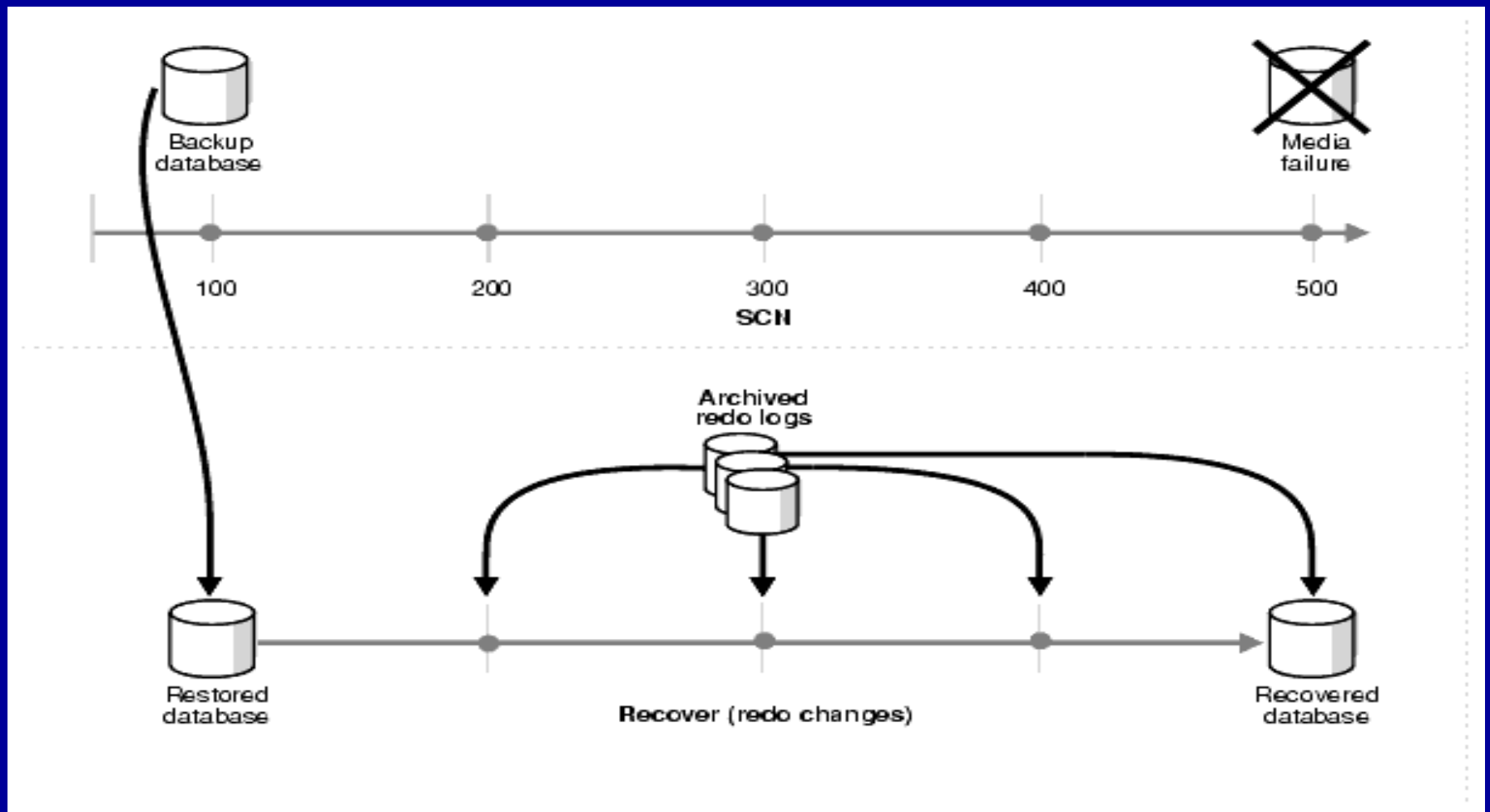
$ORACLE_HOME/bin/rman target sys/$SYS_PASSWORD@$DATABASE_NAME NOCATALOG <<EOF
RUN
{
#the following configuration should be run only once, the first time to backup
#CONFIGURE RETENTION POLICY TO REDUNDANCY 1;
#CONFIGURE DEFAULT DEVICE TYPE TO DISK; #BACKUP TYPE TO COPY
#CONFIGURE CONTROLFILE AUTOBACKUP ON;
#CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '$BACKUP_DIR/cf%f';
#CONFIGURE DEVICE TYPE DISK PARALLELISM 1;
#CONFIGURE CHANNEL 1 DEVICE TYPE DISK FORMAT '$BACKUP_DIR/backup_db_%U_$BACKUP_DAY' MAXPIECESIZE 4G;
#CONFIGURE CHANNEL 2 DEVICE TYPE DISK FORMAT '$BACKUP_DIR/disk2/backup_db_%d_S_%s_P_%p_T_%t' MAXPIECESIZE 4G;
#end of the configuration
backup database;
backup archivelog all format '$BACKUP_DIR/arc_%U_$BACKUP_DAY' delete all input;
crosscheck backup;
delete noprompt force obsolete;
sql 'create pfile from spfile';
host 'cp $ORACLE_HOME/network/admin/tnsnames.ora $BACKUP_DIR';
host 'cp $ORACLE_HOME/network/admin/listener.ora $BACKUP_DIR';
host 'cp $ORACLE_HOME/dbs/inito10gr2.ora $BACKUP_DIR';
}
exit;
EOF
```

# Revoking RMAN Script

The next step is to schedule the Linux script

```
00 1 * * 6 root /opt/oracle/cronJobs/rmanArchive.sh >/dev/null 2>& 1
```

# Database Restore and Recover



# SCN – DBA Must Understand

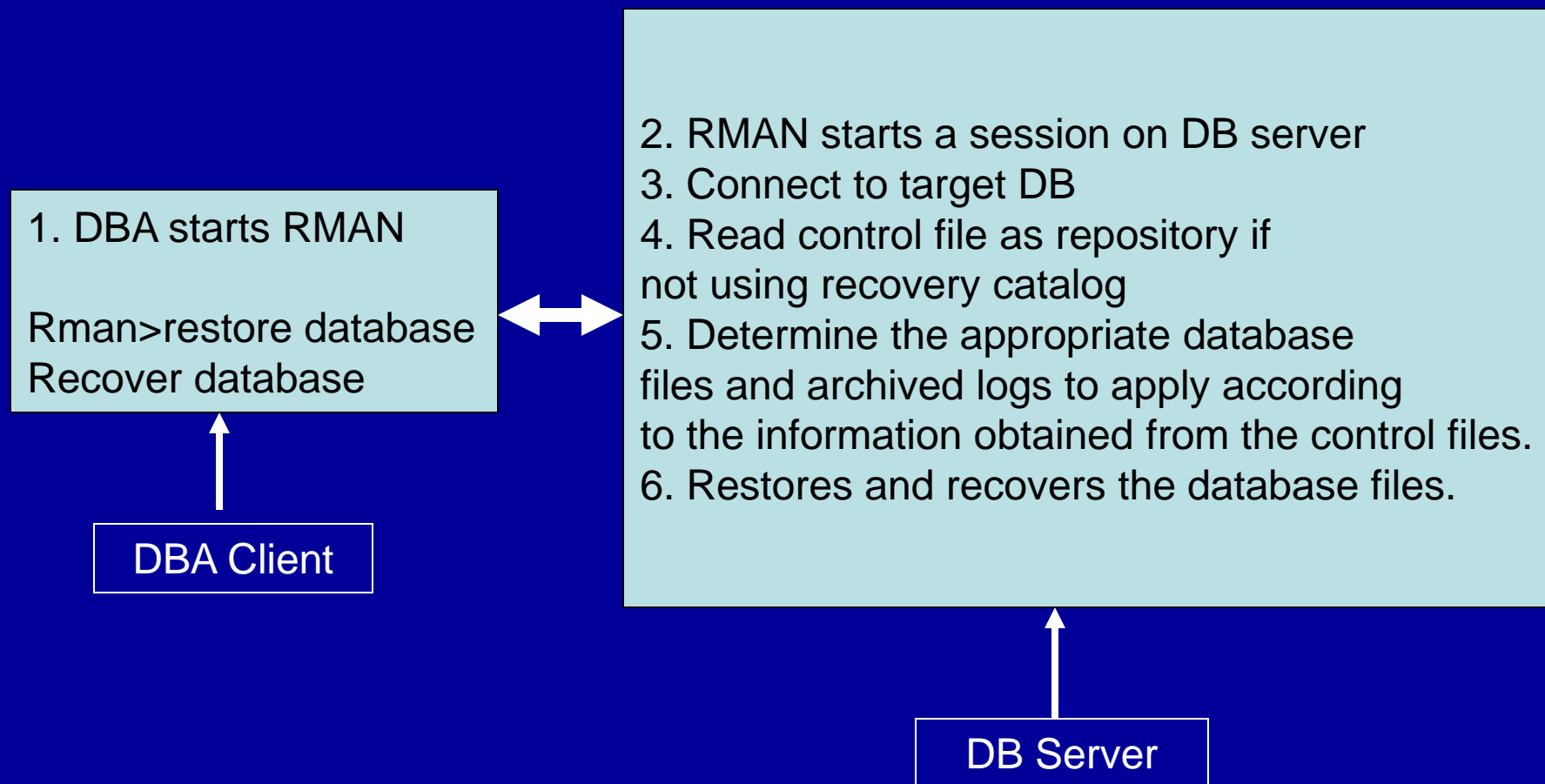
- SCN Saarbruecken, Germany - Ensheim (Airport Code)
- SCN Saskatchewan Communications Network
- SCN Satellite Communications Network
- SCN Satellite Control Network
- SCN Scan
- SCN Schwarzkopf Coaster Net (website)
- SCN Scientology
- SCN Scottish Candidate Number (unique serial number given to each student sitting Scottish Examinations)
- SCN Search Control Number
- SCN Sears Communications Network
- SCN Sensor Control Network
- SCN Sequential Contact Number
- SCN Service Channel Network (Ciena)
- SCN Service Circuit Node (AT&T)
- SCN Service Convergence Network (Pannaway Technologies)
- SCN Severe Congenital Neutropenia
- SCN Shanghai Cable Networks
- SCN Shipbuilding & Conversion, Navy
- SCN Shipbuilding and Conversion, Navy
- SCN Shipping Control Note
- SCN Shipping Control Number
- SCN Ships Construction, Navy
- SCN Software Change Notice
- SCN Southern Command Network
- SCN Special Care Nursery
- SCN Specification Change Notice
- SCN Spoken Called Number (Sprint - Voicemail)
- SCN Starting Cluster Number
- SCN Stock Code Number
- SCN Structured Cable Network
- SCN Student Center Network (forum)
- SCN Student Club Nights
- SCN Subcontract Change Notice
- SCN Subcutaneous Nodule
- SCN Supply Chain Navigator
- SCN Suprachiasmatic Nucleus
- SCN Surrender Charge Notice (insurance)
- SCN Sustainable Communities Network
- SCN Switched Circuit Network
- SCN Symmetrical Condensed Node
- SCN System Change Notice
- SCN **System Change Number (Oracle)**

# Case Study Assumptions

---

1. The database host server is still up and running
2. The last full backup is available on disk
3. All archived logs since the last backup are available on disk
4. RMAN is the tool for database recovery

# RMAN Recovery Process



# Case 1. Recovery From Missing/Corrupted Datafile

```
SQL> connect / as sysdba
```

```
Connected to an idle instance
```

```
SQL> startup
```

```
ORACLE instance started.
```

```
Total System Global Area 131555128 bytes
```

```
Fixed Size 454456 bytes
```

```
Variable Size 88080384 bytes
```

```
Database Buffers 41943040 bytes
```

```
Redo Buffers 1077248 bytes
```

```
Database mounted.
```

```
ORA-01157: cannot identify/lock data file 4 - see DBWR trace file
```

If you know the data file name, you can find out the file number by:

```
Select file# from v$datafile where name = 'Your datafile name';
```

# Case 1 - *continued*

*RMAN> restore datafile 4;*

*RMAN> recover datafile 4;*

*RMAN> alter database open;*

# Case 1 – *Continued*

The database must be mounted before any datafile recovery can be done.

In the above scenario, the database is already in the mount state before the RMAN session is initiated. If the database is not mounted, you should issue a "startup mount" command before attempting to restore the missing datafile.

If the database is already open when datafile corruption is detected, you can recover the datafile without shutting down the database. The only additional step is to take the relevant tablespace offline before starting recovery. In this case you would perform recovery at the tablespace level. The commands are:

```
RMAN> sql 'alter tablespace USERS offline immediate';  
RMAN> recover tablespace USERS;  
RMAN> sql 'alter tablespace USERS online';
```

Here we have used the SQL command, which allows us to execute arbitrary SQL from within RMAN.

# Case 2: Recovery From Block Corruption

---

- It is possible to recover corrupted blocks using RMAN backups
- RMAN> **blockrecover datafile 4 block 2015;**

## Case 2 - *continued*

### Important points regarding block recovery

1. Block recovery can only be done using RMAN.
2. The entire database can be open while performing block recovery.
3. To verify using RMAN simply do a complete database backup with default settings. If RMAN detects block corruption, it will exit with an error message pointing out the guilty file/block.

## Case 3. Recovery From Missing One Redo Log

If a redo log is missing, it should be restored from a multiplexed copy, if possible. This is the only way to recover without any losses. Here's an example, where I attempt to startup from SQLPlus when a redo log is missing:

```
SQL> startup
```

```
ORACLE instance started.
```

```
Total System Global Area 131555128 bytes
```

```
Fixed Size 454456 bytes
```

```
Variable Size 88080384 bytes
```

```
Database Buffers 41943040 bytes
```

```
Redo Buffers 1077248 bytes
```

```
Database mounted.
```

```
ORA-00313: open failed for members of log group 3 of thread 1
```

```
ORA-00312: online log 3 thread 1: '/redoDir/REDO03A.LOG'
```

```
SQL>
```

## Case 3 - continued

- To fix this we simply copy REDO03A.LOG from its multiplexed location. After copying the file, we issue an "alter database open" from the above SQLPlus session:

```
SQL> alter database open;
```

```
Database altered.
```

```
SQL>
```

# Case 4. Recovery From Missing All Log Files

In this case an incomplete recovery is the best we can do. We will lose all transactions from the missing log and all subsequent logs.

The error message indicates that members of log group 3 are missing. We don't have a copy of this file, so we know that an incomplete recovery is required.

The first step is to determine how much can be recovered. In order to do this, we query the V\$LOG view (when in the mount state) to find the system change number (SCN) that we can recover to (Reminder: the SCN is a monotonically increasing number that is incremented whenever a commit is issued):

*--The database should be in the mount state for v\$log access*

```
SQL> select first_change# from v$log where group#=3 ;
```

```
FIRST_CHANGE#
```

```
-----  
370255
```

```
SQL>
```

## Case 4 - *continued*

- The **FIRST\_CHANGE#** is the first SCN stamped in the missing log. This implies that the last SCN stamped in the previous log is 370254 (**FIRST\_CHANGE#-1**). This is the highest SCN that we can recover to. In order to do the recovery we must first restore ALL datafiles to this SCN, followed by recovery (also up to this SCN). This is an incomplete recovery, so we must open the database resetlogs after we're done. Here's a transcript of the recovery session (typed commands in bold, comments in italics, all other lines are RMAN feedback):

```
C:\>rman target /
```

```
Recovery Manager: Release 9.2.0.4.0 - Production
```

```
Copyright (c) 1995, 2002, Oracle Corporation. All rights reserved.
```

```
connected to target database: ORCL (DBID=1507972899)
```

```
--Restore ENTIRE database to determined SCN
```

## Case 4 - continued

- **RMAN> restore database until scn 370254;**  
*--Recover database*
- **RMAN> recover database until scn 370254;**
- *--open database with RESETLOGS (see comments below)*

**RMAN> alter database open resetlogs;**

## Case 4 - *continued*

1. The entire database must be restored to the SCN that has been determined by querying v\$log.
2. All changes beyond that SCN are lost. This method of recovery should be used only if you are sure that you cannot do better. Be sure to multiplex your redo logs, and (space permitting) your archived logs!
3. The database must be opened with RESETLOGS, as a required log has not been applied. This resets the log sequence to zero, thereby rendering all prior backups worthless. Therefore, the **first step** after opening a database RESETLOGS is to **take a fresh backup**. Note that the RESETLOGS option must be used for any incomplete recovery.

# Case 5. Recovery From Corrupted One Control File

- On startup Oracle must read the control file in order to find out where the datafiles and online logs are located. Oracle expects to find control files at locations specified in the CONTROL\_FILE initialisation parameter. The instance will fail to mount the database if any one of the control files are missing or corrupt

*SQL> startup*

*ORACLE instance started.*

*ORA-00205: error in identifying controlfile, check alert log for more info*

## Case 5 - *continued*

---

### Solution:

1. On checking the alert log
2. Replace the corrupted control file with a copy using operating system commands
3. Remember to rename the copied file

# Case 6. Recovery From Missing All Control Files

- Requires that all logs (archived and current online logs) since the last backup are available.
- The logs are required because all datafiles must also be restored from backup.
- The database will then have to be recovered up to the time the control files went missing.
- This can only be done if all intervening logs are available.

# Case 6 - continued

- *-- Connect to RMAN*  
C:\>rman target /  
  
RMAN> **set dbid 1507972899**
- *--restore controlfile from autobackup. The backup is not at the default --location so the path must be specified*  
  
RMAN> **restore controlfile from /'backupDir/CTL\_SP\_BAK\_C-1507972899-20050124-00';**
- RMAN> **mount database;**
- RMAN> **restore database;**
- *--Database must be recovered because all datafiles have been restored from -- backup*  
  
RMAN> **recover database;**
- *-- Recovery completed. The database must be opened with RESETLOGS -- because a backup control file was used. Can also use -- "alter database open resetlogs" instead.*  
  
RMAN> **open resetlogs database;**

## Case 6 - *continued*

- Recovery using a backup controlfile should be done only if a current control file is unavailable.
- All datafiles must be restored from backup. This means the database will need to be recovered using archived and online redo logs. These **MUST** be available for recovery until the time of failure.
- As with any database recovery involving RESETLOGS, **take a fresh backup immediately.**
- Technically the above is an example of complete recovery - since all committed transactions were recovered.
- However, some references consider this to be incomplete recovery because the database log sequence had to be reset.
- After recovery using a backup controlfile, all temporary files associated with locally-managed tablespaces are no longer available. You can check that this is so by querying the view V\$TEMPFILE - no rows will be returned. Therefore tempfiles must be added (or recreated) before the database is made available for general use. In the case at hand, the tempfile already exists so we merely add it to the temporary tablespace. This can be done using SQLPlus or any tool of your choice:

```
SQL> alter tablespace temp add tempfile  
'/DBfileDir/TEMP01.DBF';
```

# Case 7. Recovery From Missing spfile

1. Set DBID

```
RMAN> set dbid 1507972899
```

2. To restore the spfile, you first need to startup the database in the *nomount* state. This starts up the database using a dummy parameter file.

```
RMAN> startup nomount
```

3. Restore spfile from backup

```
RMAN> restore spfile from '/backupDir/CTL_BAK_C-1507972899-20050228-00';
```

4. Restart database

```
RMAN> startup force nomount
```

The instance is now started up with the correct initialisation parameters.

# Road Map for Disaster Recovery

1. Copy password file and tnsnames file from backup
2. Set ORACLE\_SID environment variables
3. Invoke RMAN and set dbid
4. Restore spfile
5. Restore control file
6. Restore all datafiles
7. Recover database

