

12. User-managed complete recovery and RMAN complete recovery

Abstract: In this lesson we focus on media failures and how to recover from them. There are two methods that can be used to recover from media failures: user-managed recovery and RMAN-based recovery. This lesson uses examples to demonstrate the differences between each type. As discussed, the mode you decide to operate in, whether ARCHIVELOG mode or NOARCHIVELOG mode, determines the recovery options that you can perform. This lesson covers these options and the modes in which a DBA operates in further detail. You will also become familiar with examples of both ARCHIVELOG mode and NOARCHIVELOG mode media recoveries using both user-managed and RMAN methods of recovery. In addition to this detailed discussion of recovery methods, you also will look at recovery situations in which the relocation of files is required, and you will learn how to handle read-only tablespace recovery in different situations. Media recoveries are critical tasks in the testing process and workplace. How media recovery situations are handled depends on the DBA performing the recovery. You can improve your ability to perform such recoveries by testing various media recovery situations so that you have a degree of confidence. As a result of this practice, when you need to perform a media recovery, your uncertainties will be significantly reduced. Testing media recovery situations will also prepare you for the real-life situations that you will experience as a DBA.

Contents

1. Defining Media Recovery	2
2. Recovering Using NOARCHIVELOG and ARCHIVELOG Modes.....	2
2.1. Performing User-Managed Recovery in NOARCHIVELOG Mode	5
2.2. Performing User-Managed Complete Recovery in ARCHIVELOG Mode	10
3. Restoring Data Files to Different Locations.....	16
4. Relocate and Recover a Tablespace by Using Archived Redo Logs	17
5. Describe Read-Only Tablespace Recovery	21
6. Using RMAN for Restoration and Recovery	22
7. Performing RMAN Recovery in NOARCHIVE Mode.....	22
8. Performing RMAN Complete Recovery in ARCHIVELOG Mode.....	26
9. Using RMAN to Restore Data Files to Different Locations	31
10. Use RMAN to Relocate and Recover a Tablespace Using Archived Logs	35
11. Summary	37
References	38

Objectives:

- Describe media recovery.
- Perform recovery in Noarchivelog mode.
- Perform complete recovery in Archivelog mode.
- Restore datafiles to different locations.
- Relocate and recover a tablespace by using archived redo log files.
- Describe read-only tablespace recovery.
- Describe the use of RMAN for restoration and recovery.

1. Defining Media Recovery

Media recovery is a type of recovery used for recovering any currently used data file, control file, or online redo log file that becomes unavailable. The data file or control file may become unavailable for a number of reasons — it may have been lost, deleted, or moved from its original location, or it may have been damaged by data corruption or a hardware failure. All of these situations result in the Oracle database not being able to read or write to this file.

When a situation requiring media recovery occurs, the DBA must restore the unavailable file or files. If the database is in ARCHIVELOG mode, you must then recover these files by applying archived logs to the restored files. This will make the restored files as current as the rest of the database files.

2. Recovering Using NOARCHIVELOG and ARCHIVELOG Modes

One of the most significant backup and recovery decisions a DBA can make is whether to operate in ARCHIVELOG mode or NOARCHIVELOG mode. The outcome of this decision dramatically affects the backup and recovery options available.

When the database is in *ARCHIVELOG mode*, it generates historical changes in the form of offline redo logs, or archived logs. That is, the database doesn't write over the online redo logs until a copy is made, and this copy is called an offline redo log, or archived log. These logs can be applied to backups of the data files to recover the database up to the point of a failure. Figure 1 illustrates complete recovery in ARCHIVELOG mode.

When the database is in *NOARCHIVELOG mode*, it does not generate historical changes; as a result, there is no archive logging. In this mode, the database writes over the online redo logs without creating an archived log. Thus, no historical information is generated and saved for later use. Figure 2 illustrates incomplete recovery in NOARCHIVELOG mode. Even though this is called a complete recovery, the recovery process will be missing transactions because you are not generating archived logs to be applied in the recovery process.

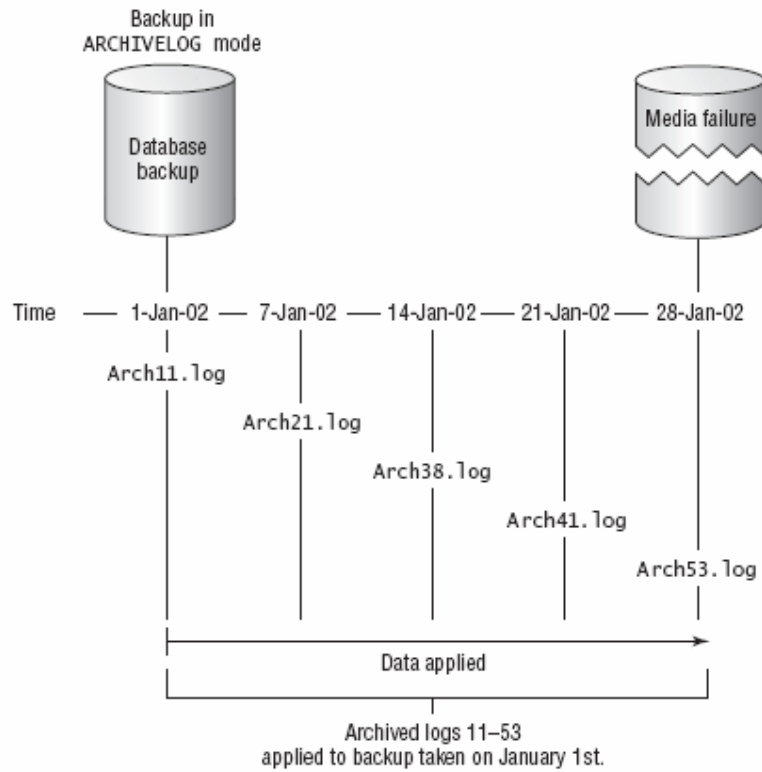


FIGURE 1. Complete recovery in ARCHIVELOG mode for media failure on January 28th

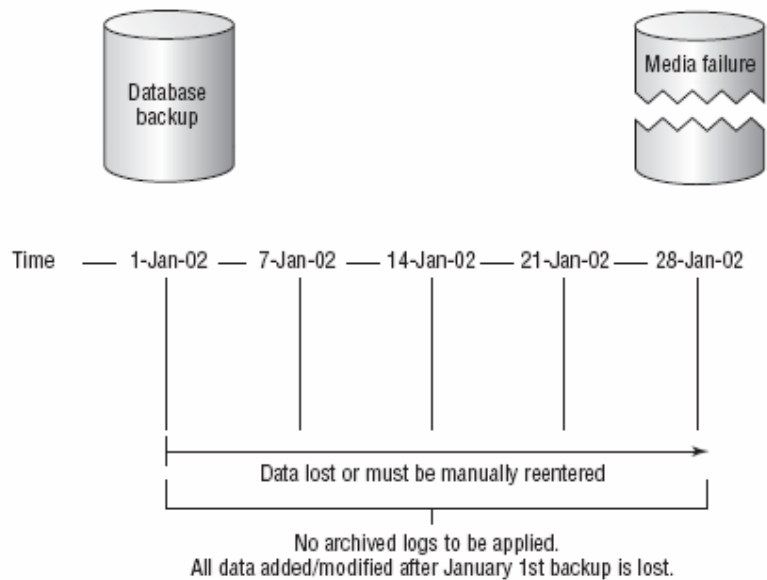


FIGURE 2. Complete recovery in NOARCHIVELOG mode for media failure on January 28th (transactions lost)

A significant type of failure is a media failure because in most cases, it requires that you restore all or some of the data files and the application of the redo logs during recovery. As recalled, media failure occurs when a database file cannot be accessed for some reason. The usual reason is a disk crash or controller failure.

REAL WORLD SCENARIO

The Dangers of Media Failures and Recovery

A media failure is generally considered the most dangerous type of failure and it is the most touchy to recover from as well. The severity of this type of failure may vary from a lost or accidentally removed data file to a severe hardware failure.

No matter what type of media failure the DBA is handling, they must contribute more analysis and thought to a media failure than they would to most other failure situations, such as those associated with instance recovery or basic user error. In fact, in certain situations, a severe hardware failure could cause a significant amount of the physical database to be relocated to a new file-system. And in some cases, new file-systems may need to be re-created and properly configured for stripe, load, and redundancy. This can be a difficult task to perform when the database is down, especially if minimal downtime cannot be tolerated by the users. In less demanding environments, the database may remain unavailable for an excessive period of time.

A case in point is a small nonprofit organization that lost a disk controller for its financial database application. As in many small companies, this organization was concerned about keeping IT costs to a minimum. As a result, most resources that had been purchased were in use, such as servers and disk space.

Another result of this policy was that extra capacity, such as multiple controllers to disk arrays, was not always purchased in an effort to reduce costs. Because of this, when the disk controller was lost, the financial instance was unavailable until a new disk controller could be purchased, delivered, and installed in the disk storage array. This application was unavailable for one business day until the hardware was successfully installed and the database was successfully restored and recovered.

Media failure requires database recovery. If the database is in ARCHIVELOG mode, complete recovery can be performed. This means that a backup can be restored to the affected file-system, and archived logs can be applied up to the point of failure. Thus, no data is lost.

If the database is in NOARCHIVELOG mode, however, a complete recovery cannot be performed without some transactions being lost. This means that a backup can be restored to the affected file-system, but there won't be any archived logs or historical changes saved. Thus, the database will have only the transactions that were available at the time of the backup. If backups were scheduled every night, the business would only lose one day's worth of transactions. If backups were scheduled weekly, on the other hand, the business would lose one week's worth of transactions. The end result is that, in almost all cases, some data will be lost. This is a complete recovery of the database but the database does not contain all the

transactions up to the failure point. The end result is that the database will be similar to an *incomplete recovery*, which will be covered latter on. The differences are that an incomplete recovery will be intentionally stopped before all the transactions are applied to the database, and an incomplete recovery requires the database to be in ARCHIVELOG mode and some other specific actions to be performed.

2.1. Performing User-Managed Recovery in NOARCHIVELOG Mode

This is an example of a user-managed recovery when the database is in NOARCHIVELOG mode. In this case, the database cannot be completely recovered. The database is available all day during the week. Every Saturday, the database is shut down, and a complete, *cold backup* (offline backup) is performed. The database is restarted when this activity is completed.

Diagnosing the Failure

On Wednesday morning, there is a lost or deleted data file in the database. The error received upon startup is as follows:

```
SQL> startup
ORACLE instance started.

Total System Global Area 19504528 bytes
Fixed Size 64912 bytes
Variable Size 16908288 bytes
Database Buffers 2457600 bytes
Redo Buffers 73728 bytes
Database mounted.
ORA-01157: cannot identify/lock data file 4 - see
  DBWR trace file
ORA-01110: data file 4:
  '/db01/ORACLE/tst9/users01.dbf'
```

Because you are operating in NOARCHIVELOG mode, you must perform a full database restore from the previous weekend. You cannot perform a tablespace or data file recovery in NOARCHIVELOG mode because you have no ability to roll back and roll forward historical changes. There are no archived logs to apply to the data file to make the data file current with the rest of the database. Data entered into the database on Monday, Tuesday, and Wednesday are lost and must be reentered, if possible.

To perform the database restore, you will need to copy all the data files, online redo logs, and control files from the last Saturday backup and perform a full database restore. You will then copy these files back to their original locations.

Step-by-Step Recovery

To recover a lost data file when you are operating in NOARCHIVELOG mode, take the following steps:

1. Perform a *cold backup* of the database to simulate the Saturday backup. The following is a sample script, which performs a cold backup by shutting down the database and copying the necessary data files, redo logs, and control files.

```
# User-managed backup script
# Cold backup script for tst9
#
echo ''
echo 'starting cold backup...'
echo ''
# Script to stop database!
./stopdb_tst9.sh
echo ''
echo 'tst9 shutdown...'
echo ''
echo 'clean up last backup in staging directory'
rm /staging/cold/tst9/*
rm /staging/cold/tst9/*
echo ''
echo 'copying files to staging...'
echo ''
cp /db01/oracle/tst9/* /staging/cold/tst9/.
cp /db02/oracle/tst9/* /staging/cold/tst9/.
cp /oracle/admin/tst9/arch/* /staging/cold/tst9/.
echo ''
echo 'tst9 starting up.....'
echo ''
```

Script to startup database!

```
./startdb_tst9.sh
```

2. Validate that the user TEST's objects exist in the USERS tablespace. This is the tablespace that you will remove to simulate a lost or deleted data file.

```
SQL> select username, default_tablespace,  
       temporary_tablespace from dba_users;
```

```
USERNAME DEFAULT_TABLESPACE TEMPORARY_TABLESPACE
```

```
-----  
SYS       SYSTEM                TEMP  
SYSTEM    TOOLS                  TEMP  
OUTLN     SYSTEM                SYSTEM  
DBSNMP    SYSTEM                SYSTEM  
TEST     USERS                  TEMP
```

5 rows selected.

```
SQL>
```

3. Create a table and insert data to simulate data being entered after Saturday's cold backup. This is the data that would be entered on Monday through Wednesday, before the failure, but after the cold backup. The user TEST was created before the cold backup with a default tablespace of USERS. The account has connect and resource privileges.

```
SQL> connect test/test
```

```
SQL> create table t1 (c1 number, c2 char (50));
```

Statement processed.

```
SQL> insert into t1 values (1, 'This is a test!');
```

1 row processed.

```
SQL> commit;
```

Statement processed.

```
SQL>
```

4. Verify the data file location of the USERS tablespace. Then remove or delete this file.

```
SQL> select name from v$datafile;
```

```
NAME
```

```
-----  
/db01/ORACLE/tst9/system01.dbf  
/db01/ORACLE/tst9/rbs01.dbf  
/db01/ORACLE/tst9/temp01.dbf  
/db01/ORACLE/tst9/users01 .dbf  
/db01/ORACLE/tst9/tools01.dbf  
/db01/ORACLE/tst9/data01.dbf  
/db01/ORACLE/tst9/indx01.dbf
```

```
7 rows selected.
```

```
SQL>
```

```
rm /db01/ORACLE/tst9/users01 .dbf
```

5. Start the database and verify that the “cannot identify/lock data file” error occurs.

```
[oracle@DS-LINUX tst9]$ sqlplus /nolog
```

```
SQL*Plus: Release 9.0.1.0.0 - Production on Thu Nov 1 21:04:10 2001
```

```
(c) Copyright 2001 Oracle Corporation. All rights reserved.
```

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

```
Connected.
```

```
SQL> startup
```

```
ORACLE instance started.
```

```
Total System Global Area 19504528 bytes
```

```
Fixed Size 64912 bytes
```

```
Variable Size 16908288 bytes
```

```
Database Buffers 2457600 bytes
```

```
Redo Buffers 73728 bytes
```

```
Database mounted.
```

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

```
see DBWR trace file
```

```
ORA-01110: data file 4:
```

```
'/db01/ORACLE/tst9/users01.dbf'
```

6. Shut down the database to perform a complete database restore. The database must be shut down to restore a cold backup.

```
SQL> shutdown
```

```
ORA-01109: database not open
```

```
Database dismounted.
```

```
ORACLE instance shut down.
```

```
SQL>
```

7. Perform a complete database restore by copying all data files, redo logs, and control files to their original locations.

```
cp /staging/cold/tst9/* /db01/ORACLE/tst9
```

8. Start the database and check to see whether the data entered after the cold backup is there. When you do, you will see that Table t1 and the data do not exist. All data entered after the last backup will have to be reentered.

```
[oracle@DS-LINUX backup]$ sqlplus /nolog
```

```
SQL*Plus: Release 9.0.1.0.0 - Production on Thu Nov 1 21:04:10 2001
```

```
(c) Copyright 2001 Oracle Corporation. All rights reserved.
```

```
SQL> connect test/test
```

```
Connected.
```

```
SQL> select * from t1;
```

```
select * from t1
```

```
*
```

```
ORA-00942: table or view does not exist
```

```
SQL>
```

Conclusions

The most notable observation about this scenario is that when the database is in NOARCHIVELOG mode, data is lost. All data entered after the backup, but before the failure, is lost and must be reentered. To recover it, you will have to shut down the database. Furthermore, you must restore the whole database instead of just the one data file that was lost or removed, which could increase the recovery time.

2.2. Performing User-Managed Complete Recovery in ARCHIVELOG Mode

In this example, the database is completely recovered because it is in ARCHIVELOG mode. This database is available 24 hours a day, 7 days a week, with the exception of scheduled maintenance periods. Every morning at 1 A.M., a *hot backup* is performed. The data files, archived logs, control files, backup control files, and init.ora files are copied to a staging directory, and from there, they are then copied to tape. The copy also remains on the disk until the next morning, when the hot backup runs again. This allows for quick access in the event of failure. When the backup runs again, the staging directory is purged and rewritten.

Diagnosing the Failure

On Wednesday morning, there is a lost or deleted data file in the database. The error received upon startup is as follows:

```
SQL> startup
ORACLE instance started.

Total System Global Area 19504528 bytes
Fixed Size 64912 bytes
Variable Size 16908288 bytes
Database Buffers 2457600 bytes
Redo Buffers 73728 bytes
Database mounted.
ORA-01157: cannot identify/lock data file 4 -
  see DBWR trace file
ORA-01110: data file 4:
  '/db01/ORACLE/tst9/users01.dbf'
```

In this case, you are operating in ARCHIVELOG mode, so you only need to replace the damaged or lost file: /db01/oracle/tst9/users01.dbf. Then, with the database open, the archived logs can be applied to the database. This archived log action reapplies all the changes to the database; therefore, no data will be lost.

Step-by-Step Recovery

To recover the lost data file, take these steps:

1. Connect to user TEST and enter data in Table t1 in the tablespace USERS, which consists of the data file users01.dbf. This will simulate the data that is in the hot backup of the USERS tablespace.

```
[oracle@DS-LINUX backup]$ sqlplus /nolog
SQL*Plus: Release 9.0.1.0.0 - Production on Thu Nov 1
21:04:10 2001
```

(c) Copyright 2001 Oracle Corporation. All rights reserved.

```
SQL> connect test/test
```

Connected.

```
SQL> insert into t1 values (1,'This is test one -
before hot backup');
```

1 row processed.

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

```
SQL> commit;
```

Statement processed.

```
SQL> select username,default_tablespace from
  2> dba_users where username = 'TEST';
```

```
USERNAME  DEFAULT_TABLESPACE
```

```
-----
```

```
TEST      USERS
```

1 row selected.

2. Perform a hot backup of the USERS tablespace by placing it in backup mode. Proceed to copy the data file users01.dbf to a staging directory. Then, end the backup of the USERS tablespace.

```
SQL>
```

```
[oracle@DS-LINUX backup]$ sqlplus /nolog
```

```
SQL*Plus: Release 9.0.1.0.0 - Production on Thu Nov 1 21:04:10 2001
```

(c) Copyright 2001 Oracle Corporation. All rights reserved.

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

Connected.

```
SQL> alter tablespace users begin backup;
Statement processed.
SQL> ! cp /db01/ORACLE/tst9/users01.dbf /stage
SQL> alter tablespace users end backup;
Statement processed.
SQL> alter system switch logfile;
Statement processed.
```

3. Connect to the user TEST and add more data to Table t1. This data is in rows 2 and 3. This data has been added after the backup of the users01.dbf data file, therefore, the data is not part of the data file copied earlier. After this is done, perform log switches to simulate normal activity in the database. This activates the archiver process to generate archived logs for the newly added data.

```
[oracle@DS-LINUX backup]$ sqlplus /nolog
SQL*Plus: Release 9.0.1.0.0 - Production on Thu Nov 1 21:04:10 2001
(c) Copyright 2001 Oracle Corporation. All rights reserved.
SQL> connect test/test
Connected.
SQL> insert into t1 values(2,'This is test two -
after hot backup');
1 row processed.
SQL> insert into t1 values(3,'This is test three -
after hot backup');
1 row processed.
SQL> commit;
Statement processed.
SQL> connect / as sysdba
Connected.
SQL> alter system switch logfile;
Statement processed.
SQL> alter system switch logfile;
Statement processed.
```

```
SQL> alter system switch logfile;
```

```
Statement processed.
```

```
SQL> alter system switch logfile;
```

```
Statement processed.
```

4. Verify the location of the data file of the USERS tablespace. Then remove or delete this file.

```
SQL> ! rm /db01/ORACLE/tst9/users01.dbf
```

5. Shut down the database. Upon restarting, verify that the missing data file error occurs.

```
[oracle@DS-LINUX tst9]$ sql
```

```
SQL*Plus: Release 9.0.1.0.0 - Production on Thu Nov 1 21:04:10 2001
```

```
(c) Copyright 2001 Oracle Corporation. All rights reserved.
```

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

```
Connected to an idle instance.
```

```
SQL> startup
```

```
ORACLE instance started.
```

```
Total System Global Area 19504528 bytes
```

```
Fixed Size 64912 bytes
```

```
Variable Size 16908288 bytes
```

```
Database Buffers 2457600 bytes
```

```
Redo Buffers 73728 bytes
```

```
Database mounted.
```

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

```
see DBWR trace file
```

```
ORA-01110: data file 4:
```

```
'/db01/ORACLE/tst9/users01.dbf'
```

```
SQL>
```

6. Take the recovered data file offline. This will enable you to recover this data file and tablespace while the rest of the database is available for user access.

```
SQL> alter database datafile '/db01/oracle/tst9/
users01.dbf' offline;
Statement processed.
```

7. Restore the individual data file by copying the data file users01.dbf back to the original location.

```
[oracle@DS-LINUX tst9]$ cp /stage/users01.dbf /db01/oracle/tst9
```

8. With the database open, begin the recovery process by executing the RECOVER DATABASE command. Then, apply all the available redo logs; this should result in a complete recovery. Finally, bring the data file online so that it is available for access by users.

```
SQL> connect /as sysdba
Connected.
```

```
SQL> recover datafile '/db01/ORACLE/tst9/users01.dbf';
ORA-00279: change 48323 generated at 03/29/00 22:04:25
needed for thread 1
ORA-00289: suggestion :
  /oracle/admin/tst9/arch1/archtst9_84.log
ORA-00280: change 48323 for thread 1 is in sequence #84
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
Log applied.
ORA-00279: change 48325 generated at 03/29/00 22:05:25
needed for thread 1
ORA-00289: suggestion :
  /oracle/admin/tst9/arch1/archtst9_85.log
ORA-00280: change 48325 for thread 1 is in sequence #85
ORA-00278: log file
  '/oracle/admin/tst9/arch1/archtst9_84.log'
no longer needed for this recovery
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
Log applied.
```

ORA-00279: change 48330 generated at 03/29/00 22:08:41
needed for thread 1

ORA-00289: suggestion :

/oracle/admin/tst9/arch1/archtst9_86.log

ORA-00280: change 48330 for thread 1 is in sequence #86

ORA-00278: log file

'/oracle/admin/tst9/arch1/archtst9_85.log'

no longer needed for this recovery

Specify log: {<RET>=suggested | filename | AUTO |
CANCEL}

ORA-00279: change 48330 generated at 03/29/00 22:08:41
needed for thread 1

ORA-00289: suggestion :

/oracle/admin/tst9/arch1/archtst9_86.log

ORA-00280: change 48330 for thread 1 is in sequence #86

ORA-00278: log file

'/oracle/admin/tst9/arch1/archtst9_85.log'

no longer needed for this recovery

Specify log: {<RET>=suggested | filename | AUTO | CANCEL}

Log applied.

Media recovery complete.

SQL>

SQL> alter database datafile

'/db01/ORACLE/tst9/users01.dbf' online;

Statement processed.

9. Verify that there is no data loss, even though records 2 and 3 were added after the hot backup. The data for these records were applied from the offline redo logs (archived logs).

SQL> select * from t1;

C1	C2
----	----

- ```

```
- 1 This is a test one - before hot backup
  - 2 This is a test two - after hot backup
  - 3 This is a test three - after hot backup

3 rows selected.

SQL>

### Conclusions

The most notable observation about this scenario is that when the database is in ARCHIVELOG mode, no data is lost. All data entered after the hot backup into the USERS tablespace, but before the failure, is not lost. Only the data file users01.dbf must be restored, which takes less time than restoring all the data files. Therefore, by applying the archived logs during the recovery process, you can salvage all changes that occur after a hot backup of a data file. Another equally important feature is that the database can remain open to users while the one tablespace and associated data file(s) are being recovered. This allows users to access data in other tablespaces of the database not affected by the failure.

## 3. Restoring Data Files to Different Locations

Restoring data files to a different location in both ARCHIVELOG mode and NOARCHIVELOG mode can be performed in a similar manner. The main difference is that like any NOARCHIVELOG mode recovery, the database in most cases cannot be completely recovered to the point of failure. The only time a database can be completely recovered in NOARCHIVELOG mode is when the database has not cycled through all of the online redo logs since the last complete backup.

To restore the files to a different location, you would perform an OS copy from the backup location to the new location, and then start the database at the mount stage. After that, you would update the control file with the ALTER DATABASE RENAME FILE command to designate the new location. Let's walk through this procedure.

1. Use OS commands to restore files to new locations.

```
cp /db01/oracle/tst9/data01.dbf /db02/oracle/tst9/data01.dbf
```

2. Start up the database instance and mount the database.

```
oracle@octilli:~ > oraenv
```

```
ORACLE_SID = [tst9] ?
```

```
oracle@octilli:~ > sqlplus /nolog
```

```
SQL*Plus: Release 9.0.1.0.0 - Production on Mon Oct 29 23:26:23 2001
```



(c) Copyright 2001 Oracle Corporation. All rights reserved.

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

```
Connected.
```

```
SQL> startup mount
```

```
ORACLE instance started.
```

```
Total System Global Area 75854976 bytes
```

```
Fixed Size 279680 bytes
```

```
Variable Size 71303168 bytes
```

```
Database Buffers 4194304 bytes
```

```
Redo Buffers 77824 bytes
```

```
Database mounted.
```

```
SQL>
```

3. Use the ALTER DATABASE RENAME FILE command to designate the new location.

```
SQL> ALTER DATABASE RENAME FILE
```

```
2> '/db01/oracle/tst9/data01.dbf' to
```

```
3> '/db02/oracle/tst9/data01.dbf';
```

4. Use the ALTER DATABASE OPEN command to open the database.

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

```
Database altered.
```

## 4. Relocate and Recover a Tablespace by Using Archived Redo Logs

In this example, during the recovery process, you will relocate a tablespace to a new file-system by restoring the tablespace's data files to a new file-system. You will use the RECOVER DATABASE command to determine which archived logs you will need to apply to the newly relocated data files. This type of recovery can be performed at the tablespace level or at the database level. If you perform it at the tablespace level, you will need to take the tablespace offline; at the database level, you will need to start and mount the database.

Below is an example of this recovery procedure at the database level.

1. Set ORACLE\_SID to ORCL, which is your target database, so that the database can be started or mounted with SQL\*Plus.

```
oracle@octilli:~ > oraenv
```

```
ORACLE_SID = [tst9] ?
```

2. Run the appropriate user-managed script to back up the tst9 database to disk. This customized script shuts down the database and then copies the data files, control files, redo logs, and archived log files to a staging directory. After this is done, database tst9 is restarted.

```
User-managed backup script
Cold backup script for tst9
#
echo ''
echo 'starting cold backup...'
echo ''
Script to stop database!
./stopdb_tst9.sh
echo ''
echo 'tst9 shutdown...'
echo ''
echo 'clean up last backup in staging directory'
rm /staging/cold/tst9/*
rm /staging/cold/tst9/*
echo ''
echo 'copying files to staging...'
echo ''
cp /db01/oracle/tst9/* /staging/cold/tst9/.
cp /db02/oracle/tst9/* /staging/cold/tst9/.
cp /oracle/admin/tst9/arch/* /staging/cold/tst9/.
echo ''
echo 'tst9 starting up.....'
echo ''
```

---

# Script to startup database!

```
./startdb_tst9.sh
```

3. Now start up the database to demonstrate the INDX tablespace failure that will need to be restored, recovered, and relocated to the new file-system.

```
oracle@octilli:/db01/oracle/tst9 > sqlplus /nolog
```

```
SQL*Plus: Release 9.0.1.0.0 - Production on Thu Nov 1 21:04:10 2001
```

```
(c) Copyright 2001 Oracle Corporation. All rights reserved.
```

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

```
Connected to an idle instance.
```

```
SQL> startup mount
```

```
ORACLE instance started.
```

```
Total System Global Area 75854976 bytes
```

```
Fixed Size 279680 bytes
```

```
Variable Size 71303168 bytes
```

```
Database Buffers 4194304 bytes
```

```
Redo Buffers 77824 bytes
```

```
Database mounted.
```

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

```
alter database open
```

```
*
```

```
ERROR at line 1:
```

```
ORA-01157: cannot identify/lock data file 7 - see DBWR
```

```
trace file
```

```
ORA-01110: data file 7: '/db01/oracle/tst9/indx01.dbf'
```

```
SQL>
```

4. Next, shut down the database.

```
SQL> shutdown
```

```
ORA-01109: database not open
```

```
Database dismounted.
```

ORACLE instance shut down.

5. After this is accomplished, restore the backup indx01.dbf file from the online backup directory located in /staging/cold/tst9/indx01.dbf to the new filesystem /db02/oracle/tst9.

```
oracle@octilli:/staging/cold/tst9 > cp indx01.dbf /db02/oracle/tst9/.
```

6. Next, start up and mount the database, and then use the RENAME command to update the control file with the indx01.dbf data file's new location.

```
SQL> startup mount
```

ORACLE instance started.

Total System Global Area 75854976 bytes

Fixed Size 279680 bytes

Variable Size 71303168 bytes

Database Buffers 4194304 bytes

Redo Buffers 77824 bytes

Database mounted.

```
SQL> alter database rename file
```

```
2 '/db01/oracle/tst9/indx01.dbf' to
```

```
3 '/db02/oracle/tst9/indx01.dbf';
```

Database altered.

```
SQL>
```

7. Then recover the database and apply the necessary archived logs to make the indx01.dbf data file in the INDX tablespace current. Then open the database.

```
SQL> recover database;
```

ORA-00279: change 153845 generated at 10/31/2001

23:12:23 needed for thread 1

ORA-00289: suggestion : /oracle/admin/tst9/arch/archtst9\_12.log

ORA-00280: change 153845 for thread 1 is in sequence #12

Specify log: {<RET>=suggested | filename | AUTO | CANCEL}

Log applied.

Media recovery complete.

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

Database altered.

8. Verify that the INDX tablespace and its associated data file have been moved from file-system /db01/oracle/tst9 to /db02/oracle/tst9.

```
SQL> select name, status from v$datafile;
```

| NAME                           | STATUS |
|--------------------------------|--------|
| -----                          | -----  |
| /db01/oracle/tst9/system01.dbf | SYSTEM |
| /db01/oracle/tst9/rbs01.dbf    | ONLINE |
| /db01/oracle/tst9/temp01.dbf   | ONLINE |
| /db01/oracle/tst9/users01.dbf  | ONLINE |
| /db02/oracle/tst9/tools01.dbf  | ONLINE |
| /db01/oracle/tst9/data01.dbf   | ONLINE |
| /db02/oracle/tst9/indx01.dbf   | ONLINE |
| /db02/oracle/tst9/data02.dbf   | ONLINE |

8 rows selected.

```
SQL>
```

## 5. Describe Read-Only Tablespace Recovery

There are three scenarios that can occur with read-only tablespace recovery. These are as follows:

- Read-only backup and read-only recovery
- Read-only backup and read-write recovery
- Read-write backup and read-only recovery

The first scenario is the most straightforward because no recovery is needed. The SCN does not change because the tablespace is read-only. In this type of recovery, the only activity you need to do is restore the data files associated with the read-only tablespaces; thus no archived logs need to be applied.

The second scenario will require a more complex recovery process because the tablespace is being recovered to a read-write state in which the SCN number has changed or transactions have been made in the tablespace. In this case, you would restore the tablespace from backup and apply archived logs from the point at which the table was made read-write.

The final scenario will also require recovery because the tablespace is restored in a read-write state and then recovered to read-only. In this case, you will need to restore the backup of the tablespace in read-write mode and apply archived logs up to the point where the tablespace was made read-only. You should always perform a backup after making a tablespace read-only because doing so eliminates the need to restore the tablespace.

## 6. Using RMAN for Restoration and Recovery

The restore and recovery considerations for using RMAN consist of how you will restore databases, tablespaces, data files, control files, and archived logs from RMAN. Restores and recoveries can be performed from backups on both disk and tape devices.

There are two main backup sources that can be the basis for the RMAN recovery process. These sources are image copies and backup sets. *Image copies* can be stored only on disk. Image copies are actual copies of the database files, archived logs, or control files and are not stored in a special RMAN format. An image copy in RMAN is equivalent to an OS copy command, such as cp or dd in Unix, or COPY in Windows NT/2000/XP.

In Oracle9i, the RESTORE command will determine the best available backup set or image copy to use in the restoration and the file will only be restored if a restoration is necessary. In prior Oracle versions, the files were always restored, even if it wasn't necessary.

The RECOVER command applies the necessary changes from the online redo logs and archived log files to recover the restored files. If you are using incremental backups, the online redo logs and archived log files will be applied to recover the database.

## 7. Performing RMAN Recovery in NOARCHIVE Mode

As the first example of using RMAN for restores and recoveries, you will restore a database in NOARCHIVELOG mode. To restore a database in this mode, you must first make sure that the database was shut down cleanly so that you are sure to get a consistent backup. This means the database should be shut down with a SHUTDOWN NORMAL, IMMEDIATE, or TRANSACTIONAL command, but the ABORT command should not be used. The database should then be started in MOUNT mode, but not opened. This is because the database files cannot be backed up when the database is opened and not in ARCHIVELOG mode.

Next, while in the RMAN utility, you must connect to the target database, which in our example, is tst9 in the Unix environment. Then you can connect to the recovery catalog in the rcat database.

Once you are connected to the proper target and catalog, you can execute the appropriate RMAN backup script. This script will back up the entire database. After this has been done, the database can be restored with the appropriate RMAN script. Finally, the database can be opened for use.

Let's walk through this example:

1. Set the ORACLE\_SID to tst9, which is your target database, so that the database can be started in MOUNT mode with SQL\*Plus.

```
oracle@octilli:~ > oraenv
ORACLE_SID = [tst9] ?
oracle@octilli:~ > sqlplus /nolog
SQL*Plus: Release 9.0.1.0.0 - Production on Mon Oct 29 23:36:19 2001
(c) Copyright 2001 Oracle Corporation. All rights reserved.
SQL> connect /as sysdba
Connected to an idle instance.
SQL> startup mount
ORACLE instance started.
Total System Global Area 75854976 bytes
Fixed Size 279680 bytes
Variable Size 71303168 bytes
Database Buffers 4194304 bytes
Redo Buffers 77824 bytes
SQL>
```

2. Start the RMAN utility at the command prompt and connect to the target and the recovery catalog database rcat.

```
oracle@octilli:~ > rman
Recovery Manager: Release 9.0.1.0.0 - Production
(c) Copyright 2001 Oracle Corporation. All rights reserved.
RMAN> connect target
connected to target database: tst9 (not mounted)
RMAN> connect catalog rman/rman@rcat
connected to recovery catalog database
RMAN>
```

3. Once you are connected to the target and recovery catalog, you can back up the target database to disk or tape. In this example, you choose disk. You give the database name a format of backupset unique identifier, and then you concatenate to the database name with the backupset number.

```
RMAN> run
```

```
2> {
3> allocate channel c1 type disk;
4> backup database format 'db_%u_%d_%s';
5> release channel c1;
6> }
```

allocated channel: c1

channel c1: sid=11 devtype=DISK

Starting backup at 29-OCT-01

channel c1: starting full datafile backupset

channel c1: specifying datafile(s) in backupset

including current controlfile in backupset

input datafile fno=00001 name=/db01/oracle/tst9/system01.dbf

input datafile fno=00006 name=/db01/oracle/tst9/data01.dbf

input datafile fno=00002 name=/db01/oracle/tst9/rbs01.dbf

input datafile fno=00008 name=/db01/oracle/tst9/data02.dbf

input datafile fno=00003 name=/db01/oracle/tst9/temp01.dbf

input datafile fno=00004 name=/db01/oracle/tst9/users01.dbf

input datafile fno=00007 name=/db01/oracle/tst9/indx01.dbf

input datafile fno=00005 name=/db01/oracle/tst9/tools01.dbf

channel c1: starting piece 1 at 29-OCT-01

channel c1: finished piece 1 at 29-OCT-01

piece handle=/oracle/product/9.0.1/dbs/db\_0jd7r8e3\_

TST9\_19 comment=NONE

channel c1: backup set complete, elapsed time: 00:01:57

Finished backup at 29-OCT-01

released channel: c1

4. Once the backup has completed, the database may be restored. It must be mounted but not opened. In the restore script, choose three disk channels to utilize



parallelization of the restore process. The RESTORE DATABASE command is responsible for the restore process within RMAN. No recovery is required because the database was in NOARCHIVELOG mode and the complete database was restored.

```
RMAN> run
{
allocate channel c1 type disk;
allocate channel c2 type disk;
allocate channel c3 type disk;
restore database;
}
```

5. Once the database has been restored, it can be opened and then shut down normally. At this point, a startup should be performed to make sure the restore process was successful.

```
SQL> alter database open;
Database altered.
SQL> shutdown
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup
ORACLE instance started.
Total System Global Area 75854976 bytes
Fixed Size 279680 bytes
Variable Size 71303168 bytes
Database Buffers 4194304 bytes
Redo Buffers 77824 bytes
Database mounted.
Database opened.
SQL>
```

## 8. Performing RMAN Complete Recovery in ARCHIVELOG Mode

As the second example of using RMAN for restores and recoveries, you will restore a database in ARCHIVELOG mode. In this case, the database can be mounted or opened. This is because the database files can be backed up when the database is opened and in ARCHIVELOG mode in a similar manner to the way the user-managed ALTER TABLESPACE <tablespace\_name> BEGIN BACKUP command is used.

To perform this, you must connect to the target database (tst9 in the Unix environment in our example). Then you can connect to the recovery catalog in the rcat database.

Once you are connected to the proper target and catalog, you can execute the appropriate RMAN backup script. This script will back up the entire database. After this is done, you can restore the database with the appropriate RMAN script and then open it for use. Let's walk through this example:

1. Set the ORACLE\_SID to tst9, which is your target database, so that the database can be started in MOUNT mode with SQL\*Plus.

```
oracle@octilli:~ > oraenv
```

```
ORACLE_SID = [tst9] ?
```

```
oracle@octilli:~ > sqlplus /nolog
```

```
SQL*Plus: Release 9.0.1.0.0 - Production on Mon Oct 29 23:36:19 2001
```

```
(c) Copyright 2001 Oracle Corporation. All rights reserved.
```

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

```
Connected to an idle instance.
```

```
SQL> startup mount
```

```
ORACLE instance started.
```

```
Total System Global Area 75854976 bytes
```

```
Fixed Size 279680 bytes
```

```
Variable Size 71303168 bytes
```

```
Database Buffers 4194304 bytes
```

```
Redo Buffers 77824 bytes
```

```
SQL>
```

2. Start the RMAN utility at the command prompt and connect to the target and the recovery catalog database rcat.

```
oracle@octilli:~ > rman
```

Recovery Manager: Release 9.0.1.0.0 - Production

(c) Copyright 2001 Oracle Corporation. All rights reserved.

```
RMAN> connect target
```

```
connected to target database: tst9 (not mounted)
```

```
RMAN> connect catalog rman/rman@rcat
```

```
connected to recovery catalog database
```

```
RMAN>
```

3. Once you are connected to the target and recovery catalog, you can back up the target database, including archived logs, to disk or tape. In this example, you choose disk. You give the database name a format of `db_%u_%d_%s`, which means that a `db_` will be concatenated to the backupset unique identifier and then concatenated to the database name with the backupset number.

```
RMAN> run
```

```
2> {
```

```
3> allocate channel c1 type disk;
```

```
4> backup database format 'db_%u_%d_%s';
```

```
5> backup format 'log_t%t_s%s_p%p'
```

```
6> (archivelog all);
```

```
7> }
```

```
allocated channel: c1
```

```
channel c1: sid=11 devtype=DISK
```

```
Starting backup at 30-OCT-01
```

```
channel c1: starting full datafile backupset
```

```
channel c1: specifying datafile(s) in backupset
```

```
including current controlfile in backupset
```

```
input datafile fno=00001 name=/db01/oracle/tst9/system01.dbf
```

```
input datafile fno=00006 name=/db01/oracle/tst9/data01.dbf
```

```
input datafile fno=00002 name=/db01/oracle/tst9/rbs01.dbf
```

```
input datafile fno=00008 name=/db01/oracle/tst9/data02.dbf
```

```
input datafile fno=00003 name=/db01/oracle/tst9/temp01.dbf
```

```
input datafile fno=00004 name=/db01/oracle/tst9/users01.dbf
input datafile fno=00007 name=/db01/oracle/tst9/indx01.dbf
input datafile fno=00005 name=/db01/oracle/tst9/tools01.dbf
channel c1: starting piece 1 at 30-OCT-01
channel c1: finished piece 1 at 30-OCT-01
piece handle=/oracle/product/9.0.1/dbs/db_0kd7tqts_
TST9_20 comment=NONE
channel c1: backup set complete, elapsed time: 00:01:48
Finished backup at 30-OCT-01
Starting backup at 30-OCT-01
current log archived
channel c1: starting archive log backupset
channel c1: specifying archive log(s) in backup set
input archive log thread=1 sequence=1 recid=8
stamp=442361669
input archive log thread=1 sequence=2 recid=9
stamp=442361872
input archive log thread=1 sequence=3 recid=10
stamp=442362056
input archive log thread=1 sequence=4 recid=11
stamp=442362297
input archive log thread=1 sequence=5 recid=12
stamp=442362415
input archive log thread=1 sequence=6 recid=13
stamp=442792220
input archive log thread=1 sequence=7 recid=14
stamp=443231077
input archive log thread=1 sequence=8 recid=15
stamp=444439517
input archive log thread=1 sequence=9 recid=16
```

```
stamp=444512889
input archive log thread=1 sequence=10 recid=17
stamp=444525609
channel c1: starting piece 1 at 30-OCT-01
channel c1: finished piece 1 at 30-OCT-01
piece handle=/oracle/product/9.0.1/dbs/log_+444525610_
s21_p1 comment=NONE
channel c1: backup set complete, elapsed time: 00:00:04
Finished backup at 30-OCT-01
released channel: c1
```

4. Once the backup has completed, the database may be restored and recovered. The database must be mounted but not opened. In the restore and recovery script, choose three disk channels to utilize parallelization of the restore process. This is not necessary, but it improves the restore and recovery time. The RESTORE DATABASE command is responsible for the restore process within RMAN; this command is required because the database was in ARCHIVELOG mode and these files need to be applied to the data files in order for a complete recovery to be performed. Finally, the database is opened.

```
RMAN> run
2> {
3> allocate channel c1 type disk;
4> allocate channel c2 type disk;
5> allocate channel c3 type disk;
6> restore database;
7> recover database;
8> alter database open;
9> }

allocated channel: c1
channel c1: sid=11 devtype=DISK
allocated channel: c2
channel c2: sid=12 devtype=DISK
```

```
allocated channel: c3
channel c3: sid=13 devtype=DISK
Starting restore at 30-OCT-01
channel c1: starting datafile backupset restore
channel c1: specifying datafile(s) to restore from backup set
restoring datafile 00001 to /db01/oracle/tst9/system01.dbf
restoring datafile 00002 to /db01/oracle/tst9/rbs01.dbf
restoring datafile 00003 to /db01/oracle/tst9/temp01.dbf
restoring datafile 00004 to /db01/oracle/tst9/users01.dbf
restoring datafile 00005 to /db01/oracle/tst9/tools01.dbf
restoring datafile 00006 to /db01/oracle/tst9/data01.dbf
restoring datafile 00007 to /db01/oracle/tst9/indx01.dbf
restoring datafile 00008 to /db01/oracle/tst9/data02.dbf
channel c1: restored backup piece 1
piece handle=/oracle/product/9.0.1/dbs/db_0kd7tqts_
TST9_20 tag=null params=NULL
channel c1: restore complete
Finished restore at 30-OCT-01
Starting recover at 30-OCT-01
starting media recovery
media recovery complete
Finished recover at 30-OCT-01
database opened
released channel: c3
released channel: c1
released channel: c2
RMAN>
```

5. Once the database has been restored, recovered, and opened, it should be shut down normally. A startup should be performed to make sure the restore process was successful.

```
SQL> shutdown
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup
ORACLE instance started.
Total System Global Area 75854976 bytes
Fixed Size 279680 bytes
Variable Size 71303168 bytes
Database Buffers 4194304 bytes
Redo Buffers 77824 bytes
Database mounted.
Database opened.
SQL>
```

## 9. Using RMAN to Restore Data Files to Different Locations

As the third example, you will restore and recover a data file by using RMAN. In this case, the database will also be in ARCHIVELOG mode because an individual data file will be backed up. As in the previous tablespace example, the database will be backed up while it is open.

First, within RMAN, you must perform the appropriate data file backup script. For this example, you will select the data file for the DATA tablespace. You will back up the current control file as an extra precaution.

Once the data file is backed up, you can begin the restore and recovery process. For this process, the database should be mounted, but not open. You will also need to use the SET NEWNAME command to identify the new data file location, and the SWITCH command to record the location change in the control file. With the database mounted, you can execute the appropriate RMAN script to restore and recover the data file. The steps are as follows:

1. Set ORACLE\_SID to ORCL, which is your target database, so that the database can be started or mounted with SQL\*Plus.

```
oracle@octilli:~ > oraenv
```

```
ORACLE_SID = [tst9] ?
```

2. Connect to RMAN, the target database, and the recovery catalog in one step.

```
oracle@octilli:~ > rman target / catalog rman/rman@rcat
Recovery Manager: Release 9.0.1.0.0 - Production
(c) Copyright 2001 Oracle Corporation. All rights reserved.
connected to target database: TST9 (DBID=1268700551)
connected to recovery catalog database
```

3. Run the appropriate RMAN script to back up the DATA data file to disk.

```
RMAN> run
2> {
3> allocate channel ch1 type disk;
4> backup
5> format '%d_%u'
6> (datafile '/db01/oracle/tst9/data02.dbf');
7> release channel ch1;
8> }
```

```
allocated channel: ch1
channel ch1: sid=12 devtype=DISK
allocated channel: ch1
channel ch1: sid=12 devtype=DISK
Starting backup at 30-OCT-01
channel ch1: starting full datafile backupset
channel ch1: specifying datafile(s) in backupset
input datafile fno=00008 name=/db01/oracle/tst9/data02.dbf
channel ch1: starting piece 1 at 30-OCT-01
channel ch1: finished piece 1 at 30-OCT-01
piece handle=/oracle/product/9.0.1/dbs/TST9_Ond7tstb
comment=NONE
channel ch1: backup set complete, elapsed time:
00:00:01
Finished backup at 30-OCT-01
```



released channel: ch1

RMAN>

4. Once the data file has been backed up, you can restore and recover the data file with the appropriate RMAN script. The RMAN script uses the SET NEWNAME command to designate the new location of the data file that will be relocated; then database will be restored. Next, the SWITCH command will record the location change in the control file. Finally, the database will be recovered and opened.

RMAN> run

2> {

3> set newname for datafile 8 to '/db02/oracle/tst9/  
data02.dbf';

4> restore database;

5> switch datafile all;

6> recover database;

7> alter database open;

8> }

executing command: SET NEWNAME

Starting restore at 30-OCT-01

allocated channel: ORA\_DISK\_1

channel ORA\_DISK\_1: sid=11 devtype=DISK

channel ORA\_DISK\_1: starting datafile backupset restore

channel ORA\_DISK\_1: specifying datafile(s) to restore

from backup set

restoring datafile 00001 to /db01/oracle/tst9/system01.dbf

restoring datafile 00002 to /db01/oracle/tst9/rbs01.dbf

restoring datafile 00003 to /db01/oracle/tst9/temp01.dbf

restoring datafile 00004 to /db01/oracle/tst9/users01.dbf

restoring datafile 00005 to /db01/oracle/tst9/tools01.dbf

restoring datafile 00006 to /db01/oracle/tst9/data01.dbf

restoring datafile 00007 to /db01/oracle/tst9/indx01.dbf

```
channel ORA_DISK_1: restored backup piece 1
piece handle=/oracle/product/9.0.1/dbs/db_0kd7tqts_
TST9_20 tag=null params=NULL
channel ORA_DISK_1: restore complete
channel ORA_DISK_1: starting datafile backupset restore
channel ORA_DISK_1: specifying datafile(s) to restore
from backup set
restoring datafile 00008 to /db02/oracle/tst9/data02.dbf
channel ORA_DISK_1: restored backup piece 1
piece handle=/oracle/product/9.0.1/dbs/TST9_0nd7tstb
tag=null params=NULL
channel ORA_DISK_1: restore complete
Finished restore at 31-OCT-01
datafile 8 switched to datafile copy
input datafilecopy recid=32 stamp=444528057 filename=/
db02/oracle/tst9/data02.df
starting full resync of recovery catalog
full resync complete
Starting recover at 31-OCT-01
using channel ORA_DISK_1
starting media recovery
media recovery complete
Finished recover at 31-OCT-01
database opened
RMAN>
```

5. Once the database has been restored, shut it down normally. Then perform a startup to make sure the restore process was completed.

```
SQL> shutdown
```

```
SQL> startup
```

## 10. Use RMAN to Relocate and Recover a Tablespace Using Archived Logs

In this example, you will relocate a tablespace to a new file-system during recovery. You can perform this using the SET NEWNAME and SWITCH commands that were mentioned earlier. In addition, the RECOVER command applies the necessary backup of data files and archived logs. The major difference between this process and that of relocating a data file is that the tablespace needs to be taken offline before the associated data files can be moved to a new location. The database, however, can be opened during this process. Below is an example of this procedure.

1. Set ORACLE\_SID to ORCL, which is your target database, so that the database can be started or mounted with SQL\*Plus.

```
oracle@octilli:~ > oraenv
ORACLE_SID = [tst9] ?
```

2. Connect to RMAN, the target database, and the recovery catalog in one step.

```
oracle@octilli:~ > rman target / catalog rman/rman@rcat
Recovery Manager: Release 9.0.1.0.0 - Production
(c) Copyright 2001 Oracle Corporation. All rights reserved.
connected to target database: TST9 (DBID=1268700551)
connected to recovery catalog database
```

3. Run the appropriate RMAN script to back up the tst9 database to disk.

```
RMAN> run
2> {
3> allocate channel c1 type disk;
4> backup database format 'db_%u_%d_%s';
5> backup format 'log_t%t_s%s_p%p'
6> (archivelog all);
7> }
```

4. Then issue the recovery script, which will utilize the SET NEWNAME, RESTORE, SWITCH, and RECOVER commands. Finally, bring the tablespace online.

```
RMAN> run
2> {
```

```
3> sql 'alter tablespace tools offline immediate';
4> set newname for datafile '/db01/oracle/tst9/tools01.dbf' to
5> '/db02/oracle/tst9/tools01.dbf';
6> restore (tablespace tools);
7> switch datafile 5;
8> recover tablespace tools;
9> sql 'alter tablespace tools online';}
```

sql statement: alter tablespace tools offline immediate

executing command: SET NEWNAME

Starting restore at 31-OCT-01

allocated channel: ORA\_DISK\_1

channel ORA\_DISK\_1: sid=11 devtype=DISK

channel ORA\_DISK\_1: starting datafile backupset restore

channel ORA\_DISK\_1: specifying datafile(s) to restore from backup set

restoring datafile 00005 to /db02/oracle/tst9/tools01.dbf

channel ORA\_DISK\_1: restored backup piece 1

piece handle=/oracle/product/9.0.1/dbs/db\_0kd7tqts\_

TST9\_20 tag=null params=NULL

channel ORA\_DISK\_1: restore complete

Finished restore at 31-OCT-01

datafile 5 switched to datafile copy

input datafilecopy recid=34 stamp=444610301 filename=/

db02/oracle/tst9/tools01.f

starting full resync of recovery catalog

full resync complete

Starting recover at 31-OCT-01

using channel ORA\_DISK\_1

starting media recovery

media recovery

```
archive log thread 1 sequence 16 is already on disk as
file /oracle/admin/tst9/arch/archtst9_16.log
archive log thread 1 sequence 17 is already on disk as
file /oracle/admin/tst9/arch/archtst9_17.log
archive log thread 1 sequence 18 is already on disk as
file /oracle/admin/tst9/arch/archtst9_18.log
archive log
filename=/oracle/admin/tst9/arch/archtst9_16.log
thread=1 sequence=6
media recovery complete
Finished recover at 31-OCT-01
sql statement: alter tablespace tools online
RMAN>
```

## 11. Summary

In this lesson, we emphasized media recoveries. We described the two methods of performing Oracle database recovery for media failures (user-managed and RMAN-based recoveries) and we performed specific examples of each.

In addition, we identified the differences between ARCHIVELOG mode and NOARCHIVELOG mode and we described the significant implications that each mode has on the backup and recovery process. We also showed examples of both ARCHIVELOG mode and NOARCHIVELOG mode recoveries using both user-managed and RMAN methods of recovery.

We then discussed read-only tablespace recovery and the three recovery scenarios associated with it. Each of these scenarios requires different recovery actions. We also performed both user-managed and RMAN-based recovery situations in which file relocation was required.

Media recoveries are an important topic in testing and in real work situations. How media recovery situations are handled depends on the confidence of the DBA performing the media recovery. You can obtain confidence by practicing media recoveries in all of the above-mentioned situations.

Then, when you need to perform a media recovery, your uncertainties will have been significantly reduced. Such practice situations will also prepare you for testing and situations you will encounter as a DBA.

## References

- [1] Oracle9i DBA Fundamentals II.