

## Practice 14

# Performing Recovery Part IV

### Practice Target

In this practice you will perform further recovery scenarios in ORADB database.

### Practice Overview

In this practice, you will implement the following recovery tasks:

- Restore SPFILE from memory
- Restore SPFILE from **AUTOBACKUP**
- Recover from losing subset of control files
- Recover from losing all control files
- Recover objects with **NOLOGGING** operations

### Assumptions

This practice assumes the `srv1` appliance is up and running and its database `ORADB` is running in `OPEN` state.

### Pre-requisites

Take a **snapshot** of `srv1` appliance. Name the snapshot "**Practice 14 Start**".

### Caution!

Do not implement the practice without creating the snapshot as instructed by the previous step.

## Recovery Scenario 9:

### Restore the SPFILE from Memory

#### Scenario Target:

- In this scenario, we assume the SPFILE is lost while the database is open. The recovery aims at restoring the SPFILE.

#### Simulating the data loss

1. Start SQL\*Plus and login to ORADB as sysdba.

```
sqlplus / as sysdba
```

2. Display the current full filename of SPFILE and take a note of it.

```
show parameter SPFILE
```

3. Delete the SPFILE.

```
host rm /u01/app/oracle/product/12.2.0/db_1/dbs/spfileORADB.ora
```

4. In SQL\*Plus session, verify that the database is still running.

```
SELECT COUNT(*) FROM DBA_OBJECTS ;
```

Now, the question is: how can you know that the SPFILE has been lost and that the database is running without one? Technically speaking, unless you're specifically monitoring the file, you will not know about the SPFILE loss until you try accessing the SPFILE or starting up the database. We will cover in this section of the practice the scenario where you know about the SPFILE loss when trying to access it.

5. Try to change a parameter value in the SPFILE.

You will receive the error "ORA-27037: unable to obtain file status".

```
ALTER SYSTEM SET NLS_DATE_FORMAT='YYYY-MM-DD:HH24:MI:SS' SCOPE=SPFILE;
```

#### Recovery Actions

Perform the following actions to restore the SPFILE while the database is still running.

6. Create a copy of the SPFILE from memory. You cannot create the default SPFILE name even though it is deleted.

```
CREATE SPFILE = '/u01/app/oracle/product/12.2.0/db_1/dbs/spfileORADB.staging' FROM MEMORY;
```

7. Shutdown the database. Unfortunately, recovering SPFILE requires restarting the database.

```
SHUTDOWN IMMEDIATE
```

8. Rename the SPFILE that you created to the default SPFILE full name.

```
host mv /u01/app/oracle/product/12.2.0/db_1/dbs/spfileORADB.staging  
/u01/app/oracle/product/12.2.0/db_1/dbs/spfileORADB.ora
```

9. Start up the database.

```
STARTUP
```



## Recovery Scenario 10:

### Restore the SPFILE from AUTOBACKUP

#### Scenario Target:

- In this scenario, we assume the SPFILE is lost and the database is down at the time of recovery.

#### Scenario assumptions:

- The database is running in ARCHIVELOG mode
- Backup of the database is available and AUTOBACKUP is configured

#### Preparing for the scenario

10. Invoke RMAN with connecting to ORADB as target and **obtain the DBID value** and take a note of it.

```
rman target /
```

11. Take a full backup of the database as backupset.

```
BACKUP DATABASE TAG 'DB_FULL';  
ALTER SYSTEM SWITCH LOGFILE;
```

12. Invoke SQL\*Plus and obtain the FRA destination. Take a note of it.

```
SHOW PARAMETER DB_RECOVERY_FILE_DEST
```

#### Simulating the data loss

13. Delete the SPFILE.

```
host rm /u01/app/oracle/product/12.2.0/db_1/dbs/spfileORADB.ora
```

14. In the SQL\*Plus session, shutdown the database and try opening it again.

The following error will be reported when you startup the database:

```
LRM-00109: could not open parameter file
```

When you restarted the database, you discovered that the SPFILE is lost. You, therefore, decided to recover it. The recovery procedure that you implemented in the previous section cannot be used in this scenario because the database is down.

```
SHUTDOWN IMMEDIATE  
STARTUP
```

**Recovery Actions**

Perform the following action to restore the SPFILE from the most recent `AUTOBACKUP` file

- 15.** Start RMAN with connecting to the local instance as target then execute the commands that follows:

```
rman target /  
  
STARTUP FORCE NOMOUNT;  
RUN{  
  SET DBID <dbid>;  
  RESTORE SPFILE FROM AUTOBACKUP  
  DB_RECOVERY_FILE_DEST='/u01/app/oracle/fra/ORADB'  
  DB_NAME='ORADB';  
}
```

- 16.** Restart the database.

```
SHUTDOWN IMMEDIATE  
STARTUP
```

## Recovery Scenario 11:

### Recovery from the Loss of Subset of the Control Files

#### Scenario Target:

- In this scenario, we assume that subset of the multiplexed control files is lost. In other words, at least one control file is still available.

When one or more of the multiplexed control files is lost, the database instance may or may not abort.

#### Scenario assumptions:

- The database is running in ARCHIVELOG mode
- Backup of the database is available

#### Preparing for the scenario

17. Invoke SQL\*Plus and obtain the control files in the database. Take a note of the command output.

```
sqlplus / as sysdba
SHOW PARAMETER CONTROL_FILES
```

#### Simulating the data loss

18. Delete **one** of the control files.

```
host rm -f /u01/app/oracle/fra/ORADB/ORADB/controlfile/<name of the control file>
```

19. In the SQL\*Plus session, run the following query to check on the status of the instance.

Although the instance is still running, so many SQL commands fail when a control file is lost.

```
SELECT * FROM V$VERSION;
SELECT OPEN_MODE FROM V$DATABASE;
```

#### Recovery Actions

Perform the following action to restore the lost control file from one of the surviving control files.

20. Abort the database.

You cannot cleanly shut down the database when a control file is not available.

```
SHUTDOWN ABORT
```

21. Copy the intact control file to the missing control file.

```
host cp /u01/app/oracle/oradata/ORADB/controlfile/<available control file>
/u01/app/oracle/fra/ORADB/ORADB/controlfile/<missing control file>
```

22. Startup the database

```
STARTUP
```

## Recovery Scenario 12:

### Restore the Loss of All Control Files

#### Scenario Target:

- In this scenario, we assume that all control files are lost.

#### Scenario assumptions:

- The database is running in ARCHIVELOG mode
- Backup of the database is available and AUTOBACKUP is configured
- AUTOBACKUP format is set to the default in FRA
- Online redo logs are available

#### Preparing for the scenario

- 23.** Invoke RMAN, connect to local database and obtain the DBID.

```
rman target /
```

- 24.** If you lost the control file full names obtained in previous section, invoke SQL\*Plus and obtain the control file full names in the database. Take a note of the command output.

```
sqlplus / as sysdba  
SHOW PARAMETER CONTROL_FILES
```

#### Simulating the data loss

- 25.** Delete all the control files.

```
host rm -f /u01/app/oracle/fra/ORADB/ORADB/controlfile/<first control file>.ctl  
host rm -f /u01/app/oracle/oradata/ORADB/controlfile/<second control file>.ctl
```

- 26.** In the SQL\*Plus session, run the following queries to check on the status of the instance.

```
host ps -ef|grep smon  
SELECT * FROM V$VERSION;  
SELECT OPEN_MODE FROM V$DATABASE;
```

- 27.** Exit from SQL\*Plus, invoke RMAN and try connecting to the local database as target.  
RMAN should fail to startup because the control file is not available.

```
rman target /
```

- 28.** Check out the last lines of the alertlog file.

```
vi /u01/app/oracle/diag/rdbms/oradb/ORADB/trace/alert_ORADB.log
```

### Recovery Actions

Perform the following actions to restore the lost control files.

- 29.** Invoke SQL\*Plus, connect to local database as sysdba and abort the database.

```
SHUTDOWN ABORT
```

- 30.** Exit from SQL\*Plus, invoke RMAN and connect to the local database instance as target.

```
rman target /
```

- 31.** Execute the following commands

```
SET DBID <dbid>;  
STARTUP NOMOUNT  
RESTORE CONTROLFILE FROM AUTOBACKUP;  
SHUTDOWN  
STARTUP MOUNT  
RECOVER DATABASE;  
ALTER DATABASE OPEN RESETLOGS;
```

### Clean Up

- 32.** Shutdown `srv1` and restore it to the snapshot "**Practice 14 Start**".



## Recovery Scenario 13:

### Recovering NOLOGGING Operations

#### Scenario Target:

If you have an object that received data using `NOLOGGING` operation after the last backup was taken, you cannot recover that object from the last backup until you include it in a new backup after the `NOLOGGING` operation.

If the object is affected by `NOLOGGING` operation and is not included in the last backup after the operation is finished, the only way to recover it is to recreate it again.

- In this scenario, you will examine the recovery scenario on objects containing data inserted by `NOLOGGING` operations.

#### Scenario assumptions:

- The database is running in `ARCHIVELOG` mode
- Backup of the database is available

#### Preparing for the scenario

33. Invoke RMAN, connect to local database, and take online backup of the entire database.

```
rman target ''/ as SYSBACKUP''  
BACKUP DATABASE TAG 'FULL_DB';  
ALTER SYSTEM SWITCH LOGFILE;
```

#### Simulating the data loss

34. Invoke SQL\*Plus and connect as `soe` user

```
sqlplus soe/soe@oradb
```

35. Create the following reporting table.

```
CREATE TABLE RPT_ORDER_MONTH NOLOGGING AS  
SELECT SUM(ORDER_TOTAL) ORDERS_SUM, TO_CHAR(ORDER_DATE, 'YYYY-MM') ORDER_MONTH  
FROM ORDERS  
GROUP BY TO_CHAR(ORDER_DATE, 'YYYY-MM')  
ORDER BY TO_CHAR(ORDER_DATE, 'YYYY-MM');
```

Now, assume we faced a situation that requires to restore and recover the database.

#### Recovery Actions

36. Invoke RMAN and connect to the local database instance as target.

```
rman target ''/ as SYSBACKUP''
```

- 37.** Execute the following commands to recover the entire database.

```
SHUTDOWN IMMEDIATE
STARTUP MOUNT
RESTORE DATABASE;
RECOVER DATABASE;
ALTER DATABASE OPEN;
```

- 38.** Invoke SQL\*Plus, connect as `soe` user, and try accessing the table.

The query should report the error "ORA-01578: ORACLE data block corrupted (file# *m*, block# *n*)".

```
sqlplus soe/soe@oradb
SELECT * FROM RPT_ORDER_MONTH;
```

- 39.** The only way to recover the table is to re-create it.

That is why if a production database includes objects with NOLOGGING operations that should be recovered, you should take backup of them after the NOLOGGING operations are finished.

```
DROP TABLE RPT_ORDER_MONTH PURGE;

CREATE TABLE RPT_ORDER_MONTH NOLOGGING AS
SELECT SUM(ORDER_TOTAL) ORDERS_SUM, TO_CHAR(ORDER_DATE, 'YYYY-MM') ORDER_MONTH
FROM ORDERS
GROUP BY TO_CHAR(ORDER_DATE, 'YYYY-MM')
ORDER BY TO_CHAR(ORDER_DATE, 'YYYY-MM');

SELECT * FROM RPT_ORDER_MONTH;
```

## Clean Up

- 40.** Restore `srv1` to the snapshot "**Practice 14 Start**". Start `srv1`.
- 41.** Delete the snapshot "Practice 14 Start"

## Summary

In this practice, you performed the following recovery tasks:

- Restore SPFILE from memory
- Restore SPFILE from **AUTOBACKUP**
- Recover from losing subset of control files
- Recover from losing all control files
- Recover objects with **NOLOGGING** operations

