

# Oracle Database Disaster Recovery 19c

Majid Shabani

[majid.shabani@live.com](mailto:majid.shabani@live.com)

June 8, 2023

## **I will Do the Following Steps to Perform the Recovery Action:**

- Overview
- Recovery Prerequisites
- Recovery Consideration
- Creating Required Scripts
- Obtaining Backups Details
- Performing Restore Operation
- Monitoring Restore Operation
- Recovering Database & Monitoring it
- Performing Post Task after Recover Operation

## **Disclaimer**

Do not consider this document as a reference to restore a production database. The author is not responsible for any losses or damage resulted from using this article.

None of the advice is warranted to be free of errors. Please use this article at your own risk and after thorough testing in your environment.

If anyone wishes to correct information presented in this article, please message me.

## Overview:

In this article I will restore a full backup of the Oracle database with ASM release 19c non-CDB. To do so I've got to perform multiple steps. In the following I will explain each step in detail.

As mentioned, this article is related to restoring full Oracle database into a new server after disaster but you can use it when you want migrate the database from the current server to another new server. In addition to that if you use this article for CDB databases, you are unlikely to face any issue.

Before starting I'd like to tell you the backup is taken and saved using Veritas NetBackup which is a third-party tool. So, the used commands to restore and recover database are a little bit different from the case the backup is not taken using third party tools. And regarding the backup itself, a full backup scheduled to be taken every 2 days and backup of archive log files is scheduled to be taken every 4 hours.

This article divided into multiple sections, at first, I will create a bunch of scripts, second, I use those scripts to obtain required details of database backup, after that I will perform the restore operation, then I monitor restore operation using created scripts and finally I recover the database to the desired SCN and I will start monitoring it. When recover operation is finished, I will do some post action in the database host.

In the case that recovery catalog not used, you should follow method other than the one presented in this article to obtain information about the backup.

Finally, you may ask what about RTO and RPO, and I should say RPO in this case can be up to 4 hours because the archive log files were backed up every 4 hours and on the other hand RTO which includes problem identification, recovery planning, and recovery time may take about a week.

Please be aware, for some scripts I put sample output, just to shed some light on them. They are not real.

## Specification of oracle software products

<b>Source Oracle Home</b>	Oracle Grid Home	/u01/app/19.0.0/grid
	Oracle Database Home	/u01/app/oracle/product/19.0.0/db_1
<b>Target Oracle Home</b>	Oracle Grid Home	/u01/app/19.0.0/grid
	Oracle Database Home	/u01/app/oracle/product/19.0.0/db_1
<b>OS</b>	RHEL Linux	7.6

The backup scheduled to be taken with the following specification.

## Specification of backup

<b>Backup Strategy</b>	Whole database
<b>Backup Type</b>	Full Backup
<b>Backup Mode</b>	Online
<b>Used Tool</b>	RMAN on Data Domain using NetBackup
<b>If RMAN catalog DB used?</b>	Yes, used

To obtain more detail about the NetBackup, please use this [link](#).

## Recovery Prerequisites:

Before you can proceed with the restore procedure, you should perform the following tasks on the destination host:

- Make the backup files available to the destination host.
- Make the Keystore file (if any) available to the destination host and copy it to its default location.
- Configure the operating system users and permissions.
- Configure the operating system environment variables.
- The Oracle database software that you want to install in the new server should be same release as source database.
- A LISTENER must've created before starting.
- Create required disk groups same as source database.
- The NetBackup agent must be installed on the target host.
- The network must properly be configured to be able to connect to the NetBackup from the DB host.
- Also, the configuration at the master server must be done.

## Recovery Consideration

- Don't connect to recovery catalog if you are restoring the database into a new machine for testing purposes only, because the restored database has the same DBID as the source database.
- Replace name of backup pieces in this document with its correct one.
- The database will be restored in same directory structure and disk group name same as source server.
- The source database name is **proddb**

## Creating Required Scripts:

To be able to obtain information about the last backup of the database, you should use commands. In the following steps, I'll create them.

1. the following script named 'cat\_db\_detail.sql' and displays the database ID and DATABASE key of the given database name.

```
cat > cat_db_detail.sql<<EOL
/* Author: Majid Shabani */
SET LIN 500 VERIFY OFF
col name for a20

SELECT DBID, NAME, DB_KEY FROM RC_DATABASE WHERE UPPER(NAME) = UPPER('&1');
SET VERIFY ON
EOL

Usage:
cat_db_detail.sql <DB Name>
```

2. This script named 'cat\_bkp\_detail.sql' and shows the backup detail of the given database name for specified time period.

```
cat > cat_bkp_detail.sql <<EOL
/* Author: Majid Shabani */
SET LINES 300 PAGES 30 VERIFY OFF
COL SESSION_RECID FOR 9999999 HEADING S_RECID
COL STATUS FOR A9
COL START_TIME FORMAT A14
COL OUTPUT_DEVICE_TYPE FOR A8 HEADING DEVICE
COL END_TIME FOR A14
col IN_SIZE for a12
col OUT_SIZE for a12
col COMP for 99.9999
col IN_SPEED for a10
col OUT_SPEED for a10

SELECT DB_NAME, DB_KEY, SESSION_KEY, SESSION_STAMP, STATUS, INPUT_TYPE,
       STARTING_TIME START_TIME, OUTPUT_DEVICE_TYPE,
       END_TIME, HOURS, IN_SIZE, OUT_SIZE, IN_SPEED, OUT_SPEED, COMP
FROM (
  SELECT DB_NAME, DB_KEY, SESSION_KEY, SESSION_STAMP, STATUS, INPUT_TYPE,
         TO_CHAR(START_TIME, 'DD/MM/YY HH24:MI') STARTING_TIME,
         OUTPUT_DEVICE_TYPE, TO_CHAR(END_TIME, 'DD/MM/YY HH24:MI') END_TIME,
         ROUND(ELAPSED_SECONDS/60/60,1 )HOURS,
         INPUT_BYTES_DISPLAY IN_SIZE, OUTPUT_BYTES_DISPLAY OUT_SIZE,
         INPUT_BYTES_PER_SEC_DISPLAY IN_SPEED, OUTPUT_BYTES_PER_SEC_DISPLAY OUT_SPEED,
         ROUND(COMPRESSION_RATIO,4) COMP
  FROM RC_RMAN_BACKUP_JOB_DETAILS
  WHERE UPPER(DB_NAME) = UPPER('&1')
        AND TRUNC(START_TIME) >= TRUNC(SYSDATE) - &2
  ORDER BY START_TIME);
SET VERIFY ON
EOL

Usage:
cat_bkp_detail.sql <DB Name> <how long time in day>
```

**3. create the script named as 'cat\_list\_ctl\_bkps.sql'.**

It shows the backup set details of controlfile for the given DB name using the value obtained from SESSION\_KEY column of the previous script or by specifying time period.

```
cat > cat_list_ctl_bkps.sql <<EOL
/* Author: Majid Shabani */
SET LIN 500 PAGES 200 VERIFY OFF
COL TAG FOR A20
COL HANDLE FOR A25
COL START_TIME FOR A20
COL DEVICE_TYPE FOR A12
COL FILESIZE_DISPLAY FOR A15

SELECT SESSION_KEY, SESSION_STAMP, TO_CHAR(P.COMPLETION_TIME, 'DD/MM/YY
HH24:MI') CREATION_TIME, HANDLE, FILESIZE_DISPLAY,
       ROUND(ELAPSED_SECONDS) ELAPSED_SECONDS, STATUS, IS_RECOVERY_DEST_FILE,
CHECKPOINT_CHANGE# || '' CHECKPOINT_CHANGE
FROM RC_BACKUP_CONTROLFILE_DETAILS C, RC_BACKUP_PIECE P
WHERE C.ID1 = P.SET_STAMP
      AND C.ID2 = P.SET_COUNT
      AND C.BTYPE = 'BACKUPSET'
      AND UPPER(DB_NAME) = UPPER('&1')
      AND (SESSION_KEY = &2 OR TRUNC(START_TIME) >= TRUNC(SYSDATE) - &3)
ORDER BY COMPLETION_TIME;
SET VERIFY ON
EOL
```

**Usage:**

```
cat_list_ctl_bkps.sql <DB Name> <specific backup session key> null
OR
cat_list_ctl_bkps.sql <DB Name> null <how long time in day>
```

#### 4. Verify the controlfile backup piece contains spfile.

The script returns no row If the backup piece doesn't contain spfile.

```
cat > cat_list_spfile_bkps.sql <<EOL
/* Author: Majid Shabani */
SET LIN 500 PAGES 50 VERIFY OFF
COL HANDLE FOR A25
COL NAME FOR A65
COL TABLESPACE_NAME FOR A25
COL TS# FOR 9999
COL HOURS FOR A10
COL SPFILE_SIZE FOR A20

SELECT S.DB_NAME, S.DB_KEY, P.HANDLE,
DECODE(P.BACKUP_TYPE, 'D', 'FULL', 'L', 'ARCHBKP', 'I', 'DBINC') BACKUP_TYPE,
TO_CHAR(P.START_TIME, 'YYYY-MM-DD HH24:MI') START_TIME,
TO_CHAR(P.COMPLETION_TIME, 'DD/MM/YY HH24:MI') COMPLETION_TIME,
ROUND(P.BYTES/1024/1024) PIECE_SIZE_MB, TIME_TAKEN_DISPLAY HOURS,
ORIGINAL_INPRATE_BYTES_DISPLAY IN_SPEED, OUTPUT_RATE_BYTES_DISPLAY OUT_SPEED
, FILESIZE_DISPLAY SPFILE_SIZE, TO_CHAR(MODIFICATION_TIME, 'YYYY-MM-DD
HH24:MI:SS') MODIFICATION_TIME
FROM RC_BACKUP_SET_DETAILS S, RC_BACKUP_PIECE P,
RC_BACKUP_SPFILE_DETAILS SP
WHERE S.BS_KEY = P.BS_KEY
AND SP.SET_STAMP = P.SET_STAMP
AND SP.SET_COUNT = P.SET_COUNT
AND UPPER(S.DB_NAME) = UPPER('&1')
AND (S.SESSION_KEY = &2 OR P.HANDLE = '&3');
SET VERIFY ON
EOL

Usage:
cat_list_spfile_bkps.sql <DB Name> <specific backup session key> null
OR
cat_list_spfile_bkps.sql <DB Name> null <piece name>
```

5. The following script named as 'cat\_show\_arch\_scn.sql'.

It shows the archive log in the specified backup using of the value obtained from SESSION\_KEY column of the 'cat\_bkp\_detail.sql' script or by specifying time period.

```
cat > cat_show_arch_scn.sql <<EOL
/* Author: Majid Shabani */
SET LIN 500 PAGES 50 VERIFY OFF
COL HANDLE FOR A50
COL NEXT_CHANGE FOR A18
COL DEVICE_TYPE FOR A8
COL START_TIME FOR A20

SELECT SESSION_KEY, SESSION_STAMP, A.DB_KEY, DB_NAME, THREAD#, SEQUENCE#,
NEXT_CHANGE#||'|'NEXT_CHANGE,
       TO_CHAR(NEXT_TIME,'YYYY-MM-DD HH24:MI:SS')LAST_TIME, B.DEVICE_TYPE,
       HANDLE, TO_CHAR(B.START_TIME,'YYYY-MM-DD HH24:MI:SS')START_TIME,
       TO_CHAR(B.COMPLETION_TIME,'YYYY-MM-DD HH24:MI:SS')COMPLETION_TIME, B.STATUS
FROM RC_BACKUP_ARCHIVELOG_DETAILS A, RC_BACKUP_PIECE B
WHERE A.ID1 = B.SET_STAMP
      AND A.ID2 = B.SET_COUNT
      AND UPPER(DB_NAME) = UPPER('&1')
      AND (SESSION_KEY = &2 OR FIRST_TIME >= SYSDATE - &3)
ORDER BY NEXT_TIME;
SET VERIFY ON
EOL

Usage:
cat_list_ctl_bkps.sql <DB Name> <specific backup session key> null
OR
cat_list_ctl_bkps.sql <DB Name> null <how long time in day>
```

6. This script shows the number of pieces in the specified backup for specific database.

```
cat > cat_piece_count.sql <<EOL
/* Author: Majid Shabani */
SET VERIFY OFF
COMP SUM OF PIECE_CNT ON REPORT
BREAK ON REPORT

SELECT DECODE(BACKUP_TYPE,'D','Full Backup','I','Incremental Backup'
              ,'L','Backup of ArchiveLog')BACKUP_TYPE, COUNT(*) PIECE_CNT
FROM RC_BACKUP_SET_DETAILS
WHERE UPPER(DB_NAME)=UPPER('&1')
      AND SESSION_KEY=&2
GROUP BY DECODE(BACKUP_TYPE,'D','Full Backup','I','Incremental Backup','L','Backup of
ArchiveLog');
CLEAR BREAK
SET VERIFY ON
EOL

Usage:
cat_piece_count.sql <DB Name> <specific backup session key>
```



7. Also, you can obtain all information obtained so far by running the following script, so using this script alone is fair enough. It shows number of files in each backup piece, thread#, sequence# and SCN value in each archive log backup file and piece name in which controlfile is backed up.

```
cat > cat_bkpset_details.sql<<EOL
/* Author: Majid Shabani */
SET LIN 500 PAGES 10000 VERIFY OFF
COL STARTING_TIME FOR A19
COL COMPLETION_TIME FOR A22
COL HANDLE FOR A24
COL TIME_TAKEN_DISPLAY FOR A10
COL HOURS FOR A10
COL SIZE_GB FOR 999,999
COL FIRST_SEQ FOR 999999999
COL LAST_SEQ FOR 999999999
COL FIRST_SCN FOR 999999999999999
COL LAST_SCN FOR 999999999999999
COL LAST_TIME FOR A24
col CHECKPOINT_CHANGE for a20
COMP COUNT LABEL 'TPiece#:' -
      MAX LABEL 'MAXPS-G:' -
      SUM LABEL 'TSIZE_G:' OF SIZE_GB ON BACKUP_TYPE FOR A20
BREAK ON BACKUP_TYPE SKIP 2

SELECT DECODE(S.BACKUP_TYPE, 'D', 'FULL', 'L', 'ARCHBKP', 'I', 'DBINC') BACKUP_TYPE,
       TO_CHAR(S.START_TIME, 'DD/MM/YY DY HH24:MI') STARTING_TIME,
       TO_CHAR(S.COMPLETION_TIME, 'DD/MM/YY DY HH24:MI:SS') COMPLETION_TIME,
       S.TIME_TAKEN_DISPLAY, ROUND(P.BYTES/POWER(1024,3)) SIZE_GB,
       P.HANDLE, COUNT(*) FILE#, THREAD#,
       MIN(A.SEQUENCE#) FIRST_SEQ, MAX(A.SEQUENCE#) LAST_SEQ,
       MIN(A.NEXT_CHANGE#) FIRST_SCN, MAX(A.NEXT_CHANGE#) LAST_SCN,
       TO_CHAR(MAX(A.NEXT_TIME), 'YYYY-MM-DD DY HH24:MI:SS') LAST_TIME,
       MAX(C.CHECKPOINT_CHANGE#) || ' ' CHECKPOINT_CHANGE
FROM RC_BACKUP_SET_DETAILS S, RC_BACKUP_PIECE P,
RC_BACKUP_DATAFILE BD, RC_BACKUP_ARCHIVELOG_DETAILS A, RC_BACKUP_CONTROLFILE_DETAILS C
WHERE S.BS_KEY = P.BS_KEY
      AND BD.SET_STAMP (+) = P.SET_STAMP
      AND BD.SET_COUNT (+) = P.SET_COUNT
      AND A.btype_key(+) = P.BS_KEY
      AND C.ID1 (+) = P.SET_STAMP
      AND C.ID2 (+) = P.SET_COUNT
      AND C.BTYPE (+) = 'BACKUPSET'
      AND UPPER(S.DB_NAME) = UPPER('&1')
      AND S.SESSION_KEY = &2
GROUP BY DECODE(S.BACKUP_TYPE, 'D', 'FULL', 'L', 'ARCHBKP', 'I', 'DBINC'),
       TO_CHAR(S.START_TIME, 'DD/MM/YY DY HH24:MI'),
       TO_CHAR(S.COMPLETION_TIME, 'DD/MM/YY DY HH24:MI:SS'),
       S.TIME_TAKEN_DISPLAY, S.ORIGINAL_INPUT_BYTES_DISPLAY,
       S.OUTPUT_BYTES_DISPLAY, P.HANDLE, THREAD#, P.BYTES
ORDER BY STARTING_TIME, MAX(A.NEXT_TIME), LAST_SEQ;
CLEAR BREAK;
CLEAR COMP;
SET VERIFY ON;
PAUSE
EOL
```

**Usage:**

```
cat_bkpset_details.sql <DB Name> <specific backup session key>
```

8. Create the following scripts, to display the log of RMAN job saved in the catalog db.

```
cat > cat_bkp_log.sql <<EOL
/* Author: Majid Shabani */
SET PAGES 1000 VERIFY OFF
SELECT OUTPUT
FROM RC_RMAN_OUTPUT
WHERE DB_KEY = &1
      AND SESSION_KEY = &2
ORDER BY RECID;
SET VERIFY ON
EOL
```

**Usage:**

```
cat_bkp_log.sql <DB key> <specific backup session key>
```

9. And finally, the following scripts show the backup script saved in the catalog DB of the given database name.

```
cat > cat_db_scp.sql <<EOL
/* Author: Majid Shabani */
SET LIN 500 PAGES 200 VERIFY OFF
COL SCRIPT_NAME FOR A23
COL SCRIPT_BODY FOR A120
BREAK ON SCRIPT_NAME SKIP 2

SELECT NAME DBNAME, D.DB_KEY, DBID, SCRIPT_NAME, LINE, TEXT SCRIPT_BODY
FROM RC_DATABASE D, RC_STORED_SCRIPT_LINE S
WHERE D.DB_KEY = S.DB_KEY
      AND UPPER(NAME) = UPPER('&1')
ORDER BY SCRIPT_NAME, LINE;
CLEAR BREAK
SET VERIFY ON
EOL
```

**Usage:**

```
cat_db_scp.sql <DB Name>
```

10. This script shows RMAN session and time needed to perform its operation.

```
cat > rman_sess.sql <<EOL
PROMPT
PROMPT This query displays the progress of the RMAN job.
PROMPT
SET LIN 500
col SID for 999999999

SELECT INST_ID, SID, SERIAL#, CONTEXT, SOFAR, TOTALWORK,
       ROUND(SOFAR/TOTALWORK*100,2) "%_COMPLETE",
       ROUND(TIME_REMAINING/60,2) TIME_REMAINING_MIN
FROM GV$SESSION_LONGOPS
WHERE OPNAME LIKE 'RMAN%'
      AND OPNAME NOT LIKE '%aggregate%'
      AND TOTALWORK != 0
      AND SOFAR <> TOTALWORK;
EOL
```

11. The following script shows how much of data restored by all channels.

```
cat > rman_tio.sql <<EOL
/* Author: Majid Shabani */
PROMPT
PROMPT READ WRITE PERFORMED BY ALL CHANNELS
PROMPT
COL NAME FOR A30
COL SID FOR 99999
COL VAL_GB FOR 999,999,999,999,999

SELECT N.NAME, ROUND(SUM(S.VALUE)/POWER(1024,3),2) VAL_GB FROM
V\\$SESSTAT S, V\\$STATNAME N, V\\$SESSION SS
WHERE S.STATISTIC# = N.STATISTIC#
      AND S.SID = SS.SID
      AND CLIENT_INFO LIKE 'rman%'
      AND NAME = 'physical write total bytes'
GROUP BY N.NAME
ORDER BY NAME, VAL_GB DESC;
EOL
```

12. The following script shows how much of data restored by each channel.

```
cat > rman_sess_io.sql <<EOL
/* Author: Majid Shabani */
SET LIN 500
COL NAME FOR A30
COL SID FOR 99999
COL VAL_GB FOR 999,999,999,999,999
COMPUTE SUM OF VAL_GB ON REPORT
BREAK ON REPORT

SELECT S.SID, N.NAME, ROUND(S.VALUE/1024/1024/1024,2) VAL_GB FROM
V\\$SESSTAT S, V\\$STATNAME N, V\\$SESSION SS
WHERE S.STATISTIC# = N.STATISTIC#
      AND S.SID = SS.SID
      AND CLIENT_INFO LIKE 'rman%'
      AND NAME = 'physical write total bytes'
ORDER BY SID, NAME, VAL_GB DESC;
CLEAR BREAK
EOL
```

You can download all of those scripts from my GitHub using this [link](#). There is one more script but very useful. Also, you can download new scripts and the latest version of them.

## Obtaining Backup Details:

In the following steps, I will obtain the following information about the database backups from catalog Database.

- Size of the last full backup of database.
- The last controlfile piece name
- The last SCN of the available archive log in the backups.
- The number of backup pieces have to be restored.
- The backup scripts (optional)

**Note:** to be able to restore the specific controlfile that was taken and saved in Data Domain, you should use the piece name of it. That's because you perform the 16th step.

### 13. Invoke SQLPLUS and connect to the RMAN catalog DB.

```
sqlplus rman/oracle@catdb
```

### 14. Obtain the database ID and DB KEY.

```
@ cat_db_detail prod_db
```

#### Sample output:

```
SQL> @cat_db_detail proddb
```

DBID	NAME	DB_KEY
832131506	PRODDb	2494554

### 15. Display the backup of the database was taken in last 4 days.

```
@ cat_bkp_detail <DB name> <how long time in day>
```

#### Sample output:

```
SQL> @cat_bkp_detail proddb 5
```

DB_NAME	DB_KEY	SESSION_KEY	SESSION_STAMP	STATUS	INPUT_TYPE	START_TIME	DEVICE	END_TIME	HOURS	IN_SIZE	OUT_SIZE	IN_SPEED	OUT_SPEED	COMP
proddb	2494554	11667316	1138323604	COMPLETED	DB FULL	01/06/23 01:00	SBT_TAPE	01/06/23 19:53	19	52.13T	51.03T	804.11M	787.11M	1.0216
proddb	2494554	11671076	1138328404	COMPLETED	ARCHIVELOG	01/06/23 02:20	SBT_TAPE	01/06/23 02:27	0	107.99G	107.99G	254.79M	254.79M	1.0000
proddb	2494554	11677393	1138339205	COMPLETED	ARCHIVELOG	01/06/23 05:20	SBT_TAPE	01/06/23 05:28	0	143.42G	143.42G	293.72M	293.72M	1.0000

### 16. Obtain the name of the controlfile backup piece of the last backup by SESSION KEY obtained in the previous step.

```
@ cat_bkp_detail <DB name> <session_key> OR <how long time in day>
```

#### Sample output:

```
SQL> @ cat_list_ctl_bkps proddb 11677393 null
```

SESSION_KEY	SESSION_STAMP	CREATION_TIME	HANDLE	FILESIZE_DISPLA	ELAPSED_SECONDS	STATUS	IS_CHECKPOINT_CHANGE
11677393	1138339205	01/06/23 05:28	c-832131506-20230601-01	39.36M	19	A	NO 425341897771

### 17. Verify the controlfile backup piece contains spfile.

```
@ cat_list_spfile_bkps <DB name> <SESSION KEY> <HANDLE>
```

**Note:** if the controlfile piece file doesn't contains spfile, the query returns no row.

### Sample output:

```
SQL> @ cat_list_spfile.bkps proddb null c-832131506-20230601-01
```

DB_NAME	DB_KEY_HANDLE	BACKUP_	START_TIME	COMPLETION_TIM	PIECE_SIZE_MB	HOURS	IN_SPEED	OUT_SPEED	SPFILE_SIZE	MODIFICATION_TIME
proddb	2494554	<b>c-832131506-20230601-01</b>	FULL	2023-06-01 05:28	01/06/23 05:28	40 00:00:18	2.19M	2.19M	32.00K	2023-05-25 11:08:37

**18.** Also, you can use the log of the rman job to obtain the controlfile backup piece name (optional).

```
@ cat_bkp_log <DB name> <SESSION KEY>
```

**19.** Obtain the SCN in the last archivelog in the last backup using SESSION KEY obtained in the step 15.

```
@ cat_show_arch_scn <DB name> <SESSION KEY> <how long time in day>
```

### Sample output:

```
SQL> @ cat_show_arch_scn proddb 11677393
```

SESSION_KEY	SESSION_STAMP	DB_KEY_DB_NAME	THREAD#	SEQUENCE#	NEXT_CHANGE	LAST_TIME	DEVICE_T	HANDLE	START_TIME	COMPLETION_TIME	STATUS
11677393	1138339205	2494554	PRODD08	1	285505	425310489297	2023-06-01 02:24:15	SBT_TAPE	cditjbck_33165_1_1	2023-06-01 05:20:20	2023-06-01 05:27:57 A
11677393	1138339205	2494554	PRODD08	1	285506	425310638664	2023-06-01 02:28:36	SBT_TAPE	cditjbck_33165_1_1	2023-06-01 05:20:20	2023-06-01 05:27:57 A
11677393	1138339205	2494554	PRODD08	1	285507	425320524742	2023-06-01 02:34:24	SBT_TAPE	cditjbck_33165_1_1	2023-06-01 05:20:20	2023-06-01 05:27:57 A
11677393	1138339205	2494554	PRODD08	1	285508	425320652240	2023-06-01 02:39:24	SBT_TAPE	cditjbck_33165_1_1	2023-06-01 05:20:20	2023-06-01 05:27:57 A
11677393	1138339205	2494554	PRODD08	1	285509	425320730448	2023-06-01 02:43:54	SBT_TAPE	cditjbck_33165_1_1	2023-06-01 05:20:20	2023-06-01 05:27:57 A
11677393	1138339205	2494554	PRODD08	1	285510	425320867466	2023-06-01 02:46:24	SBT_TAPE	cditjbck_33165_1_1	2023-06-01 05:20:20	2023-06-01 05:27:57 A
11677393	1138339205	2494554	PRODD08	1	285511	425322784704	2023-06-01 02:50:42	SBT_TAPE	cditjbck_33165_1_1	2023-06-01 05:20:20	2023-06-01 05:27:57 A
11677393	1138339205	2494554	PRODD08	1	285512	425325150537	2023-06-01 02:56:15	SBT_TAPE	cditjbck_33165_1_1	2023-06-01 05:20:20	2023-06-01 05:27:57 A
11677393	1138339205	2494554	PRODD08	1	285513	425325260840	2023-06-01 03:02:43	SBT_TAPE	cditjbck_33167_1_1	2023-06-01 05:20:20	2023-06-01 05:27:58 A
11677393	1138339205	2494554	PRODD08	1	285514	425325498065	2023-06-01 03:10:25	SBT_TAPE	cditjbck_33167_1_1	2023-06-01 05:20:20	2023-06-01 05:27:58 A
11677393	1138339205	2494554	PRODD08	1	285515	425325717990	2023-06-01 03:16:25	SBT_TAPE	cditjbck_33167_1_1	2023-06-01 05:20:20	2023-06-01 05:27:58 A
11677393	1138339205	2494554	PRODD08	1	285516	425326279428	2023-06-01 03:24:28	SBT_TAPE	cditjbck_33167_1_1	2023-06-01 05:20:20	2023-06-01 05:27:58 A
11677393	1138339205	2494554	PRODD08	1	285517	425327520924	2023-06-01 03:32:22	SBT_TAPE	cditjbck_33167_1_1	2023-06-01 05:20:20	2023-06-01 05:27:58 A
11677393	1138339205	2494554	PRODD08	1	285518	425327688218	2023-06-01 03:38:37	SBT_TAPE	cditjbck_33167_1_1	2023-06-01 05:20:20	2023-06-01 05:27:58 A
11677393	1138339205	2494554	PRODD08	1	285519	425327896264	2023-06-01 03:44:52	SBT_TAPE	cditjbck_33167_1_1	2023-06-01 05:20:20	2023-06-01 05:27:58 A
11677393	1138339205	2494554	PRODD08	1	285520	425327956683	2023-06-01 03:49:55	SBT_TAPE	cditjbck_33167_1_1	2023-06-01 05:20:20	2023-06-01 05:27:58 A
11677393	1138339205	2494554	PRODD08	1	285521	425330113508	2023-06-01 03:54:34	SBT_TAPE	cditjbck_33166_1_1	2023-06-01 05:20:20	2023-06-01 05:27:58 A
11677393	1138339205	2494554	PRODD08	1	285522	425332381159	2023-06-01 04:04:08	SBT_TAPE	cditjbck_33166_1_1	2023-06-01 05:20:20	2023-06-01 05:27:58 A
11677393	1138339205	2494554	PRODD08	1	285523	425333435591	2023-06-01 04:12:32	SBT_TAPE	cditjbck_33166_1_1	2023-06-01 05:20:20	2023-06-01 05:27:58 A
11677393	1138339205	2494554	PRODD08	1	285524	425333656569	2023-06-01 04:20:44	SBT_TAPE	cditjbck_33166_1_1	2023-06-01 05:20:20	2023-06-01 05:27:58 A
11677393	1138339205	2494554	PRODD08	1	285525	425333893743	2023-06-01 04:28:23	SBT_TAPE	cditjbck_33166_1_1	2023-06-01 05:20:20	2023-06-01 05:27:58 A
11677393	1138339205	2494554	PRODD08	1	285526	425334887581	2023-06-01 04:36:11	SBT_TAPE	cditjbck_33166_1_1	2023-06-01 05:20:20	2023-06-01 05:27:58 A
11677393	1138339205	2494554	PRODD08	1	285527	425335038834	2023-06-01 04:44:29	SBT_TAPE	cditjbck_33166_1_1	2023-06-01 05:20:20	2023-06-01 05:27:58 A
11677393	1138339205	2494554	PRODD08	1	285528	425335099218	2023-06-01 04:50:30	SBT_TAPE	cditjbck_33166_1_1	2023-06-01 05:20:20	2023-06-01 05:27:58 A
11677393	1138339205	2494554	PRODD08	1	285529	425336747696	2023-06-01 04:54:27	SBT_TAPE	cditjbck_33168_1_1	2023-06-01 05:20:21	2023-06-01 05:24:19 A
11677393	1138339205	2494554	PRODD08	1	285530	425341247675	2023-06-01 05:04:10	SBT_TAPE	cditjbck_33168_1_1	2023-06-01 05:20:21	2023-06-01 05:24:19 A
11677393	1138339205	2494554	PRODD08	1	285531	425341242037	2023-06-01 05:10:55	SBT_TAPE	cditjbck_33168_1_1	2023-06-01 05:20:21	2023-06-01 05:24:19 A
11677393	1138339205	2494554	PRODD08	1	285532	425341645135	2023-06-01 05:18:16	SBT_TAPE	cditjbck_33168_1_1	2023-06-01 05:20:21	2023-06-01 05:24:19 A
11677393	1138339205	2494554	PRODD08	1	285533	<b>4253416609713</b>	2023-06-01 05:20:14	SBT_TAPE	cditjbck_33168_1_1	2023-06-01 05:20:21	2023-06-01 05:24:19 A

29 rows selected.

**20.** Instead of performing step 16, 17, and 19, you can run below script and obtain all the required information such as number of files in each backup piece, thread# and SCN value in each archivelog backup file and piece name in which controlfile is backed up.

```
@ cat_bkpset_details <DB name> <SESSION KEY>
```

### Sample output:

```
# Last Archivelog backup Details
SQL> @ cat_bkpset_details proddb 11677393
```

BACKUP_	STARTING_TIME	COMPLETION_TIME	TIME_TAKEN	SIZE_GB	HANDLE	FILE#	THREAD#	FIRST_SEQ	LAST_SEQ	FIRST_SCN	LAST_SCN	LAST_TIME	CHECKPOINT_CHANGE
ARCHBKP	01/06/23 THU 05:20	01/06/23 THU 05:27:55	00:07:35	41	cditjbck_33165_1_1	8	1	285505	285512	425310489297	425325150537	2023-06-01 THU 02:56:15	
	01/06/23 THU 05:20	01/06/23 THU 05:27:49	00:07:29	41	cditjbck_33167_1_1	8	1	285513	285520	425325260840	425327956683	2023-06-01 THU 03:49:55	
	01/06/23 THU 05:20	01/06/23 THU 05:27:38	00:07:18	41	cditjbck_33166_1_1	8	1	285521	285528	425330113508	425338922318	2023-06-01 THU 04:50:30	
	01/06/23 THU 05:20	01/06/23 THU 05:24:18	00:03:57	21	cditjbck_33168_1_1	5	1	285529	285533	425336747696	<b>4253416609713</b>	2023-06-01 THU 05:20:14	
PIECE#				4									
MAXPS-G				41									
TSIZE_G				144									
FULL	01/06/23 THU 05:28	01/06/23 THU 05:28:22	00:00:18	0	<b>c-832131506-20230601-01</b>	1							425341897771
PIECE#				1									
MAXPS-G				0									
TSIZE_G				0									

**# Also, you can use this script to obtain details about the last full backup of the database.**

```
SQL> @ cat_bkpset_details proddb 11667316
```

BACKUP_	STARTING_TIME	COMPLETION_TIME	TIME_TAKEN	SIZE_GB	HANDLE	FILE#	THREAD#	FIRST_SEQ	LAST_SEQ	FIRST_SCN	LAST_SCN	LAST_TIME	CHECKPOINT_CHANGE
ARCHBKP	01/06/23 THU 01:00	01/06/23 THU 01:09:23	00:08:53	43	aditliso_33113_1_1	8	1	285420	285427	425291870440	425292216688	2023-05-31 WED 23:40:08	
	01/06/23 THU 01:00	01/06/23 THU 01:09:58	00:09:28	43	aditliso_33112_1_1	8	1	285428	285435	425292269062	425292616179	2023-05-31 WED 23:48:21	
	01/06/23 THU 01:00	01/06/23 THU 01:10:02	00:09:31	42	aditliso_33117_1_1	8	1	285436	285443	425292682022	425301891396	2023-05-31 WED 23:56:42	
	01/06/23 THU 01:00	01/06/23 THU 01:09:50	00:09:19	42	aditliso_33115_1_1	8	1	285444	285451	425301925236	42530265757	2023-06-01 THU 00:06:43	
	01/06/23 THU 01:00	01/06/23 THU 01:09:58	00:09:27	42	aditliso_33116_1_1	8	1	285452	285459	425302297984	425302636848	2023-06-01 THU 00:15:23	
	01/06/23 THU 01:00	01/06/23 THU 01:10:06	00:09:36	41	aditliso_33114_1_1	8	1	285460	285467	425302672191	425307847989	2023-06-01 THU 00:24:38	
	01/06/23 THU 01:00	01/06/23 THU 01:09:46	00:09:14	41	aditliso_33118_1_1	8	1	285468	285475	425307808935	425308453353	2023-06-01 THU 00:41:30	
	01/06/23 THU 01:00	01/06/23 THU 01:09:18	00:08:46	41	aditliso_33119_1_1	8	1	285476	285483	425308598074	425310399285	2023-06-01 THU 01:00:19	
PIECE#				8									
MAXPS-G				43									
TSIZE_G				338									
FULL	01/06/23 THU 01:10	01/06/23 THU 02:08:10	00:57:22	351	b2itliso_33122_1_1	8							
	01/06/23 THU 01:10	01/06/23 THU 02:12:41	01:01:52	350	b2itliso_33127_1_1	8							
	01/06/23 THU 01:10	01/06/23 THU 02:03:04	00:52:16	351	b3itliso_33123_1_1	8							
	01/06/23 THU 01:10	01/06/23 THU 02:06:16	00:55:28	352	b5itliso_33125_1_1	8							
	01/06/23 THU 01:10	01/06/23 THU 02:07:06	00:56:18	351	b0itliso_33120_1_1	8							
	01/06/23 THU 01:10	01/06/23 THU 02:08:47	00:57:59	351	b1itliso_33121_1_1	8							
	01/06/23 THU 01:10	01/06/23 THU 02:04:21	00:53:32	348	b6itliso_33126_1_1	8							
	01/06/23 THU 01:10	01/06/23 THU 02:03:12	00:52:24	343	b4itliso_33124_1_1	8							
	01/06/23 THU 02:03	01/06/23 THU 03:05:03	01:01:55	352	b8itliso_33128_1_1	8							

Majid Shabani Page | 13

```

***** 01/06/23 THU 19:36 01/06/23 THU 19:51:24 00:14:30 ----- 68 h01tktm_33312_1_1 13 1 286064 286076 425558781083 425563823330 2023-06-01 THU 19:36:39
TPIECE# ----- 8
MAXPS-G 09
TSIZE_G 550

FULL 01/06/23 THU 19:52 01/06/23 THU 19:53:00 00:00:14 0 c-832131506-20230601-00 1 425572337442
01/06/23 THU 19:52 01/06/23 THU 19:52:32 00:00:15 0 h11tkufh_33313_1_1 1 425572305457
***** -----
TPIECE# ----- 2
MAXPS-G 0
TSIZE_G 0

```

## 21. Obtain number of backup piece (optional).

```
@ cat_piece_count <DB name> <SESSION KEY>
```

### Sample output:

```
SQL> @ cat_piece_count proddb 11667316
```

```

BACKUP_TYPE          PIECE_CNT
-----
Full Backup          155
Backup of Archivelog 16
-----
sum                  171

```

## 22. Display the backup script of the database if there is any (optional).

```
@ cat_db_scp <DB name>
```

In conclusion, the name of the last controlfile which was taken in the last archivelog backup is '**c-832131506-20230601-01**' also, the controlfile backup contains spfile as well. And the last SCN of the last archivelog backup is '**425341669713**'.

## Performing Restore Operation

Now, it's time to start restore operation. Let's get started. Perform the following actions to restore the database from the backup files. In this stage datafiles will be created using backup files.

Please be aware, you need to replace the value of the following parameter with the proper one.

**NB\_ORA\_SERV:** Hostname of NetBackup server

**NB\_ORA\_CLIENT:** Hostname of the database server before failure.

**23.** Create a directory under oracle home to save temporary files in it.

```
mkdir /home/oracle/temp
```

**24.** Invoke RMAN and login as target to the local instance.

```
# set environment variables:
export ORACLE_SID=proddb
export ORACLE_HOME=/u01/app/oracle/product/19.0.0/db_1
export PATH=$ORACLE_HOME/bin:$PATH
export NLS_DATE_FORMAT='YYYY-MM-DD HH24:MI:SS'

rman target /
```

**25.** Start the instance in NOMOUNT mode

```
STARTUP FORCE NOMOUNT
```

**26.** Test connection to the NetBackup server.

```
# Note:
# NB_ORA_CLIENT: Hostname of the database server before failure.
# NB_ORA_SERV: Hostname of NetBackup server

RUN{
  ALLOCATE CHANNEL C01 TYPE 'SBT_TAPE' PARMS 'ENV=(NB_ORA_SERV=<NetBackup server,
  NB_ORA_CLIENT=DB hostname)';
  RELEASE CHANNEL C01;
}
```

**27.** Restore SPFILE and also create a PFILE from the SPFILE backup.

```
RUN {
  SET DBID <DBID>;
  ALLOCATE CHANNEL C01 TYPE 'SBT_TAPE' PARMS 'ENV=(NB_ORA_SERV=NetBackup server,
  NB_ORA_CLIENT=DB hostname)';
  RESTORE SPFILE TO PFILE '/home/oracle/temp/db_pfile.ora' FROM '<controlfile piece name>';

  # controlfile piece name obtained in step 16 or 20
```



- 28.** Obtain the directory name of the directory-based initialization parameters and create them in the destination host.

A. Open the recovered PFILE with the vi editor

```
vi /home/oracle/temp/db_pfile.ora
```

B. Obtain value of the following parameters. (you can edit them if needed):

```
*.audit_file_dest
*.control_files
*.db_create_file_dest
*.db_recovery_file_dest

# verify
egrep
'audit_file_dest|control_files|db_create_file_dest|db_recovery_file_dest|log_archive_
dest_1' /home/oracle/temp/db_pfile.ora
```

C. Create the directory, based on those parameter value as `oracle` user.

```
For example:
mkdir -p /u01/app/oracle/audit
```

- 29.** create spfile from PFILE (after changing parameters value).

```
sqlplus / as sysdba

create spfile from pfile='/home/oracle/temp/db_pfile.ora';
```

- 30.** Start the database in NOMOUNT state again. The target of this step is to let the instance starts using the SPFILE.

```
SHUTDOWN IMMEDIATE
STARTUP NOMOUNT

# Verify
col name for a20
col value for a60
select name,value from v$parameter where name='spfile';
```

- 31.** Invoke RMAN and connect as target to the local instance.

```
rman target /
```

- 32.** Restore the control files from the backup of the control file.

```
RUN {
SET DBID <DB ID>;
allocate channel c01 type 'SBT_TAPE' PARMS 'ENV=(NB_ORA_SERV=NetBackup server,
NB_ORA_CLIENT=DB hostname)';
restore controlfile from 'c-1250117420-20220124-0c';
}

# controlfile piece name obtained in step 16 or 20
# Also, you can use the following syntax
# RESTORE CONTROLFILE FROM AUTOBACKUP;
```

**33. Mount the database.**

```
SHUTDOWN IMMEDIATE
STARTUP MOUNT
```

**34. Generate `allocate channel` command.**

**Note:** In my opinion, each channel scans one backup piece at the time so the number of allocated channels could be up to the number of backup piece of datafiles in the database backup and also less than 255. To obtain number of backup piece, run `cat_piece_count.sql` script as shown in the step 21. Also, consult with net backup admin for setting channel.

```
sqlplus / as sysdba
set pages 500 line 500
SELECT 'ALLOCATE CHANNEL C'|| level || ' DEVICE TYPE ''SBT_TAPE'';' FROM DUAL CONNECT BY
LEVEL <= 20;
```

**35. create the following scripts to restore database.**

```
vi ~/temp/restore_db.sh
# NB_ORA_SERV: the hostname of the NetBackup server
# NB_ORA_CLIENT: the hostname of the server in which the backup was taken.

export ORACLE_SID=proddb
export ORACLE_HOME=/u01/app/oracle/product/19.0.0/db_1
export PATH=$ORACLE_HOME/bin:$PATH
export NLS_DATE_FORMAT='YYYY-MM-DD HH24:MI:SS'

$ORACLE_HOME/bin/rman target/ log=~/temp/restore_db.log <<EOL
RUN {
# replace channel commands
send 'NB_ORA_SERV=NetBackup server, NB_ORA_CLIENT=DB Hostname';
RESTORE DATABASE;
}
exit;
EOL
# save & exit.
# make it executable.
chmod u+x ~/temp/restore_db.sh
```

**36. Run the below command to validate the syntax of created script in the previous step.**

```
# Syntax validation
export ORACLE_SID=proddb
export ORACLE_HOME=/u01/app/oracle/product/19.0.0/db_1
export PATH=$ORACLE_HOME/bin:$PATH
export NLS_DATE_FORMAT='YYYY-MM-DD HH24:MI:SS'

$ORACLE_HOME/bin/rman checksyntax log=~/temp/restore_db_chk.log <<EOL
RUN {
# replace channel commands
send 'NB_ORA_SERV=NetBackup server, NB_ORA_CLIENT=DB hostname';
RESTORE DATABASE;
}
exit;
EOL
# check the output for any error
cat ~/temp/restore_db_chk.log
```

**37.** Now everything is ready to restore the database.

```
# run script in background.  
nohup ~/temp/restore_db.sh > ~/temp/restore_db.log 2>&1 &  
  
# check the log file.  
tail -100f ~/temp/restore_db.log
```

## Monitoring RMAN Restore Operation

38. Run below script and verify how many channels allocated.

```
SQL> @ rman_sess
```

39. Run the following script to see how much data is restored by RMAN.

```
SQL> @ rman_tio
```

Sample output:

READ WRITE PERFORMED BY ALL CHANNELS

NAME	VAL_GB
-----	-----
physical write total bytes	2560

Unfortunately, **V\$SYSMETRIC** doesn't show any data when the database is in mount state, which is supposed to be when restoring it. So, it is not possible to determine the total I/O per second performed by the database. One idea is calculating it by yourself, to do so you can calculate the average total I/O per second using this formula,

***Total write by RMAN (output of rman\_tio.sql) / (current time - start time of restore operation)***

If anyone has any other idea, please inform me.

40. Display how many data restored by each channel (optional).

```
SQL> @ rman_sess_io
```

41. It's been seen in some cases the SCN of some datafiles have significant differences from the other one, specifically when restoring large databases, so in that case the recovery operation will face error, and you need to restore those files again. I recommend you to check the datafile header SCN before starting recovery operation.

```
SQL> SELECT CHECKPOINT_CHANGE#, COUNT(*) FILE_NO FROM V$DATAFILE_HEADER GROUP BY
CHECKPOINT_CHANGE# ORDER BY FILE_NO;
```

Sample output:

CHECKPOINT_CHANGE#	FILE_NO
-----	-----
425310399285	760
416199026710	450
2539169	8

For the case like above, you need to restore those files have lower SCN again otherwise, you will face error when recovering the database.

## Recovering Database & Monitoring it

Please be aware, you need to replace the value of the following parameter with the proper one. In this stage changes will be applied on the datafiles by reading the archive logs and the incremental backups if they exist.

**NB\_ORA\_SERV:** Hostname of NetBackup server

**NB\_ORA\_CLIENT:** Hostname of the database server before failure.

**42.** After, finishing restore operation, recover database until obtained SCN in step 19 or 20.

```
# Generate allocate channel command.:
sqlplus / as sysdba
set pages 500 line 500
SELECT 'ALLOCATE CHANNEL C'|| level || ' DEVICE TYPE ''SBT_TAPE'';' FROM DUAL CONNECT BY
LEVEL <= 10;

# Create recover script. Replace SCN value with <last NEXT_SCN value> obtained from step
19 or 20.

vi ~/temp/recover_db.sh
# NB_ORA_SERV: the hostname of the NetBackup server
# NB_ORA_CLIENT: the hostname of the server in which the backup was taken.

export ORACLE_SID=proddb
export ORACLE_HOME=/u01/app/oracle/product/19.0.0/db_1
export PATH=$ORACLE_HOME/bin:$PATH
export NLS_DATE_FORMAT='YYYY-MM-DD HH24:MI:SS'

$ORACLE_HOME/bin/rman target/ log=~/temp/recover_db.log <<EOL
RUN {
# Replace channel commands
send 'NB_ORA_SERV=<NetBackup server>, NB_ORA_CLIENT=<DB hostname>';
RECOVER DATABASE UNTIL SCN <last NEXT_SCN value>;
}
exit;
EOL

# make it executable.
chmod u+x ~/temp/recover_db.sh
```

**43.** Validate the syntax of the script command before running (optional).

```
# validate syntax
export ORACLE_SID=proddb
export ORACLE_HOME=/u01/app/oracle/product/19.0.0/db_1
export PATH=$ORACLE_HOME/bin:$PATH
export NLS_DATE_FORMAT='YYYY-MM-DD HH24:MI:SS'

$ORACLE_HOME/bin/rman checksyntax log=~/temp/recover_db_chk.log <<EOL
RUN {
# replace channel commands
send 'NB_ORA_SERV=NetBackup server, NB_ORA_CLIENT=DB hostname';
RECOVER DATABASE UNTIL SCN <last NEXT_SCN value>;
}
exit;
EOL

# check for any error
cat ~/temp/recover_db_chk.log
```

**44.** Run the recover script command.

```
nohup ~/temp/recover_db.sh > ~/temp/recover_db.log 2>&1 &
```

**45.** Run below script and verify how many channels allocated.

```
SQL> @ rman_sess
```

**46.** Run the following script to see how much data is restored by RMAN.

```
SQL> @ rman_tio
```

Sample output:

READ WRITE PERFORMED BY ALL CHANNELS

NAME	VAL_GB
-----	-----
physical write total bytes	1743

**47.** Display how many data restored by each channel (optional).

```
SQL> @ rman_sess_io
```

## Post Action

### 48. Disable BCT and enable it again. (If needed)

```
# check if it is enabled.
set lin 500
col filename for a50
SELECT * FROM V$BLOCK_CHANGE_TRACKING;

# re-enable it again
ALTER DATABASE DISABLE BLOCK CHANGE TRACKING;
ALTER DATABASE ENABLE BLOCK CHANGE TRACKING;
```

### 49. Open the database with RESETLOGS option.

```
SELECT OPEN_RESETLOGS FROM V$DATABASE;

ALTER DATABASE OPEN RESETLOGS;
```

### 50. Retrieve the new database file names (optional).

```
SET LIN 500 PAGES 500
COL NAME FOR A90
SELECT NAME FROM V$DATAFILE ORDER BY CON_ID;
```

### 51. Add the following line to the /etc/oratab file as oracle user.

```
vi /etc/oratab

# add the following line
proddb:/u01/app/oracle/product/19.0.0/db_1:Y
```

### 52. Re-create password file.

As you know, password file cannot be backed up by RMAN. So, you must re-create it manually.

```
orapwd file=$ORACLE_HOME/dbs/orapwproddb password=Proddb@1234 entries=5 format=12.2
```

**Note:** using the "format=12" option, you can set simple password.

**Note:** name of the file must follow orapwORACLE\_SID naming format.

As you know, having experience about backup and recovery task is very crucial for every DBA to know. So, in this article, I've tried to demonstrate how to restore full database into a new server by a backup was taken using third-party tool. I hope it will be helpful.

If anyone has suggestion about the information explained in this article or event about the scripts, please send me message.

**Thanks,**

Majid Shabani