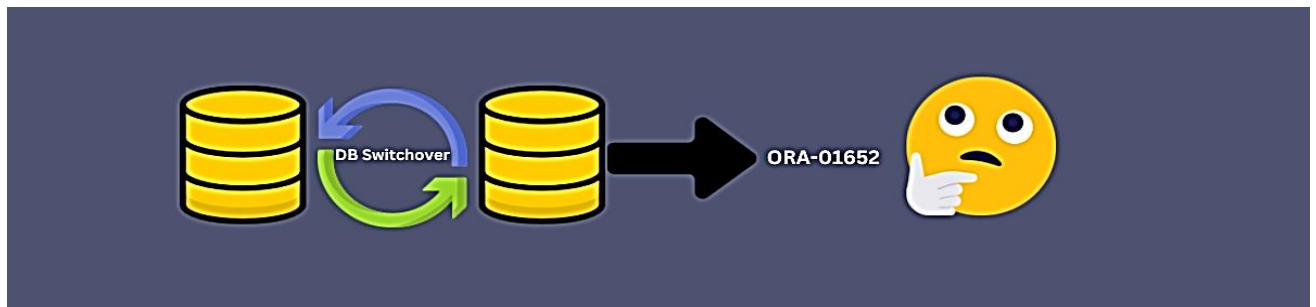


Why Sometimes, Successful DB Switchover Led Us to Temporary Tablespace Issue in Future



Step 1st: Create a blackout for the primary & standby database both

It's very important to create a blackout for both the primary & standby database, so that we can prevent the noise alert in a real-time environment.

Step 2nd: Disable All Database jobs in primary & standby both

Prior to start database switchover operation we must have to disable database's jobs in both primary & standby side whether we've schedule it through cron or any other third-party tool.

Step 3rd: Cross Verify if temporary tablespace's space is same in between primary and standby database

NAME	DATABASE_ROLE	OPEN_MODE
-----	-----	-----
US PRIM	PRIMARY	READ WRITE

NOTE: Verifying the size of temporary tablespace in both primary & standby side is one of the important steps because if the size of the temporary tablespace in standby database is smaller than the size of the temporary tablespace in primary database then even after a successful database switchover may led us to temporary tablespace issue which is **ORA-01652** in future.

REASON: Alter any temporary tablespace doesn't create redo due to which it'll not get replicated into the standby side automatically irrespective of the standby_file_management parameter's value, so for e.g. If we resize tempfile from 8G To 16G due to the lack insufficient temp segment for executing one of the application code then in that case we also need to increase the size of tempfile in standby side too otherwise after role transition the same application code will get fail again with same temp space's issue

Solution: whenever increase the side of tempfile in primary side try to do the same in standby side too.

```
SQL> show pdbs
```

CON_ID	CON_NAME	OPEN MODE	RESTRICTED
2	PDB\$SEED	READ ONLY	NO
3	PDB1	READ WRITE	NO

```
SQL> set lines 300 colsep '|'
```

```
SQL> col tablespace_name for a30
```

```
SQL> col file_name for a80
```

```
SQL> col con_id for 9999
```

```
SQL> col bytes for 999999.99999
```

```
SQL> select con_id,tablespace_name,file_name,sum(bytes/1024/1024/1024) as  
"SIZE_IN_GB" from cdb_temp_files group by con_id,tablespace_name,file_name;
```

CON_ID	TABLESPACE_NAME	FILE_NAME	SIZE_IN_GB
3	TEMP	/u01/app/oracle/oradata/US_PRIM/0D40A9D4D52D26BDE0630901A8C0E8F7/datafile/ o1_mf_ .03515625	
		temp_lrj53jhk_.dbf	
1	TEMP	/u01/app/oracle/oradata/US_PRIM/datafile/o1_mf_temp_lrj366m3_.tmp	
		.03125	

In Standby Database

NAME	DATABASE_ROLE	OPEN_MODE
US_PRIM	PHYSICAL STANDBY	READ ONLY WITH APPLY

```
SQL> show pdbs
```

CON_ID	CON_NAME	OPEN MODE	RESTRICTED
2	PDB\$SEED	READ ONLY	NO
3	PDB1	READ ONLY	NO

```
SQL> select con_id,tablespace_name,file_name,sum(bytes/1024/1024/1024) as
"SIZE_IN_GB" from cdb_temp_files group by con_id,tablespace_name,file_name;
```

CON_ID	TABLESPACE_NAME	FILE_NAME	SIZE_IN_GB
1	TEMP	/u01/app/oracle/oradata/IND_DR/datafile/TEMP01.dbf	.03125
3	TEMP	/u01/app/oracle/oradata/IND_DR/datafile/PDB1_TEMP01	.03515625

Step 4th: Verify If Standby Redo Logs are configure in current primary

```
SQL> select group#,sequence#,status,bytes from v$standby_log;
```

GROUP#	SEQUENCE#	STATUS	BYTES
4	0	UNASSIGNED	209715200
5	0	UNASSIGNED	209715200
6	0	UNASSIGNED	209715200
7	0	UNASSIGNED	209715200

Execute The Same In Standby Database

```
SQL> select group#,sequence#,status,bytes from v$standby_log;
```

GROUP#	SEQUENCE#	STATUS	BYTES
4	10	ACTIVE	209715200
5	0	UNASSIGNED	209715200
6	0	UNASSIGNED	209715200
7	0	UNASSIGNED	209715200

Step 5th: Verify if primary & standby are in sync

Execute Below Command In Both Primary & Standby Side.

NOTE: Check **async ORL single** in primary side and **recovery logmerger** in standby side and the sequence# in both the side must be same

```
SQL> select name,role,thread#,sequence#,block# from
v$logdataguard_process;
```

NAME	ROLE	THREAD#	SEQUENCE#	BLOCK#
LGWR	log writer		0	0
TMON	redo transport monitor		0	0
TT00	gap manager	1	10	0
TT01	redo transport timer	0	0	0
ARC0	archive local	0	0	0
ARC1	archive redo	0	0	0
ARC2	archive redo	0	0	0
ARC3	archive redo	0	0	0
TT02	async ORL multi	1	10	0
TT03	heartbeat redo informer	0	0	0
TT04	async ORL single	1	10	230556

11 rows selected.

Step 6th: Connect with primary & Execute below command to convert it into standby

```
SQL> alter database commit to switchover to physical standby;
```

Database altered.

NOTE: After executing switchover command in the primary side, **End-Of-Redo Branch archival of T-1.S-10** must be shipped into the standby side and apply it too in order to make the database switchover operation successful.

Alert log of primary database

```
ALTER DATABASE COMMIT TO SWITCHOVER TO PHYSICAL STANDBY [Process
Id: 4734] (USPRIM)
```

```
.... (PID:4734): Active, synchronized Physical Standby switchover target has been
identified
```

Switchover End-Of-Redo Log thread 1 sequence 10 has been fixed

Switchover: Primary highest seen SCN set to 0x0000000000024411c

NET (PID:4734): End-Of-Redo Branch archival of T-1.S-10

NET (PID:4734): LGWR is scheduled to archive to LAD:2 after log switch

Backup controlfile written to trace file

/u01/app/oracle/diag/rdbms/us_prim/USPRIM/trace/USPRIM_ora_4734.trc

NET (PID:4734): Converting the primary database to a new standby database

Clearing standby activation ID 688636239 (0x290bc14f)

Step 7th: Connect to physical standby & execute below command to convert it into primary role.

SQL> alter database commit to switchover to primary;

Database altered.

Alert log of standby database

alter database commit to switchover to primary

2023-12-30T16:11:44.314590+05:30

ALTER DATABASE SWITCHOVER TO PRIMARY (INDDR)

Maximum wait for role transition is 15 minutes.

.... (PID:4579): Database role cleared from PHYSICAL STANDBY
[kcvs.c:1030]

Switchover: Complete - Database mounted as primary

**TMI: kvc_commit_to_so_to_primary Switchover from physical END
2023-12-30 16:11:55.230816**

Completed: alter database commit to switchover to primary

Step 8th: Start database in their respective mode after role transition

SQL> alter database open;

Database altered.

SQL> select name,database_role,open_mode from v\$database;

NAME	DATABASE_ROLE	OPEN_MODE
------	---------------	-----------

US_PRIM	PRIMARY	READ WRITE
---------	---------	------------

NAME	TYPE	VALUE
------	------	-------

db_unique_name	string	IND_DR
----------------	--------	--------

```
SQL> show pdbs
```

CON_ID	CON_NAME	OPEN	MODE	RESTRICTED
2	PDB\$SEED	READ ONLY	NO	
3	PDB1	READ WRITE	NO	

Start New Standby Database In Mount Mode/Read-Only Mode

```
SQL> startup
```

ORACLE instance started.

Total System Global Area 1728050736 bytes

Fixed Size 9135664 bytes

Variable Size 436207616 bytes

Database Buffers 1275068416 bytes

Redo Buffers 7639040 bytes

Database mounted.

Database opened.

```
SQL> select name,database_role,open_mode from v$database;
```

NAME	DATABASE_ROLE	OPEN_MODE

US_PRIM PHYSICAL STANDBY READ ONLY

NAME	TYPE	VALUE

db_unique_name	string	US_PRIM
-----------------------	---------------	----------------

```
SQL> show pdbs
```

CON_ID	CON_NAME	OPEN	MODE	RESTRICTED
2	PDB\$SEED	READ ONLY	NO	
3	PDB1	READ ONLY	NO	

Step 9th : Start MRP in new standby database

```
SQL> alter database recover managed standby database disconnect from session;
```

Database altered.

Step 10 : Verify if new standby is in sync now with its new primary

NOTE: Check **async ORL single** in primary side and **recovery logmerger** in standby side and the sequence# in both the side must be same

```
SQL> select role,thread#,sequence#,block# from v$dataguard_process;
```

ROLE	THREAD#	SEQUENCE#	BLOCK#
-----	-----	-----	-----
log writer	0	0	0
redo transport monitor	0	0	0
gap manager	0	0	0
redo transport timer	0	0	0
archive local	0	0	0
archive redo	0	0	0
archive redo	0	0	0
archive redo	0	0	0
managed recovery	0	0	0
recovery logmerger	1	29	1661
recovery apply slave	0	0	0
recovery apply slave	0	0	0
RFS ping	1	29	0
RFS async	1	29	1661
RFS archive	0	0	0
RFS archive	0	0	0
RFS archive	0	0	0

17 rows selected.

Step 11th: Enable All Database Jobs

Step 12: Remove Blackout after completing role transition operation

Step 13: Release the database for the application team to validate their data and Connectivity.