Oracle DataGuard Scenarios

Version 0.9 Hemant K Chitale. Last Updated 11-November-2022

Table of Contents

Contents

Introduction

This document covers Physical Standby which uses Redo Copy (aka “Redo Shipping”) and Apply. Logical Standby actual transfer SQL statements --- Golden Gate would be preferable to building a Logical Standby

About ORACLE_SID

Note : I always setup the Primary and Standby with *different* ORACLE_SIDs. Some sites may prefer use the same ORACLE_SID on both servers. This can be a bit confusing.
In this document Primary SID is "ORCL12C" and the Standby SID is "STDB"
Different ORACLE_SIDs can be used when applications connect using the Service Names registered through the listener – as wen a Switchover or Failover is done, the Service Name remains the same (except that HostName/IP are different). Going forward with MultiTenant architecture (nonCDB is deprecated in 19c and desupported in 21c), connecting via Service Name will have to be the connection method so ORACLE_SID will be less important.

For steps on how to actual build and start the Standby see the documents in Appendix-1.

Important

The database configuration FORCE LOGGING and the instance parameter STANDBY_FILE_MANAGEMENTT are quite important. The first ensures that even for NoLogging operations, Redo shipping is done – else the Standby will stop applying Redo and will lag when it sees the first NoLogging operation being shipped. The second ensures that when new datafiles are added at the Primary they are also added automatically at the Standby.
These two should be configured for both the Primary and [all] Standby[s] so that the behaviour remains the same even after a Switchover / Failover / Sandbox Testing / DR Testing

When the Standby encounters markers for NoLogging Transactions in the Redo Stream simply because FORCE LOGGING was not set on the Primary, it will halt recovery of the whole Standby Database The datafiles where NoLogging was done can be refreshed using the "Partial Refresh – one or more datafiles" until FORCE LOGGING is configured.

Loss of a single datafile which did not get create on the Standby simply because the parameter STANDBY_FILE_MANAGEMENT was set at 'MANUAL' will halt recovey of the whole Standby Database See the section "Partial Refresh – one or more datafiles"

Startup

Primary

```
SQL> startup
ORACLE instance started.

Total System Global Area  838860800 bytes
Fixed Size            8798312 bytes
Variable Size             343936920 bytes
Database Buffers    478150656 bytes
Redo Buffers                7974912 bytes
Database mounted.
Database opened.
SQL>
```

Messages in the Primary alert log that show it can’t connect to the Standby

(messages will repeat every minute)

If the Standby Listener is down

```
***********************************************************************

Fatal NI connect error 12541, connecting to:

(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=<address>)(PORT=<port>))(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=STDB)(CID=(PROGRAM=oracle)(HOST=<hostname>)(USER=oracle))))

  VERSION INFORMATION:
      TNS for Linux: Version 12.2.0.1.0 - Production
      TCP/IP NT Protocol Adapter for Linux: Version 12.2.0.1.0 - Production
  Time: <timestamp>
  Tracing not turned on.
  Tns error struct:
    ns main err code: 12541

TNS-12541: TNS:no listener
    ns secondary err code: 12560
    nt main err code: 511

TNS-00511: No listener
    nt secondary err code: 111
    nt OS err code: 0

Error 12541 received logging on to the standby
Check whether the listener is up and running
```

If the Standby Listener is up but the Standby Database Instance is not Mounted

```
Error 1034 received logging on to the standby
```

After the Standby database is Mounted

```
TT04: Standby redo logfile selected for thread 1 sequence 106 for
destination LOG_ARCHIVE_DEST_2
```

```
ARC3: Standby redo logfile selected for thread 1 sequence 105 for
destination LOG_ARCHIVE_DEST_2
```

Standby

```
SQL> startup mount;
ORACLE instance started.

Total System Global Area  419430400 bytes
Fixed Size            8793496 bytes
Variable Size             167772776 bytes
Database Buffers    234881024 bytes
Redo Buffers                7983104 bytes
Database mounted.
SQL> alter database recover managed standby database
  2  using current logfile
  3  disconnect from session
  4  /

Database altered.
```

SQL>

Merely doing a "STARTUP" or "STARTUP MOUNT" does not start recovery. The "… RECOVER MANAGED STANDBY DATABASE …" statement is required, else the Standby doe not apply any Redo from the Primary and starts lagging behind the Primary. See the section on Transaction Lag for how to monitor / catch-up and/or Refresh the Standby if the lag is very high

Messages from the Standby alert log

```
Completed: ALTER DATABASE   MOUNT
2021-03-06T17:38:11.053123+08:00
Warning: ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING CURRENT
LOGFILE has been deprecated.
Warning: ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING CURRENT
LOGFILE has been deprecated.
alter database recover managed standby database
using current logfile
disconnect from session
2021-03-06T17:38:11.057863+08:00
Attempt to start background Managed Standby Recovery process (STDB)
Starting background process MRP0
2021-03-06T17:38:11.067475+08:00
MRP0 started with pid=42, OS id=8557
2021-03-06T17:38:11.069006+08:00
MRP0: Background Managed Standby Recovery process started (STDB)
2021-03-06T17:38:16.099370+08:00
 Started logmerger process
2021-03-06T17:38:16.120914+08:00
Managed Standby Recovery starting Real Time Apply
2021-03-06T17:38:16.183209+08:00
Parallel Media Recovery started with 2 slaves
2021-03-06T17:38:16.378846+08:00
Recovery of Online Redo Log: Thread 1 Group 5 Seq 106 Reading mem 0
  Mem# 0: /STANDBY/database/STDB/onlinelog/o1_mf_5_j46l4pyq_.log
  Mem# 1:
/STANDBY/fast_recovery_area/stdb/STDB/onlinelog/o1_mf_5_j46l4tvl_.log
```

```
2021-03-06T17:38:17.077942+08:00
Completed: alter database recover managed standby database
using current logfile
disconnect from session
2021-03-06T17:38:23.216807+08:00
Primary database is in MAXIMUM PERFORMANCE mode
RFS[1]: Assigned to RFS process (PID:8595)
RFS[1]: Selected log 5 for T-1.S-106 dbid 768045447 branch 937554761
2021-03-06T17:57:53.110199+08:00
Archived Log entry 3 added for T-1.S-106 ID 0x2dc76487 LAD:1
2021-03-06T17:57:53.492744+08:00
Media Recovery Waiting for thread 1 sequence 107 (in transit)
2021-03-06T17:57:53.493962+08:00
Recovery of Online Redo Log: Thread 1 Group 4 Seq 107 Reading mem 0
  Mem# 0: /STANDBY/database/STDB/onlinelog/o1_mf_4_j46l4bns_.log
  Mem# 1:
/STANDBY/fast_recovery_area/stdb/STDB/onlinelog/o1_mf_4_j46l4hg6_.log
```

Note the warning about the "USING CURRENT LOGFILE" clause being deprecated. So, we don't need to specify it on a 12c Standby

Standby starts using the Online Redo Log file present at the Standby

(Primary can be in either "Maximum Protection" [Primary goes down if Standby goes down], "Maximum Performance" [Primary can keep running by creating ArchiveLogs even when Standby is down] or "Maximum Availability" [Primary waits for acknowledgement of transaction commit from Standby, however, if it doesn't get acknowledgement, it switches to "Maximum Performance" mode]. "Maximum Performance" is the default.

T-1 is "Thread-1 for the first instance in RAC and the only instance in non-RAC, S-106 is the ArchiveLog SequenceNumber in that thread , dbid is a unique identifier for each database (Primary and Standby have the same DBID and DB_NAME but can have different DB_UNIQUE_NAME and ORACLE_SID, branch is a the Resetlogs_ID when a Resetlogs is done and a new Incarnation of the database is created (DBID and RESETLOGS_ID are visible from V$DATABASE and V$INCARNATION)

Depending on the volume of changes at the Primary you will see multiple "Archive Log entry xx added" and "Media Recovery Waiting for thread x sequence xx (in transit)" and "Recovery of Onlne Redo Log: Thread x Group x Seq xx" messages
Once recovery of Sequence 106 is completed, it starts waiting for Sequence 107 and so on for all subsequent Archive Logs.

Additional message possible if Standby instance is auto-managing the FRA (Fast Recovery Area) to limit space usage :

```
Deleted Oracle managed file
/STANDBY/fast_recovery_area/stdb/STDB/archivelog/2021_03_06/o1_mf_0_0_j46md
h6o_.arc
```

Verification to re-enable Redo Shipping on the Primary after Standby Startup

```
SQL> show parameter log_archive_dest_2

NAME                                 TYPE        VALUE
------------------------------------ ----------- --------------------------
----
log_archive_dest_2                   string      service=STDB async affirm reop
```

```
                                             en=15 valid_for=(all_logfiles,
                                             primary_role) db_unique_name=S
                                             TDB
SQL> show parameter log_archive_dest_state_2

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
----
log_archive_dest_state_2             string      enable
```

If the dest_state was set to DEFER

```
NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
----
log_archive_dest_state_2             string      DEFER
```

You must re-enable after the Standby is startup and recovery at the Standby is begun again

```
SQL> alter system set log_archive_dest_state_2='ENABLE';
```

Shutdown

Primary-Only

```
SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL>
```

Messages on the Primary

```
Completed: ALTER DATABASE CLOSE NORMAL
ALTER DATABASE DISMOUNT
Shutting down archive processes
Archiving is disabled
Completed: ALTER DATABASE DISMOUNT
2021-03-06T17:54:48.464694+08:00
ARCH: Archival disabled due to shutdown: 1089
Shutting down archive processes
Archiving is disabled
2021-03-06T17:54:49.479559+08:00
JIT: pid 8738 requesting stop
2021-03-06T17:54:50.480736+08:00
ARCH: Archival disabled due to shutdown: 1089
Shutting down archive processes
Archiving is disabled
JIT: pid 8738 requesting stop
2021-03-06T17:54:50.501073+08:00
Stopping background process VKTM
2021-03-06T17:54:54.767896+08:00
Instance shutdown complete (OS id: 8738)
```

(all normal messages)

Messages on the Standby

```
RFS[2]: Possible network disconnect with primary database
```

Standby-Only

Command to be issued on Primary before Standby is shutdown

```
SQL> show parameter log_archive_dest_2

NAME                             TYPE    VALUE
------------------------------------ ----------- --------------------------
----
log_archive_dest_2                 string  service=STDB async affirm reop
                                     en=15 valid_for=(all_logfiles,
                                     primary_role) db_unique_name=S
                                     TDB
SQL> show parameter log_archive_dest_state_2

NAME                             TYPE    VALUE
------------------------------------ ----------- --------------------------
----
log_archive_dest_state_2           string  enable
SQL> alter system set log_archive_dest_state_2='DEFER';

System altered.
```

```
SQL>
```

Note : Specify for multiple log_archive_dest_state_xx if there are more than 1 Standby Databases. DEST_2 and DEST_STATE_2 are always for the First Standby.

Commands on Standby

```
SQL> alter database recover managed standby database cancel;

Database altered.

SQL> shutdown immediate;
ORA-01109: database not open


Database dismounted.
ORACLE instance shut down.
SQL>
```

The “CANCEL” of the Recovery is optional (was required in earlier versions). The SHUTDOWN IMMEDIATE does this automatically

Messages on the Standby alert log

```
alter database recover managed standby database cancel
2021-03-06T18:06:19.864296+08:00
MRP0: Background Media Recovery cancelled with status 16037
2021-03-06T18:06:19.889906+08:00
Errors in file
/u01/app/oracle/diag/rdbms/stdb/STDB/trace/STDB_pr00_8573.trc:
ORA-16037: user requested cancel of managed recovery operation
Managed Standby Recovery not using Real Time Apply
Recovery interrupted!
Recovered data files to a consistent state at change 3431769
2021-03-06T18:06:20.346831+08:00
Errors in file
/u01/app/oracle/diag/rdbms/stdb/STDB/trace/STDB_pr00_8573.trc:
ORA-16037: user requested cancel of managed recovery operation
2021-03-06T18:06:20.347628+08:00
MRP0: Background Media Recovery process shutdown (STDB)
2021-03-06T18:06:20.845677+08:00
Managed Standby Recovery Canceled (STDB)
Completed: alter database recover managed standby database cancel
2021-03-06T18:06:25.583666+08:00
Shutting down instance (immediate) (OS id: 9582)
Stopping background process SMCO
2021-03-06T18:06:27.250552+08:00
Shutting down instance: further logons disabled
Stopping background process MMNL
2021-03-06T18:06:28.276685+08:00
Stopping background process MMON
2021-03-06T18:06:29.277045+08:00
License high water mark = 6
2021-03-06T18:06:29.428138+08:00
Dispatchers and shared servers shutdown
2021-03-06T18:06:31.433584+08:00
ALTER DATABASE CLOSE NORMAL
2021-03-06T18:06:31.434615+08:00
```

```
Stopping Emon pool
ORA-1109 signalled during: ALTER DATABASE CLOSE NORMAL...
ALTER DATABASE DISMOUNT
Shutting down archive processes
2021-03-06T18:06:31.464933+08:00
TT00: Gap Manager exiting (PID:8546)
2021-03-06T18:06:32.465227+08:00
Archiving is disabled
2021-03-06T18:06:32.466388+08:00
ARC1: ARCH shutting down
ARC1: Archival stopped
2021-03-06T18:06:32.470283+08:00
ARC0: ARCH shutting down
ARC0: Archival stopped
2021-03-06T18:06:33.482000+08:00
Completed: ALTER DATABASE DISMOUNT
2021-03-06T18:06:33.482912+08:00
ARCH: Archival disabled due to shutdown: 1089
Shutting down archive processes
Archiving is disabled
2021-03-06T18:06:34.496840+08:00
JIT: pid 9582 requesting stop
2021-03-06T18:06:35.280762+08:00
OS process OFSD (ospid 8487) idle for 30 seconds, exiting
2021-03-06T18:06:35.496992+08:00
ARCH: Archival disabled due to shutdown: 1089
Shutting down archive processes
Archiving is disabled
JIT: pid 9582 requesting stop
2021-03-06T18:06:35.539832+08:00
Stopping background process VKTM
2021-03-06T18:06:40.935194+08:00
Instance shutdown complete (OS id: 9582)
```

The [optional] manually issued CANCEL RECOVERY is logged in the alert log.
The ORA-16037 and "Recovery Interrupted" are normal, expected messages when CANCEL RECOVERY and SHUTDOWN IMMEDIATE are manually issued (or issued through a script)
The alert log also shows the last SCN to which all datafiles are consistent at the standby (this can be compared with `current_scn from v$database` on the Primary later)

Both

If shutting down both the databases, always shutdown the Primary first wit

```
SQL> alter system switch logfile
SQL> alter system archive log all
SQL>shutdown immediate
```

This ensures that *all* pending Redo is shipped to the Standby before the Primary is shutdown.

When restarting, business users will want that the Primary is started first. However, if you can get enough time, you can start the Standby first and have it ready to receive Redo from the Primary.
Remember that the listener on both servers must be started before the database instance.

Transaction Lag

When Standby is Lagging after Long Shutdown

Current ArchiveLog Status at Primary

```
SQL> archive log list;
Database log mode        Archive Mode
Automatic archival             Enabled
Archive destination            USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence     111
Next log sequence to archive   113
Current log sequence           113
SQL>
```

Remark : Do not run an “archive log list” query at a Standby. It will not show the correct sequence numbers

Current SCN at Primary

```
SQL> set numwidth 32
SQL> select current_scn from v$database;

                     CURRENT_SCN
--------------------------------
                         3493117

SQL>
```

Last ArchiveLog applied at Standby

These messages from the Standby alert log show that Sequence 107 was the last one being applied

```
Media Recovery Waiting for thread 1 sequence 107 (in transit)
```

And

```
Recovery of Online Redo Log: Thread 1 Group 4 Seq 107 Reading mem 0
```

(V$ views cannot be queried at the Standby when it is down)

Initial Actions

Startup on Standby

```
SQL> startup mount
SQL> alter database recover managed standby database disconnect from
session
```

Re-enable Redo Shipping from Primary

```
SQL> alter system set log_archive_dest_state_2='enable';
```

Messages in the Standby alert log

```
alter database recover managed standby database disconnect from session
Attempt to start background Managed Standby Recovery process (STDB)
Starting background process MRP0
MRP0 started with pid=35, OS id=15410
MRP0: Background Managed Standby Recovery process started (STDB)
 Started logmerger process
```

```
Managed Standby Recovery starting Real Time Apply
Parallel Media Recovery started with 2 slaves
Recovery of Online Redo Log: Thread 1 Group 4 Seq 107 Reading mem 0
  Mem# 0: /STANDBY/database/STDB/onlinelog/o1_mf_4_j46l4bns_.log
  Mem# 1:
/STANDBY/fast_recovery_area/stdb/STDB/onlinelog/o1_mf_4_j46l4hg6_.log
Completed: alter database recover managed standby database disconnect from
session
RFS[1]: Assigned to RFS process (PID:15494)
RFS[1]: Opened log for T-1.S-108 dbid 768045447 branch 937554761
RFS[2]: Assigned to RFS process (PID:15496)
RFS[2]: Opened log for T-1.S-109 dbid 768045447 branch 937554761
Archived Log entry 4 added for thread 1 sequence 108 rlc 937554761 ID
0x2dc76487 LAD2 :
Archived Log entry 5 added for thread 1 sequence 109 rlc 937554761 ID
0x2dc76487 LAD2 :
Primary database is in MAXIMUM PERFORMANCE mode
RFS[3]: Assigned to RFS process (PID:15498)
RFS[3]: Selected log 4 for T-1.S-107 dbid 768045447 branch 937554761
RFS[1]: Opened log for T-1.S-110 dbid 768045447 branch 937554761
RFS[4]: Assigned to RFS process (PID:15501)
RFS[4]: Selected log 5 for T-1.S-114 dbid 768045447 branch 937554761
2021-03-06T22:39:33.357142+08:00
RFS[2]: Opened log for T-1.S-111 dbid 768045447 branch 937554761
Archived Log entry 6 added for thread 1 sequence 110 rlc 937554761 ID
0x2dc76487 LAD2 :
Archived Log entry 7 added for thread 1 sequence 111 rlc 937554761 ID
0x2dc76487 LAD2 :
RFS[1]: Opened log for T-1.S-112 dbid 768045447 branch 937554761
2021-03-06T22:39:39.219578+08:00
Archived Log entry 8 added for thread 1 sequence 112 rlc 937554761 ID
0x2dc76487 LAD2 :
RFS[2]: Opened log for T-1.S-113 dbid 768045447 branch 937554761
Archived Log entry 9 added for thread 1 sequence 113 rlc 937554761 ID
0x2dc76487 LAD2 :
db_recovery_file_dest_size of 5120 MB is 28.03% used. This is a
user-specified limit on the amount of space that will be used by this
database for recovery-related files, and does not reflect the amount of
space available in the underlying filesystem or ASM diskgroup.
Archived Log entry 10 added for T-1.S-107 ID 0x2dc76487 LAD:1
Media Recovery Log
/STANDBY/fast_recovery_area/stdb/STDB/archivelog/o1_mf_1_108_j47511jw_.arc
Media Recovery Log
/STANDBY/fast_recovery_area/stdb/STDB/archivelog/o1_mf_1_109_j47511kt_.arc
Media Recovery Log
/STANDBY/fast_recovery_area/stdb/STDB/archivelog/o1_mf_1_110_j47513sf_.arc
Media Recovery Log
/STANDBY/fast_recovery_area/stdb/STDB/archivelog/o1_mf_1_111_j4751533_.arc
Media Recovery Log
/STANDBY/fast_recovery_area/stdb/STDB/archivelog/o1_mf_1_112_j47517dj_.arc
Media Recovery Log
/STANDBY/fast_recovery_area/stdb/STDB/archivelog/o1_mf_1_113_j4751c6d_.arc
Media Recovery Waiting for thread 1 sequence 114 (in transit)
Recovery of Online Redo Log: Thread 1 Group 5 Seq 114 Reading mem 0
  Mem# 0: /STANDBY/database/STDB/onlinelog/o1_mf_5_j46l4pyq_.log
```

(I have removed all the timestamps and shortened the path to the archivelog files on the Standby so that the messags are more readable)

Oracle on the Standby retrieves all the “missing” (i.e. UnApplied) ArchiveLogs (Sequences 108 to 113) with the “Archived log entry xx added for ..” messages and then applies them to the Standby with the “Media Recovery Log” message.
Once the Standby has completed a catch-up, it reverts to the normal continuous recovery mode with “Media Recovery Waiting for …” and “Recovery of Online Redo Log” messages from Sequence 114.

```
SQL> select current_scn from v$database;

CURRENT_SCN
-----------
    3492795

SQL>
```

So, now, based on the SCN, the Standby lag from the Primary is low.

Query To verify the SCNs

One way to verify the SCNs is to query the primary (you can add FIRST_TIME to the SELECT list)

```
SQL> l
  1  select sequence#, first_change#, next_change#
  2  from v$archived_log
  3  where first_time > sysdate-1
  4  and dest_id=1  -- dest_id 1 for ArchiveLogs generated on the Primary
only
  5* order by 1
SQL> /

 SEQUENCE# FIRST_CHANGE# NEXT_CHANGE#
---------- ------------- ------------
       103       3404372      3417093
       104       3417093      3423619
       105       3423619      3424798
       106       3424798      3429952
       107       3429952      3479339
       108       3479339      3479344
       109       3479344      3479355
       110       3479355      3479358
       111       3479358      3479369
       112       3479369      3492796
       113       3492796      3498235

11 rows selected.

SQL>
```

Note : IF the database running RAC, you must also include THREAD# in the query

Periodically monitoring the “catch-up”

This query on the Standby can show the lag (it will work only if both the listeners are running and the Primary has log_archive_dest_state_2 set to ‘enable’)

```
SQL> l
  1  select name, value, unit
  2  from v$dataguard_stats
  3* where name like '%lag'
SQL> /

NAME                         VALUE                          UNIT
```

```
------------------------ -------------------------- -----------------------
-------
transport lag            +00 00:00:00               day(2) to second(0)
apply lag                +00 00:04:14               day(2) to second(0)

SQL>
```

(I have removed the word “interval” from the UNIT column listing so that the output can be shown in a single line for each entry)

“transport lag” shows the archive log transfer gap – e.g if network is too slow for the volume (total bytes) of archivelogs being transferred

“apply lag”shows the Redo Apply lag – e.g. if the received ArchiveLogs/Redo can’t be written to the Standby quickly enough

Another useful query (when you have only 1 Standby database) which shows an *approximate* Lag between the Primary and the Standby is :

```
select scn_to_timestamp(current_scn)
          - scn_to_timestamp(applied_scn)
          as Time_Diff
from v$database d,
(select applied_scn from  v$archive_dest a where target = 'STANDBY')
```

This query relies on information that the Standby database periodically sends to the Primary, so it expects that the Standby is up and running.

More Useful Queries at the Primary

```
select client_process, process, thread#, sequence#, status
from v$managed_standby
where
(
client_process='LGWR'
or
process='MRP0'
)
/


select status, gap_status
from v$archive_dest_status
where dest_id=2   --- for first Standby
/
(sample output
STATUS    GAP_STATUS
--------- ------------------------
VALID     NO GAP
)
```

Remark : The query on v$archive_dest_status only shows whether Redo Shipping (i.e transfer of ArchiveLogs to the Standby) is working properly. It does NOT confirm that Redo Apply is running. Redo Apply (i.e. actual Recovery of the Standby database) may have halted (e.g. if one or more datafiles is missing and/or Recovery has not been started or has been stopped after it was started). So you must also check the Redo Apply from the alert log messages at the Standby and from v$dataguard_status at the Standby.

```
select sequence#-1 Last_Sequence_Applied,
first_change#-1 Last_Txn_Applied_SCN,
substr(scn_to_timestamp(first_change#-1),1,24)
Last_Txn_Applied_Time_Approx,
to_char(sysdate,'DD-MON-RR HH24:MI:SS') Current_Time
```

```
from v$log
where sequence# =
  (select max(sequence#)+1 from v$archived_log where applied='YES')
/
tti off
(sample output
LAST_SEQUENCE_APPLIED LAST_TXN_APPLIED_SCN
--------------------- --------------------
LAST_TXN_APPLIED_TIME_APPROX
CURRENT_TIME
----------------------------------------------------------------------------------------------
-- ---------------------------
                  125            3550118
07-MAR-21 11.07.30.00000
07-MAR-21 23:23:57


SQL>
)

select current_scn Primary_SCN, to_char(sysdate,'DD-MON-RR HH24:MI:SS')
Current_Time  from v$database;
```

Other Useful Queries at the Standby

```
select client_process, process, thread#, sequence#, status
from v$managed_standby
where
(
client_process='LGWR'
or
process='MRP0'
)
/

select thread#, low_sequence#, high_sequence#
from v$archive_gap
/
(sample output
no rows selected
)
```

Remark : The query for v$archive_gap should show “no rows selected”. However, the result can be misleading if Redo Shipping is happening but Redo Apply is not happening. It does NOT confirm that Redo Apply is running. Redo Apply (i.e. actual Recovery of the Standby database) may have halted (e.g. if one or more datafiles is missing and/or Recovery has not been started or has been stopped after it was started). So you must also check the Redo Apply from the alert log messages at the Standby and from v$dataguard_status at the Standby.

```
select sequence#, to_char(first_time,'DD-MON-RR HH24:MI:SS')
First_Txn_Time,
to_char(completion_time,'DD-MON-RR HH24:MI:SS') Arch_Time,
applied Applied,
to_char(sysdate,'DD-MON-RR HH24:MI:SS') Current_Time
from v$archived_log
where sequence#  >=
(select max(sequence#)-5 from v$archived_log)
order by sequence#
/
(sample output :
SEQUENCE# FIRST_TXN_TIME             ARCH_TIME                  APPLIED   CURRENT_TIME
---------- -------------------------- -------------------------- --------- -----------------
```

```
       121 06-MAR-21 23:15:35       06-MAR-21 23:43:57       YES      07-MAR-21 23:20:50
       122 06-MAR-21 23:43:54       07-MAR-21 22:40:12       YES      07-MAR-21 23:20:50
       123 07-MAR-21 22:33:06       07-MAR-21 23:03:33       YES      07-MAR-21 23:20:50
       124 07-MAR-21 23:03:17       07-MAR-21 23:03:36       YES      07-MAR-21 23:20:50
       125 07-MAR-21 23:03:26       07-MAR-21 23:07:39       YES      07-MAR-21 23:20:50
       126 07-MAR-21 23:07:32       07-MAR-21 23:10:12       IN-MEMORY 07-MAR-21 23:20:50

6 rows selected.
)
```

```
alter session set nls_date_format='DD-MON-RR HH24:MI:SS';
select facility, severit, message_num, error_code, timestamp, message from
v$dataguard_status
where message_num >=
(select max(message_num)-5 from v$dataguard_status)
order by message_num
/
(sample output
FACILITY                SEVERITY      MESSAGE_NUM ERROR_CODE TIMESTAMP
----------------------- ------------- ----------- ---------- ------------------
MESSAGE
----------------------------------------------------------------------------------------------
--------------------------------------
Log Transport Services  Control                28          0 07-MAR-21 23:10:11
ARC1: Beginning to archive T-1.S-126 (3550119-3550458)

Log Transport Services  Control                29          0 07-MAR-21 23:10:12
ARC1: Completed archiving T-1.S-126 (0-0)

Log Apply Services      Informational          30          0 07-MAR-21 23:10:13
Media Recovery Log
/STANDBY/fast_recovery_area/stdb/STDB/archivelog/2021_03_07/o1_mf_1_124_j49tt8lp_.arc

Log Apply Services      Informational          31          0 07-MAR-21 23:10:15
Media Recovery Log
/STANDBY/fast_recovery_area/stdb/STDB/archivelog/2021_03_07/o1_mf_1_125_j49v1v1x_.arc

Log Apply Services      Informational          32          0 07-MAR-21 23:10:16
Media Recovery Log
/STANDBY/fast_recovery_area/stdb/STDB/archivelog/2021_03_07/o1_mf_1_126_j49v6m7b_.arc

Log Apply Services      Warning                33          0 07-MAR-21 23:10:18
Media Recovery Waiting for thread 1 sequence 127 (in transit)


6 rows selected.
)
```

```
select name,time_computed, value
from v$dataguard_stats
where name like '% lag'
order by name
/
(sample output
NAME                           TIME_COMPUTED                VALUE
------------------------------ ------------------------------ ------------------------------
----------------------------------
apply lag                      03/07/2021 23:30:29          +00 00:00:00
transport lag                  03/07/2021 23:30:29          +00 00:00:00
)
```

```
tti off
select current_scn Standby_SCN, to_char(sysdate,'DD-MON-RR HH24:MI:SS')
Current_Time  from v$database;
```

Refreshing Full Standby Database

Available Methods

If the Standby is lagging behind the Primary by a large number of hours or even days (e.g. if the Standby Server or Storage or Network link were down for a long time), the Standby can be refreshed over the network using either of four methods

a. [If all ArchiveLogs that are required are available] Backup the ArchiveLogs from the Primary manually, copy them to the Standby and issue the RECOVER DATABASE command at the Standby (See "Demonstration 3" at https://hemantoracledba.blogspot.com/2021/01/rmans-catalog-command.html)
b. Full Backup from Primary and Copy the backup to Standby, CATALOG the Backups and issue the RECOVER DATABASE command at the Standby
c. Backup with "FROM SCN …" clause from Primary and Copy the backup to the Standby, CATALOG the backup and issue the RECOVER DATABASE at the Standby
d. Issue the "RESTORE DATABASE FROM SERVICE" command at the Standby to automatically resync from the Primary See https://hemantoracledba.blogspot.com/2020/06/full-recovery-of-standby-database-over.html

Forced Delayed Shipping or Apply -- Advanced Topic

This can be done when you want build two or more Standby databases such that at least one Standby is lagging behind the Primary – e.g. by 1hour or 4hours. Such a scenario is useful in case of "user error. If a user accidentally deletes data at the Primary, by default, the delete is also applied to the Standby. However, if you have a second (or third) Standby that is deliberately lagging by 1hour or 4hours, it gives you enough time to stop this additional standby and recover the "deleted" data from the Standby.

See https://hemantoracledba.blogspot.com/2021/07/a-standby-that-lags-primary-by.html

Multiple Standbys, Cascaded Standby and Far Sync – Advanced Topic

Multiple Standbys allows you to build Standby Databases at multiple Data Centres and even some with Delayed Shipping or Delayed Apply (see the earlier section). Oracle Supports 30 Standby Databases.

Cascaded Standby can be used to build a Standby at a far distance where the network latency is very high. From the Primary Data Centre a Standby is built at a DC with low latency (i.e. closer) and from that Standby DC, another Standby is built further away (with higher latency). A US example would be Primary at New York, first Standby at Chicago and Cascaded Standby at Los Angeles. Thus, the data is available at all 3 sites but the NY site does not have to suffer the latency of shipping Redo across the longer distance to Los Angeles.

In recent versions, Oracle is now packing Far Sync where a near Standby is built on Synchronous replication (but requiring much lesser disk space as only the Control File and Archive Logs are present, no datafiles) and a further Standby "cascades" from this but also has the actual Data Files.

See https://hemantoracledba.blogspot.com/2021/07/creating-1-or-more-additional-standby.html for an example of Multiple Standbys

Partial Refresh – one or more datafiles

Note : Also see the sub-section "Refreshing Full Standby Database" under "Transaction Lag"

This may be required when one or more datafiles is corrupt or missing/deleted.

Loss of a single datafile will halt recovey of the whole database

There are two methods :

Using RMAN BACKUP, CATALOG, RESTORE and Resuming Recovery

Take an RMAN Backup of the datafile from the Primary, copy to the Standby, CATALOG and then RESTORE and RECOVER the datafile

Sequence of Actions for Recovery

Assume that the SYSAUX datafile in the Standby Database is lost (corrupt or mistakenly deleted)

The Standby alert log shows errors :

```
Errors in file
/u01/app/oracle/diag/rdbms/stdb/STDB/trace/STDB_dbw0_5612.trc:
ORA-01157: cannot identify/lock data file 3 - see DBWR trace file
ORA-01110: data file 3:
'/STANDBY/database/STDB/datafile/o1_mf_sysaux_j46kt553_.dbf'
ORA-27037: unable to obtain file status
Linux-x86_64 Error: 2: No such file or directory
Additional information: 7
MRP0: Background Media Recovery terminated with error 1110

Recovery Slave PR00 previously exited with exception 1110

MRP0: Background Media Recovery process shutdown (STDB)

Checker run found 1 new persistent data failures
```

(timestamps have been removed from the listing above to improve readability)

This query executed at the Standby also confirms that datafile 3 has an error

```
SQL> select file#, error
  2  from v$recover_file
  3  where error is not null
  4  /

     FILE# ERROR
---------- ----------------------------------------------------------------
         3 FILE NOT FOUND

SQL>
```

Datafile backup taken from Primary :

(File# 3 is identified from the error message in the Standby)

```
$rman target /
```

```
Recovery Manager: Release 12.2.0.1.0 - Production on Sun Mar 7 22:57:20
2021

Copyright (c) 1982, 2017, Oracle and/or its affiliates.  All rights
reserved.

connected to target database: ORCL12C (DBID=768045447)

RMAN> backup as compressed backupset datafile 3 format
'/tmp/df3_for_standby.bkp';

Starting backup at 07-MAR-21
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=32 device type=DISK
channel ORA_DISK_1: starting compressed full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00003
name=/u01/app/oracle/oradata/orcl12c/sysaux01.dbf
channel ORA_DISK_1: starting piece 1 at 07-MAR-21
channel ORA_DISK_1: finished piece 1 at 07-MAR-21
piece handle=/tmp/df3_for_standby.bkp tag=TAG20210307T225813 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:25
Finished backup at 07-MAR-21

Starting Control File and SPFILE Autobackup at 07-MAR-21
piece
handle=/u01/app/oracle/fast_recovery_area/orcl12c/ORCL12C/autobackup/2021_0
3_07/o1_mf_s_1066604319_j49tk24c_.bkp comment=NONE
Finished Control File and SPFILE Autobackup at 07-MAR-21

RMAN>
```

It is then copied to the Standby Server (location folder /tmp/From_Primary)

Actions on the Standby Database

```
$rman target /

Recovery Manager: Release 12.2.0.1.0 - Production on Sun Mar 7 23:06:29
2021

Copyright (c) 1982, 2017, Oracle and/or its affiliates.  All rights
reserved.

connected to target database: ORCL12C (DBID=768045447, not open)

RMAN> catalog start with '/tmp/From_Primary';

using target database control file instead of recovery catalog
searching for all files that match the pattern /tmp/From_Primary

List of Files Unknown to the Database
=====================================
File Name: /tmp/From_Primary/df3_for_standby.bkp

Do you really want to catalog the above files (enter YES or NO)? YES
cataloging files...
cataloging done

List of Cataloged Files
```

```
=======================
File Name: /tmp/From_Primary/df3_for_standby.bkp

RMAN>
RMAN> restore datafile 3;

Starting restore at 07-MAR-21
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=185 device type=DISK

channel ORA_DISK_1: starting datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
channel ORA_DISK_1: restoring datafile 00003 to
/STANDBY/database/STDB/datafile/o1_mf_sysaux_j46kt553_.dbf
channel ORA_DISK_1: reading from backup piece
/tmp/From_Primary/df3_for_standby.bkp
channel ORA_DISK_1: piece handle=/tmp/From_Primary/df3_for_standby.bkp
tag=TAG20210307T225813
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:25
Finished restore at 07-MAR-21

RMAN>
```

Messages in the Standby alert log

```
Full restore complete of datafile 3
/STANDBY/database/STDB/datafile/o1_mf_sysaux_j49v16z5_.dbf.  Elapsed time:
0:00:14
  checkpoint is 3547268
  last deallocation scn is 3519705
```

Resume Recovery at the Standby

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

Database altered.

SQL>
```

Messages in the Standby alert log

```
alter database recover managed standby database disconnect from session
Attempt to start background Managed Standby Recovery process (STDB)
Starting background process MRP0
MRP0 started with pid=43, OS id=6216
MRP0: Background Managed Standby Recovery process started (STDB)
Started logmerger process
Managed Standby Recovery starting Real Time Apply
Parallel Media Recovery started with 2 slaves
Completed: alter database recover managed standby database disconnect from
session
Media Recovery Log
/STANDBY/fast_recovery_area/stdb/STDB/archivelog/2021_03_07/o1_mf_1_123_j49
tsqkq_.arc
RFS[2]: Selected log 5 for T-1.S-127 dbid 768045447 branch 937554761
Archived Log entry 23 added for T-1.S-126 ID 0x2dc76487 LAD:1
Media Recovery Log
/STANDBY/fast_recovery_area/stdb/STDB/archivelog/2021_03_07/o1_mf_1_124_j49
tt8lp_.arc
Media Recovery Log
/STANDBY/fast_recovery_area/stdb/STDB/archivelog/2021_03_07/o1_mf_1_125_j49
v1v1x_.arc
```

```
Media Recovery Log
/STANDBY/fast_recovery_area/stdb/STDB/archivelog/2021_03_07/o1_mf_1_126_j49
v6m7b_.arc
Media Recovery Waiting for thread 1 sequence 127 (in transit)
Recovery of Online Redo Log: Thread 1 Group 5 Seq 127 Reading mem 0
  Mem# 0: /STANDBY/database/STDB/onlinelog/o1_mf_5_j46l4pyq_.log
  Mem# 1:
/STANDBY/fast_recovery_area/stdb/STDB/onlinelog/o1_mf_5_j46l4tvl_.log
```

(timestamps have been removed from the listing to improve readability)

The “Media Recovery Log” messages are for ArchiveLogs Sequences 123 to 126 that were generated on the Primary while Database Recovery had halted at the Standby. (**Loss of a single datafile will halt recovey of the whole database**)
Once it has applied all the ArchiveLogs, it has completed a catch-up and is now using the Online Redo Log.

This method can also be used to restore a corrupt datafile at the Primary by taking a backup from the Standby and applying it to the Primary. The last backup of the datafile at the Primary may have been a number of hours ago (e.g. upto 24hours if running daily Full / Level-0 backup) and may need time to apply ArchiveLogs in the RECOVER phase. On the other hand, the backup copy from the Standby may be lagging by only a few seconds before the time of the loss/corruption of the datafile at the Primary, so it might be quicker to backup the datafile from the Standby, copy it over the network to the Primary and restore it at the Primary.

See https://hemantoracledba.blogspot.com/2021/01/rmans-catalog-command.html (Demonstration 3 in this post) which shows backup from Standby taken to the Primary

Using RMAN RESTORE .. FROM SERVICE

Use RMAN’s “RESTORE DATAFILE FROM SERVICE” command. This method is available only from 19c onwards. See https://hemantoracledba.blogspot.com/2020/05/restoring-lost-datafile-on-standby.html which shows RMAN restoring a Primary Datafile to the Standby

ArchiveLog Deletion Policy

The ArchiveLog Deletion Policy ensures that RMAN DELETE commands do NOT delete ArchiveLogs at the Primary until and unless they have been applied at the Standby
See https://hemantoracledba.blogspot.com/2020/07/archive-log-deletion-policy-with.html

Opening a Standby for Access

Read Only with Stopped Recovery

This method does not require any additional licencing.
Issue the commands

```
SQL> alter database recover managed standby database cancel
SQL> alter database open read only
```

This allows developers/DBAs to query the database. However, since Recovery is stopped, you must consider the issue of "Transaction Lag"

Read Only with Continued Recovery ("Active DataGuard")

This method requires the Active DataGuard licence
Issue the commands

```
SQL> alter database recover managed standby database cancel
SQL> alter database open read only
SQL> alter database recover managed standby database disconnect from
session
```

With the Active DataGuard licence, the database can continue to apply Redo and yet allow developers/DBAs to query for updated data.

Read Only with DML Support ["DML Redirection"]

Available in 18c/19c with the Active DataGuard licence
See https://hemantoracledba.blogspot.com/2020/11/active-data-guard-and-dml-redirection.html

Backups of the Standby

The Standby Database can have independent RMAN Backups and also independently manage the FRA for Backups and ArchiveLogs.

See https://hemantoracledba.blogspot.com/2020/05/rman-backup-of-standby-database.html

Graceful Switchover

On Primary Database

```
CONNECT / AS SYSDBA

ALTER DATABASE COMMIT TO SWITCHOVER TO STANDBY;

SHUTDOWN IMMEDIATE;

STARTUP NOMOUNT;

ALTER DATABASE MOUNT STANDBY DATABASE;

ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT FROM SESSION;
```

DataGuard Broker Command

`switchover to STDBY;` - if the registered name of the Standby is “STBY”

On the original standby database

issue the following commands to convert standby database to primary

```
CONNECT / AS SYSDBA

ALTER DATABASE COMMIT TO SWITCHOVER TO PRIMARY;

SHUTDOWN IMMEDIATE;

STARTUP;
```

Failover with DataLoss

On the standby database when the Primary is *unavailable*

ALTER DATABASE RECOVER MANAGED STANDBY DATABASE FINISH;

ALTER DATABASE ACTIVATE STANDBY DATABASE;

DataGuard Broker Command

`failover to STDBY;` - if the registered name of the Standby is “STBY”

Opening the Standby for “Sandbox” testing

Note : This requires “db_recovery_file_dest_size” and “db_recovery_file_dest” to be configure for the Guaranteed Restore Point even if they are not used for normal Flashback Logging or Archive Logs or Backups

Create a GRP at the Standby

```
Standby > recover managed standby database cancel;

Standby > create restore point forsandbox guarantee flashback database;
```

Disable Redo Shipping from the Primary

```
Primary > alter system archive log current;

Primary > alter system set log_archive_dest_state_2=’DEFER’;
```

OPEN the Standby in Read-Write Mode

```
Standby > alter database set standby database to maximize performance;

Standby > alter database activate standby database;

Standby > alter database open;


... use your Standby READ WRITE as you like, then ...
```

Revert after completing Testing

```
Standby > startup mount force;

Standby > flashback database to restore point forsandbox;

Standby > alter database convert to physical standby;

Standby > startup mount force;
```

Re-enable Redo Shipping from the Primary

```
Primary > alter system set log_archive_dest_state_2=’enable’;
```

See the section on Transaction Lag for monitoring the Redo Shipping Status or to Refresh the Full Standby if the lag is very high

List of Blog Posts

A number of my Blog Posts on Oracle DataGuard and Standby Databases are listed here :

https://hemantoracledba.blogspot.com/2021/10/my-posts-on-standby-databases-data-guard.html

Reference : Oracle Documentation on the SET STANDBY TO MAXIMIZE clause

maximize_standby_db_clause

Use this clause to specify the level of protection for the data in your database environment. You specify this clause from the primary database.

Note:

The `PROTECTED` and `UNPROTECTED` keywords have been replaced for clarity but are still supported. `PROTECTED` is equivalent to `TO MAXIMIZE PROTECTION`. `UNPROTECTED` is equivalent to `TO MAXIMIZE PERFORMANCE`.

TO MAXIMIZE PROTECTION

This setting establishes **maximum protection mode** and offers the highest level of data protection. A transaction does not commit until all data needed to recover that transaction has been written to at least one physical standby database that is configured to use the `SYNC` log transport mode. If the primary database is unable to write the redo records to at least one such standby database, then the primary database is shut down. This mode guarantees zero data loss, but it has the greatest potential impact on the performance and availability of the primary database.

Restriction on Establishing Maximum Protection Mode

You can specify `TO MAXIMIZE PROTECTION` on an open database only if the current data protection mode is `MAXIMUM AVAILABILITY` and there is at least one synchronized standby database.

TO MAXIMIZE AVAILABILITY

This setting establishes **maximum availability mode** and offers the next highest level of data protection. A transaction does not commit until all data needed to recover that transaction has been written to at least one physical or logical standby database that is configured to use the `SYNC` log transport mode. Unlike maximum protection mode, the primary database does not shut down if it is unable to write the redo records to at least one such standby database. Instead, the protection is lowered to maximum performance mode until the fault has been corrected and the standby database has caught up with the primary database. This mode guarantees zero data loss unless the primary database fails while in maximum performance mode. Maximum availability mode provides the highest level of data protection that is possible without affecting the availability of the primary database.

TO MAXIMIZE PERFORMANCE

This setting establishes **maximum performance mode** and is the default setting. A transaction commits before the data needed to recover that transaction has been written to a standby database. Therefore, some transactions may be lost if the primary database fails and you are unable to recover the redo records from the primary database. This mode provides the highest level of data protection that is possible without affecting the performance of the primary database.

To determine the current mode of the database, query the `PROTECTION_MODE` column of the `V$DATABASE` dynamic performance view.