**How to Roll Forward a Standby Database Using Recover Database From Service (Doc ID 2850185.1)**

**Example:**

**HEPYDEV2\_xhepydbw02d - Primary database**

**HEPYDEV2\_xhepydbw04d - Physical standby database**

**Simulate gap by removing some archivelog from primary.**

**From HEPYDEV2\_xhepydbw02d - Primary database**

dgmgrl /

edit database 'HEPYDEV2\_xhepydbw02d' set state=trasport-off;

show database verbose 'HEPYDEV2\_xhepydbw02d';

sqlplus / as sysdba

alter system switch logfile;

. oraenv

+ASM

asmcmd

cd +ARCH/HEPYDEV2\_XHEPYDBW02D/ARCHIVELOG/2024\_12\_23

ls

mv thread\_1\_seq\_63279.256.xxxx thread\_1\_seq\_63279.256.BKP\_1

mv thread\_1\_seq\_63279.257.xxxx thread\_1\_seq\_63279.25&.BKP\_1

dgmgrl /

edit database 'HEPYDEV2\_xhepydbw02d' set state=transport-on;

show database verbose 'HEPYDEV2\_xhepydbw02d';

**From HEPYDEV2\_xhepydbw04d - Physical standby database**

**Should report some gap**

select \* from v$archive\_gap;

**(Doc ID 2850185.1)**

**Stop Managed Recovery and Redo Transport**

via Data Guard Broker connected to any database in the configuration:

DGMGRL> edit database <primary> set state=TRANSPORT-OFF;  
Succeeded.  
  
DGMGRL> edit database <standby> set state=APPLY-OFF;  
Succeeded.

**From HEPYDEV2\_xhepydbw02d - Primary database**

dgmgrl /

edit database 'HEPYDEV2\_xhepydbw02d' set state=transport-off;

show database verbose 'HEPYDEV2\_xhepydbw02d';

**From HEPYDEV2\_xhepydbw04d - Physical standby database**

dgmgrl /

edit database 'HEPYDEV2\_xhepydbw04d' set state=apply-off;

show database verbose 'HEPYDEV2\_xhepydbw04d';

OR

via SQLPLUS (if Data Guard Broker is not configured):

From the Primary database:

SQL> alter system set LOG\_ARCHIVE\_DEST\_STATE\_<#>=DEFER;

System altered.

SQL> alter system archive log current;

From the Standby database:

SQL> recover managed standby database cancel;  
Media recovery complete.

**Save Existing Logfile Names**

This roll forward process will create new online redo logs and standby redo logs, leaving the existing logs as orphans using up space.  Save the logfile names with the following query to be removed later in the process.

SQL> set heading off linesize 999 pagesize 0 feedback off trimspool on  
SQL> spool /tmp/delete\_logfiles.log  
SQL> select member from v$logfile;  
SQL> spool off

**From HEPYDEV2\_xhepydbw02d - Primary database**

set heading off linesize 999 pagesize 0 feedback off trimspool on

spool /tmp/delete\_logfiles\_Primary.log

select member from v$logfile;

spool off

**From HEPYDEV2\_xhepydbw04d - Physical standby database**

set heading off linesize 999 pagesize 0 feedback off trimspool on

spool /tmp/delete\_logfiles\_Stdby.log

select member from v$logfile;

spool off

**Refresh Standby Control File From Primary**

The standby database control file must be refreshed to advance the control file SCN.  This new standby control file will contain information about any new files added to the database since the current standby SCN.  Additional steps are described to update the locations of the standby files in the control file and restore any of those newly added files.

***Document the Current SCN for the Standby***

Run the command in the example below on the standby database and save the SCN for later use.

RMAN> select current\_scn from v$database;

CURRENT\_SCN#  
------------------  
644203931

**From HEPYDEV2\_xhepydbw04d - Physical standby database**

***SET NUMWIDTH 20***

***select current\_scn from v$database;***

***Save Current RMAN Configuration Settings***

Refreshing the standby control file from the primary control file overwrites the RMAN configuration items specific to the standby database.

Create a script to restore the current RMAN configuration settings.  This will be used after the control file is restored from the primary.

NOTE: This step is when using target database control file instead of recovery catalog

On the standby:

$ $ rman target / nocatalog log=/tmp/RMAN\_settings.log <<EOF  
show all;  
EOF

To remove the unnecessary commands output and the RETENTION POLICY command, which cannot be executed on a standby, run the following:

$ grep ^CONFIGURE /tmp/RMAN\_settings.log | grep -v 'RETENTION POLICY' >/tmp/RMAN\_settings.rman

$ rm /tmp/RMAN\_settings.log

***rmanc***

***SPOOL LOG TO*** /tmp/RMAN\_settings.log

***show all;***

***spool off***

***Refresh the Standby Control File***

Start one instance in nomount and restore the control file from the primary database.

$ srvctl stop database -d <db> -o immediate

$ rman target / nocatalog

RMAN> startup nomount

RMAN> restore standby controlfile from service <tns alias for primary database>;

Starting restore at <date>  
allocated channel: ORA\_DISK\_1  
channel ORA\_DISK\_1: SID=2824 instance=<standby instance> device type=DISK

channel ORA\_DISK\_1: starting datafile backup set restore  
channel ORA\_DISK\_1: using network backup set from service <primary service>  
channel ORA\_DISK\_1: restoring control file  
channel ORA\_DISK\_1: restore complete, elapsed time: 00:00:10  
output file name=<standby control file>  
Finished restore at <date>

**From HEPYDEV2\_xhepydbw04d - Physical standby database**

***srvctl stop database -d HEPYDEV2\_xhepydbw04d -o immediate***

***rman target / nocatalog***

***startup nomount***

***restore standby controlfile from service HEPYDEV2\_xhepydbw02d;***

***Replace RMAN Configuration Settings and Remove Orphaned Online and Standby Redo Log Files***

Mount the database and replace the RMAN configuration with the saved script

RMAN> alter database mount;

released channel: ORA\_DISK\_1  
Statement processed

RMAN> @/tmp/RMAN\_settings.rman

<output from CONFIGURE commands in the script>

RMAN> exit

$ rm /tmp/RMAN\_settings.rman

Then remove the orphaned log files saved in */tmp/delete\_logfiles.log* and delete the */tmp/delete\_logfiles.log*file

RMAN> alter database mount;

show all;

--> if something diff restore rman settings

RMAN> @/tmp/RMAN\_settings.rman

***Catalog Standby Database Files to the Refreshed Control File***

At this point, the refreshed control file has file locations and metadata from the primary database.  Use the RMAN CATALOG command to update the control file with standby database file locations.

RMAN> catalog start with '<DATA DISKGROUP>/<standby db\_unique\_name/';

Starting implicit crosscheck backup at 24-FEB-22

<...listing of files found...>

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

[There may be error reported for files which cannot be cataloged.  These can be ignored.]

RMAN> catalog start with '<RECO DISKGROUP>/<standby db\_unique\_name>/';

searching for all files that match the pattern <RECO DISKGROUP>/<standby db\_unique\_name>  
no files found to be unknown to the database    ***<-- often times there are no files found on RECO***

NOTE: If the standby was used as a snapshot standby check that the standby controlfile incarnation matches that of the primary.  It is possible that the incarnation was not set correctly during the catalog command, especially if backups were taken during the snapshot database and not deleted.   
Use RMAN> LIST INCARNATION on each database to compare incarnations and RMAN> RESET DATABASE INCARNATION TO x to set the standby database incarnation if necessary.  Cleaning up any backups from orphaned incarnations before executing this process is recommended.

***RMAN> catalog start with '+DATA1/HEPYDEV2\_XHEPYDBW04/';***

***RMAN> catalog start with '+REDO1/HEPYDEV2\_XHEPYDBW04D/';***

***RMAN> catalog start with '+REDO2/HEPYDEV2\_XHEPYDBW04D/';***

***Restore Any Missing Files***

Files would be missing if they were created between the SCN of the standby and the SCN of the refreshed control file.

Using the SCN documented in step 'Document the Current SCN for the Standby', identify any missing files and restore them.

RMAN> select file# from v$datafile where creation\_change# >= 644203931;

* If files are returned by the query, files must be restored from the primary and the existing files.

**Restore Missing Datatfiles**

RMAN> run {  
allocate channel c1 type disk;  
allocate channel c2 type disk;  
allocate channel c3 type disk;  
allocate channel c4 type disk;  
allocate channel c5 type disk;  
allocate channel c6 type disk;  
allocate channel c7 type disk;  
allocate channel c8 type disk;  
set newname for database to NEW;  
restore datafile <comma separate list of files> from service <tns alias for primary database> section size <section size>;  
}

NOTE: For SECTION SIZE, query the primary database datafile sizes for those files being restored.   
If the largest file is

* <15TB use section size of 64GB
* >15TB and <30TB used section size of 128G
* >30TB and <60TB used section size of 256G
* >60TB use section size of 512G

NOTE: For cases where the primary database is not encrypted and the standby in encrypted, the files should be restored using the AS ENCRYPTED clause. This clause is not valid on RESTORE DATAFILE so the tablespace(s) of the missing datafile(s) must be restored using RESTORE TABLESPACE instead and incorporate the 'AS ENCRYPTED' clause.

For example: RESTORE TABLESPACE <tablespace name> FROM SERVICE <tns alias for primary database> SECTION SIZE <section size> AS ENCRYPTED;

***Switch Database to Copy***

To make the restored and pre-existing standby datafile locations permanent in the controlfile, switch to the cataloged copies of the datafiles.

RMAN> switch database to copy;

RMAN> switch database to copy;

***Clean Up Orphaned Files***

In the event that any file was deleted between the SCN of the standby and the SCN of the refreshed control file it will remain a datafilecopy in RMAN.  Remove any datafilecopy to remove the unneeded file.

RMAN> list datafilecopy all;

using target database control file instead of recovery catalog  
List of Datafile Copies  
=======================

Key File S Completion Time Ckp SCN Ckp Time Sparse  
------- ---- - --------------- ---------- --------------- ------  
<#> <#> A <date> <#> <date> NO  
Name: <orphaned datafilecopy name>

RMAN> delete datafilecopy all;

allocated channel: ORA\_DISK\_1  
channel ORA\_DISK\_1: SID=2620 instance=<instance name> device type=DISK  
allocated channel: ORA\_DISK\_2  
channel ORA\_DISK\_2: SID=1737 instance=<instance name> device type=DISK  
List of Datafile Copies  
=======================

Key File S Completion Time Ckp SCN Ckp Time Sparse  
------- ---- - --------------- ---------- --------------- ------  
<#> <#> A <date> <#> <date> NO  
Name: <orphaned datafilecopy name>

Do you really want to delete the above objects (enter YES or NO)? YES  
deleted datafile copy  
datafile copy file name=<orphaned datafilecopy name> RECID=<#> STAMP=<#>  
Deleted 1 objects

**Clear Online Redo Logs and Standby Redo Logs**

Execute the following query in SQL\*PLUS on the standby to create new logfiles.

NOTE: The code below clears Standby Redo Logs only.  When managed recovery starts for the first time, it will clear the Online Redo Logs for the standby database automatically as long as the Oracle Managed Files (OMF) naming convention is used on the primary and standby databases.  If OMF is not used, the Online Redo Logs for the standby database should be manually created by the user during instantiation.

SQL> begin

for log\_cur in ( select group# group\_no from v$standby\_log )  
loop  
execute immediate 'alter database clear logfile group '||log\_cur.group\_no;  
end loop;  
end;  
/

**From HEPYDEV2\_xhepydbw04d - Physical standby database**

**begin**

**for log\_cur in ( select group# group\_no from v$standby\_log )**

**loop**

**execute immediate 'alter database clear logfile group '||log\_cur.group\_no;**

**end loop;**

**end;**

**/**

**Roll the Standby Database Forward**

The standby database is now prepared to execute the recover from service incremental roll forward.

***Restart All Instances to Mount***

All instances can be used during the recovery.  restart the database mounting all instances.

$ srvctl stop database -db <dbname> -o immediate

$ srvctl start database -db <dbname> -o mount

**From HEPYDEV2\_xhepydbw04d - Physical standby database**

***srvctl stop database -d HEPYDEV2\_xhepydbw04d -o immediate***

***srvctl start database -d HEPYDEV2\_xhepydbw04d -o mount***

***Re-Enable Redo Transport***

Archived logs created during the roll forward will be needed to make the database consistent.  It is more efficient to let the primary ship the redo to the standby while the roll forward is running than to wait for those logs to be transported at the end of the roll forward.

via Data Guard Broker from either database:

DGMGRL> edit database <primary> set state=TRANSPORT-ON;  
Succeeded.

**From HEPYDEV2\_xhepydbw02d - Primary database**

dgmgrl /

edit database 'HEPYDEV2\_xhepydbw02d' set state=trasport-on;

show database verbose 'HEPYDEV2\_xhepydbw02d';

via SQLPLUS From the Primary database:

SQL> alter system set LOG\_ARCHIVE\_DEST\_STATE\_<#>=ENABLE;

System altered.

NOTE: Do NOT start managed recovery at this point.

***Determine Section Size to be Used***

On the primary, query the largest datafile size to determine the section size to be used for the recover command.

SQL> select max(bytes)/1073741824 GB from v$datafile;

If the largest file is:

* <15TB use section size of 64GB
* >15TB and <30TB used section size of 128G
* >30TB and <60TB used section size of 256G
* >60TB use section size of 512G

**From HEPYDEV2\_xhepydbw02d - Primary database**

***select max(bytes)/1073741824 GB from v$datafile;***

***Run Recover From Service***

The recover command can utilize all instances of the standby during recovery.  This approach spreads the resource utilization across hardware and potentially increases parallelization and throughput for the process as a whole.  The example below utilizes this method but parallelization where all channels run on one instance can also be used.

$ rman target sys/<password>  **<- It is necessary to connect with the password**

RMAN > run {  
allocate channel c1 type disk connect '/@<standby instance 1 SID\_NAME>';  
allocate channel c2 type disk connect '/@<standby instance 1 SID\_NAME>';  
allocate channel c3 type disk connect '/@<standby instance 1 SID\_NAME>';  
allocate channel c4 type disk connect '/@<standby instance 1 SID\_NAME>';  
allocate channel c5 type disk connect '/@<standby instance 2 SID\_NAME>';  
allocate channel c6 type disk connect '/@<standby instance 2 SID\_NAME>';  
allocate channel c7 type disk connect '/@<standby instance 2 SID\_NAME>';  
allocate channel c8 type disk connect '/@<standby instance 2 SID\_NAME>';  
recover database from service '<primary unique name>' section size <section size>;  
}

NOTE: Additional channels can be allocated per instance or on additional instances.

**From HEPYDEV2\_xhepydbw04d - Physical standby database**

rman target sys/<password>

run {

recover database from service 'HEPYDEV2\_xhepydbw02d' section size 128G;

}

Estimating the time which the recover will take is difficult due to the number of variables however, once started, progress and estimates can be monitored with GV$SESSION\_LONGOPS on the primary database.

select INST\_ID,SID,SERIAL#,OPNAME, (sofar/totalwork)\*100 as "%complete", ELAPSED\_SECONDS, TIME\_REMAINING  
from gv$session\_longops  
where sofar<>totalwork  
and totalwork<>0  
and SID||SERIAL# in (select SID||SERIAL# from v$session);

INST\_ID SID SERIAL# OPNAME %complete ELAPSED\_SECONDS TIME\_REMAINING

---------- ---------- ---------- ---------------------------------------------------------------- ---------- --------------- --------------

1 1177 56089 RMAN: incremental datafile backup 2.36167908 55 2274

1 2415 52071 RMAN: incremental datafile backup 12.9760623 394 2642

1 2541 26066 RMAN: incremental datafile backup .274372101 22 7996

1 2808 18883 RMAN: incremental datafile backup .262057781 21 7992

1 2670 65170 RMAN: incremental datafile backup 8.32093954 386 4253

1 1900 48392 RMAN: incremental datafile backup 5.93750477 197 3121

***Recover Until Consistent***

At the completion of the recover database from service command, in order to re-enable flashback database and open the standby read-only more recovery is required to make the database consistent, meaning the control file and all datafiles are at the same SCN.  The amount of recovery required will be a function of how long the initial recovery took and how active the primary was during that time.

First switch a log on the primary database to archive the last of the redo during the recover command.

From the primary:

SQL> alter system archive log current;

**From HEPYDEV2\_xhepydbw02d - Primary database**

alter system archive log current;

Then from **SQL\*PLUS** on the standby issue the command below.  The UNTIL CONSISTENT clause cannot be used in RMAN.

SQL> recover automatic standby database until consistent;  
Media recovery complete.

NOTE: If the recover until consistent results in the error below, all required logs are not present at the standby.

ORA-01547: warning: RECOVER succeeded but OPEN RESETLOGS would get error below  
ORA-01152: file 1 was not restored from a sufficiently old backup

Monitor the standby alert logs and as logs arrive at the standby, reissue the recover until consistent until it complete successfully as indicated by 'Media recovery complete.'

**From HEPYDEV2\_xhepydbw04d - Physical standby database**

recover automatic standby database until consistent;

**Re-Enable Flashback Database**

Restoring the standby control file automatically disables flashback.  Once the database is in a consistent state, flashback can be re-enabled.

SQL> alter database flashback on;

Database altered.

**Restart the Standby and Managed Recovery**

Once the standby database is consistent, it can be opened read-only.  Restart the database to open all instances read-only.

$ srvctl stop database -db <dbname> -o immediate

$ srvctl start database -db <dbname> -o 'read only'

**From HEPYDEV2\_xhepydbw04d - Physical standby database**

srvctl stop database -d HEPYDEV2\_xhepydbw04d -o immediate

srvctl start database -d HEPYDEV2\_xhepydbw04d -o 'read only'

Then restart managed recovery

via Data Guard Broker from either database:

DGMGRL> edit database <standby> set state=APPLY-ON;  
Succeeded.

**From HEPYDEV2\_xhepydbw04d - Physical standby database**

dgmgrl /

edit database 'HEPYDEV2\_xhepydbw04d' set state=apply-off;

show database verbose 'HEPYDEV2\_xhepydbw04d';

OR

via SQLPLUS (if Data Guard Broker is not configured):

From the Standby database:

SQL> recover managed standby database disconnect;

**Take a New Backup (If Needed)**

After the roll forward, if the standby database is being backed up, a new backup is suggested.