**Rollback databases during the HRP Application Upgrade**

--🡪 PY

Shutdown the **Primary** Database

sqlplus / as sysdba

sql => shutdown immediate;

OR

run => srvctl stop database -d HEPYPRD\_xhepydbm21p -o immediate

Startup mount the **Primary** Database

sqlplus / as sysdba

sql => startup mount;

OR

run => srvctl start database -d HEPYPRD\_xhepydbm21p -o mount

Flashback the **Primary** Database to the desired Restore Point:

sqlplus / as sysdba

sql => flashback database to restore point HEPYPRD\_Primary\_flashback\_YYYYMMDD;

Open the **primary** database READ ONLY to confirm restore time

sqlplus / as sysdba

sql => alter database open read only;

sql => select \* from aedba.rman\_heartbeat;

Confirm restored to appropriate timeframe

DBNAME TIMESTMP

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

Open the **primary** database with RESETLOGS (if above restore time was correct)

sqlplus / as sysdba

sql => shutdown immediate;

sql => startup mount;

sql => alter database open resetlogs;

Flashback the Standby Database to the desired Restore Point: (Will need to mount the database if this is an ADG standby)

sqlplus / as sysdba

sql => flashback database to restore point HEPYPRD\_Standby\_flashback\_YYYYMMDD;

-- double check **Standby**

alter database open read only;

SQL> select \* from aedba.rman\_heartbeat;

DBNAME TIMESTMP

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

sqlplus / as sysdba

sql => shutdown immediate;

sql => startup mount;

Enable Log Transport Services to the Standby Database from the **Primary** server

$SCRIPTS/ENable\_log\_shipping.sh HEPYPRD

Perform a Log Switch on **Primary** Database

sqlplus / as sysdba

sql => alter system switch logfile;

sql => select group#,thread#,sequence# from v$log where status = 'CURRENT';

Confirm log sent to standby server

run => view $BDUMP/alert\_HEPYPRD.log

Once Log Transport Services is running again, enable Managed Recovery on the **Standby** Database:

dgmgrl /

edit database 'HEPYPRD\_xhepydbw21p' set state = 'APPLY-ON';

show database verbose 'HEPYPRD\_xhepydbw21p';

Verify the Standby Database is now following the Primary Database into the new Incarnation

Run the following against the primary and standby databases

sqlplus / as sysdba

select incarnation# from V$DATABASE\_INCARNATION where status = 'CURRENT';

Check BCT

SELECT status, filename FROM V$BLOCK\_CHANGE\_TRACKING;

May be needed

ALTER DATABASE ENABLE BLOCK CHANGE TRACKING;

**Drop restore points**

Primary:

sqlplus / as sysdba

sql => DROP RESTORE POINT HEPYPRD\_Primary\_flashback\_YYYYMMDD;

sql => select name,scn,time from v$restore\_point where guarantee\_flashback\_database='YES';

no rows selected

sql => select flashback\_on from v$database;

FLASHBACK\_ON

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

NO

Standby:

sqlplus / as sysdba

sql => DROP RESTORE POINT HEPYPRD\_Standby\_flashback\_YYYYMMDD;

sql => select name,scn,time from v$restore\_point where guarantee\_flashback\_database='YES';

no rows selected

sql => select flashback\_on from v$database;

FLASHBACK\_ON

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

NO

Resync RMAN catalog from primary server while connected to the primary database and catalog

rmanc

rman => resync catalog;

new incarnation of database registered in recovery catalog

starting full resync of recovery catalog

Take an archivelog backup (Mimic prod OEM activity)

/orahome/u01/app/oracle/local/scripts/backup\_archive\_logs.sh HEPYPRD

Take a differential level 1 database backup (optional)

Resume OEM archivelog backup job BACKUP\_HEPYPRD\_XHEPYDBM21P\_ARCHIVE\_LOG for primary database

Resume OEM standby archive delete job HEPYPRD\_STDBY2\_ARCHIVE\_LOG\_PURGE for standby database

Uncomment heartbeat in cron on primary server

Uncomment PY stats job

--🡪DW

Shutdown the **Primary** Database

sqlplus / as sysdba

sql => shutdown immediate;

OR

run => srvctl stop database -d HEDWPRD\_xhedwdbm21p -o immediate

Startup mount the **Primary** Database

sqlplus / as sysdba

sql => startup mount;

OR

run => srvctl start database -d HEDWPRD\_xhedwdbm21p -o mount

Flashback the Primary Database to the desired Restore Point:

sqlplus / as sysdba

sql => flashback database to restore point HEDWPRD\_Primary\_flashback\_YYYYMMDD;

Open the **primary** database READ ONLY to confirm restore time

sqlplus / as sysdba

sql => alter database open read only;

sql => select \* from aedba.rman\_heartbeat;

Confirm restored to appropriate timeframe

DBNAME TIMESTMP

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

Open the primary database with RESETLOGS (if above restore time was correct)

sqlplus / as sysdba

sql => shutdown immediate;

sql => startup mount;

sql => alter database open resetlogs;

Flashback the Standby Database to the desired Restore Point: (Will need to mount the database if this is an ADG standby)

sqlplus / as sysdba

sql => shutdown immediate;

sql => startup mount;

sqlplus / as sysdba

sql => flashback database to restore point HEDWPRD\_Standby\_flashback\_YYYYMMDD;

-- double check Standby

alter database open read only;

SQL> select \* from aedba.rman\_heartbeat;

DBNAME TIMESTMP

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

sqlplus / as sysdba

sql => shutdown immediate;

sql => startup mount;

Enable Log Transport Services to the Standby Database from the Primary server

$SCRIPTS/ENable\_log\_shipping.sh HEDWPRD

Perform a Log Switch on Primary Database

sqlplus / as sysdba

sql => alter system switch logfile;

sql => select group#,thread#,sequence# from v$log where status = 'CURRENT';

Confirm log sent to standby server

run => view $BDUMP/alert\_HEDWPRD.log

Once Log Transport Services is running again, enable Managed Recovery on the Standby Database:

dgmgrl /

edit database 'HEDWPRD\_xhedwdbw21p' set state = 'APPLY-ON';

show database verbose 'HEDWPRD\_xhedwdbw21p';

Verify the Standby Database is now following the Primary Database into the new Incarnation

Run the following against the primary and standby databases

sqlplus / as sysdba

select incarnation# from V$DATABASE\_INCARNATION where status = 'CURRENT';

Check BCT

SELECT status, filename FROM V$BLOCK\_CHANGE\_TRACKING;

May be needed

ALTER DATABASE ENABLE BLOCK CHANGE TRACKING;

**Drop restore points**

**Primary**:

sqlplus / as sysdba

sql => DROP RESTORE POINT HEDWPRD\_Primary\_flashback\_YYYYMMDD;

sql => select name,scn,time from v$restore\_point where guarantee\_flashback\_database='YES';

no rows selected

sql => select flashback\_on from v$database;

FLASHBACK\_ON

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

NO

Standby:

sqlplus / as sysdba

sql => DROP RESTORE POINT HEDWPRD\_Standby\_flashback\_YYYYMMDD;

sql => select name,scn,time from v$restore\_point where guarantee\_flashback\_database='YES';

no rows selected

sql => select flashback\_on from v$database;

FLASHBACK\_ON

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

NO

Resync RMAN catalog from primary server while connected to the primary database and catalog

rmanc

rman => resync catalog;

new incarnation of database registered in recovery catalog

starting full resync of recovery catalog

Take an archivelog backup (Mimic prod OEM activity)

/orahome/u01/app/oracle/local/scripts/backup\_archive\_logs.sh HEDWPRD

Take a differential level 1 database backup (optional)

!! !! Run **UNLOCKUSER** procedure in HEDWPRD

!! Uncomment DW stats crontab jobs

Resume OEM archivelog backup job BACKUP\_HEDWPRD\_XHEDWDBM21P\_ARCHIVED\_LOGS for primary database

Resume OEM standby archive delete job HEDWPRD\_STDBY2\_ARCHIVE\_LOG\_PURGE for standby database

Uncomment heartbeat in cron on primary server