**PY and DW Friday/Saturday pre steps**

-- Comment out PY and DW crontab stats job. (Do this Friday at the end of the day or Saturday morning this will prevent later ones to start)

-- Add DATA, INDEX space to HEPYPRD and HEDWPRD (Do this Friday at the end of the day or any time before upgrade starts on Saturday if needed)

-- Check make sure PY Level 0 completed (should start 1 AM on Saturday and should be done way before 11:45 PM). If backup still running prior to Upgrade start time let it run.

-- Check make sure DW Level 0 completed (should start 9 AM on Saturday and should be done way before 11:45PM). If backup still running prior to Upgrade start time let it run.

-- Check make sure last PY and DW Archivelog jobs completed ( Last backup should kick off at 11:40 PM on Saturday)

Run manually if needed - BACKUP\_HEPYPRD\_XHEPYDBM21P\_ARCHIVE\_LOG

Run manually if needed - BACKUP\_HEDWPRD\_XHEDWDBM21P\_ARCHIVED\_LOGS

-- Check make sure FLASH space freed up on both PY and DW servers.

. oraenv

+ASM

sqlplus / as sysdba

@showasm

**PY Saturday upgrade steps – 11:45 PM**

Primary : xhepydbm21p HEPYPRD

StandBy : xhepydbw21p HEPYPRD

Run OEM **Standby** archivelog backup job HEPYPRD\_STDBY2\_ARCHIVE\_LOG\_PURGE for **Standby** database. Or adjust schedule prior to upgrade to finish right before upgrade start.

--Comment heartbeat in cron on **Primary** server

#\*/5 \* \* \* \* /home/oracle/tls/rman/heartbeat.ksh HEPYPRD AEDBA > /dev/null 2>&1

--Manually update RMAN HeartBeat Table in **Primary** Database

/home/oracle/tls/rman/heartbeat.ksh HEPYPRD AEDBA

DBNAME Start Time

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

Previous TimeStamp

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

Current TimeStamp

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

End Time

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

--Get Flashback Database status in **Primary** database

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

FLASHBACK\_ON

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

NO

--Get Flashback Database status in **Standby** database

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

FLASHBACK\_ON

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

NO

--**DISABLE DG Transport for PY**

--From **Primary** xhepydbm21p for PAYOR (PY)

--Get oracle password from TPAM for oracle

. oraenv

HEPYPRD

cd $SCRIPTS

-- Check make sure not N/A sessions. (Kill it if any still connected)

-- Check make sure all apps connection down and nothing else running.

./DISable\_log\_shipping.sh HEPYPRD

-- Check logs

cd /orahome/u01/app/oracle/local/logs

ls -altr

--Perform a Log Switch on **Primary** Database

sql => alter system switch logfile;

--Stop Managed Recovery on the **Standby** Database from Standby server

Confirm MRP0 Process is up

sqlplus / as sysdba

set line 140

select thread#,

sequence# as "REDO LOG BEING APPLIED",

process,

status,

block#,

blocks

from v$managed\_standby;

--Stop Managed Recovery on the **Standby** server

dgmgrl /

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

show database verbose 'HEPYPRD\_xhepydbw21p';

--Confirm MRP0 Process is down

sqlplus / as sysdba

set line 140

select thread#,

sequence# as "REDO LOG BEING APPLIED",

process,

status,

block#,

blocks

from v$managed\_standby;

--Note: FRA was already defined in HEPYPRD (Otherwise would need to create)

From **Primary** and **Standby** database

alter system set db\_recovery\_file\_dest\_size = 400G scope=both;

SQL> show parameter db\_recovery\_file\_dest

NAME TYPE VALUE

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

db\_recovery\_file\_dest string +FLASH\_01

db\_recovery\_file\_dest\_size big integer 400 GB

--Create restore points.

**Standby** first:

sqlplus / as sysdba

sql => CREATE RESTORE POINT HEPYPRD\_Standby\_flashback\_20210327 GUARANTEE FLASHBACK DATABASE;

col name for a25

col SCN for 9999999999999999

col TIME for a35

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

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

FLASHBACK\_ON

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

RESTORE POINT ONLY

**Primary** second:

sqlplus / as sysdba

sql => CREATE RESTORE POINT HEPYPRD\_Primary\_flashback\_20210327 GUARANTEE FLASHBACK DATABASE;

col name for a25

col SCN for 9999999999999999

col TIME for a35

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

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

FLASHBACK\_ON

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

RESTORE POINT ONLY

**DW Saturday upgrade steps – 11:45 PM**

Primary: xhedwdbm21p HEDWPRD

Standby: xhedwdbw21p HEDWPRD

Run OEM **standby** archivelog backup job HEDWPRD\_STDBY2\_ARCHIVE\_LOG\_PURGE for **standby** database. Or adjust schedule to be completed right before upgrade starts.

--Comment heartbeat in cron on **Primary** server

#\* /home/oracle/tls/rman/heartbeat.ksh HEDWPRD AEDBA

--Manually update RMAN HeartBeat Table in **Primary** Database

/home/oracle/tls/rman/heartbeat.ksh HEDWPRD AEDBA

DBNAME Start Time

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

HEDWPRD

Previous TimeStamp

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

Current TimeStamp

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

End Time

-- Get Flashback Database status in **Primary** database

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

FLASHBACK\_ON

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

NO

--Get Flashback Database status in **Standby** database

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

FLASHBACK\_ON

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

NO

--Get oracle password from TPAM for oracle

. oraenv

HEDWPRD

cd $SCRIPTS

./DISable\_log\_shipping.sh HEDWPRD

-- Check logs

cd /orahome/u01/app/oracle/local/logs

ls -altr

--Perform a Log Switch on **Primary** Database

sql => alter system switch logfile;

--Run **LockUser** procedure in HEDWPRD. (Use DBArtisan or sqlplus)

--Manually kill N and A ID connections

DROP TABLESPACE PERFSTAT INCLUDING CONTENTS AND DATAFILES

/

--Stop Managed Recovery on the **Standby** Database from Standby server

Confirm MRP0 Process is up

sqlplus / as sysdba

set line 140

select thread#,

sequence# as "REDO LOG BEING APPLIED",

process,

status,

block#,

blocks

from v$managed\_standby;

Stop Managed Recovery on **Standby** server

dgmgrl /

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

show database verbose 'HEDWPRD\_xhedwdbw21p';

Confirm MRP0 Process is down

sqlplus / as sysdba

set line 140

select thread#,

sequence# as "REDO LOG BEING APPLIED",

process,

status,

block#,

blocks

from v$managed\_standby;

--Note: FRA was already defined in HEDWPRD (Otherwise would need to create)

From **Primary** and **Standby** database

alter system set db\_recovery\_file\_dest\_size = 400G scope=both;

SQL> show parameter db\_recovery\_file\_dest

NAME TYPE VALUE

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

db\_recovery\_file\_dest string +FLASH\_01

db\_recovery\_file\_dest\_size big integer 400G

Create restore points.

**Standby** first:

sqlplus / as sysdba

sql => CREATE RESTORE POINT HEDWPRD\_Standby\_flashback\_20210327 GUARANTEE FLASHBACK DATABASE;

col name for a25

col SCN for 9999999999999999

col TIME for a35

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

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

FLASHBACK\_ON

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

RESTORE POINT ONLY

**Primary** second:

sqlplus / as sysdba

sql => CREATE RESTORE POINT HEDWPRD\_Primary\_flashback\_20210327 GUARANTEE FLASHBACK DATABASE;

col name for a25

col SCN for 9999999999999999

col TIME for a35

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

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

FLASHBACK\_ON

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

RESTORE POINT ONLY

**-----------------------------> Sunday morning steps around 8:00 AM**

* Check how much logs being generated during upgrade. If FLASH space does not have much free drop restore points first.

. oraenv

+ASM

--- To monitor FLASH space in general

sqlplus / as sysdba

@showasm

exit

--To Monitor flashback logs size that being generated during upgrade

asmcmd

cd +FLASH\_01/HEPYPRD\_XHEPYDBM21P/FLASHBACK/

du

Used\_MB Mirror\_used\_MB

86128 86128

or

. oraenv

HEPYPRD

SELECT

log# as "Log No",

thread# as "Thread No",

sequence# as "Seq No",

name,

bytes/1024/1024/1024 as "Size(GB)",

first\_change# as "First Chg No",

first\_time

FROM

v$flashback\_database\_logfile

ORDER BY first\_time;

--**ENABLE DG Trasport for PY**

In **Primary** xhepydbm21p for PAYOR (PY)

--Get oracle password from TPAM for oracle

. oraenv

HEPYPRD

cd $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 from **Primary** database

sqlplus / as sysdba

SELECT DB\_NAME, HOSTNAME, LOG\_ARCHIVED, LOG\_APPLIED,APPLIED\_TIME,

LOG\_ARCHIVED-LOG\_APPLIED LOG\_GAP

FROM

(

SELECT NAME DB\_NAME

FROM V$DATABASE

),

(

SELECT UPPER(SUBSTR(HOST\_NAME,1,(DECODE(INSTR(HOST\_NAME,'.'),0,LENGTH(HOST\_NAME),

(INSTR(HOST\_NAME,'.')-1))))) HOSTNAME

FROM V$INSTANCE

),

(

SELECT MAX(SEQUENCE#) LOG\_ARCHIVED

FROM V$ARCHIVED\_LOG WHERE DEST\_ID=1 AND ARCHIVED='YES'

),

(

SELECT MAX(SEQUENCE#) LOG\_APPLIED

FROM V$ARCHIVED\_LOG WHERE DEST\_ID=2 AND APPLIED='YES'

),

(

SELECT TO\_CHAR(MAX(COMPLETION\_TIME),'DD-MON/HH24:MI') APPLIED\_TIME

FROM V$ARCHIVED\_LOG WHERE DEST\_ID=3 AND APPLIED='YES'

);

**Drop restore points**

**Primary**:

sqlplus / as sysdba

sql => DROP RESTORE POINT HEPYPRD\_Primary\_flashback\_20210327;

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\_20210327;

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

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

--Check logs

cd /orahome/u01/app/oracle/local/logs

ls -altr

Uncomment heartbeat in cron on **Primary** server

~~Uncomment PY stats job~~

**--ENABLE DG Trasport for DW**

* Check how much logs being generated during upgrade. If FLASH space does not have much free drop restore points first.

. oraenv

+ASM

--- To monitor FLASH space in general

sqlplus / as sysdba

@showasm

exit

--To Monitor flashback logs size that being generated during upgrade

asmcmd

cd +FLASH\_01/HEDWPRD\_XHEDWDBM21P/FLASHBACK

du

Used\_MB Mirror\_used\_MB

24592 24592

or

. oraenv

HEDWPRD

SELECT

log# as "Log No",

thread# as "Thread No",

sequence# as "Seq No",

name,

bytes/1024/1024/1024 as "Size(GB)",

first\_change# as "First Chg No",

first\_time

FROM

v$flashback\_database\_logfile

ORDER BY first\_time;

--In **Primary** machine for Data Warehouse (DW)

--Get oracle password from TPAM for oracle

. oraenv

HEDWPRD

cd $SCRIPTS

./ENable\_log\_shipping.sh HEDWPRD

--Check logs

cd /orahome/u01/app/oracle/local/logs

ls -altr

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 from **Primary** database

sqlplus / as sysdba

SELECT DB\_NAME, HOSTNAME, LOG\_ARCHIVED, LOG\_APPLIED,APPLIED\_TIME,

LOG\_ARCHIVED-LOG\_APPLIED LOG\_GAP

FROM

(

SELECT NAME DB\_NAME

FROM V$DATABASE

),

(

SELECT UPPER(SUBSTR(HOST\_NAME,1,(DECODE(INSTR(HOST\_NAME,'.'),0,LENGTH(HOST\_NAME),

(INSTR(HOST\_NAME,'.')-1))))) HOSTNAME

FROM V$INSTANCE

),

(

SELECT MAX(SEQUENCE#) LOG\_ARCHIVED

FROM V$ARCHIVED\_LOG WHERE DEST\_ID=1 AND ARCHIVED='YES'

),

(

SELECT MAX(SEQUENCE#) LOG\_APPLIED

FROM V$ARCHIVED\_LOG WHERE DEST\_ID=3 AND APPLIED='YES'

),

(

SELECT TO\_CHAR(MAX(COMPLETION\_TIME),'DD-MON/HH24:MI') APPLIED\_TIME

FROM V$ARCHIVED\_LOG WHERE DEST\_ID=3 AND APPLIED='YES'

);

**Drop restore points**

**Primary**:

sqlplus / as sysdba

sql => DROP RESTORE POINT HEDWPRD\_Primary\_flashback\_20210327;

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\_20210327;

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

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

-- Run **UNLOCKUSER** procedure in HEDWPRD (Use DBArtisan or sqlplus)

~~--Uncomment DW stats crontab jobs~~

--Uncomment heartbeat in cron on DW primary server