**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 HEPYQA and HEDWQA ~~(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 11PM check around 1 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 11PM check around 1 PM)~~. If backup still running prior to Upgrade start time let it run.

-- Check make sure last PY and DW Archivelog jobs completed ~~( Do this around 10:45 PM on Saturday)~~

Run manually if needed - BACKUP\_HEPYQA\_XHEPYDBW21Q\_ARCHIVE\_LOG

Run manually if needed - BACKUP\_HEDWQA\_XHEDWDBW21Q\_ARCHIVE

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

. oraenv

+ASM

sqlplus / as sysdba

@showasm

**PY ~~Saturday upgrade steps – 11 PM~~**

Primary : xhepydbw21q HEPYQA

StandBy : xhepydbm21q HEPYQA

Run OEM Standby archivelog backup job HEPYQA\_STDBY1\_ARCHIVE\_LOG\_PURGE for Standby database. Or adjust check if it ran recently prior to upgrade.

--Comment heartbeat in cron on Primary server

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

#0 \* \* \* \* /home/oracle/tls/rman/confirm\_heartbeat\_active\_in\_cron.sh HEPYQA > /dev/null 2>&1

--Manually update RMAN HeartBeat Table in Primary Database

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

DBNAME Start Time

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

HEPYQA 01-28-2021 09:32:35 AM

Previous TimeStamp

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

01-28-2021 09:30:01 AM

Current TimeStamp

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

01-28-2021 09:32:35 AM

End Time

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

01-28-2021 09:32:35 AM

--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

--In Prod on primary machine xhepydbw21qfor PAYOR (PY)

--Get oracle password from TPAM

. oraenv

HEPYQA

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 HEPYQA

-- 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

dgmgrl /

edit database 'HEPYQA\_xhepydbm21q' set state = 'APPLY-OFF';

show database verbose 'HEPYQA\_xhepydbm21q';

--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 = 100G 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 100 GB

--Create restore points.

Standby:

sqlplus / as sysdba

sql => CREATE RESTORE POINT HEPYQA\_Standby\_flashback\_20210128 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;

Primary:

sqlplus / as sysdba

sql => CREATE RESTORE POINT HEPYQA\_Primary\_flashback\_20210128 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;

**DW ~~Saturday upgrade steps – 11 PM~~**

Primary: xhedwdbw21q HEDWQA

Standby: xhedwdbm21q HEDWQA

Run OEM standby archivelog backup job HEDWQA\_STDBY1\_ARCHIVE\_LOG\_PURGEfor standby database. Or adjust schedule to be completed right before upgrade starts.

--Comment heartbeat in cron on primary server

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

#0 \* \* \* \* /home/oracle/tls/rman/confirm\_heartbeat\_active\_in\_cron.sh HEDWQA > /dev/null 2>&1

--Manually update RMAN HeartBeat Table in Primary Database

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

-- 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

. oraenv

HEDWQA

cd $SCRIPTS

./DISable\_log\_shipping.sh HEDWQA

-- 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 HEDWQA. (Use DBArtisan or sqlplus)

--Manually kill N and A ID connections

--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

dgmgrl /

edit database 'HEDWQA\_xhedwdbm21q' set state = 'APPLY-OFF';

show database verbose 'HEDWQA\_xhedwdbm21q';

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 HEDWQA (Otherwise would need to create)

From primary and standby database

alter system set db\_recovery\_file\_dest\_size = 60G 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 7000M

Create restore points.

Standby:

sqlplus / as sysdba

sql => CREATE RESTORE POINT HEDWQA\_Standby\_flashback\_20210128 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;

Primary:

sqlplus / as sysdba

sql => CREATE RESTORE POINT HEDWQA\_Primary\_flashback\_20210128 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;

**-------------------------------------------------------> Steps once Upgrade completed with no issues.**

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

. oraenv

+ASM

sqlplus / as sysdba

@showasm

--ENABLE DG Trasport for PY

From Primary xhepydbw21q for PAYOR (PY)

. oraenv

HEPYQA

cd /orahome/u01/app/oracle/local/scripts

./ENable\_log\_shipping.sh HEPYQA

Perform a Log Switch on Primary Database

sqlplus / as sysdba

sql => alter system switch logfile;

Confirm log sent to Standby server

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

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

dgmgrl /

edit database 'HEPYQA\_xhepydbm21q' set state = 'APPLY-ON';

show database verbose 'HEPYQA\_xhepydbm21q';

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=2 AND APPLIED='YES'

);

Run the following against the primary and standby databases

Drop restore points

Primary:

sqlplus / as sysdba

sql => DROP RESTORE POINT HEPYQA\_Primary\_flashback\_20210128;

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 HEPYQA\_Standby\_flashback\_20210128;

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

sqlplus / as sysdba

@showasm

--In Primary machine for Data Warehouse (DW)

. oraenv

HEDWQA

cd /orahome/u01/app/oracle/local/scripts

./ENable\_log\_shipping.sh HEDWQA

--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 'HEDWQA\_xhedwdbm21q' set state = 'APPLY-ON';

show database verbose 'HEDWQA\_xhedwdbm21q';

--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 HEDWQA\_Primary\_flashback\_20210128;

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 HEDWQA\_Standby\_flashback\_20210128;

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 HEDWQA (Use DBArtisan or sqlplus)

~~--Uncomment DW stats crontab jobs~~

--Uncomment heartbeat in cron on DW primary server