**PPM EBIZ Database Upgrade from 11.2.0.3 to 12.1.0.2**

Application Remedy ticket: **CRQ########**

DBA Remedy ticket: **CRQ######**

This document uses Metalink doc **1503653.1 Complete Checklist for Manual Upgrades to 12cR1** as its basis.

**Pre-Upgrade Steps to be done for each database**

NOTE: Scripts cannot always be run more than once and get the desired results.

Use a sysdba id for all scripts unless otherwise noted.

1. Send request to App Admin requesting the apps be shut down for this database to begin the upgrade. Note: it is helpful to create an EAE Service request a few days before the upgrade is planned.
2. Copy scripts to new upgrade directory:

~~From same server:~~

~~cp -rp $SCRIPTS/upgrade/upgora12102 $HOME/admin/$ORACLE\_SID/~~

**~~OR~~**

~~From remote server:~~

~~cd $SCRIPTS~~

~~scp -rp dppdev:/u01/app/oracle/aetna/db\_mgmt/upgrade/upgora12102 .~~

From aorinfw1d to aorinfm1p Server (completed)

scp -rp /orahome/allu01/app/oracle/admin/PPMDEBIZ/upgora12102 aorinfm1p:/orahome/u01/app/oracle/admin/PPMPEBIZ/upgora12102

## Set up Variables

1. Set variables for database being upgraded

. $SETUPENV $ORACLE\_SID 🡸 based on what to be upgraded.

cd $HOME/admin/$ORACLE\_SID/upgora12102

#Make change based in this section case $UPGDB in… based on environment.

vi set\_upgrade\_env\_variables.ksh

run => . set\_upgrade\_env\_variables.ksh $ORACLE\_SID

1. **Make or double check if you need changes in scripts below to make sure all necessary environment settings are present in .ksh scripts. (Scripts were copied from different server originally)**

cd $HOME/admin/$ORACLE\_SID/upgora12102

vi upgrade\_db\_to\_12c.ksh

vi archsize.ksh

vi rman\_catlgparms.ksh

vi verify\_psu\_applied.ksh

vi run\_utlrp.ksh

cd $HOME/admin/$ORACLE\_SID/upgora12102/stats

vi run\_stats.ksh

cd $HOME

ls –altr .${ORACLE\_SID}sys.pw

ls –altr .${ORACLE\_SID}.pw

#if not present create one.or copy from other server and change name to reflect Database being upgraded.

1. Start grid control blackout period for database and listener being upgraded in OEM

Suspend all applicable jobs in OEM

1. Set environment and $ORACLE\_SID to instance being upgraded

. $SETUPENV $ORACLE\_SID

1. Capture RMAN configuration parameters prior to the upgrade

cd $HOME/admin/$ORACLE\_SID/upgora12102

rman\_catlgparms.ksh $ORACLE\_SID ${RCATDB} b4ora12102

1. To check for any cron jobs for this database that need to be disabled:

crontab –l > crontab\_` date '+%Y%m%d'`\_b4ora12102upg\_${ORACLE\_SID}

crontab -l |grep $ORACLE\_SID|grep -v ^#

PPMDEBIZ

#20 07 \* \* 1-6 /orahome/allu01/aetna/scripts/maint/ora\_aud\_maint.ksh +14 PPMDEBIZ > /dev/null 2>&1

# Rename database alert log for all running databases

59 23 \* \* 6 /orahome/allu01/aetna/scripts/maint/rename\_db\_alertlog.ksh > /dev/null 2>&1

05 00 1 \* \* /orahome/allu01/app/oracle/relink/save\_OH\_b4patch.ksh

## Added by E.B. after PPMDEBIZ upgrade to 12c

01 00 1 \* \* /orahome/allu01/aetna/scripts/maint/move\_lsnr\_log.sh $ORACLE\_HOME/network/admin/PPMDEBIZ\_aorinfw1d/diag/tnslsnr/aorinfw1d/ppmdebiz/trace ppmdebiz.log

05 00 1 \* \* /orahome/allu01/aetna/scripts/maint/purge\_trace\_files.ksh /orahome/u01/app/oracle/admin/PPMDEBIZ/diag/rdbms/ppmdebiz/PPMDEBIZ/trace

Save crontab and edit if changes are required

See also: [**Note 404238.1**](https://support.oracle.com/oip/faces/secure/km/DocumentDisplay.jspx?id=404238.1): How to Disable an Entry from DBMS\_SCHEDULER [**Note 1335741.1**](https://support.us.oracle.com/oip/faces/secure/km/DocumentDisplay.jspx?id=1335741.1): How To Stop A Running Job Using DBMS\_JOB [**Note 67695.1**](https://support.us.oracle.com/oip/faces/secure/km/DocumentDisplay.jspx?id=67695.1) **:** PROCEDURE DBMS\_JOB.BROKEN Specification

1. Save and update oratab indicator to **N**

cp –p ${ORATAB} oratab\_` date '+%Y%m%d'`\_b4ora12102upg\_${ORACLE\_SID}

vi /etc/oratab

1. Take a backup of your source database and save

For Production only:

Configure RMAN tape channels to 6 before taking backup

rman

connect target /

connect catalog ID/PW@RCATPROD

CONFIGURE DEVICE TYPE 'SBT\_TAPE' PARALLELISM **6** BACKUP TYPE TO BACKUPSET;

show all;

exit

Check to see if any backup jobs are in flight and make sure all are

complete before taking the backup.

ps –ef|grep rman|grep –v grep

**Take fresh backup**

cd $SCRIPTS/backup

~~./rman\_general\_purpose\_oem.ksh $ORACLE\_SID RCATDB incremental\_level0\_backup\_tape\_oem.rman~~

./rman\_general\_purpose\_oem.ksh PPMDEBIZ RCATDEV incremental\_level0\_backup\_tape\_oem.rman Y

#Once backup completed. Get latest log file name

cd $INSTANCE\_OUTPUTS/rman\_backups

ls -altr \*.log

#Save Backup

$BKPSCR/savebkp.ksh $ORACLE\_SID PPMDEBIZ\_20180524\_08:06:45\_incremental\_level0\_backup\_tape\_oem.rman.log b4upgora12102

PPMDEBIZ (had issues with savebkp.ksh just extracted from backup log)

##Restore DataBase

#connect target /

#connect catalog PPMDEBIZ/PPMDEBIZnamr@RCATDEV

#startup force nomount;

#set DBID=3566175702;

#reset database to incarnation 615632;

#restore controlfile from 'c-3566175702-20180524-01\_PPMDEBIZ';

#alter database mount;

#run{

#restore database until sequence 101;

#recover database until sequence 101;

#}

#alter database open resetlogs;

##End Restore DataBase

**Archivelog backup (if applicable)**

For prod, a full backup will be taken 2 hours prior

to the start of the upgrade. Therefore, an archivelog

backup will be taken here.



Check if any job maybe in scheduler

PPMDEBIZ

select \* from dba\_scheduler\_jobs;

only sys jobs. Nothing to do

select \* from dba\_jobs;

JOB LOG\_USER

48 APPS

104 APPS

Login as APPS/Ac\_nGe\_d

BEGIN

DBMS\_JOB.BROKEN(104, TRUE);

END;

/

commit;

BEGIN

DBMS\_JOB.BROKEN(48, TRUE);

END;

/

commit:

select \* from dba\_jobs;

Broken should return Y

**Also complete steps from Doc ID 1524398.1**

**Doc ID 1524398.1**

**EBIZ Pre Upgrade Steps…**

Step 10: Drop SYS.ENABLED$INDEXES (conditional)

If the SYS.ENABLED$INDEXES table exists, use SQL\*Plus to connect to the database as SYSDBA and running the following command to drop it:

SQL> drop table sys.enabled$indexes;

PPMDEBIZ – n/a

sqlplus / as sysdba

drop table sys.enabled$indexes

\*

ERROR at line 1:

ORA-00942: table or view does not exist

DESC SYS.ENABLED$INDEXES

ORA-04043: object SYS.ENABLED$INDEXES does not exist

Step 11: Disable Database Vault (conditional)

If you have Database Vault installed, perform steps 1 to 6 of Part 2 of document 1091083.1 on My Oracle Support to disable Database Vault.

PPMDEBIZ – n/a

SELECT \* FROM V$OPTION WHERE PARAMETER = 'Oracle Database Vault';

--FALSE

Step 12.Export OLAP analytical workspaces (conditional)

If your database server node is on a 32-bit platform, perform detailed steps 1 to 4 of Upgrading OLAP from 32 to 64 bits on My Oracle Support.

PPMDEBIZ – n/a

Step 13. Remove the MGDSYS schema (conditional)

If you are upgrading from an RDBMS version prior to 12c, on the old database server node, use SQL\*Plus to connect to the database as SYSDBA and run the $ORACLE\_HOME/md/admin/catnomgdidcode.sql script. This drops the MGDSYS schema.

$ sqlplus "/ as sysdba" @?/md/admin/catnomgdidcode.sql

PPMDEBIZ

sqlplus / as sysdba

@?/md/admin/catnomgdidcode.sql

User dropped.

Synonym dropped.

Synonym dropped….

1. Verify all processes disconnected:

sqlplus “/ as sysdba”

select username from v$session; 🡸 Should only see SYS and DBSNMP users

exit

~~Send email to Bruce Rose to stop Tivoli process.~~

~~Sample email:~~

~~Subject: Upgrading PPMTEST on aorinfw1d to Oracle 12.1.0.2~~

~~Please shut down the Tivoli process for PPMTEST database on aorinfw1d. I will be upgrading it today and will let you know when it is complete. There will be a new oracle home directory. Thanks.~~

To kill any lingering processes:

Start and shutdown are time consuming so instead do the following:

sqlplus “/ as sysdba”

select \* from dba\_sys\_privs

where privilege = 'RESTRICTED SESSION'; 🡸 to identify any users or roles with RESTRICTED SESSION granted

should only be DBA and ORATIVOLI

revoke RESTRICTED SESSION from ORATIVOLI;

kill any processes remaining using DBArtisan

Put database in restricted session:

alter system enable restricted session;

select username from v$session;

ALTER USER ORATIVOLI ACCOUNT LOCK

/

1. Alter all system, sysaux datafiles and last TEMP file to be autoextend if needed

cd $HOME/admin/$ORACLE\_SID/upgora12102

sqlplus / as sysdba

@ext\_alter\_system\_datafiles\_autoextend.sql

!view alter\_system\_datafiles\_autoextend.sql 🡸 verify expected

datafiles to be set

to autoextend.

if not, manually modify

alter\_system\_datafiles\_autoextend.sql and

alter\_system\_datafiles\_autoextend\_off.sql

files

@alter\_system\_datafiles\_autoextend.sql

exit

view alter\_system\_datafiles\_autoextend.out 🡸 verify that system, sysaux and temp changed to autoextend if were not already autoextend

1. Run Script to Collect DB Upgrade/Migrate Diagnostic Information (dbupgdiag.sql)

Note 556610.1

run => sqlplus / as sysdba

sql => @dbupgdiag\_pv.sql 🡸 respond

$HOME/admin/$ORACLE\_SID/upgora12102

PPMDEBIZ - db\_upg\_diag\_PPMDEBIZ\_24\_May\_2018\_1055.log

view db\_upg\_diag\_PPMDEBIZ\_24\_May\_2018\_1055.log

1. Check for INVALID database components and objects in the Source database

set pagesize500 linesize 100 trimspool on

spool check\_for\_invalids\_b4\_upgrade.out

select substr(comp\_name,1,40) comp\_name, status, substr(version,1,10) version

from dba\_registry order by comp\_name;

select substr(object\_name,1,40) object\_name,substr(owner,1,15) owner,object\_type

from dba\_objects where status='INVALID' order by owner,object\_type;

select owner,object\_type,count(\*) from dba\_objects

where status='INVALID' group by owner,object\_type order by owner,object\_type ;

exit

If you find invalid objects and/or database components then try to VALIDATE the

invalid objects and/or database components by executing the following steps:

Run $ORACLE\_HOME/rdbms/admin/utlrp.sql to validate the invalid objects in the

database. You can execute the utlrp.sql scripts multiple times to validate the invalid

objects.

$HOME/admin/$ORACLE\_SID/upgora12102/run\_utlrp.ksh ${ORACLE\_SID} $HOME/admin/$ORACLE\_SID/upgora12102

Or

@?/rdbms/admin/utlrp.sql

select count(\*)

from dba\_objects

where status = 'INVALID';

PPMDEBIZ - 34

view `ls -lrt utlrp\_\*.out|awk '{print $9}'|tail -1` (view most

recent - file name has timestamp included)

Resolve any invalid object issues

e.g. - alter materialized view DM.DM\_MVW\_ST\_STRUCTURE compile;

run => sqlplus / as sysdba

sql => set trimspool on

sql => spool utluiobj\_b4\_upgrade.out

sql => @?/rdbms/admin/utluiobj.sql

PPMTEST

None

PPMDEBIZ - utluiobj\_b4\_upgrade.out

1. Determine hidden parameters and events that need to be commented during upgrade

run > sqlplus / as sysdba

sql => @hidden\_parms\_events.sql

run => view hidden\_parms\_events.out

PPMTEST

None

PPMDEBIZ

\_disable\_fast\_validate

\_system\_trig\_enabled

\_sort\_elimination\_cost\_ratio

\_b\_tree\_bitmap\_plans

\_fast\_full\_scan\_enabled

\_index\_join\_enabled

\_sqlexec\_progression\_cost

\_like\_with\_bind\_as\_equality

\_sort\_elimination\_cost\_ratio

\_b\_tree\_bitmap\_plans

\_fast\_full\_scan\_enabled

\_index\_join\_enabled

\_sqlexec\_progression\_cost

\_like\_with\_bind\_as\_equality

\_optimizer\_autostats\_job

\_trace\_files\_public

1. Disable custom triggers that would fire before/after DDL and enable them after the

upgrade is complete.

Check with DBArtisan

PPMTEST - None

1. Check for database corruption (5 minutes)

run => sqlplus / as sysdba

sql => drop table INVALID\_ROWS;

sql => @?/rdbms/admin/utlvalid.sql

sql => @ext\_analyze.sql

sql => @analyze.sql

sql => exit

run => grep ORA- analyze.out 🡸 should not see any errors

1. Gather dictionary stats

run => sqlplus / as sysdba

sql => EXEC DBMS\_STATS.GATHER\_DICTIONARY\_STATS;

sql => @check\_schema\_stale\_stats.sql

1. Confirm database is in archivelog mode

sql => select name,log\_mode from v$database;

1. Save characterset information

run => sqlplus / as sysdba

sql => set trimspool on

sql => spool characterset.out

select parameter, value

from nls\_database\_parameters

where parameter like '%CHARACTERSET%';

sql => spool off

1. Ensure users SYS and SYTEM have system as default tablespace

SELECT username, default\_tablespace

FROM dba\_users

WHERE username in ('SYS','SYSTEM');

If DEFAULT\_TABLESPACE is anything other than SYSTEM tablespace, modify the

default tablespace for user SYS and SYSTEM to SYSTEM by using the command below:

sql => ALTER user SYS default tablespace SYSTEM;

sql => ALTER user SYSTEM default tablespace SYSTEM;

1. Purge recyclebin

run => sqlplus / as sysdba

sql => @purge\_recyclebin.sql

1. Collect segment statistics prior to upgrading the database

run => cd $HOME/admin/$ORACLE\_SID/upgora12102/stats

run => vi run\_stats.ksh 🡸 edit MAILID and uncomment row counts if

desired (for PLNVD001 without row\_counts

took 1 minute, **Caution**: with row counts

can take more than 9 minutes)

run => nohup run\_stats.ksh ${ORACLE\_SID} b4ora12upgrade `pwd` & 🡸 sends

email when complete

nohup run\_stats\_eb.ksh ${ORACLE\_SID} b4ora12upgrade `pwd` system/Z00mZ00m! &

run => view \*b4ora12upgrade\*

run => sqlplus / as sysdba

sql => @dbsize.sql b4ora12upgrade

**~~Take a DBArtisan Report~~**

1. Run 12c preupgrade script

cd $HOME/admin/$ORACLE\_SID/upgora12102

run => sqlplus / as sysdba

sql => @preupgrd.sql

PPMDEBIZ (original attempt)

Log file - /orahome/allu01/app/oracle/cfgtoollogs/PPMDEBIZ/preupgrade/preupgrade.log

\_b\_tree\_bitmap\_plans = FALSE

\_disable\_fast\_validate = TRUE

\_fast\_full\_scan\_enabled = FALSE

\_index\_join\_enabled = FALSE

\_like\_with\_bind\_as\_equality = TRUE

\_optimizer\_autostats\_job = FALSE

\_sort\_elimination\_cost\_ratio = 5

\_sqlexec\_progression\_cost = 2147483647

\_system\_trig\_enabled = TRUE

\_trace\_files\_public = TRUE

WARNING: --> job\_queue\_processes set too low

sec\_case\_sensitive\_logon 12.1 DESUPPORTED

ERROR: --> Invalid Oracle supplied table data found in your database.

Invalid data can be seen prior to the database upgrade

or during PDB plug in. This table data must be made

valid BEFORE upgrade or plug in.

- To fix the data, load the Preupgrade package and execute

the fixup routine.

For plug in, execute the fix up routine in the PDB.

---Doc ID 2009405.1

cd /orahome/u01/app/oracle/product/12.1.0.2.180116/db\_1/rdbms/admin

cp utluppkg.sql /orahome/allu01/app/oracle/product/11.2.0/db\_2/rdbms/admin/utluppkg.sql

@?/rdbms/admin/utluppkg.sql

SET SERVEROUTPUT ON;

exec dbms\_preup.run\_fixup\_and\_report('INVALID\_SYS\_TABLEDATA')

SET SERVEROUTPUT OFF;

SELECT rpad(u.name,128) TABLENAME, rpad(o.name,128) OWNER,

rpad(c.name,128) COLNAME FROM SYS.OBJ$ o, SYS.COL$ c, SYS.COLTYPE$ t,

SYS.USER$ u

WHERE o.OBJ# = t.OBJ# AND c.OBJ# = t.OBJ# AND c.COL# = t.COL#

AND t.INTCOL# = c.INTCOL# AND BITAND(t.FLAGS, 256) = 256

AND o.OWNER# = u.USER# AND o.OWNER# NOT IN

(SELECT UNIQUE (d.USER\_ID) FROM SYS.DBA\_USERS d, SYS.REGISTRY$ r

WHERE d.USER\_ID = r.SCHEMA# and r.NAMESPACE='SERVER');

Should be no rows.

PPMDEBIZ (latest)

ACTIONS REQUIRED:

1. Review results of the pre-upgrade checks:

/orahome/allu01/app/oracle/cfgtoollogs/PPMDEBIZ/preupgrade/preupgrade.log

2. Execute in the SOURCE environment BEFORE upgrade:

/orahome/allu01/app/oracle/cfgtoollogs/PPMDEBIZ/preupgrade/preupgrade\_fixups.s

l

3. Execute in the NEW environment AFTER upgrade:

/orahome/allu01/app/oracle/cfgtoollogs/PPMDEBIZ/preupgrade/postupgrade\_fixups.

Ql

view /orahome/allu01/app/oracle/cfgtoollogs/PPMDEBIZ/preupgrade/preupgrade.log

WARNING: --> "pga\_aggregate\_limit" needs to be increased to at least 2147483648

--> sec\_case\_sensitive\_logon 12.1 DESUPPORTED

ERROR: --> Invalid Oracle supplied table data found in your database.

Invalid data can be seen prior to the database upgrade

or during PDB plug in. This table data must be made

valid BEFORE upgrade or plug in.

- To fix the data, load the Preupgrade package and execute

the fixup routine.

For plug in, execute the fix up routine in the PDB.

@?/rdbms/admin/utluppkg.sql

SET SERVEROUTPUT ON;

exec dbms\_preup.run\_fixup\_and\_report('INVALID\_SYS\_TABLEDATA')

SET SERVEROUTPUT OFF;

---Doc ID 2009405.1

cd /orahome/u01/app/oracle/product/12.1.0.2.180116/db\_1/rdbms/admin

cp utluppkg.sql /orahome/allu01/app/oracle/product/11.2.0/db\_2/rdbms/admin/utluppkg.sql

@?/rdbms/admin/utluppkg.sql

SET SERVEROUTPUT ON;

exec dbms\_preup.run\_fixup\_and\_report('INVALID\_SYS\_TABLEDATA')

SET SERVEROUTPUT OFF;

SELECT rpad(u.name,128) TABLENAME, rpad(o.name,128) OWNER,

rpad(c.name,128) COLNAME FROM SYS.OBJ$ o, SYS.COL$ c, SYS.COLTYPE$ t,

SYS.USER$ u

WHERE o.OBJ# = t.OBJ# AND c.OBJ# = t.OBJ# AND c.COL# = t.COL#

AND t.INTCOL# = c.INTCOL# AND BITAND(t.FLAGS, 256) = 256

AND o.OWNER# = u.USER# AND o.OWNER# NOT IN

(SELECT UNIQUE (d.USER\_ID) FROM SYS.DBA\_USERS d, SYS.REGISTRY$ r

WHERE d.USER\_ID = r.SCHEMA# and r.NAMESPACE='SERVER');

WARNING: --> "DMSYS" schema exists in the database

The DMSYS schema (Oracle Data Mining) will be removed

from the database during the database upgrade.

All data in DMSYS will be preserved under the SYS schema.

Refer to the Oracle Data Mining User's Guide for details.

WARNING: --> Database contains INVALID objects prior to upgrade

The list of invalid SYS/SYSTEM objects was written to

registry$sys\_inv\_objs.

The list of non-SYS/SYSTEM objects was written to

registry$nonsys\_inv\_objs unless there were over 5000.

Use utluiobj.sql after the upgrade to identify any new invalid

objects due to the upgrade.

INFORMATION: --> OLAP Catalog(AMD) exists in database

Starting with Oracle Database 12c, OLAP Catalog component is desupported.

If you are not using the OLAP Catalog component and want

to remove it, then execute the

ORACLE\_HOME/olap/admin/catnoamd.sql script before or

after the upgrade.

INFORMATION: --> Older Timezone in use

Database is using a time zone file older than version 18.

After the upgrade, it is recommended that DBMS\_DST package

be used to upgrade the 11.2.0.3.0 database time zone version

to the latest version which comes with the new release.

Please refer to My Oracle Support note number 1509653.1 for details

INFORMATION: --> There are existing Oracle components that will NOT be

upgraded by the database upgrade script. Typically, such components

have their own upgrade scripts, are deprecated, or obsolete.

Those components are: OLAP Catalog

\_b\_tree\_bitmap\_plans = FALSE

\_disable\_fast\_validate = TRUE

\_fast\_full\_scan\_enabled = FALSE

\_index\_join\_enabled = FALSE

\_like\_with\_bind\_as\_equality = TRUE

\_optimizer\_autostats\_job = FALSE

\_sort\_elimination\_cost\_ratio = 5

\_sqlexec\_progression\_cost = 2147483647

\_system\_trig\_enabled = TRUE

\_trace\_files\_public = TRUE

1. Disable cron and OEM jobs for database being upgraded
2. Remove EM database control repository if needed

run => sqlplus / as sysdba

sql => spool emremoval.log

sql => @emremove.sql

View output log emremoval.log

1. Remove or comment out obsolete, hidden, and deprecated initialization parameters

in parameter file in ${NEWOHDIR}/dbs directory

PPMDEBIZ

Determine if pfile or spfile

SELECT DECODE(value, NULL, 'PFILE', 'SPFILE') "Init File Type"

FROM sys.v\_$parameter WHERE name = 'spfile';

If spfile

create pfile from spfile;

For init.ora

cp -p $ORACLE\_HOME/dbs/init${ORACLE\_SID}.ora /orahome/u01/app/oracle/product/12.1.0.2.180116/${NEWOHDIR}/dbs/init${ORACLE\_SID}.ora

cp -p $ORACLE\_HOME/dbs/init${ORACLE\_SID}.ora /orahome/u01/app/oracle/product/12.1.0.2.180116/${NEWOHDIR}/dbs/init${ORACLE\_SID}.ora\_11203

PPMDEBIZ

cd /orahome/u01/app/oracle/product/12.1.0.2.180116/db\_1

vi

#change this parameter to new ORACLE\_BASE

diagnostic\_dest = /orahome/u01/app/oracle/admin/PPMDEBIZ

oracle\_home = /orahome/u01/app/oracle

audit\_file\_dest=/orahome/u01/app/oracle/admin/PPMDEBIZ/adump

#**Create new directories if not already there**

cd /orahome/u01/app/oracle

mkdir admin

cd admin

mkdir PPMDEBIZ

cd PPMDEBIZ

mkdir adump

**Make all necessary changes identified in item #30 and #31**

PPMDEBIZ

cd /orahome/u01/app/oracle/product/12.1.0.2.180116/${NEWOHDIR}/dbs

vi init${ORACLE\_SID}.ora

--change

job\_queue\_processes =17

--Comment out old

#\*.audit\_file\_dest='/orahome/allu01/app/oracle/admin/PPMDEBIZ/adump'

#\*.diagnostic\_dest='/orahome/allu01/app/oracle'

#PPMDEBIZ.\_\_oracle\_base='/orahome/allu01/app/oracle'#ORACLE\_BASE set from environment

#sec\_case\_sensitive\_logon = false

--Add new

\*.audit\_file\_dest='/orahome/u01/app/oracle/admin/PPMDEBIZ/adump'

\*.diagnostic\_dest='/orahome/u01/app/oracle/admin/PPMDEBIZ'

PPMDEBIZ.\_\_oracle\_base='/orahome/u01/app/oracle'

--Comment out

#PPMDEBIZ.\_\_pga\_aggregate\_target=1493172224

--Add

PPMDEBIZ.\_\_pga\_aggregate\_target=2207483648

--comment out

\_disable\_fast\_validate

\_system\_trig\_enabled

\_sort\_elimination\_cost\_ratio

\_b\_tree\_bitmap\_plans

\_fast\_full\_scan\_enabled

\_index\_join\_enabled

\_sqlexec\_progression\_cost

\_like\_with\_bind\_as\_equality

\_optimizer\_autostats\_job

\_trace\_files\_public

1. Verify That Materialized View Refreshes Have Completed Before Upgrading

sql => SELECT o.name FROM sys.obj$ o, sys.user$ u, sys.sum$ s

WHERE o.type# = 42 AND bitand(s.mflags, 8) =8;

If the above query returns any row, see Metalink Note 1442457.1

1. Ensure That No Files Need Media Recovery Before Upgrading

sql => SELECT \* FROM v$backup WHERE status != 'NOT ACTIVE';

1. Resolve Outstanding Distributed Transactions Before Upgrading

sql => SELECT \* FROM dba\_2pc\_pending;

If the above sql returns rows run:

sql => SELECT local\_tran\_id FROM dba\_2pc\_pending;

sql => EXECUTE dbms\_transaction.purge\_lost\_db\_entry('');

sql => COMMIT;

1. Check the XDB ACLs has start\_date and end\_date ACE attributes

Before upgrading the database to 12c, please run the below query as SYS:

SQL> select aclid, start\_date, end\_date from xds\_ace where start\_date is not null;

If the query returns any row, then please follow Note 1958876.1 Upgrade to 12.1 fails with ORA-01830 date format picture ends before converting entire input string ORA-06512: at "SYS.XS\_OBJECT\_MIGRATION" to avoid failure in XDB's upgrade.

1. Note the location of datafiles, redo logs and control files.

cd $HOME/admin/$ORACLE\_SID/upgora12102

run => sqlplus / as sysdba

sql => set echo on trimspool on

sql => spool database\_files.out

sql => select name as "Control Files" from v$controlfile;

sql => select file\_name as "Datafiles" from dba\_data\_files order by file\_id;

sql => select group#, member as "Logfiles" from v$logfile order by group#;

sql => exit;

1. Select users and current status

sql => set echo on trimspool on line 200

sql => spool database\_users\_b4upgrade.out

sql => select username, account\_status from dba\_users order by 1;

sql => exit;

1. Run the preupgrade fixup script

cd $HOME/cfgtoollogs/$ORACLE\_SID/preupgrade

sqlplus / as sysdba

sql => @preupgrade\_fixups.sql

run => view preupgrade.log

If message

WARNING: --> Enterprise Manager Database Control repository found in

the database

Run the following to confirm no longer exists

select substr(comp\_name,1,40) comp\_name, status, substr(version,1,10) version

from dba\_registry order by comp\_name;

1. Shutdown the database

sqlplus / as sysdba

sql => select name from v$database;

sql => shutdown immediate;

1. Rename alertlog

cd $BDUMP

mv alert\_${ORACLE\_SID}.log alert\_${ORACLE\_SID}.log\_`date +%m:%d:%Y:Time:%H:%M:%S`

1. Stop the listener

PPMDEBIZ

run => lsnrctl stop PPMDEBIZ

E.B.

1. Update the oratab to point to the new $ORACLE\_HOME

e.g.

Add new entry ~~PPMTEST:/orahome/u01/app/oracle/product/12.1.0.2.5/db\_1:Y:/orahome/allu01:/orabackup/ppmu001~~

PPMDEBIZ:/orahome/u01/app/oracle/product/12.1.0.2.180116/db\_1:Y:/orahome/allu01:/orabackup/ppmu001

Comment old entry

**Make sure the startup flag is set to Y**

1. Set to new Oracle environment

. $SETUPENV $UPGDB

Confirm all environment variables set accordingly for new oracle home

echo $ORA\_NLS10

if not set do following

export ORA\_NLS10=$ORACLE\_HOME/nls/data/9idata

. $SETUPENV $UPGDB

cd $HOME/admin/$ORACLE\_SID/upgora12102

. set\_upgrade\_env\_variables.ksh $ORACLE\_SID

1. Update the listener.ora to reflect the new $ORACLE\_HOME directory

for the database being upgraded

cd $ORACLE\_HOME/network/admin

cp listener.ora listener.ora\_b412cUpgrade

cp tnsnames.ora tnsnames.ora\_ b412cUpgrade

cp sqlnet.ora sqlnet.ora\_b412cUpgrade

#comment out some old stuff and change Oracle Home

vi listener.ora

vi sqlnet.ora

~~cp /orahome/allu01/app/oracle/product/11.2.0/db\_2/network/admin/PPMDEBIZ\_aorinfw1d/listener\_ifile.ora /orahome/u01/app/oracle/product/12.1.0.2.180116/db\_1/network/admin/PPMDEBIZ\_aorinfw1d/listener\_ifile.ora~~

1. Check for the SQLNET.ALLOWED\_LOGON\_VERSION Parameter Behavior

Starting with Oracle Database 12c, the default value for the

SQLNET.ALLOWED\_LOGON\_VERSION parameter has changed from 8 to 11.

The use of this parameter has been deprecated, and it is now replaced

with the SQLNET.ALLOWED\_LOGON\_VERSION\_SERVER and

SQLNET.ALLOWED\_LOGON\_VERSION\_CLIENT parameters.

If you have not explicitly set the SQLNET.ALLOWED\_LOGON\_VERSION\_SERVER

parameter in the upgraded database, then connections from clients

earlier than release 10g will fail with the error ORA-28040: No

matching authentication protocol. For better security, check the

password verifiers of your database users, and then configure the

database to use the correct password verifier by setting the

SQLNET.ALLOWED\_LOGON\_VERSION\_SERVER and

SQLNET.ALLOWED\_LOGON\_VERSION\_CLIENT parameters.

If you have password-protected roles (secure roles) in your existing

database and if you upgrade to Oracle Database 12c with the default

SQLNET.ALLOWED\_LOGON\_VERSION\_SERVER setting of 11, because those secure

roles only have release 10g verifiers, the password for each secure

role must be reset by the administrator so that the secure roles will

remain usable after the upgrade.

Check for the SQLNET IFILE parameter and set to new location

PPMTEST – Did not modify sqlnet.ora

1. Start or reload the listener

PPMDEBIZ

run => lsnrctl start PPMDEBIZ

~~Create init file if spfile being used~~

**~~If spfile exists for database, rename so it won’t be used~~**

~~Rename pfile if one exists in the directory~~

~~cd $ORACLE\_HOME/dbs~~

~~sqlplus / as sysdba~~

~~CREATE PFILE='$ORACLE\_HOME/dbs/init$ORACLE\_SID.ora' FROM SPFILE;~~

~~mv spfile{ORACLE\_SID}.ora spfile{ORACLE\_SID}.ora\_b412102upg~~

1. Upgrade the database

run => cd $HOME/admin/$ORACLE\_SID/upgora12102

sql => startup UPGRADE

sql => exit

run => nohup upgrade\_db\_to\_12c.ksh ${ORACLE\_SID} &

Logs are written to $HOME/admin/$ORACLE\_SID/upgora12102/logs

Open new SSH window

cd $HOME/admin/$ORACLE\_SID/upgora12102/logs

tail –f catupgrd0.log

**41 min for PPMDEBIZ**

View the upgrade summary report

view $ORACLE\_HOME/cfgtoollogs/$ORACLE\_SID/upgrade/upg\_summary.log

Check all applicable logs in all applicable directories

cd $ORACLE\_BASE/cfgtoollogs/sqlpatch

ls -ltr

cd to each applicable directory and run below find command

e.g.

find . -type f -exec grep -i "ORA-" {} \;

Save the output logs (if exists)

cd $ORACLE\_HOME/cfgtoollogs/$ORACLE\_SID/upgrade

cp -p upg\_summary.log $HOME/admin/$ORACLE\_SID/upgora12102/logs

Check for errors

cd $HOME/admin/$ORACLE\_SID/upgora12102/logs

egrep -i "ora-" \*cat\* |egrep -iv "\-\-|rem|doc|\\*"

View the database alertlog

view $BDUMP/alert\_$ORACLE\_SID.log

These type of messages can be ignored

Some indexes or index [sub]partitions of table SYS.WRI$\_OPTSTAT\_HISTGRM\_HISTORY have been marked unusable

Important: The catuppst.sql script is run as part of the upgrade

process unless the upgrade returns errors during the

process. Check the log file for "BEGIN catuppst.sql" to

verify that catuppst.sql ran during the upgrade process.

If catuppst.sql has not run, then proceed to run

catuppst.sql as shown in this step. Warning messages are

also displayed when running catctl.pl indicating that

catuppst.sql was not run during the upgrade.

Run catuppst.sql, located in the $ORACLE\_HOME/rdbms/admin

directory, to perform upgrade actions that do not require

the database to be in UPGRADE mode.

$ORACLE\_HOME/perl/bin/perl catcon.pl -n 1 -e -b catuppst -d '''.''' catuppst.sql

Check if catuppst.sql was run

grep "catuppst" \*

grep "BEGIN catuppst.sql" \*

Should see things like

catupgrd0.log:catuppst: Dropping library DBMS\_DDL\_INTERNAL\_LIB

catupgrd0.log:catuppst: Dropping view \_CURRENT\_EDITION\_OBJ\_MIG

catupgrd0.log:catuppst: Dropping view \_ACTUAL\_EDITION\_OBJ\_MIG

SQL> Rem BEGIN catuppst.sql

1. Run **catcon.pl** to invoke **utlrp.sql** to recompile any remaining stored PL/SQL and Java code.

Use the following syntax

cd $ORACLE\_HOME/rdbms/admin

run => sqlplus /as sysdba

sql => STARTUP

sql => exit

$ORACLE\_HOME/perl/bin/perl catcon.pl -n 1 -e -b utlrp -d '''.''' utlrp.sql

To monitor this open new SSH (count should be deacreasing…)

sqlplus / as sysdba

select count(\*) from dba\_objects where status = 'INVALID';

The log file **utlrp0.log** is generated with the results of the recompilations.

**1 hour 25 min for PPMDEBIZ**

1. Run the post fixup script

run => cd $HOME/cfgtoollogs/$ORACLE\_SID/preupgrade

sql => @postupgrade\_fixups.sql

1. Run the post upgrade status script

run => cd $HOME/admin/$ORACLE\_SID/upgora12102

run => sqlplus /as sysdba

sql => @utlu121s\_pv.sql

sql => exit

Query upgrade summary (Use DBArtisan to validate remote connections at the same time)

run => sqlplus /as sysdba

sql => select \* from sys.registry$upg\_summary;

select \* from dba\_registry;

1. Run the April PSU upgrade script

There may or may not be anything applied as the upgrade may have handled it all

run => cd $ORACLE\_HOME/OPatch

run => datapatch -verbose

**Check log files for any errors:**

cd $ORACLE\_BASE/cfgtoollogs/sqlpatch

ls –ltr

cd to all applicable directories and check for errors

1. Confirm the Oracle Home PSU applied to the database and check for errors

run => cd $HOME/admin/$ORACLE\_SID/upgora12102

run => verify\_psu\_applied.ksh

run => cd $ORACLE\_BASE/cfgtoollogs/sqlpatch

Check all files in directories created during upgrade time

e.g.

for i in `find . -type d -name '\*2015\_07\_28\*' -print`  
 do  
    grep -i "ORA-" $i/\*  
 done

The datapatch utility will then run the necessary apply scripts to load the modified SQL

files into the database. An entry will be added to the dba\_registry\_sqlpatch view

reflecting the patch application. In the dba\_registry\_sqlpatch view, verify the Status

for the APPLY is "SUCCESS". For any other status, refer to the following My Oracle

Support note for additional information and actions: Document [1609718.1](https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1609718.1) Datapatch

Known Issues.

1. Check for deprecated initialization parameters

SELECT name FROM v$parameter WHERE isdeprecated = 'TRUE' order by name;

Comment or remove any identified that are in use

PPMTEST – none

1. Run Diagnostic Information (dbupgdiag.sql)

Note 556610.1

run => sqlplus / as sysdba

sql => @dbupgdiag\_pv.sql 🡸 respond

$HOME/admin/$ORACLE\_SID/upgora12102

view log

1. Query current state of the dictionary

sql => set echo on trimspool on lines 80 pages 100

sql => spool regInvalid.out

sql => select substr(comp\_id,1,15) comp\_id,substr(comp\_name,1,30)

comp\_name,substr(version,1,10) version,status

from dba\_registry order by modified;

1. Create a new password file

run => cd $DBS

## Use actual sys password

orapwd file=$ORACLE\_HOME/dbs/orapw$ORACLE\_SID password=Z00mZ00m! entries=20

1. Enable any hidden parameters that were disabled

PPMTEST

PPMDEBIZ

cd $ORACLE\_HOME/dbs

vi initPPMDEBIZ.ora

\_disable\_fast\_validate

\_system\_trig\_enabled

\_sort\_elimination\_cost\_ratio

\_b\_tree\_bitmap\_plans

\_fast\_full\_scan\_enabled

\_index\_join\_enabled

\_sqlexec\_progression\_cost

\_like\_with\_bind\_as\_equality

\_optimizer\_autostats\_job

\_trace\_files\_public

Bounce the database

shutdown immediate;

startup;

1. ~~Create spfile if one was originally being used~~

~~cd $ORACLE\_HOME/dbs~~

~~sqlplus / as sysdba~~

~~CREATE SPFILE FROM PFILE;~~

1. Change password for users with default password

select 'alter user ' ||username ||' identified by "Add\_Password\_Here" PASSWORD EXPIRE;' from dba\_users where username in (select username from dba\_users\_with\_defpwd) order by username;

1. Select users and current status

run => cd $HOME/admin/$ORACLE\_SID/upgora12102

sql => set trimspool on line 200

sql -> col username for a30

sql => spool database\_users\_aftupgrade.out

sql => SELECT username, account\_status FROM dba\_users ORDER BY 1;

sql => exit;

Compare to list taken above.

Depending on the release from which you upgraded, there might be new

Oracle-supplied accounts. Oracle recommends that you lock all Oracle

supplied accounts except for SYS and SYSTEM, and expire their

passwords, thus requiring new passwords to be specified when the

accounts are unlocked.

To lock and expire passwords, issue the following SQL statement:

sql => alter user username password expire account lock;

1. Enable custom triggers that were disabled above

PPMTEST – None

1. Check for INVALID database components and objects in the Source database

set pagesize500 linesize 100

spool check\_for\_invalids\_aft\_upgrade.out

select substr(comp\_name,1,40) comp\_name, status, substr(version,1,10) version

from dba\_registry order by comp\_name;

select substr(object\_name,1,40) object\_name,substr(owner,1,15) owner,object\_type

from dba\_objects where status='INVALID' order by owner,object\_type;

select owner,object\_type,count(\*) from dba\_objects

where status='INVALID' group by owner,object\_type order by owner,object\_type ;

If you find invalid objects and/or database components then try to VALIDATE the

invalid objects and/or database components by executing the following steps:

Run $ORACLE\_HOME/rdbms/admin/utlrp.sql to validate the invalid objects in the

database. You can execute the utlrp.sql scripts multiple times to validate the invalid

objects.

$HOME/admin/$ORACLE\_SID/upgora12102/run\_utlrp.ksh ${ORACLE\_SID} $HOME/admin/$ORACLE\_SID/upgora12102

view `ls -lrt utlrp\_\*.out|awk '{print $9}'|tail -1` (view most

recent - file name has timestamp included)

or

@?/rdbms/admin/utlrp.sql

select count(\*)

from dba\_objects

where status = 'INVALID';

32

Compare to invalid list taken above

To determine new invalid objects

cd $ORACLE\_HOME/rdbms/admin

run => $ORACLE\_HOME/perl/bin/perl catcon.pl -n 1 -e -b utluiobj -d '''.''' utluiobj.sql

This tool lists post-upgrade invalid objects that were not invalid

prior to upgrade (it ignores pre-existing pre-upgrade invalid objects).

PPMTEST

None

1. Export RMAN recovery catalog for this schema

Modify highlighted items to reflect the schema and date of the

**For None Prod**

Log on to XORACLDW2D/ Bl0ckhe@d

Or

xoracldm2d

or

xoraclddbw1d

. oraenv

RCATDEV

cd /orahome/u01/app/oracle/admin/${ORACLE\_SID}/exports 🡸 if directory

does not exist

create it

echo $ORACLE\_HOME

echo $ORACLE\_SID

DBNAME=*name-of-db-being-upgraded 🡸* replace with name of database being

upgraded

REASON=*reason-for-backup* 🡸 replace with reason for backup

either

b4upgora12c

**or**

b4dwngora12c

ext\_export\_par\_gen.ksh ${DBNAME} ${REASON} 🡸 this script creates a

base par file for this

schema name

vi exp\_schema\_${DBNAME}\_${REASON}.par (view the par file to verify the

schema name, and file names)

exp parfile=exp\_schema\_${DBNAME}\_${REASON}.par 🡸 enter RMAN schema name

and password for

database being upgraded

verify dump file was created:

ls –l export\_schema\_${DBNAME}\_${REASON}\_`date +'%Y%m%d’`.dmp

compress dump file:

gzip export\_schema\_${DBNAME}\_${REASON}\_`date +'%Y%m%d’`.dmp

Log off XORACLDW2D

**For XPVDBM1P XPVDBM2P**:

Log on to XORACLDM2P

Or

xoraclddbm1p

. $SETUPENV RCATPROD

cd /orahome/u01/app/oracle/admin/RCATPROD/exports 🡸 if directory does

not exist, create

it

echo $ORACLE\_HOME

echo $ORACLE\_SID

DBNAME=*name-of-db-being-upgraded 🡸* replace with name of database

being upgraded

REASON=*reason-for-backup* 🡸 replace with reason for backup either

b4upgora12c

**or**

b4dwngora12c

Copy export script from the development cloud server:

scp –rp xoracldw2d:/orahome/u01/app/oracle/admin/RCATDEV/exports/ext\_export\_par\_gen.ksh .

ext\_export\_par\_gen.ksh ${DBNAME} ${REASON} 🡸 this script creates a

base par file for this

schema name

vi exp\_schema\_${DBNAME}.par (verify the par file for the database

name, date and reason for the export)

exp parfile=exp\_schema\_${DBNAME}.par 🡸 enter RMAN schema name and

password for database being

upgraded verify dump file was

created:

ls –l export\_schema\_${DBNAME}\_${REASON}\_`date +'%Y%m%d'`\_b4upgora112.dmp

compress dump file:

gzip export\_schema\_${DBNAME}\_${REASON}`date +'%Y%m%d'`\_b4upgora112.dmp

~~Backup RMAN export file to Planview db server:~~

~~For PLNVP001~~

Log off XORACLDM2P

1. Copy RMAN catalog export dump file to database server

PPMTEST

Run below from XORACLDM2P server

scp -rp export\_schema\_PPMTEST\_b4upgora12c\_20151218.dmp.gz aorinfw1d:/orahome/allu01/app/oracle/admin/PPMTEST/dpdump/export\_schema\_PPMTEST\_b4upgora12c\_20151218.dmp.gz

PPMDEBIZ

Run below from xoraclddbw1d server

cd /orahome/u01/app/oracle/admin/RCATDEV/exports

scp -rp export\_schema\_PPMDEBIZ\_b4upgora12c\_20170602.dmp.gz aorinfw1d:/oraexport/ppmu800/PPMDEBIZ/export\_schema\_PPMDEBIZ\_b4upgora12c\_20170602.dmp.gz

1. Upgrade RMAN catalog

cd $HOME/admin/$ORACLE\_SID/upgora12102

run => . set\_upgrade\_env\_variables.ksh $ORACLE\_SID

run => cd $HOME/admin/$ORACLE\_SID/upgora12102

rman catalog ${ORACLE\_SID}/${ORACLE\_SID}namr@${RCATDB} 2>&1 |tee rmanupg.out

or

rman

connect catalog xxx/xxx@RCATDEV

Expect this message:

PL/SQL package PLNVD001.DBMS\_RCVCAT version 11.02.00.03 in RCVCAT database is too old

**For all servers and databases:**

RMAN> UPGRADE CATALOG;

Enter again to confirm catalog upgrade

RMAN> UPGRADE CATALOG;

RMAN> exit

Expect output:

recovery catalog upgraded to version 12.01.00.02

DBMS\_RCVMAN package upgraded to version 12.01.00.02

DBMS\_RCVCAT package upgraded to version 12.01.00.02

1. Take a full database backup

**Take fresh backup**

cd $SCRIPTS/backup

./rman\_general\_purpose\_oem.ksh PPMDEBIZ RCATDEV incremental\_level0\_backup\_tape\_oem.rman Y

#Once backup completed. Get latest log file name

cd $INSTANCE\_OUTPUTS/rman\_backups

ls -altr \*.log

#Save Backup

$BKPSCR/savebkp.ksh $ORACLE\_SID <backup log file name> postupgora12102

PPMTEST –

1. Upgrade the timezone version

Note 1585343.1

cd $HOME/admin/$ORACLE\_SID/upgora12102/dst/DBMS\_DST\_scriptsV1.9

Check for tstz columns

run => sqlplus / as sysdba

sql => set trimspool on

sql => spool timezone\_stats.out

sql => @countstarTSTZ.sql

sql => @countstatsTSTZ.sql

sql => exit

Run checker

run => sqlplus / as sysdba

sql => set trimspool on

sql => spool upgtzv\_check.log

sql => @upg\_tzv\_check.sql

sql => exit

Upgrade files (Note: The database will be bounced twice)

Disconnect all other connections to the database

run => sqlplus / as sysdba

sql => set trimspool on

sql => spool upgtzv\_apply.log

sql => @upg\_tzv\_apply.sql

sql => exit

A successful run will show at the end:

INFO: The RDBMS DST update is successfully finished.

INFO: Make sure to exit this sqlplus session.

INFO: Do not use it for timezone related selects.

Get version

SELECT version FROM v$timezone\_file;

Check database alert log for errors

view $BDUMP/alert\_${ORACLE\_SID}.log

1. Run the post fixup script

run => cd $HOME/cfgtoollogs/$ORACLE\_SID/preupgrade

sql => @postupgrade\_fixups.sql

If the following is returned:

Fixup Returned Information:

INFORMATION: --> Older Timezone in use

then Run

SELECT PROPERTY\_NAME, SUBSTR(property\_value, 1, 30) value

FROM DATABASE\_PROPERTIES  
 WHERE PROPERTY\_NAME LIKE 'DST\_%'

ORDER BY PROPERTY\_NAME;

1. **Upgrade Statistics Tables Created by the DBMS\_STATS Package**

run => sqlplus / as sysdba

select owner, table\_name

from dba\_tab\_cols

where column\_name in ('STATID','TYPE','CH1','CL1')

and owner != 'SYS'

group by owner,table\_name

having count(column\_name) = 4;

If rows returned, run below for each replacing owner and table\_name

EXECUTE DBMS\_STATS.UPGRADE\_STAT\_TABLE('OWNER','TABLE\_NAME');

PPMTEST – No rows

PPMDEBIZ

GL STAT

XLA STAT

AEXLA STAT

EXECUTE DBMS\_STATS.UPGRADE\_STAT\_TABLE('GL','STAT');

EXECUTE DBMS\_STATS.UPGRADE\_STAT\_TABLE('XLA','STAT');

EXECUTE DBMS\_STATS.UPGRADE\_STAT\_TABLE('AEXLA','STAT');

1. Alter all system, sysaux datafiles and last TEMP file to autoextend options as

prior to the upgrade

run => cd $HOME/admin/$ORACLE\_SID/upgora12102

run => sqlplus / as sysdba

sql => @alter\_system\_datafiles\_autoextend\_off.sql

sql => exit

run => view alter\_system\_datafiles\_autoextend\_off.out 🡸 to ensure no

errors

1. Collect segment statistics after upgrading the database

run => cd $HOME/admin/$ORACLE\_SID/upgora12102/stats

run => nohup run\_stats\_eb.ksh ${ORACLE\_SID} aftora12upgrade `pwd` &

🡸 sends email when complete

nohup run\_stats\_eb.ksh ${ORACLE\_SID} b4ora12upgrade `pwd` system/Z00mZ00m! &

run => view \*aftora12upgrade\*

run => sqlplus / as sysdba

sql => @dbsize.sql aftora12upgrade

if you want run from DBArtisan to make sure

select \* from DBA\_REGISTRY\_SQLPATCH;

1. Fulfill ICR requirement and other security tasks

Starting 12c, the SELECT ANY DICTIONARY privilege no longer permits access to security sensitive data dictionary tables DEFAULT\_PWD$, ENC$, LINK$, USER$, USER\_HISTORY$, and XS$VERIFIERS. This change increases the default security of the database by not allowing access to a subset of data dictionary tables through the SELECT ANY DICTIONARY privilege.

grant select on sys.user$ to ICR\_AUDIT;

GRANT SELECT ON SYS.USER$ TO SYSTEM;

alter system disable restricted session;

1. To address bug described in (Doc ID 1676376.1) during Full DB export

# Do this if only dpadmin user exist or other account to be used for datapump job.

If other account in use modify script.

cd $HOME/admin/$ORACLE\_SID/upgora12102

run => sqlplus / as sysdba

sql => @exportgrants.sql

1. Update compatible in initialization file and bounce the database

\*.compatible='12.1.0'

Reference - 733987.1

run => vi $DBS/init${ORACLE\_SID}.ora

e.g.

## 04/09/2015 TS Changes in support of 12c upgrade

## \*.compatible='11.2.0' to \*.compatible='12.1.0'

run => sqlplus / as sysdba

sql => select name from v$database;

sql => shutdown immediate;

sql => startup;

sql => show parameter compatible

Check alertlog

view $BDUMP/alert\_${ORACLE\_SID}.log

1. Set database parameters per note 2034610.1

If spfile in use

alter system set "\_optimizer\_aggr\_groupby\_elim"=false scope=both;

alter system set "\_optimizer\_reduce\_groupby\_key"=false scope=both;

if pfile in use

run => vi $DBS/init${ORACLE\_SID}.ora

\_optimizer\_aggr\_groupby\_elim=false

\_optimizer\_reduce\_groupby\_key=false

--Uncomment for now

\*.sec\_case\_sensitive\_logon=FALSE

run => sqlplus / as sysdba

sql => select name from v$database;

sql => shutdown immediate;

sql => startup;

1. If need create SPFILE and restart DB

run => sqlplus / as sysdba

sql => create spfile from pfile;

sql => shutdown immediate;

sql => startup;

sql => show parameter spfile

**Doc ID 1524398.1**

**After the Database Upgrade…steps**

~~recyclebin=off~~

#\_sqlexec\_progression\_cost = 2147483647

Sqlnet.ora

SQLNET.ALLOWED\_LOGON\_VERSION\_SERVER = 8

If you are on Oracle E-Business Suite Release 12.1, ensure that your sqlnet\_ifile.ora has the line:

SQLNET.ALLOWED\_LOGON\_VERSION\_SERVER = 8 (if the initialization parameter SEC\_CASE\_SENSITIVE\_LOGON is set to FALSE)

SQLNET.ALLOWED\_LOGON\_VERSION\_SERVER = 10 (if SEC\_CASE\_SENSITIVE\_LOGON is set to TRUE)

DEV login to Linux App server and copy: xppmw3d

--Get oracle password from TPAM

PROD login to Linux App server and copy: xppmm3p

PROD login to Linux App server and copy: xppmw3p

E.B.

scp -rp /aetnas28/asrxoracle/DBA\_REQ\_FILES aorinfw1d:/orahome/allu01/app/oracle/admin/PPMDEBIZ/upgora12102/DBA\_REQ\_FILES

Item 21. Copy $APPL\_TOP/admin/adgrants.sql (adgrants\_nt.sql for Windows) from the administration server node to the database server node. Use SQL\*Plus to connect to the database as SYSDBA and run the script using the following command:

--check with Hameed which version to run

cd $HOME/admin/$ORACLE\_SID/upgora12102/DBA\_REQ\_FILES

or

cd $ORACLE\_HOME/appsutil/sql

$ sqlplus "/ as sysdba" @adgrants.sql (or adgrants\_nt.sql)

[APPS schema name]

sqlplus / as sysdba

@adgrants.sql

=> APPS

Item22. Grant create procedure privilege on CTXSYS

Copy $AD\_TOP/patch/115/sql/adctxprv.sql from the administration server node to the database server node. Use SQL\*Plus to connect to the database as APPS and run the script using the following command:

$ sqlplus apps/[APPS password] @adctxprv.sql \

[SYSTEM password] CTXSYS

sqlplus

APPS/Ac\_nGe\_d

@adctxprv.sql

=> Z00mZ00m! (or get from TPAM for PROD)

=> CTXSYS

Item 23. Set CTXSYS parameter

Use SQL\*Plus to connect to the database as SYSDBA and run the following command:

$ sqlplus "/ as sysdba"

SQL> exec ctxsys.ctx\_adm.set\_parameter('file\_access\_role', 'public');

sqlplus / as sysdba

exec ctxsys.ctx\_adm.set\_parameter('file\_access\_role', 'public');

Item 24: Validate Workflow ruleset

On the administration server node, use SQL\*Plus to connect to the database as APPS and run the $FND\_TOP/patch/115/sql/wfaqupfix.sql script using the following command:

Login to xppmw3d

cd /aetnas28/asrxoracle/DBA\_REQ\_FILES

$ sqlplus [APPS user]/[APPS password] @wfaqupfix.sql\

[APPLSYS user] [APPS user]

sqlplus

apps/Ac\_nGe\_d@PPMDEBIZ

@wfaqupfix.sql

=> APPLSYS

=> APPS

Item 25: Gather statistics for SYS schema

Copy $APPL\_TOP/admin/adstats.sql from the administration server node to the database server node. Note that adstats.sql has to be run in restricted mode. Use SQL\*Plus to connect to the database as SYSDBA and use the following commands to run adstats.sql in restricted mode:

sqlplus / as sysdba

SQL> alter system enable restricted session;

SQL> @adstats.sql

sqlplus / as sysdba

SQL> alter system disable restricted session;

SQL> exit;

Item 26: Deregister the current database server (conditional)

If you plan to change the database port, host, SID, or database name parameter on the database server, you must also update AutoConfig on the database tier and deregister the current database server node.

sqlplus

apps/Ac\_nGe\_d

SQL> exec fnd\_conc\_clone.setup\_clean;

Item 27: Implement and run AutoConfig

See Using AutoConfig to Manage System Configurations in Oracle E-Business Suite Release 12 on My Oracle Support,

especially section 3.2, for instructions on how to implement and run AutoConfig.

Ask Hameed to provide file and Login back to Dev App box to copy it.

scp -p /aetnas28/asrxoracle/DBA\_REQ\_FILES/appsutil.zip aorinfw1d:/orahome/allu01/app/oracle/admin/PPMDEBIZ/upgora12102/DBA\_REQ\_FILES/appsutil.zip

Login back to oracle box

cd /orahome/allu01/app/oracle/admin/PPMDEBIZ/upgora12102/DBA\_REQ\_FILES

cp appsutil.zip $ORACLE\_HOME/appsutil.zip

cd $ORACLE\_HOME

unzip -o appsutil.zip

Attention: After creating the appsutil directory and before running AutoConfig, copy orai18n.jar from $ORACLE\_HOME/jlib to $ORACLE\_HOME/jdk/jre/lib/ext

cd $ORACLE\_HOME/jlib

cp orai18n.jar $ORACLE\_HOME/jdk/jre/lib/ext/orai18n.jar

Generate the Database Context File

Execute the following command to create your Database Context File:

E.B.

cd $ORACLE\_HOME/appsutil/bin/

export DISPLAY=localhost:0

export TNS\_ADMIN=/orahome/u01/app/oracle/product/12.1.0.2.180116/db\_1/network/admin

perl adbldxml.pl

--> Ac\_nGe\_d

--> aorinfw1d

--> 1551

--> PPMDEBIZ

The context file has been created at:

/orahome/u01/app/oracle/product/12.1.0.2.180116/db\_1/appsutil/PPMDEBIZ\_aorinfw1d

.xml

Run AutoConfig on the Database tier

cd $ORACLE\_HOME/network/admin

backupa listener, tnsnames.r ans seln net.ora

####-- To avoid ORA-12705: Cannot access NLS data files or invalid environment specified

####--Doc ID 2165613.1

####Run the following script after taking a backup:

##### perl $ORACLE\_HOME/nls/data/old/cr9idata.pl

### Once the scripts complete, confirm that the data is corrected.

#### $ORACLE\_HOME/nls/data/9idata

### ls -ltr

Attention: After creating the XML context file, ensure the variable s\_jretop points to $ORACLE\_HOME/jdk/jre and is not altered to any other value.

export s\_jretop=$ORACLE\_HOME/jdk/jre

cd $ORACLE\_HOME/appsutil/bin/

./adconfig.sh contextfile=/orahome/u01/app/oracle/product/12.1.0.2.180116/db\_1/appsutil/PPMDEBIZ\_aorinfw1d.xml

--> Ac\_nGe\_d

--Check for errors

The log file for this session is located at: /orahome/u01/app/oracle/product/12.1.0.2.180116/db\_1/appsutil/log/PPMDEBIZ\_aorinfw1d/05291609/adconfig.log

**--AUtoconfi will overlay listener.ora, tnsnames.ora and slqnet.ora**

**-- Restore from saved version**

cp listener.ora listener.ora\_afterAutoConfig

cp tnsnames.ora tnsnames.ora\_afterAutoConfig

cp sqlnet.ora sqlnet.ora\_afterAutoConfig

mv listener.ora\_b4AutoConfig listener.ora

mv tnsnames.ora\_b4AutoConfig tnsnames.ora

mv sqlnet.ora\_b4AutoConfig sqlnet.ora

lsnrctl reload PPMDEBIZ

Item 28: Create the new MGDSYS schema (conditional)

If you upgraded from an RDBMS version prior to 12c, use SQL\*Plus to connect to the database as SYSDBA and run the $ORACLE\_HOME/rdbms/admin/catmgd.sql script. This creates the new MGDSYS schema.

$ sqlplus "/ as sysdba" @?/rdbms/admin/catmgd.sql

alter profile STANDARD limit PASSWORD\_VERIFY\_FUNCTION null

/

alter profile DEFAULT limit PASSWORD\_VERIFY\_FUNCTION null

/

sqlplus / as sysdba

@?/rdbms/admin/catmgd.sql

ALTER PROFILE STANDARD LIMIT PASSWORD\_VERIFY\_FUNCTION AI\_PASSWORD\_VALIDATE

/

ALTER PROFILE DEFAULT LIMIT PASSWORD\_VERIFY\_FUNCTION AI\_PASSWORD\_VALIDATE

/

Sslplus

PPMFIN/ppmfin91

drop database link FAHDEV.AETNA.COM;

select \* from global\_name@FAHDEV.AETNA.COM

1. Take a full database backup

**Take fresh backup**

cd $SCRIPTS/backup

./rman\_general\_purpose\_oem.ksh PPMDEBIZ RCATDEV incremental\_level0\_backup\_tape\_oem.rman Y

#Once backup completed. Get latest log file name

cd $INSTANCE\_OUTPUTS/rman\_backups

ls -altr \*.log

#Save Backup

$BKPSCR/savebkp.ksh $ORACLE\_SID <backup log file name> postupgora12102\_tstz

PPMDEBIZ (had issues with savebkp.ksh just extracted from backup log)

##Restore DataBase

#connect target /

#connect catalog PPMDEBIZ/PPMDEBIZnamr@RCATDEV

#startup force nomount;

#set DBID=3566175702;

#reset database to incarnation 615632;

#restore controlfile from 'c-3566175702-20180529-01\_PPMDEBIZ';

#alter database mount;

#run{

#restore database until sequence 284;

#recover database until sequence 284;

#}

#alter database open resetlogs;

##End Restore DataBase

1. For Production only:

Configure RMAN tape channels back to 4

rman

connect target /

connect catalog ID/PW@RCATPROD

CONFIGURE DEVICE TYPE 'SBT\_TAPE' PARALLELISM **4** BACKUP TYPE TO BACKUPSET;

show all;

exit

1. Remove exit 0 from backup script

**For production only**

cd $BKPSCR

~~cp –p rman\_general\_purpose\_4upg.ksh rman\_general\_purpose\_oem.ksh~~

1. Log on to appropriate application server

~~old servers: XPSAPM2D, XPSAPM1Q, XPSAPW1Q, XPSAPM1P, XPSAPW1P~~

~~new servers: xpsdppw2d, xpsdppm1q, xpsdppw1q~~

1. Change $ORACLE\_HOME directory for database and listener in OEM
2. Stop Grid Control Blackout Period via OEM

**Resume any suspended jobs in OEM**

1. Enable any disabled cron entries

crontab -e

~~vi /orahome/u01/aetna/scripts/setupenv.ksh~~

vi /orahome/allu01/aetna/scripts/setupenv.ksh

alias bdump='cd $ORACLE\_BASE/admin/${ORACLE\_SID}/diag/rdbms/${lc\_instance}/${ORACLE\_SID/trace'

export BDUMP=$ORACLE\_BASE/admin/${ORACLE\_SID}/diag/rdbms/${lc\_instance}/${ORACLE\_SID}/trace

. $SETUPENV $ORACLE\_SID

echo $BDUMP

1. Send an email to App admin that the database is now at 12.1.0.2 and let them know the OH to use on the db server and the app server
2. Gather Fixed Objects Statistics with DBMS\_STATS

**A few days after upgrading Oracle Database**, a best practice is to

Gather fixed objects statistics with the

DBMS\_STATS.GATHER\_FIXED\_OBJECTS\_STATS

PL/SQL procedure. This can have a positive impact on overall database

performance. DBMS\_STATS.GATHER\_FIXED\_OBJECTS\_STATS also displays a

recommendation to remove all hidden or underscore parameters and events

from init.ora/spfile. Because of the transient nature of the x$ tables,

it is important that you gather fixed objects statistics when there is a

representative workload on the system. This may not always be feasible on

large systems due to additional resources needed to gather the statistics.

If you cannot do this during peak load, then you should do it after

the system has warmed up and the key types of fixed object tables have

been populated. To gather statistics for fixed objects, run the following

PL/SQL procedure:

SQL> execute dbms\_stats.gather\_fixed\_objects\_stats;

**\*\*\*\*\*\*\*\*\*Hameed asked for PPMDEBIZ license issue\*\*\***

DATA\_PUMP\_DIR3

/oraexport/meru276/PPMDEBIZ

1. Export tables below from PPMRX schema

KNTA\_LICENSE\_DATASOURCE

KNTA\_SECRETKEY\_DATASOURCE

expdp SYSTEM/Z00mZ00m! DIRECTORY=DATA\_PUMP\_DIR3 DUMPFILE=PPMRX\_KNTA\_LICENSE\_DATASOURCE.dmp TABLEs=PPMRX.KNTA\_LICENSE\_DATASOURCE

expdp SYSTEM/Z00mZ00m! DIRECTORY=DATA\_PUMP\_DIR3 DUMPFILE=PPMRX\_KNTA\_SECRETKEY\_DATASOURCE.dmp TABLEs=PPMRX.KNTA\_SECRETKEY\_DATASOURCE

2. Export tables below from PPMFIN schema

KNTA\_LICENSE\_DATASOURCE

KNTA\_SECRETKEY\_DATASOURCE

expdp SYSTEM/Z00mZ00m! DIRECTORY=DATA\_PUMP\_DIR3 DUMPFILE=PPMFIN\_KNTA\_LICENSE\_DATASOURCE.dmp TABLEs=PPMFIN.KNTA\_LICENSE\_DATASOURCE

expdp SYSTEM/Z00mZ00m! DIRECTORY=DATA\_PUMP\_DIR3 DUMPFILE=PPMFIN\_KNTA\_SECRETKEY\_DATASOURCE.dmp TABLEs=PPMFIN.KNTA\_SECRETKEY\_DATASOURCE

3. Truncate tables below from PPMFIN schema

TRUNCATE TABLE PPMFIN.KNTA\_LICENSE\_DATASOURCE;

TRUNCATE TABLE PPMFIN.KNTA\_SECRETKEY\_DATASOURCE;

4. Import export file from step 1 (from PPMFX to PPMFIN schema)

SQL> select count(\*) from PPMFIN.KNTA\_LICENSE\_DATASOURCE

COUNT(\*)

----------

0

impdp SYSTEM/Z00mZ00m! DIRECTORY=DATA\_PUMP\_DIR3 DUMPFILE=PPMRX\_KNTA\_LICENSE\_DATASOURCE.dmp REMAP\_SCHEMA=PPMRX:PPMFIN TABLE\_EXISTS\_ACTION=APPEND

SQL> select count(\*) from PPMFIN.KNTA\_LICENSE\_DATASOURCE

2 ;

COUNT(\*)

----------

9

**Backout Procedures**

The below steps can be used for down grading the database from 12.1.0.2 to 11.2.0.3

The assumption here is compatible was changed and a database backup was taken at release level 12.1.0.2

1. Set Oracle Environment for database being downgraded.

cd $HOME

. SETUPENV UPGDB

**Set up Backout Variables**

1. Set variables for database being downgraded

UPGDB=name\_of\_db\_being\_upgraded 🡸 replace with database name

ORATAB=/etc/oratab

RCATDB=name\_of\_rman\_catalog\_for\_db\_being\_upgraded

REASON=b4dwngora12102 reason\_for\_RMAN\_catalog\_backup

1. Check for cron jobs for this database that need to be disabled during the downgrade:

cd $HOME/admin/$ORACLE\_SID/upgora12102 crontab -l |grep $ORACLE\_SID|grep -v ^#

Save crontab and edit if changes are required

crontab –l > crontab\_` date '+%Y%m%d'`\_b4ora12102dwng\_${ORACLE\_SID}

crontab -e

1. Save and update oratab indicator to N to disable cron jobs.

cd $HOME/admin/$ORACLE\_SID/upgora12102

cp –p ${ORATAB} oratab\_` date '+%Y%m%d'`\_b4ora12102dwng\_${ORACLE\_SID}

vi /etc/oratab

1. Have app admins shut down all application processes connected to the database

Verify all application processes disconnected:

sqlplus “/ as sysdba”

Select username from v$session;

exit

no rows with user ids should be returned

1. Make sure backup script will not fire

**For production only**

~~vi rman\_general\_purpose.ksh 🡸 Add exit 0 to top of script~~

1. Take RMAN archive log backup (if applicable)

cd $SCRIPTS/backup

./rman\_general\_purpose\_oem.ksh $ORACLE\_SID RCATDB archivelog\_backup\_tape\_oem.rman

Email will be sent when complete

ls –lrt $BKPLOG/\* |tail -10 🡸 verify that there is an output log for the current time

view `ls -lrt $BKPLOG/\*log|awk '{print $9}'|tail -1` 🡸 view most recent backup log to verify successful completion

Copy log output to upgrade directory to identify when taken

cp –p `ls -rt ${BKPLOG}/\*archivelog\_backup\_tape.rman.log|tail -1` archivelog\_backup\_b4dwngora12102.log

1. Export RMAN catalog schema from RMAN catalog server (Be sure to change par file

content so previous

info is not overwritten. This was

performed above as part of the upgrade.

See steps above for details).

**Note:** If any RMAN configuration parameters were changed for the upgrade you may want to modify them for the period during the backout and modify them back later.

Need to run export as sysdba

exp \'/ as sysdba\' parfile=some\_parfile\_name.par

1. Collect RMAN catalog parameters before backout

rman\_catlgparms.ksh $UPGDB $RCATDB b4backout

1. Start grid control blackout period for database and listener being downgraded via OEM

Be sure to uncheck jobs running during blackout period box

1. Set environment and $ORACLE\_SID to instance being downgraded

**. $SETUPENV $UPGDB** 🡸 set up environment for the database being downgraded

1. Shutdown listener

***For listeners with only one database:***

*lsnrctl stop* ${LSNRNAME} 🡸replace listener name as appropriate

***For listeners listening for more than one database:***

*set environment for Oracle Home where listener is running – needs to be 12c*

*. $SETUPENV TNS11D 🡸 specify a SID that will set the OH to the listener home*

*cd $TNS\_ADMIN*

*LSNRNAME=*${LSNRNAME}🡸 replace with listener name

*~~cp –p listener.ora listener.ora\_b4${LSNRNAME}dwngrd\_`date +'%Y%m%d`~~*

*~~vi listener.ora 🡸 comment out the entries for the database SID being downgraded~~*

*~~add this parameter if not already in the listener.ora~~*

~~DYNAMIC\_REGISTRATION\_~~*~~PLANVIEWD~~*~~=OFF 🡸 replace with listener name~~

*lsnrctl status ${LSNRNAME} 🡸 see database SID in service list*

*lsnrctl stop* ${LSNRNAME}*;*

*Change $ORACLE\_HOME back to 11.2.0.3 location for database being downgraded*

*lsnrctl start* ${LSNRNAME}*; 🡸 see The command completed successfully*

*lsnrctl status ${LSNRNAME} 🡸 verify that the SID is not in the service list*

1. Set environment and $ORACLE\_SID to instance being downgraded

**. $SETUPENV $UPGDB** 🡸 set up environment for the database being downgraded

1. Shut down the database

sqlplus “/ as sysdba”

shutdown immediate

exit

1. Remove all applicable database files from the database server

data files

control Files

temp files

redo logs

ls –lrt /ora[drt]\*/\*/${ORACLE\_SID}/\* 🡸 verify list of database files that

will be removed

rm /ora[drt]\*/\*/${ORACLE\_SID}/\* 🡸 Takes several minutes on zLinux

1. Update entry in oratab to revert back to 11.2.0.3

vi /etc/oratab

1. Set Oracle environment to 11.2.0.4

**. $SETUPENV $UPGDB**

1. Confirm $ORACLE\_HOME set to 11.2.0.4

echo $ORACLE\_HOME

**Note: Steps 19-23 are optional. RMAN catalog can stay at 12c**

1. Extract create user statement for RMAN catalog user from RMAN catalog database via **DBArtisan**
2. Drop RMAN catalog user from RMAN catalog database

sql => drop user xxx cascade;

1. Create RMAN catalog user in RMAN catalog database
2. uncompress the RMAN catalog dump file to be imported

gunzip some\_file\_name.gz

1. Import 11.2.0.3 RMAN catalog schema taken prior to upgrade to 12.1.0.2

Sample import parameter file would look something like:

buffer=409600 file=/u01/app/oracle/admin/RCATDEV/exports/export\_schema\_DBNAME\_b4upgora12c.dmp log=/u01/app/oracle/admin/RCATDEV/exports/import\_schema\_DBNAME\_b4upgora12c.log

ignore=Y

commit=Y

fromuser=DBNAME

touser=DBNAME

run => imp parfile=some\_file\_name.par

**Note**: This will invalidate a bunch of views

select object\_name from dba\_objects where owner = 'DBNAME' and status <> 'VALID'; 🡸 replace DBNAME with database being downgraded

The views will need to be recompiled.

e.g.

alter view DBNAME.RC\_UNUSABLE\_BACKUPFILE\_DETAILS compile;

To build the recompile commands run:

select 'alter '||object\_type||' '||owner||'.'||object\_name||' compile;'

from dba\_objects

where owner = 'XXX'

and object\_type <> ‘TYPE BODY’

and status <> 'VALID'; 🡸 replace XXX with database being downgraded

select 'alter TYPE '||owner||'.'||object\_name||' compile BODY;'

from dba\_objects

where owner = 'XXX'

and object\_type = 'TYPE BODY'

and status <> 'VALID'; 🡸 replace XXX with database being downgraded

**Note:** Restoring an RMAN catalog that does not include backups taken after the catalog was exported will result in orphan files that the TSM catalog will be aware of but the RMAN catalog will not. To clean up the TSM files a TDPOSYNC is required.

On Database Server:

1. Start listener

***For listeners with only one database:***

*lsnrctl start* ${LSNRNAME} 🡸replace listener name as appropriate

***For listeners listening for more than one database:***

*set environment for Oracle Home where listener is running – needs to be 12c*

*. $SETUPENV TNS11D 🡸 specify a SID that will set the OH to the listener home*

*cd $TNS\_ADMIN*

*vi listener.ora 🡸 uncomment the entries for the database SID being downgraded and change oracle home to match the oratab oracle home for the database*

*add this parameter if not already in the listener.ora*

~~DYNAMIC\_REGISTRATION\_~~*~~PLANVIEWD~~*~~=OFF 🡸 replace with listener name~~

*lsnrctl status ${LSNRNAME} 🡸 see database SID not in service list*

*lsnrctl stop* ${LSNRNAME}*; lsnrctl start* ${LSNRNAME}*; 🡸 see The command completed successfully*

*lsnrctl status ${LSNRNAME} 🡸 verify that the SID is not in the service list*

1. Set environment and $ORACLE\_SID to instance being downgraded

**. $SETUPENV $UPGDB** 🡸 set up environment for the database being downgraded

1. Rename alertlog

cd $BDUMP

mv alert\_${ORACLE\_SID}.log alert\_${ORACLE\_SID}.log\_`date +%m:%d:%Y:Time:%H:%M:%S`

1. Restore database backup taken prior to upgrade to 12.1.0.2

export NLS\_DATE\_FORMAT='YYYY-MM-DD HH24:MI:SS'

See RMAN commands in saved file or database backup log

1. Review alert log for errors

view $BDUMP/alert\_$ORACLE\_SID.log

1. Confirm database is in archivelog mode (if applicable)

select name, log\_mode from v$database;

1. Collect RMAN catalog parameters after backout

cd $HOME/admin/$ORACLE\_SID/upgora12102

rman\_catlgparms.ksh $ORACLE\_SID $RCATDB postbackout

1. Remove obsolete backups prior to running crosscheck further below

run => rman

rman => connect target /

rman => connect catalog $ORACLE\_SID/$ORACLE\_SIDnamr@$RCATDB

rman => report obsolete;

rman => delete noprompt obsolete;

1. Crosscheck to identify and fix any issues

rman

connect target /

connect catalog ID/PW@RCATDEV or RCATPROD 🡸 replace ID and PW and catalog

crosscheck backup;

if encounter:

**validation failed for archived log** 🡸 in backup log

**or**

**crosschecked backup piece: found to be 'EXPIRED'**

then run (may want to note file(s) to determine the sequence of events)

To clean up entries marked as expired:

delete noprompt expired backup;

1. Confirm HeartBeat Timestamp is in the general time frame of the restore point

export NLS\_DATE\_FORMAT='YYYY-MM-DD HH24:MI:SS'

sqlplus /

select \* from aetnadba.rman\_heartbeat order by timestmp;

exit

1. Change $ORACLE\_HOME directory for database and listener in OEM
2. Stop Grid Control Blackout Period via OEM
3. Set environment and $ORACLE\_SID to instance being downgraded

**. $SETUPENV $UPGDB** 🡸 set up environment for the database being downgraded

1. Take a hot backup of database to ensure recoverability

cd $SCRIPTS/backup

./rman\_general\_purpose\_oem.ksh $ORACLE\_SID RCATDB incremental\_level0\_backup\_tape\_oem.rman

May encounter errors like

RMAN-03009: failure of Control File Autobackup command on ORA\_SBT\_TAPE\_1

channel at 04/17/2015 16:02:21

ORA-19506: failed to create sequential file, name="c-3762674548-20150417-

ff\_PLNVD001", parms=""

ORA-27027: sbtremove2 returned error

ORA-19511: Error received from media manager layer, error text:

ANU2614E Invalid sequence of function calls to Data Protection for Oracle

If errors encountered, run another backup to ensure all is well

1. For Production only:

Confirm following is set

rman

connect target /

connect catalog ID/PW@RCATPROD

CONFIGURE DEVICE TYPE 'SBT\_TAPE' PARALLELISM **4** BACKUP TYPE TO BACKUPSET;

show all;

exit

1. Unsuspend backup job where applicable

**For production only**

cd $SCRIPTS/backup

cp –p rman\_general\_purpose\_4upg.ksh rman\_general\_purpose\_oem.ksh 🡸 To remove exit 0 from top of script

1. **Turn on cron for this database**

cd $HOME/admin/$ORACLE\_SID/upgora12102

crontab crontab\_` date '+%Y%m%d'`\_b4ora11gdwng

crontab –e 🡸 verify the crontab for this database

To check for cron jobs not disabled for this database:

crontab -l |grep $ORACLE\_SID|grep -v ^#

To check for cron jobs for all databases not disabled:

crontab -l |grep -v ^#

1. **Confirm startup indicator set to Y in /etc/oratab for this database**