Upgradation

Of Oracle Database 19c.

Links:

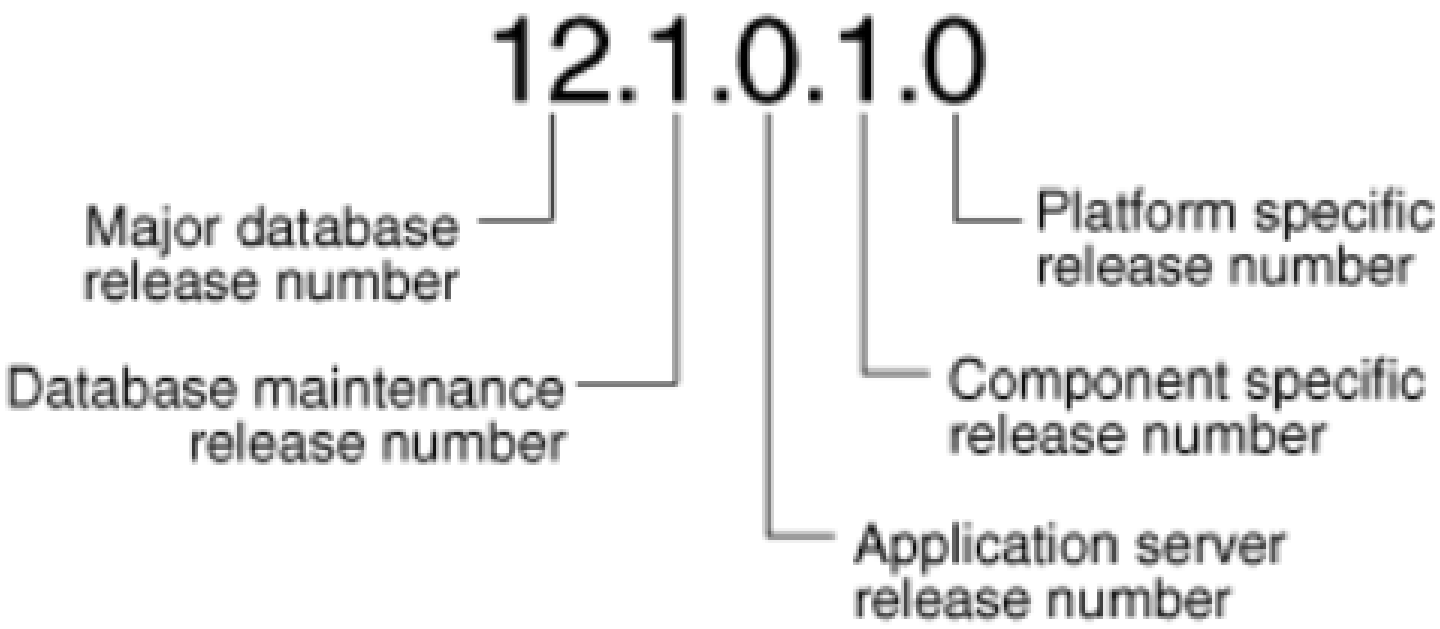
https://asrblogger.com/12cr1-to-19c-upgradation-with-auto-upgrade-feature/

https://asrblogger.com/upgrade-oracle-database-from-12-1-0-2-to-12-2-0-1-using-dbua/

Microsoft Word Document

Contents:

About Oracle Database Release Numbers



Comparing Upgradation & Migration:

- Although the terms are often used as synonyms in other contexts, in the context of Oracle Database there is a necessary distinction between database upgrade and database migration.

- Understanding this difference is the first step in choosing the best upgrade or migration method for your project.

Note:

The term “migration” can also be used when discussing the move of data from a non-Oracle database into Oracle. This white paper will cover migrations only when both the source and destination are Oracle databases.

Database Upgrade

- The act of upgrading an Oracle Database involves modifying the data dictionary to be compatible with a newer version of Oracle Database software. Typical actions that may be part of a database upgrade include:

  - Adding, dropping, or modifying columns in system tables and views
  - Creating or modifying system packages or procedures
  - Creating, modifying, or dropping database types, users, roles, and privileges
  - Modifying seed data that is used by Oracle Database components.

All of these actions affect the data dictionary of your database. They do not affect the data stored in your user or application tablespaces

Database Migration

The term “migration” applies to several different types of changes that can be applied to an Oracle database. In addition to database version, these can include a change to any or all of the following:

- Computer server (hardware or virtualized environment)
- Storage architecture 4 WHITE PAPER /Upgrade and Migrate to Oracle Database 19c
- Character set
- Operating system
- Schema topology (changing the partitioning scheme)
- Encryption » Compression
- Database architecture (moving into the multitenant database architecture)

SELECTING A DATABASE UPGRADE OR MIGRATION METHOD

Best upgrade or migration method for a particular project:

- The version from which you are upgrading or migrating, including the patch set level
- The source and destination operating system and version .
- Any plans to change the actual data layout or format, such as changing the character set, partitioning, encryption.
- Availability requirements including the amount of downtime allowed for the upgrade or migration project, fallback possibilities and disaster recovery
- The size of the database to be migrated
- The source and target database architecture, whether non-CDB or PDB.

Upgradation & Migration Methods:

There are 3 upgrade and migration methods described.

The methods are:

1. Database Upgrade, using either the command-line upgrade with dbupgrade or the new Autoupgrade, or DBUA

2. Transportable tablespaces (TTS) export and import, using the Oracle Database feature full transportable export/import, or the traditional TTS mode

3. Oracle Data Pump Export/Import, using either dump files or network mode

DATABASE UPGRADE AND MIGRATION METHODS

Method	Complexity	Speed	Minimum Source Version	Move to New Server	Change O/S	Change Data Layout, character set, encryption, compression
Unplug, Plug and Upgrade	Med	Fastest	12.1.0.2	Yes	No	No
Command-line Upgrade	Med	Fastest	11.2.0.4	Yes	No	No
Database Upgrade Assistant	Low	Fastest	11.2.0.4	No	No	No
Full Transportable Export/Import	Med	Faster	11.2.0.4	Yes	Yes	No
Transportable Tablespaces	High	Faster	8.1.5	Yes	Yes, starting with 10.1	No
Data Pump expdp/impdp	Med	Fast	10.1	Yes	Yes	Yes

Upgrade Oracle Database from 12.2.0.1 to 19.0.0.0 Using DBUA

- Important points:
  - Direct upgrade to 19c can be performed from 11.2.0.4, 12.1.0.2, 12.2.0.1 & 18c.
  - Compatible parameter should be at minimum 11.2.0
  - Post upgrade, you may not be able to login to the existing users with the password, because of new authentication method.

To fix this, sqlnet.ora file need to be update

What is Auto Upgrade Utility?

The Oracle Database AutoUpgrade utility is a new command-line tool that allows you to upgrade your databases in an unattended way.

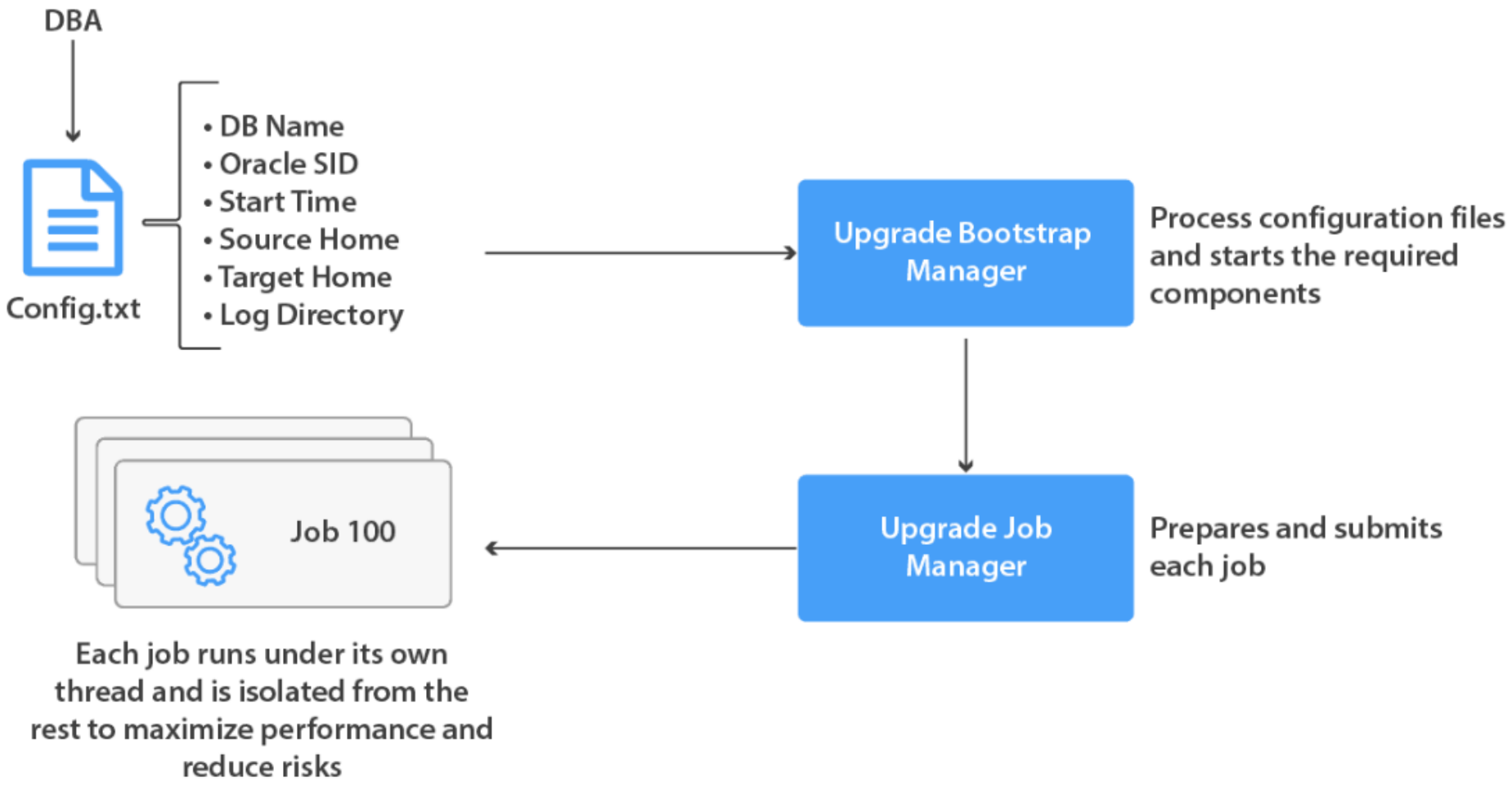
The idea of the tool is to run the prechecks against multiple databases, fix 99% of the potential issues, set a restore point in case something goes wrong – and then upgrade your databases.

And of course, do the post-upgrade, recompilation, and time zone adjustment.

Benefits of Auto Upgrade Utility

- Enables customers to upgrade one or many Oracle databases at the command-line with a single command and a single configuration file.
- Will run the pre-upgrade tasks, perform automated fixups where needed, execute the database upgrade, and finish by taking care of post-upgrade tasks.
- Includes automatic retry and fallback, the possibility to schedule upgrades for future points in time, and the ability to set, change or remove initialization parameters as desired.
- Saves time and money by upgrading hundreds of databases with one command and replacing bespoke high maintenance upgrade solutions.

Auto Upgrade Workflow



Modes of Auto Upgrade:

- **Analyze Mode:** Setup->Prechecks.
- **Fixups Mode:** Setup, Prechecks, and Prefixups.
- **Deploy Mode:** Setup, Pre upgrade, Prechecks, guaranteed to restore point (GRP), Prefixups, Drain, DB Upgrade, Post checks, and Post upgrade. You can run your own scripts before the upgrade or after the upgrade, or both before and after the upgrade.
- **Upgrade Mode:** Setup, DB Upgrade, Post checks, and Postfixups.

ANALYZE: This mode perform pre-checks on source database to see if it is ready for upgrade. It will only ANALYZE the source database and will to perform any changes.

FIXUPS: This mode performs the pre-checks on source database, also it will attempt to fix up the items that must be corrected before it can be upgraded.

DEPLOY: This mode will perform actual upgrade of the source database. If needed it can also perform mixups on database.

UPGRADE: Upgrade mode enables us to upgrade the target Oracle Home in cases where we do not have access to source Oracle Home.

We can run auto upgrade directly in DEPLOY mode without running ANALYZE and FIXUPS mode. But if we do not want to take chances to see unforeseen abortion of upgrade, we better run it in ANALYZE and FIXUPS mode before running it in DEPLOY mode

High-level Steps For Auto-Upgrade

1. Install Oracle 19.15.0.0 binaries
2. Prerequisite for Auto upgrade
3. Create the config file
4. Analyze the database
5. Deploy the upgrade
6. Post upgrade task

Pre Downtime Preparation

High Level Steps

✓ Log a proactive SR with Oracle and attach the work plan.

✓ Log a CRQ## with respective teams.

✓ Schedule a meeting with all stakeholders a day/week before Upgrade.

✓ There should be appropriate approval from client and management.

✓ Create a workplan for DB upgrade which constitutes:

-> Preupgrade

-> DBUpgrade

-> PostUpgrade

High Level Steps

1. Installation of Oracle 19c Database binaries.

- Manual Database Installation.
- DBUA
- Silent Installation

High Level Steps

2. Apply the most recent 19c Release Update and additional patches

For Grid/ASM

```
export ORACLE_HOME=/u01/grid/oracle/product/19/grid
$ORACLE_HOME/OPatch/opatch lsinventory | grep -e applied -e description
```

e.g. `Database Release Update : 19.15.0.0.`

For Database

- `export ORACLE_HOME=/u01/oracle/product/19/db`
- `$ORACLE_HOME/OPatch/opatch lsinventory | grep -e applied -e description`

If Patch is not up to date, please proceed with patching to the latest RU.

High Level Steps

3. Confirm minimum client version. (Greater than 11.2.0.3)

```
select distinct client_version from GV$SESSION_CONNECT_INFO;
```

- Client versions listed as "Unknown" are probably background processes, but this can be confirmed with the following query:
- ```
  select info.osuser, info.client_version, info.client_driver,
  sess.username, sess.schemaname, sess.process, sess.machine,
  sess.program from GV$SESSION_CONNECT_INFO info inner join gv$session
  sess on info.sid = sess.sid and info.serial# = sess.serial# and
  info.INST_ID = sess.inst_id where info.client_version='Unknown';
  ```

- ```
  select info.osuser, info.client_version, info.client_driver,
  sess.username, sess.schemaname, sess.process, sess.machine,
  sess.program from GV$SESSION_CONNECT_INFO info inner join gv$session
  sess on info.sid = sess.sid and info.serial# = sess.serial# and
  info.INST_ID = sess.inst_id;
  ```

High Level Steps

4. Confirm that no files exist outside of the disk groups.

› SQL> show parameter spfile

(The spfile should be in the primary (DG1) diskgroup)

```
› select distinct regexp_substr(name,'+.*\/') from v$datafile;
› select distinct regexp_substr(member,'+.*\/') from v$logfile;
› select distinct regexp_substr(name,'+.*\/') from v$tempfile;
```

(No datafiles, redo logs, or tempfiles should exist outside of the DG1 and DG_FLASH/ARC diskgroups).

```
› create pfile='/tmp/pfile_<db name>.ora' from spfile;
```

grep _DG /tmp/pfile_<db name>.ora

High Level Steps

5. All diskgroups must be updated to 19c compatibility for the upgrade to proceed.

› Update all of the diskgroups in ASM to an ASM compatibility of 18.0.0.0.0.

$ORACLE_HOME/bin/sqlplus / as sysasm

SQL> select name,compatibility from v$asm_diskgroup;

NAME	COMPATIBILITY
ARC_DG	11.2.0.2.0
DATA_DG	11.2.0.2.0
GRID_DG	11.2.0.2.0

alter diskgroup <diskgroup_name> set attribute 'compatible.asm' = '18.0.0.0.0';

The command below can be used to generate the update statement for all diskgroups:

select 'alter diskgroup '|| name ||' set attribute '||''''||'compatible.asm'||''''||'='||''''||'18.0.0.0.0'';' from v$asm_diskgroup where COMPATIBILITY not like '18.%' and COMPATIBILITY not like '19.%';

High Level Steps

6. Move the ASM password file to GRID_DG.

```
export ORACLE_HOME=/u01/grid/oracle/product/19/grid
export ORACLE_SID=+ASM1
$ORACLE_HOME/bin/asmcmd
ASMCMD> pwget --asm                    ---to check the location of password file

ASMCMD>
pwcopy --asm -f /u01/grid/oracle/product/19/grid/dbs/orapw+ASM +GRID_DG/orapwasm
Exit

mv /u01/grid/oracle/product/19/grid/dbs/orapw+ASM/u01/grid/oracle/product/19/grid/dbs/orapw+ASM_ORIG
Exit
```

High Level Steps

7. Run the steps below to proactively fix issues that may be flagged in the next step.

- `export ORACLE_SID=<non-cdb SID1>`
- `export ORACLE_HOME=<old DB home>`
- `$ORACLE_HOME/bin/sqlplus / as sysdba`
-
- `SQL> purge dba_recyclebin;`

- `SQL> @$ORACLE_HOME/rdbms/admin/utlrp.sql`

High Level Steps

Note: For 12.2 to 19c upgrade `preupgrade_fixups.sql script need to be run for both container & pdb database`

- **Review the files generated from the last step. Resolve any issues before moving on.**

- Preupgrade generated files:
- <stage directory>/preupgrade-utility-logs/preupgrade.log
- <stage directory>/preupgrade-utility-logs/preupgrade_fixups.sql
- <stage directory>/preupgrade-utility-logs/postupgrade_fixups.sql

- **Run preupgrade_fixups.sql**

```
SQL> @<stage directory>/preupgrade-utility-logs/preupgrade_fixups.sql
```

High Level Steps

Check for invalid objects and save a list to the staging directory for post-upgrade review.

```
$ORACLE_HOME/bin/sqlplus / as sysdba
set serveroutput on
spool <stage directory>/<non-cdb SID1>_invalid_objects_preupgrade.log;
set lines 132
set pages 100
column owner format a30;
column object_name format a60;
select owner, object_name, object_type from dba_objects where status != 'VALID' ORDER BY 1,2,3;
```

If any invalid objects are owned by SYS or SYSTEM, they must be reviewed before proceeding with the upgrade.

High Level Steps

Check for stale statistics

EXECUTE DBMS_STATS.GATHER_table_STATS(OWNNAME => 'V500', TABNAME => 'PHONE',estimate_percent => dbms_stats.auto_sample_size);

EXEC DBMS_STATS.publish_pending_stats('V500','PHONE')

Create a copy of the spfile and make a copy of the database memory settings.

- `$ORACLE_HOME/bin/sqlplus / as sysdba`
- `create pfile='/home/oracle/pfiles/pfile_<non-cdb name>_mmddyyyy.ora' from spfile;`
- `show parameter shared_pool_size;`

--NOTE: if this is below 3GB, it must be increased before dbua is run.

- `show parameter db_cache_size;`

Add these values:

- `alter system set shared_pool_size=<value from above> scope=spfile sid='*';`
- `alter system set db_cache_size=<value from above> scope=spfile sid='*';`
- `alter system set db_domain='world' scope=spfile sid='*';`
- `alter system set open_cursors=2000 scope=spfile sid='*';`
- `alter system set use_large_pages='only' scope=spfile sid='*';`
- `alter system set "_rollback_segment_count"=<match parameter from non-cdb name> container=all scope=spfile sid='*';`
- `alter system set undo_retention=86400 container=all scope=spfile sid='*';`
- `alter system set job_queue_processes=100 scope=spfile sid='*';`

High Level Steps

At least 200GB of storage is required for flashback logs and archived redo logs during the upgrade.

DBUA will create a Guaranteed Restore Point prior to upgrading the database. The flash diskgroup needs to have space for this. If there is not enough space, the upgrade may error out or it may hang.

$ORACLE_HOME/bin/sqlplus / as sysdba

Ensure archive log mode is enabled.

archive log list;

show parameter recover

alter system set DB_RECOVERY_FILE_DEST_SIZE=200G scope=both sid='*';

alter system set DB_RECOVERY_FILE_DEST='+<non-cdb name>_DG_ARC' scope=both sid='*';

High Level Steps

Configure BCT in the container database

CDB:

export ORACLE_HOME=/u01/oracle/product/19/db

export ORACLE_SID=<cdb_sid1>

$ORACLE_HOME/bin/sqlplus / as sysdba

select * from   v$block_change_tracking;

show parameter bct

Update container database BCT parameters if they do not meet the recommendations from

- $ORACLE_HOME/bin/sqlplus / as sysdba

- alter system set "_bct_public_dba_buffer_maxsize"=<value> scope=spfile sid='*';
- alter system set "_bct_buffer_allocation_max"=<value> scope=spfile sid='*';
- alter system set "_bct_public_dba_buffer_size"=<value> scope=spfile sid='*';
- alter system set large_pool_size=<value> scope=spfile sid='*';

- ALTER DATABASE ENABLE BLOCK CHANGE TRACKING USING FILE '+<non-cdb name>_DG1';

High Level Steps

All Initialization Parameters that need to be set.

Configurational Parameters..txt

High Level Steps

Stage or update the tnsnames.ora file in $ORACLE_HOME/network/admin on each host where a database instance will be upgraded.

https://asrblogger.com/connect-string/

- `cd $ORACLE_HOME/network/admin/`
- `cp -p tnsnames.ora /u01/oracle/product/19/db/network/admin/tnsnames.ora_11204`
- `cp -p sqlnet.ora /u01/oracle/product/19/db/network/admin/sqlnet.ora_11204`
- `cd /u01/oracle/product/19/db/network/admin`
- `cp tnsnames.ora_11204 newtnsnames.ora`

High Level Steps

```
alter system set o7_dictionary_accessibility= FALSE scope = spfile
sid='*';
```

Disable all audit statements during the upgrade using the commands below:

```
select 'noaudit '||audit_option||';' as "Disable Auditing Commands" from dba_stmt_audit_opts;
```

Purge the recycle bin a final time:

```
› purge dba_recyclebin;
› alter system reset sec_case_sensitive_logon;
› alter system reset service_names scope=spfile sid='*';
> alter system set use_large_pages ='TRUE' scope=spfile sid='*';
```

Downtime Event Steps..

High Level Steps

Ask Application team to bring down the domain.

Ask Golden Gate & other services team to bring down the services.

Java version
Java version should be **8** or later, which is available by default in Oracle Database homes from release 12.1.0.2 and latest.

High Level Steps

In some cases, the plug-in phase has encountered ORA-4031 errors against shared pool. To prevent this, increase shared pool size to 10G

Login to the database and disable BCT.

```
alter database disable block change tracking;
```

- Stop and start all instances of the database to load changed parameters and the updated getenv setting above.

```
export ORACLE_HOME=/u01/oracle/product/19/db
$ORACLE_HOME/bin/srvctl stop database -d <cdb_name>
$ORACLE_HOME/bin/srvctl start database -d <cdb_name>
```

- #Review the alert log on all nodes for any errors during restart.

```
# alert
```

High Level Steps

Download the latest file of **autoupgrade.jar**

- The **autoupgrade.jar** file exists by default, In Oracle Database 19c (19.3) and the later target Oracle homes.
- The autoupgrade. jar file is available in the Oracle 19c database software in the $ORACLE_HOME/rdbms/admin directory.
- Before you use AutoUpgrade, Oracle recommends that you download the latest Autoupgrade version available from

MOS Note: *2485457.1 - AutoUpgrade Tool.*

High Level Steps

Create the config file

- **Create a directory** to hold all upgrade config and log files.
- **Create the sample config file**
- **Modify the config file:** Copy the sample config file and make the necessary changes as per the database environment.

[oracle@11g upgrade_to_19c]$

java -jar /u02/app/oracle/product/19.0.0/dbhome_1/rdbms/admin/autoupgrade.jar -create_sample_file config

Created sample configuration file /u02/upgrade_to_19c/sample_config.cfg

[oracle@11g upgrade_to_19c]$ ls -lrt

total 4

-rw-r--r-- 1 oracle oinstall 1943 Dec 2 14:06 sample_config.cfg

High Level Steps

Analyze the database

Autoupgrade Analyze mode checks your database to see if it is ready for the upgrade. This will read data from the database and does not perform any updates.

Execute autoupgrade in analyze mode with the below syntax,

```
export ORACLE_HOME=/u01/app/oracle/product/19.3.0/dbhome_1

export PATH=$PATH:$ORACLE_HOME/jdk/bin

cd /u01/19c-autoupg

$ORACLE_HOME/jdk/bin/java -jar $ORACLE_HOME/rdbms/admin/autoupgrade.jar -config cdbdev_db_config.cfg -mode ANALYZE
```

Commands:

› We can monitor, manage and control the jobs from the **autoupgrade console**.

For example,

lsj – to list the jobs
status – to show the job status
tasks – shows the tasks executing

High Level Steps

Deploy the upgrade

Auto upgrade Deploy mode **performs the actual upgrade of the database from pre-upgrade source database analysis to post-upgrade checks.**

Note: Before deploying the upgrade, you must have a backup plan in place.

Execute the auto-upgrade in DEPLOY mode using the below syntax,

```
export ORACLE_HOME=/u01/app/oracle/product/19.3.0/dbhome_1export
PATH=$PATH:$ORACLE_HOME/jdk/bincd /u01/19c-autoupg$ORACLE_HOME/jdk/bin/java -jar
$ORACLE_HOME/rdbms/admin/autoupgrade.jar -config cdbdev_db_config.cfg -mode
DEPLOY
```

High Level Steps

Check the Timezone version

```
SQL> SELECT VERSION FROM V$TIMEZONE_FILE;
  VERSION
  32
```

Check the DB details

```
SQL> select name, open_mode, version, status from v$database, v$instance;

NAME      OPEN_MODE          VERSION        STATUS
CDBDEV    READ WRITE         19.0.0.0.0     OPEN
```

Post Downtime Event Steps..

High Level Steps

12c to 19c Only - Drop Guaranteed Restore Point

Once the upgrade is successful and all testing is done, drop the restore point.

SQL> select name from v$restore_point;

NAME

AUTOUPGRADE_28373

SQL> drop restore point AUTOUPGRADE_28373;

Restore point dropped.

What Is Oracle Database Compatibility?

Databases from different releases of Oracle Database software are compatible if they support the same features and those features perform the same way. When you upgrade to a new release of Oracle Database, certain new features might make your database incompatible with your earlier release.

Your upgraded database becomes incompatible with your earlier release under the following conditions:

A new feature stores any data on disk (including data dictionary changes) that cannot be processed with your earlier release.

An existing feature behaves differently in the new environment as compared to the old environment.

Oracle Database Release	Default Value	Minimum Value	Maximum Value
Oracle Database 12*c*	`12.0.0`	`11.0.0`	`12.1.0`
Oracle Database 11*g* Release 2 (11.2)	`11.2.0`	`10.0.0`	`11.2.0`
Oracle Database 11*g* Release 1 (11.1)	`11.0.0`	`10.0.0`	`11.1.0`
Oracle Database 10*g* Release 2 (10.2)	`10.2.0`	`9.2.0.`	`10.2.0`

High Level Steps

Change the compatible parameter

Note: After the upgrade, the database has to be tested properly before updating the compatible parameter. Once the parameter is updated database cannot be downgraded.

```
show parameter compatible

alter system set compatible='19.0.0' scope=spfile;

shutdown immediate;

startup;
```

High Level Steps

Compare the pre- and post- lists to see what new objects are now invalid. Resolve any new objects

$ORACLE_HOME/bin/sqlplus / as sysdba

Run utlrp.sql to recompile any invalid objects:

@$ORACLE_HOME/rdbms/admin/utlrp.sql

```
set serveroutput on
spool <stage directory>/<non-cdb name>_invalid_objects_postupgrade.log;
set lines 132
set pages 100
column owner format a10;
column object_name format a60;
column object_type format a15;
select owner, object_name, object_type from dba_objects where status != 'VALID' ORDER BY 1,2,3;
```

High Level Steps

Run Datapatch in case there are patches that need to be applied.

export ORACLE_HOME=/u01/oracle/product/19/db

export ORACLE_SID=<cdb_sid1>

$ORACLE_HOME/OPatch/datapatch -verbose

High Level Steps

Gather Fixed Object Statistics

$ORACLE_HOME/bin/sqlplus '/ as sysdba'

SQL> exec sys.dbms_stats.delete_system_stats();

SQL> exec sys.dbms_stats.gather_system_stats(gathering_mode=>'NOWORKLOAD');

To gather dictionary statistics for all PDBs in a container database, use the following syntax:

$ORACLE_HOME/perl/bin/perl $ORACLE_HOME/rdbms/admin/catcon.pl -l /tmp -b gatherstats -- --x"exec dbms_stats.gather_dictionary_stats"

Fixed Object statistics are gathered using the command below

$ORACLE_HOME/perl/bin/perl $ORACLE_HOME/rdbms/admin/catcon.pl -l /tmp -b gatherfixedstats -- --x"exec dbms_stats.gather_fixed_objects_stats"

High Level Steps

Remove messages If alerts were added to /etc/motd on database hosts, they can be removed now if appropriate.

Upgradation

Completed