Upgrade Internals

Make IT, May 2026

Photo by Robert Katzki on Unsplash

Oracle

DBAs

run the world

Daniel Overby Hansen

Distinguished Product Manager

dohdatabase

@dohdatabase.com

https://dohdatabase.com

The Basics

Oracle home

Binaries

Database

Data dictionary

Binaries and data dictionary release must match

Oracle Database 19c

Data dictionary 19c

Oracle AI Database 26ai

Data dictionary 26ai

Database Upgrade Guide

3 Upgrading to the New Oracle Database 10*g* Release

This chapter guides you through the process of upgrading a database to the new Oracle Database 10*g* release. This chapter covers the following topics:

- System Considerations and Requirements
- Install the Release 10.2 Oracle Software
- Install the Latest Available Patch Set Release and Any Required Patches

9. Set the system to spool results to a log file for later verification of success:

```
SQL> SPOOL upgrade.log
```

10. Run `catupgrd.sql`:

```
SQL> @catupgrd.sql
```

The `catupgrd.sql` script determines which upgrade scripts need to be run and then runs each necessary script. You must run the script in the new release 10.2 environment.

The upgrade script creates and alters certain data dictionary tables. It also upgrades or installs the following database components in the new release 10.2 database:

- Oracle Database Catalog Views
- Oracle Database Packages and Types
- JServer JAVA Virtual Machine

dbua

dbupgrade

catctl

@catupgrd.sql

@@catupstr.sql
@@catalog.sql
@@cdstrt.sql
@@cdfixed.sql
...
@catshutdown.sql

~~dbua~~

~~dbupgrade~~

```
java -jar autoupgrade.jar -config UPGR26.cfg -mode deploy
```

↳ catctl

↳ @catupgrd.sql

↳ @@catupstr.sql
@@catalog.sql
@@cdstrt.sql
@@cdfixed.sql
...
@catshutdown.sql

`catupgrd.sql` Overview

Initialize

- i-script / c-script

RDBMS upgrade

- catalog/catproc
- a-script

Components

- javavm,dv,ols,sdo,...

Finalizing

- bootstrap upgrade

```
$ ll $ORACLE_HOME/rdbms/admin

-rw-r-----. 1 oracle oinstall 46693 Feb 10 2025   a19.sql
-rw-r-----. 1 oracle oinstall 11261 Nov 10 2022   a20.sql
-rw-r-----. 1 oracle oinstall 113582 Jul 4 06:47  a21.sql

-rw-r-----. 1 oracle oinstall 106648 Feb 27 2025  c19.sql
-rw-r-----. 1 oracle oinstall 70597 Oct 18 2024   c20.sql
-rw-r-----. 1 oracle oinstall 406297 Jul 10 06:42 c21.sql

-rw-r-----. 1 oracle oinstall 3933 Jul 20 2020    i19.sql
-rw-r-----. 1 oracle oinstall 3712 Nov 23 2022    i20.sql
-rw-r-----. 1 oracle oinstall 13877 Oct 9 2024    i21.sql
```

```
$ ll $ORACLE_HOME/rdbms/admin

-rw-r-----. 1 oracle oinstall 46693 Feb 10 2025   a19.sql
-rw-r-----. 1 oracle oinstall 11261 Nov 10 2022   a20.sql
-rw-r-----. 1 oracle oinstall 113582 Jul 4 06:47  a21.sql

-rw-r-----. 1 oracle oinstall 106648 Feb 27 2025  c19.sql
-rw-r-----. 1 oracle oinstall 70597 Oct 18 2024   c20.sql
-rw-r-----. 1 oracle oinstall 406297 Jul 10 06:42 c21.sql

-rw-r-----. 1 oracle oinstall 3933 Jul 20 2020    i19.sql
-rw-r-----. 1 oracle oinstall 3712 Nov 23 2022    i20.sql
-rw-r-----. 1 oracle oinstall 13877 Oct 9 2024    i21.sql
```

```
$ vi i19.sql

(output truncated)

Rem ========================================================================
Rem Bug 28618619: Add new col to lob$ and lobfrag$, with values
Rem               4000 for enable and 0 for disable storage in row
Rem ========================================================================

alter table lob$ add maxinl number;
update lob$ set maxinl =
               case when bitand(property, 2) != 0 then 4000 else 0 end;
commit;
alter table lob$ modify maxinl not null;

alter table lobfrag$ add maxinl number;
update lobfrag$ set maxinl =
                  case when bitand(fragpro, 2) != 0 then 4000 else 0 end;
commit;
alter table lobfrag$ modify maxinl not null;
```


Upgrading from Oracle Database 19c to Oracle AI Database 26ai:

- 20,000 `CREATE OR REPLACE`
- 7,000 `CREATE ...`
- 1,500 `ALTER TABLE`
- 300 `ALTER TYPE`
- 7,000 `ALTER SESSION`
- … and many more statements

Parallel Upgrade

```
$ vi catupgrd.sql
```

```
Rem
Rem Initial checks and RDBMS upgrade scripts
Rem
@@catupstr.sql

Rem
Rem Execute upgrade and catalog session script
Rem
@@catupses.sql    --CATFILE -SES
@@catalogses.sql  --CATFILE -SESS
@@catalogses.sql  --CATFILE -SESS

Rem
Rem Run catalog with some multiprocess phases
Rem
@@catalog.sql     --CATFILE -X

Rem
Rem Execute catproc session script
Rem
@@catprocses.sql  --CATFILE -SESE -SESS

Rem
Rem Run catproc with some multiprocess phases
Rem
@@catproc.sql     --CATFILE -X
```

```
Rem
Rem Initial checks and RDBMS upgrade scripts
Rem
@@catupstr.sql

Rem
Rem Execute upgrade and catalog session script
Rem
@@catupses.sql    --CATFILE -SES
@@catalogses.sql  --CATFILE -SESS
@@catalogses.sql  --CATFILE -SESS

Rem
Rem Run catalog with some multiprocess phases
Rem
@@catalog.sql     --CATFILE -X

Rem
Rem Execute catproc session script
Rem
@@catprocses.sql  --CATFILE -SESE -SESS

Rem
Rem Run catproc with some multiprocess phases
Rem
@@catproc.sql     --CATFILE -X
```

```
Rem
Rem Initial checks and RDBMS upgrade scripts
Rem
@@catupstr.sql

Rem
Rem Execute upgrade and catalog session script
Rem
@@catupses.sql    --CATFILE -SES
@@catalogses.sql  --CATFILE -SESS
@@catalogses.sql  --CATFILE -SESS

Rem
Rem Run catalog with some multiprocess phases
Rem
@@catalog.sql     --CATFILE -X

Rem
Rem Execute catproc session script
Rem
@@catprocses.sql  --CATFILE -SESE -SESS

Rem
Rem Run catproc with some multiprocess phases
Rem
@@catproc.sql     --CATFILE -X
```

```
Rem
Rem Initial checks and RDBMS upgrade scripts
Rem
@@catupstr.sql

Rem
Rem Execute upgrade and catalog session script
Rem
@@catupses.sql    --CATFILE -SES
@@catalogses.sql  --CATFILE -SESS
@@catalogses.sql  --CATFILE -SESS

Rem
Rem Run catalog with some multiprocess phases
Rem
@@catalog.sql     --CATFILE -X

Rem
Rem Execute catproc session script
Rem
@@catprocses.sql  --CATFILE -SESE -SESS

Rem
Rem Run catproc with some multiprocess phases
Rem
@@catproc.sql     --CATFILE -X
```

```
--CATCTL -S -D "Catalog Core SQL"    Initial scripts single process
@@cdstrt.sql
@@cdfixed.sql
@@catcdbviews.sql
@@catblock.sql
@@cdcore.sql

--CATCTL -R
--CATCTL -M -D "Catalog Tables and Views"
@@cdplsql.sql
@@cdsqlddl.sql
@@cdmanage.sql
@@cdtxnspc.sql
@@cdenv.sql
@@cdrac.sql
@@cdsec.sql
@@cdobj.sql
@@cdjava.sql
@@cdpart.sql
@@cdrep.sql
@@cdaw.sql
@@cdsummgt.sql
@@cdtools.sql
@@cdexttab.sql
@@cddm.sql
@@cdclst.sql
```

```
--CATCTL -S -D "Catalog Core SQL"    Initial scripts single process
@@cdstrt.sql
@@cdfixed.sql
@@catcdbviews.sql
@@catblock.sql
@@cdcore.sql

--CATCTL -R
--CATCTL -M -D "Catalog Tables and Views"
@@cdplsql.sql
@@cdsqlddl.sql
@@cdmanage.sql
@@cdtxnspc.sql
@@cdenv.sql
@@cdrac.sql
@@cdsec.sql
@@cdobj.sql
@@cdjava.sql
@@cdpart.sql
@@cdrep.sql
@@cdaw.sql
@@cdsummgt.sql
@@cdtools.sql
@@cdexttab.sql
@@cddm.sql
@@cdclst.sql
```

```
--CATCTL -S -D "Catalog Core SQL"    Initial scripts single process
@@cdstrt.sql
@@cdfixed.sql
@@catcdbviews.sql
@@catblock.sql
@@cdcore.sql

--CATCTL -R
--CATCTL -M -D "Catalog Tables and Views"
@@cdplsql.sql
@@cdsqlddl.sql
@@cdmanage.sql
@@cdtxnspc.sql
@@cdenv.sql
@@cdrac.sql
@@cdsec.sql
@@cdobj.sql
@@cdjava.sql
@@cdpart.sql
@@cdrep.sql
@@cdaw.sql
@@cdsummgt.sql
@@cdtools.sql
@@cdexttab.sql
@@cddm.sql
@@cdclst.sql
```

```
$ grep "jloaiza" catalog.sql

Rem     jloaiza    06/20/94 -  fix all_tables
Rem     jloaiza    06/16/94 -  add disable dml locks
Rem     jloaiza    05/23/94 -  add new fixed views
Rem     jloaiza    10/28/92 -  add v$db_object_cache and v$open_cursor
```



Juan Loaiza
EVP, Database Technologies, Oracle

```
Rem
Rem Run catproc with some multiprocess phases
Rem
@@catproc.sql     --CATFILE -X

--CATCTL -R
--CATCTL -S -D "Final RDBMS scripts"
Rem
Rem Final RDBMS upgrade scripts
Rem
@@catupprc.sql

Rem
Rem Upgrade components with some multiprocess phases
Rem
@@cmpupgrd.sql    --CATFILE -X -SESE

--CATCTL -S -D "Final Upgrade scripts"
Rem
Rem Final upgrade scripts
Rem
@@catupend.sql
```

```
--CATCTL -M  -D "Upgrading Java and non-Java"
Rem
Rem Java upgrade (JAVAVM, XML, CATJAVA)
Rem
@@cmpupjav.sql

Rem
Rem non-Java dependent upgrades (include XDB and Dependent since all
Rem     single process while Java components being upgraded in other single
Rem     process):
Rem     APS, AMD, OLS, DV, CONTEXT, XDB, OWM, MGW, RAC
Rem
@@cmpupnjv.sql

--CATCTL -R
--CATCTL -S
Rem
Rem Extracted grants/revokes of bootstrap tables for component scripts
Rem These are run serially as to not run into timeout/deadlocking issues
Rem
@@cmpupprv.sql

--CATCTL -R -D "Upgrading XDB"
Rem
Rem Oracle XDB (uses catctl directly)
Rem
--CATCTL -CP XDB -X
```

```
--CATCTL -M  -D "Upgrading Java and non-Java"
Rem
Rem Java upgrade (JAVAVM, XML, CATJAVA)
Rem
@@cmpupjav.sql

Rem
Rem non-Java dependent upgrades (include XDB and Dependent since all
Rem     single process while Java components being upgraded in other single
Rem     process):
Rem     APS, AMD, OLS, DV, CONTEXT, XDB, OWM, MGW, RAC
Rem
@@cmpupnjv.sql

--CATCTL -R
--CATCTL -S
Rem
Rem Extracted grants/revokes of bootstrap tables for component scripts
Rem These are run serially as to not run into timeout/deadlocking issues
Rem
@@cmpupprv.sql

--CATCTL -R -D "Upgrading XDB"
Rem
Rem Oracle XDB (uses catctl directly)
Rem
--CATCTL -CP XDB -X
```

```
$ pwd
/u02/app/oracle/autoupgrade/runs/LONGHORN1/LONGHORN1/101/dbupgrade

$ vi autoupgrade*cdbroot.log
```

```
------------------------------------------------------
Phases [0-106]         Start Time:[2025_08_22 18:00:46]
Container Lists Inclusion:[CDB$ROOT] Exclusion:[NONE]
------------------------------------------------------
***********   Executing Change Scripts   ***********
Serial   Phase #:0    [CDB$ROOT] Files:1    Time: 51s
***************   Catalog Core SQL   ***************
Serial   Phase #:1    [CDB$ROOT] Files:5    Time: 40s
Restart  Phase #:2    [CDB$ROOT] Files:1    Time: 0s
***********   Catalog Tables and Views   ***********
Parallel Phase #:3    [CDB$ROOT] Files:21   Time: 18s
Restart  Phase #:4    [CDB$ROOT] Files:1    Time: 0s
*************   Catalog Final Scripts   ************
Serial   Phase #:5    [CDB$ROOT] Files:7    Time: 12s
*****************   Catproc Start   *****************
Serial   Phase #:6    [CDB$ROOT] Files:1    Time: 9s
*****************   Catproc Types   *****************
Serial   Phase #:7    [CDB$ROOT] Files:2    Time: 9s
Restart  Phase #:8    [CDB$ROOT] Files:1    Time: 0s

(output truncated)

***   End PDB Application Upgrade Post-Shutdown   **
Serial   Phase #:104  [CDB$ROOT] Files:2    Time: 101s
Serial   Phase #:105  [CDB$ROOT] Files:1    Time: 0s
Serial   Phase #:106  [CDB$ROOT] Files:1     Time: 24s
Serial   Phase #:106  [CDB$ROOT] Files:1     Time: 24s
```

```
Phases [0-106]         Start Time:[2025_08_22 18:00:46]
Container Lists Inclusion:[CDB$ROOT] Exclusion:[NONE]
**********   Executing Change Scripts   **********
Serial   Phase #:0    [CDB$ROOT] Files:1     Time: 51s
***************   Catalog Core SQL   ***************
Serial   Phase #:1    [CDB$ROOT] Files:5     Time: 40s
Restart  Phase #:2    [CDB$ROOT] Files:1     Time: 0s
**********   Catalog Tables and Views   **********
Parallel Phase #:3    [CDB$ROOT] Files:21
Restart  Phase #:4    [CDB$ROOT] Files:1
*************   Catalog Final Scripts   ****
Serial   Phase #:5    [CDB$ROOT] Files:7
*****************   Catproc Start   ********
Serial   Phase #:6    [CDB$ROOT] Files:1
*****************   Catproc Types   ********
Serial   Phase #:7    [CDB$ROOT] Files:2
Restart  Phase #:8    [CDB$ROOT] Files:1

(output truncated)

***   End PDB Application Upgrade Post-Shutd
Serial   Phase #:104  [CDB$ROOT] Files:2
Serial   Phase #:105  [CDB$ROOT] Files:1
Serial   Phase #:106  [CDB$ROOT] Files:1
Serial   Phase #:106  [CDB$ROOT] Files:1     Time: 24s
```

```
--CATCTL -S -D "Catalog Core SQL"    Initial scripts single process
@@cdstrt.sql
@@cdfixed.sql
@@catcdbviews.sql
@@catblock.sql
@@cdcore.sql

--CATCTL -R
--CATCTL -M -D "Catalog Tables and Views"
@@cdplsql.sql
@@cdsqlddl.sql
@@cdmanage.sql
@@cdtxnspc.sql
@@cdenv.sql
@@cdrac.sql
@@cdsec.sql
@@cdobj.sql
@@cdjava.sql
```

```
Phases [0-106]         Start Time:[2025_08_22 18:00:46]
Container Lists Inclusion:[CDB$ROOT] Exclusion:[NONE]
**********   Executing Change Scripts   **********
Serial   Phase #:0    [CDB$ROOT] Files:1     Time: 51s
***************   Catalog Core SQL   ***************
Serial   Phase #:1    [CDB$ROOT] Files:5     Time: 40s
Restart  Phase #:2    [CDB$ROOT] Files:1     Time: 0s
**********   Catalog Tables and Views   **********
Parallel Phase #:3    [CDB$ROOT] Files:21
Restart  Phase #:4    [CDB$ROOT] Files:1
*************   Catalog Final Scripts   ****
Serial   Phase #:5    [CDB$ROOT] Files:7
****************   Catproc Start   ********
Serial   Phase #:6    [CDB$ROOT] Files:1
****************   Catproc Types   ********
Serial   Phase #:7    [CDB$ROOT] Files:2
Restart  Phase #:8    [CDB$ROOT] Files:1

(output truncated)

***   End PDB Application Upgrade Post-Shutd
Serial   Phase #:104  [CDB$ROOT] Files:2
Serial   Phase #:105  [CDB$ROOT] Files:1
Serial   Phase #:106  [CDB$ROOT] Files:1
Serial   Phase #:106  [CDB$ROOT] Files:1     Time: 24s
```

```
--CATCTL -S -D "Catalog Core SQL"   Initial scripts single process
@@cdstrt.sql
@@cdfixed.sql
@@catcdbviews.sql
@@catblock.sql
@@cdcore.sql

--CATCTL -R
--CATCTL -M -D "Catalog Tables and Views"
@@cdplsql.sql
@@cdsqlddl.sql
@@cdmanage.sql
@@cdtxnspc.sql
@@cdenv.sql
@@cdrac.sql
@@cdsec.sql
@@cdobj.sql
@@cdjava.sql
```

```
----------------------------------------------------
Phases [0-106]         Start Time:[2025_08_22 18:00:46]
Container Lists Inclusion:[CDB$ROOT] Exclusion:[NONE]
----------------------------------------------------
**********   Executing Change Scripts   **********
Serial   Phase #:0    [CDB$ROOT] Files:1     Time: 51s
***************   Catalog Core SQL   ***************
Serial   Phase #:1    [CDB$ROOT] Files:5     Time: 40s
Restart  Phase #:2    [CDB$ROOT] Files:1     Time: 0s
**********   Catalog Tables and Views   **********
Parallel Phase #:3    [CDB$ROOT] Files:21
Restart  Phase #:4    [CDB$ROOT] Files:1
*************   Catalog Final Scripts   ****
Serial   Phase #:5    [CDB$ROOT] Files:7
*****************   Catproc Start   ********
Serial   Phase #:6    [CDB$ROOT] Files:1
*****************   Catproc Types   ********
Serial   Phase #:7    [CDB$ROOT] Files:2
Restart  Phase #:8    [CDB$ROOT] Files:1

(output truncated)

***   End PDB Application Upgrade Post-Shutd
Serial   Phase #:104  [CDB$ROOT] Files:2
Serial   Phase #:105  [CDB$ROOT] Files:1
Serial   Phase #:106  [CDB$ROOT] Files:1
Serial   Phase #:106  [CDB$ROOT] Files:1     Time: 24s
```

```
--CATCTL -S -D "Catalog Core SQL"    Initial scripts single process
@@cdstrt.sql
@@cdfixed.sql
@@catcdbviews.sql
@@catblock.sql
@@cdcore.sql

--CATCTL -R
--CATCTL -M -D "Catalog Tables and Views"
@@cdplsql.sql
@@cdsqlddl.sql
@@cdmanage.sql
@@cdtxnspc.sql
@@cdenv.sql
@@cdrac.sql
@@cdsec.sql
@@cdobj.sql
@@cdjava.sql
```

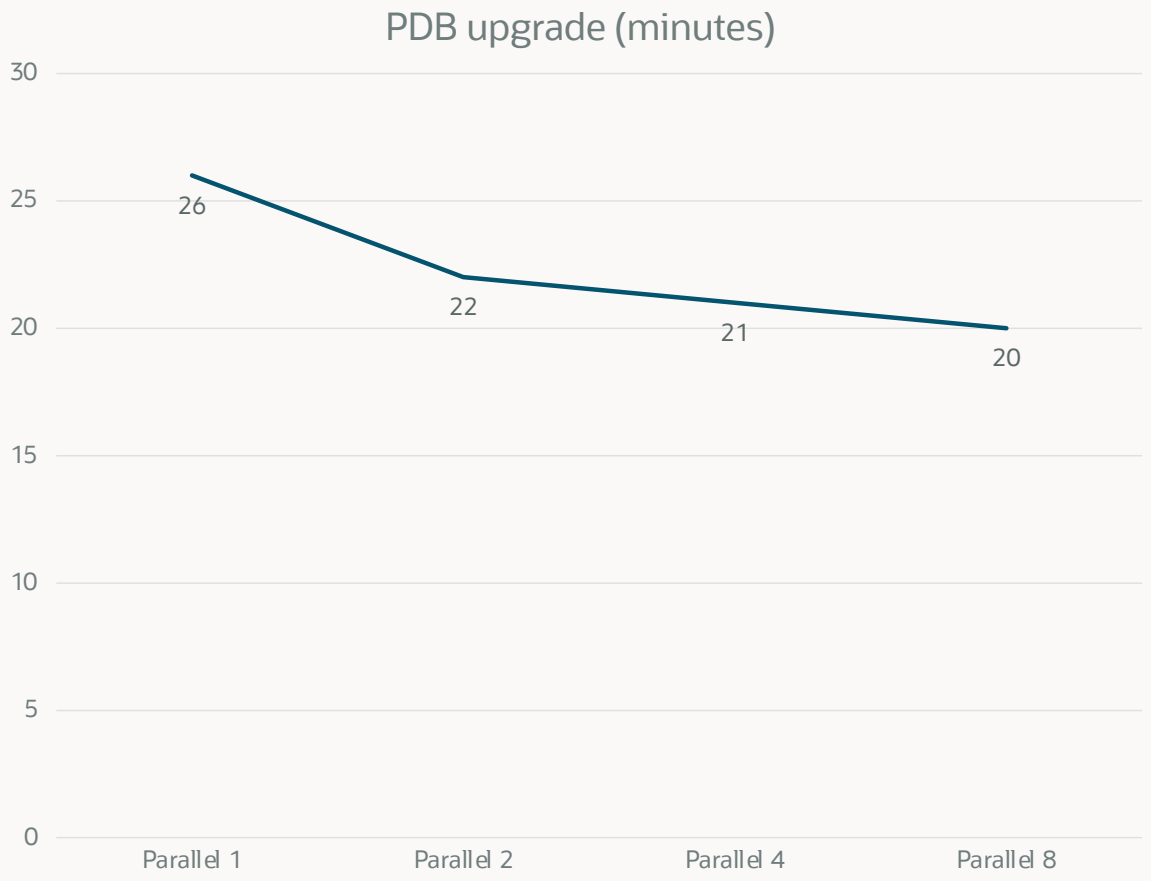
For parallel upgrade, AutoUpgrade

- Uses 8 threads for CDB$ROOT
- Uses 2 threads per PDB
- Upgrades CPU_COUNT / 2 PDBs concurrently

For parallel upgrade, AutoUpgrade

- Uses 8 threads for CDB$ROOT
- Uses 2 threads per PDB
  - Overridable with `upg1.catctl_options=-N <value>`
- Upgrades CPU_COUNT / 2 PDBs concurrently
  - Overridable with `upg1.catctl_options=-n <value>`

More threads leads to more contention



Scale by upgrading more PDBs concurrently

Upgrade Details

You must `startup upgrade` to upgrade a database

- Enables restricted session

```
ALTER SYSTEM SET "_SYSTEM_TRIG_ENABLED" = FALSE
                 "_UNDO_AUTOTUNE"       = FALSE
                 AQ_TM_PROCESSES        = 0
                 DB_SECUREFILE          = PERMITTED
                 ENABLE_DDL_LOGGING     = FALSE
                 PLSQL_WARNINGS         = 'DISABLE:ALL'
                 RECYCLEBIN             = OFF
                 RESOURCE_MANAGER_PLAN  = ''
                 UNDO_RETENTION         = 900
```

```
SQL> startup upgrade
ORACLE instance started.

Total System Global Area 1002435912 bytes
Fixed Size                  8947016 bytes
Variable Size             272629760 bytes
Database Buffers          713031680 bytes
Redo Buffers                7827456 bytes
Database mounted.
Database opened.

SQL> create table t1 (c1 number);

Table created.

SQL> create table t1 (c1 number);

Table created.

SQL> create table t1 (c1 number);

Table created.
```

A restart is the only way out of *upgrade* mode

AutoUpgrade recompiles invalid objects at the end

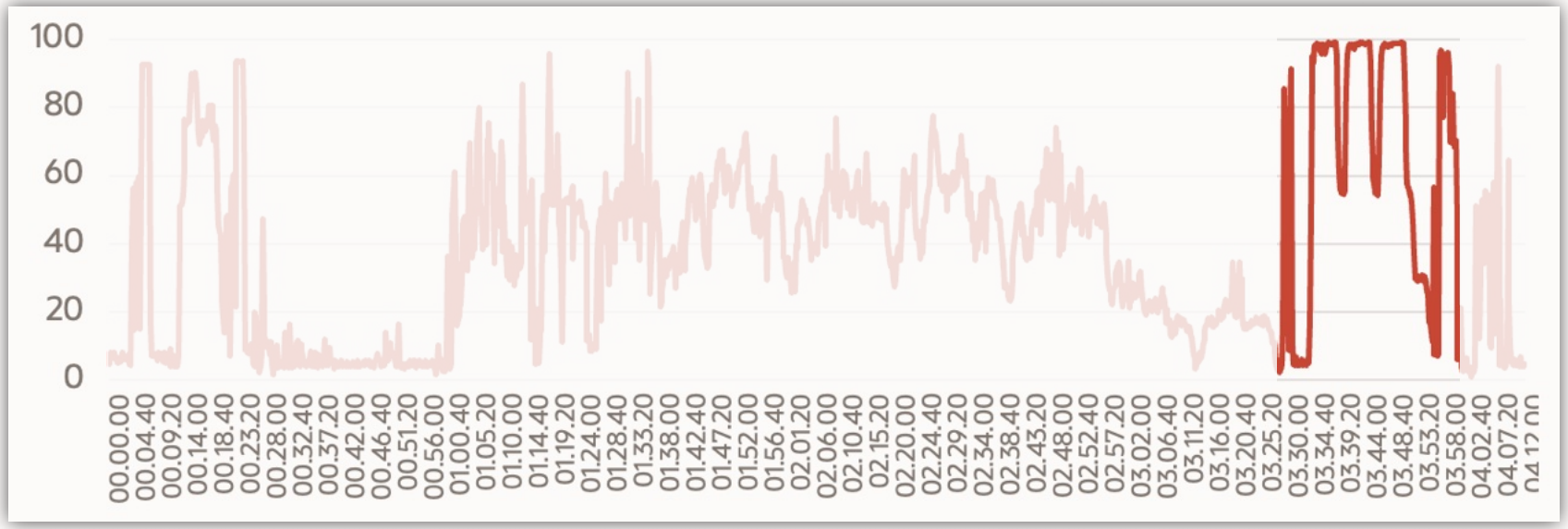
- Only those owned by Oracle-maintained schemas

```
@?/rdbms/admin/utlrp
```

~~@?/rdbms/admin/utlrp~~

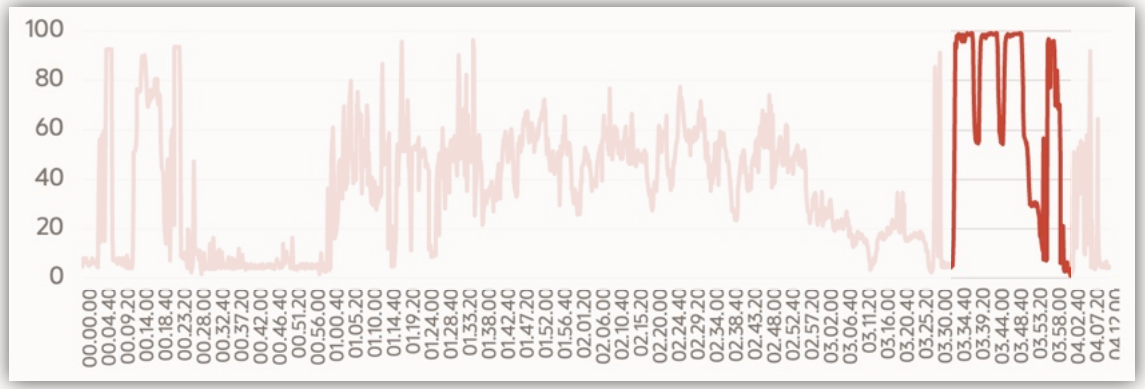
```
@?/rdbms/admin/utlprpom.sql <n>
```

Recompilation



Recompilation

- Uses the scheduler (`DBMS_SCHEDULER`) to parallelize
- Lowering `JOB_QUEUE_PROCESSES` might limit the recompilation
- Very CPU intensive



Bootstrapping

```
STARTUP MOUNT

ALTER SESSION SET EVENTS '10046 TRACE NAME CONTEXT FOREVER, LEVEL 12';

ALTER DATABASE OPEN;
```

```
grep -i "CREATE TABLE" FTEX_ora_328857.trc | cut -c1-100
```

```
create table bootstrap$ ( line#         number not null,   obj#          number not null,   sql_tex
CREATE TABLE TAB$("OBJ#" NUMBER NOT NULL,"DATAOBJ#" NUMBER,"TS#" NUMBER NOT NULL,"FILE#" NUMBER NOT
CREATE TABLE CLU$("OBJ#" NUMBER NOT NULL,"DATAOBJ#" NUMBER,"TS#" NUMBER NOT NULL,"FILE#" NUMBER NOT
CREATE TABLE FET$("TS#" NUMBER NOT NULL,"FILE#" NUMBER NOT NULL,"BLOCK#" NUMBER NOT NULL,"LENGTH" NU
CREATE TABLE UET$("SEGFILE#" NUMBER NOT NULL,"SEGBLOCK#" NUMBER NOT NULL,"EXT#" NUMBER NOT NULL,"TS#
CREATE TABLE SEG$("FILE#" NUMBER NOT NULL,"BLOCK#" NUMBER NOT NULL,"TYPE#" NUMBER NOT NULL,"TS#" NUM
CREATE TABLE UNDO$("US#" NUMBER NOT NULL,"NAME" VARCHAR2(30) NOT NULL,"USER#" NUMBER NOT NULL,"FILE#
CREATE TABLE TS$("TS#" NUMBER NOT NULL,"NAME" VARCHAR2(30) NOT NULL,"OWNER#" NUMBER NOT NULL,"ONLINE
CREATE TABLE FILE$("FILE#" NUMBER NOT NULL,"STATUS$" NUMBER NOT NULL,"BLOCKS" NUMBER NOT NULL,"TS#"
CREATE TABLE OBJ$("OBJ#" NUMBER NOT NULL,"DATAOBJ#" NUMBER,"OWNER#" NUMBER NOT NULL,"NAME" VARCHAR2(
CREATE TABLE IND$("OBJ#" NUMBER NOT NULL,"DATAOBJ#" NUMBER,"TS#" NUMBER NOT NULL,"FILE#" NUMBER NOT
CREATE TABLE ICOL$("OBJ#" NUMBER NOT NULL,"BO#" NUMBER NOT NULL,"COL#" NUMBER NOT NULL,"POS#" NUMBER
CREATE TABLE COL$("OBJ#" NUMBER NOT NULL,"COL#" NUMBER NOT NULL,"SEGCOL#" NUMBER NOT NULL,"SEGCOLLEN
CREATE TABLE USER$("USER#" NUMBER NOT NULL,"NAME" VARCHAR2(128) NOT NULL,"TYPE#" NUMBER NOT NULL,"PA
CREATE TABLE PROXY_DATA$("CLIENT#" NUMBER NOT NULL,"PROXY#" NUMBER NOT NULL,"CREDENTIAL_TYPE#" NUMBE
CREATE TABLE PROXY_ROLE_DATA$("CLIENT#" NUMBER NOT NULL,"PROXY#" NUMBER NOT NULL,"ROLE#" NUMBER NOT
CREATE TABLE CON$("OWNER#" NUMBER NOT NULL,"NAME" VARCHAR2(128) NOT NULL,"CON#" NUMBER NOT NULL,"SPA
CREATE TABLE CDEF$("CON#" NUMBER NOT NULL,"OBJ#" NUMBER NOT NULL,"COLS" NUMBER,"TYPE#" NUMBER NOT NU
CREATE TABLE CCOL$("CON#" NUMBER NOT NULL,"OBJ#" NUMBER NOT NULL,"COL#" NUMBER NOT NULL,"POS#" NUMBE
```

```
create table bootstrap$ ( line#         number not null,   obj#          number not null,   sql_tex
CREATE TABLE TAB$("OBJ#" NUMBER NOT NULL,"DATAOBJ#" NUMBER,"TS#" NUMBER NOT NULL,"FILE#" NUMBER NOT
CREATE TABLE CLU$("OBJ#" NUMBER NOT NULL,"DATAOBJ#" NUMBER,"TS#" NUMBER NOT NULL,"FILE#" NUMBER NOT
CREATE TABLE FET$("TS#" NUMBER NOT NULL,"FILE#" NUMBER NOT NULL,"BLOCK#" NUMBER NOT NULL,"LENGTH" NU
CREATE TABLE UET$("SEGFILE#" NUMBER NOT NULL,"SEGBLOCK#" NUMBER NOT NULL,"EXT#" NUMBER NOT NULL,"TS#
CREATE TABLE SEG$("FILE#" NUMBER NOT NULL,"BLOCK#" NUMBER NOT NULL,"TYPE#" NUMBER NOT NULL,"TS#" NUM
CREATE TABLE UNDO$("US#" NUMBER NOT NULL,"NAME" VARCHAR2(30) NOT NULL,"USER#" NUMBER NOT NULL,"FILE#
CREATE TABLE TS$("TS#" NUMBER NOT NULL,"NAME" VARCHAR2(30) NOT NULL,"OWNER#" NUMBER NOT NULL,"ONLINE
CREATE TABLE FILE$("FILE#" NUMBER NOT NULL,"STATUS$" NUMBER NOT NULL,"BLOCKS" NUMBER NOT NULL,"TS#"
CREATE TABLE OBJ$("OBJ#" NUMBER NOT NULL,"DATAOBJ#" NUMBER,"OWNER#" NUMBER NOT NULL,"NAME" VARCHAR2(
CREATE TABLE IND$("OBJ#" NUMBER NOT NULL,"DATAOBJ#" NUMBER,"TS#" NUMBER NOT NULL,"FILE#" NUMBER NOT
CREATE TABLE ICOL$("OBJ#" NUMBER NOT NULL,"BO#" NUMBER NOT NULL,"COL#" NUMBER NOT NULL,"POS#" NUMBER
CREATE TABLE COL$("OBJ#" NUMBER NOT NULL,"COL#" NUMBER NOT NULL,"SEGCOL#" NUMBER NOT NULL,"SEGCOLLEN
CREATE TABLE USER$("USER#" NUMBER NOT NULL,"NAME" VARCHAR2(128) NOT NULL,"TYPE#" NUMBER NOT NULL,"PA
CREATE TABLE PROXY_DATA$("CLIENT#" NUMBER NOT NULL,"PROXY#" NUMBER NOT NULL,"CREDENTIAL_TYPE#" NUMBE
CREATE TABLE PROXY_ROLE_DATA$("CLIENT#" NUMBER NOT NULL,"PROXY#" NUMBER NOT NULL,"ROLE#" NUMBER NOT
CREATE TABLE CON$("OWNER#" NUMBER NOT NULL,"NAME" VARCHAR2(128) NOT NULL,"CON#" NUMBER NOT NULL,"SPA
CREATE TABLE CDEF$("CON#" NUMBER NOT NULL,"OBJ#" NUMBER NOT NULL,"COLS" NUMBER,"TYPE#" NUMBER NOT NU
CREATE TABLE CCOL$("CON#" NUMBER NOT NULL,"OBJ#" NUMBER NOT NULL,"COL#" NUMBER NOT NULL,"POS#" NUMBE
```

Bootstrap tables are a core part of the data dictionary

- `OBJECT_ID` below 56

```
SQL> select obj#, sql_text from bootstrap$ order by 1;

   OBJ#                                                                                SQL_TEXT
_______ ________________________________________________________________________________________
      0 CREATE ROLLBACK SEGMENT SYSTEM STORAGE (  INITIAL 112K NEXT 56K MINEXTENTS 1 MAX ...
      2 CREATE CLUSTER C_OBJ#("OBJ#" NUMBER) PCTFREE 5 PCTUSED 40 INITRANS 2 MAXTRANS 25 ...
      3 CREATE INDEX I_OBJ# ON CLUSTER C_OBJ# PCTFREE 10 INITRANS 2 MAXTRANS 255 STORAGE ...
      4 CREATE TABLE TAB$("OBJ#" NUMBER NOT NULL,"DATAOBJ#" NUMBER,"TS#" NUMBER NOT NULL ...
      5 CREATE TABLE CLU$("OBJ#" NUMBER NOT NULL,"DATAOBJ#" NUMBER,"TS#" NUMBER NOT NULL ...
      6 CREATE CLUSTER C_TS#("TS#" NUMBER) PCTFREE 10 PCTUSED 40 INITRANS 2 MAXTRANS 255 ...
      7 CREATE INDEX I_TS# ON CLUSTER C_TS# PCTFREE 10 INITRANS 2 MAXTRANS 255 STORAGE ( ...
      8 CREATE CLUSTER C_FILE#_BLOCK#("TS#" NUMBER,"SEGFILE#" NUMBER,"SEGBLOCK#" NUMBER) ...
      9 CREATE INDEX I_FILE#_BLOCK# ON CLUSTER C_FILE#_BLOCK# PCTFREE 10 INITRANS 2 MAXT ...
     10 CREATE CLUSTER C_USER#("USER#" NUMBER) PCTFREE 10 PCTUSED 40 INITRANS 2 MAXTRANS ...
     11 CREATE INDEX I_USER# ON CLUSTER C_USER# PCTFREE 10 INITRANS 2 MAXTRANS 255 STORA ...
     12 CREATE TABLE FET$("TS#" NUMBER NOT NULL,"FILE#" NUMBER NOT NULL,"BLOCK#" NUMBER  ...
     13 CREATE TABLE UET$("SEGFILE#" NUMBER NOT NULL,"SEGBLOCK#" NUMBER NOT NULL,"EXT#"  ...
     14 CREATE TABLE SEG$("FILE#" NUMBER NOT NULL,"BLOCK#" NUMBER NOT NULL,"TYPE#" NUMBE ...
     15 CREATE TABLE UNDO$("US#" NUMBER NOT NULL,"NAME" VARCHAR2(30) NOT NULL,"USER#" NU ...
     16 CREATE TABLE TS$("TS#" NUMBER NOT NULL,"NAME" VARCHAR2(30) NOT NULL,"OWNER#" NUM ...
     17 CREATE TABLE FILE$("FILE#" NUMBER NOT NULL,"STATUS$" NUMBER NOT NULL,"BLOCKS" NU ...
```



```
CREATE CLUSTER C_OBJ#("OBJ#" NUMBER)
PCTFREE 5 PCTUSED 40 INITRANS 2 MAXTRANS 255
STORAGE (
      INITIAL 136K NEXT 200K
      MINEXTENTS 1 MAXEXTENTS 2147483645
      PCTINCREASE 0
      OBJNO 2
      EXTENTS (FILE 1 BLOCK 144)
      ) SIZE 800
```

```
SQL> alter table registry$sqlpatch add(daniel number);

Table altered.
```


```
SQL> alter table registry$sqlpatch add(daniel number);

Table altered.

SQL> alter table tab$ add(daniel number);
alter table tab$ add(daniel number)
*
ERROR at line 1:
ORA-00701: object necessary for warmstarting database cannot be altered
```

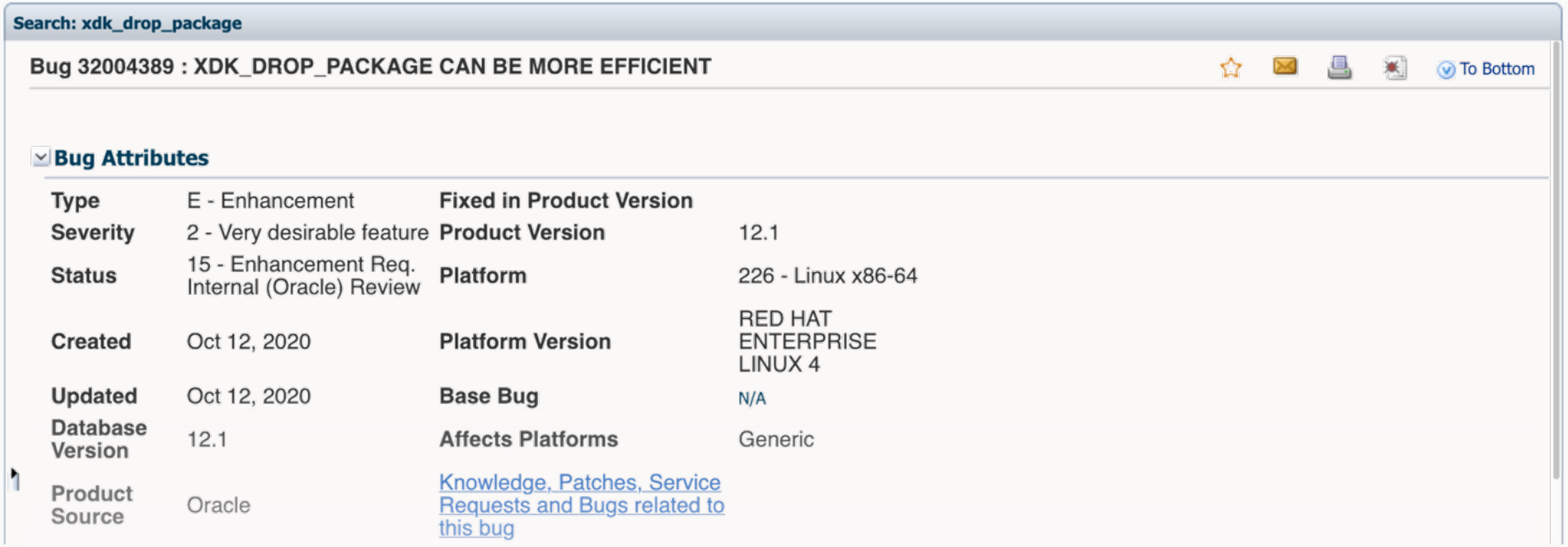
```
SQL> startup

ORA-01092: ORACLE instance terminated. Disconnection forced
ORA-00704: bootstrap process failure
ORA-39700: database must be opened with UPGRADE option
Process ID: 22211
Session ID: 1125 Serial number: 3
```

Diagnosing Upgrades

```
$ pwd
/u02/app/oracle/autoupgrade/runs/FIREBALL1/FIREBALL1/101

$ ll
total 23936
-rw-r----- 1 oracle oinstall 24279072 Aug 22 13:41 autoupgrade_20250822.log
-rw-r----- 1 oracle oinstall   110170 Aug 22 13:41 autoupgrade_20250822_user.log
-rw-r----- 1 oracle oinstall    20655 Aug 22 13:30 autoupgrade_err.log
drwxr-x--- 2 oracle oinstall    32768 Aug 22 13:04 dbupgrade
drwxr-x--- 2 oracle oinstall     4096 Aug 22 11:33 drain
drwxr-x--- 2 oracle oinstall     4096 Aug 22 13:04 postchecks
drwxr-x--- 2 oracle oinstall    12288 Aug 22 13:30 postfixups
drwxr-x--- 2 oracle oinstall     4096 Aug 22 13:31 postupgrade
drwxr-x--- 2 oracle oinstall     4096 Aug 22 11:29 prechecks
drwxr-x--- 2 oracle oinstall    12288 Aug 22 11:29 prefixups
drwxr-x--- 2 oracle oinstall     4096 Aug 22 11:20 preupgrade
drwxr-x--- 2 oracle oinstall     4096 Aug 22 13:41 sysupdates
drwxr-x--- 2 oracle oinstall     4096 Aug 22 13:04 upgstat
```

```
$ pwd
/u02/app/oracle/autoupgrade/runs/FIREBALL1/FIREBALL1/101/dbupgrade

$ grep -h "Grand Total" autoupgrade* | sort
Grand Total Time: 1456s
Grand Total Time: 1690s [PDB017]
Grand Total Time: 1691s [PDB007]
Grand Total Time: 1702s [PDB028]
Grand Total Time: 1703s [PDB019]
...
Grand Total Time: 2157s [PDB006]
Grand Total Time: 2161s [PDB008]
Grand Total Time: 2168s [PDB005]
Grand Total Time: 2180s [PDB003]
Grand Total Time: 2182s [PDB029]
```

```
$ grep -ih "Elapsed" catupgrd*pdb029*.log | sort
...
Elapsed: 00:01:06.01
Elapsed: 00:01:37.71
Elapsed: 00:01:47.52
Elapsed: 00:02:36.98
Elapsed: 00:09:59.70

$ grep -i "00:09:59.70" -B5 catupgrd*pdb0290.log
11:32:26 SQL>
11:32:26 SQL> EXECUTE xdk_drop_package;

PL/SQL procedure successfully completed.

Elapsed: 00:09:59.70
```

You can't use AWR while upgrading

- Or, can you?

```
SQL> select status from v$instance;

STATUS
------------
OPEN MIGRATE
```

```
SQL> select status from v$instance;

STATUS
------------
OPEN MIGRATE

SQL> exec dbms_workload_repository.create_snapshot();
BEGIN dbms_workload_repository.create_snapshot(); END;

*
ERROR at line 1:
ORA-13516: AWR Operation failed: CATPROC not valid
ORA-06512: at "SYS.DBMS_WORKLOAD_REPOSITORY", line 99
ORA-06512: at "SYS.DBMS_WORKLOAD_REPOSITORY", line 145
ORA-06512: at line 1
Help: https://docs.oracle.com/error-help/db/ora-13516/
```

```
--After upgrading CDB$ROOT and while PDBs are upgrading,
--you can use AWR in the root container

SQL> alter session set container=CDB$ROOT;

SQL> select status from v$instance;

STATUS
------------
OPEN

SQL> exec dbms_workload_repository.create_snapshot();

PL/SQL procedure successfully completed.
```

```
$ pwd
/u02/app/oracle/autoupgrade/runs/DB08221/DB08221/101/upgstat

$ ll
total 144
-rw-r----- 1 oracle oinstall 65575 Aug 24 08:20 fullpdbs.txt
-rw-r----- 1 oracle oinstall 49978 Aug 24 08:20 nonpdb_catproc.txt
-rw-r----- 1 oracle oinstall 21428 Aug 24 08:20 nonpdb_upg.txt
```


```
AUStats Database Upgrade Report

 Database Name  : DB0822
 Instance Name  : DB08221
 Upgrade from   : 19.28.0.0.0 to 23.9.0.25.07
 Snapshots      : 42-62

 Note: This report displays selective data from internal Oracle
 views for database upgrades via AutoUpgrade. For a full Oracle
 statistics report, use AWR (Automatic Workload Repository).

    Begin Snap ID   End Snap ID Elapsed Time
----------------- ------------- ---------------
               42            62 66 mins

Begin Snap ID Begin Time                          Description
------------- ----------------------------------- -----------------------------------
           42 2025-Aug-24 07:13:28 +00:00         PDBS_START

  End Snap ID End Time                            Description
------------- ----------------------------------- -----------------------------------
           62 2025-Aug-24 08:19:41 +00:00         PDBS_END
```

#4 Top 20 Foreground Events

- Ordered by Total Wait Time, Waits

Event	Waits	Total Wait Time (sec)	Avg Wait (ms)	% DB Time	Wait Class
Disk file operations I/O	20,968,536	7,622.0	0.36	11.0	User I/O
library cache lock	26,145	4,959.8	189.70	7.2	Concurre
latch: shared pool	518,161	2,127.7	4.11	3.1	Concurre
library cache: mutex X	180,079	436.2	2.42	.6	Concurre
cell list of blocks physical r	240,920	338.1	1.40	.5	User I/O
cell single block physical rea	65,887	319.3	4.85	.5	User I/O
latch free	1,457,213	287.4	0.20	.4	Other
cell multiblock physical read	132,588	279.5	2.11	.4	User I/O
library cache: bucket mutex X	37,309	191.3	5.13	.3	Concurre
Free private memory to OS	415,968	114.2	0.27	.2	Other
DBWR range invalidation sync	827	74.0	89.50	.1	Configur
online DDL delay	31	60.0	1935.54	.1	Other
row cache lock	15,504	58.2	3.75	.1	Concurre
Allocate CGA memory from OS	2,589,636	50.0	0.02	.1	Other
cursor: pin S wait on X	4,352	48.5	11.15	.1	Concurre
ASM IO for non-blocking poll	1,269,052	46.2	0.04	.1	User I/O

#4 Top 20 Foreground Events

- Ordered by Total Wait Time, Waits

Event	Waits	Total Wait Time (sec)	Avg Wait (ms)	% DB Time	Wait Class
Disk file operations I/O	20,968,536	7,622.0	0.36	11.0	User I/O
library cache lock	26,145	4,959.8	189.70	7.2	Concurre
latch: shared pool	518,161	2,127.7	4.11	3.1	Concurre
library cache: mutex X	180,079	436.2	2.42	.6	Concurre
cell list of blocks physical r	240,920	338.1	1.40	.5	User I/O
cell single block physical rea	65,887	319.3	4.85	.5	User I/O
latch free	1,457,213	287.4	0.20	.4	Other
cell multiblock physical read	132,588	279.5	2.11	.4	User I/O
library cache: bucket mutex X	37,309	191.3	5.13	.3	Concurre
Free private memory to OS	415,968	114.2	0.27	.2	Other
DBWR range invalidation sync	827	74.0	89.50	.1	Configur
online DDL delay	31	60.0	1935.54	.1	Other
row cache lock	15,504	58.2	3.75	.1	Concurre
Allocate CGA memory from OS	2,589,636	50.0	0.02	.1	Other
cursor: pin S wait on X	4,352	48.5	11.15	.1	Concurre
ASM IO for non-blocking poll	1,269,052	46.2	0.04	.1	User I/O

#5. Background Wait Events

Event	Waits	Total Wait Time (s)	Avg Wait Time (ms)
AQPC idle	773	3,987	5,158.37
log file parallel write	2,584,727	123	.05
Streams AQ: load balancer id	316	116	367.73
flashback log file write	118,767	115	.97
db file parallel write	209,083	68	.32
db file async I/O submit	361,607	67	.18
log file sequential read	14,937	28	1.85
latch: shared pool	2,132	21	9.73
oracle thread bootstrap	371	10	27.03
latch free	3,957	9	2.24
ASM file metadata operation	24,151	9	.36
ASM IO for non-blocking poll	189,219	7	.04
latch: MGA shared context la	28	7	265.10
LGWR wait for redo copy	259,860	5	.02
LGWR worker group ordering	8,105	4	.53
resmgr:internal state change	26	3	100.09
KSV master wait	7,368	2	.33
os thread creation	632	2	3.50

#8 Wait Event Histogram for Top 20 Foreground Events

Event	Foreground Waits	Total Waits	% of Waits <1ms	1 to <32ms	32 to <1024ms	>= 1024ms
Disk file operations I/O	20,968,536	20981704	97.45	2.46	.09	.00
library cache lock	26,145	26211	4.65	61.25	31.08	3.02
latch: shared pool	518,161	520308	70.62	26.97	2.41	.00
library cache: mutex X	180,079	180242	80.36	19.51	.13	.00
cell list of blocks physic	240,920	241951	89.49	10.06	.45	.00
cell single block physical	65,887	65886	60.97	37.49	1.54	.00
latch free	1,457,213	1461626	97.60	2.40	.00	.00
cell multiblock physical r	132,588	133264	84.26	15.20	.54	.00
library cache: bucket mute	37,309	37345	54.06	45.66	.28	.00
Free private memory to OS	415,968	415995	98.17	1.82	.01	.00
DBWR range invalidation sy	827	829	1.45	49.46	49.10	.00
online DDL delay	31	31	3.23	.00	.00	96.77
row cache lock	15,504	15539	42.29	57.30	.40	.01
Allocate CGA memory from O	2,589,636	2611823	100.00	.00	.00	.00
cursor: pin S wait on X	4,352	4553	8.41	87.44	4.13	.02
ASM IO for non-blocking po	1,269,052	1458394	99.90	.10	.00	.00

```
SQL> select table_name from dba_tables where table_name like 'UPD$%' and owner='SYS';

TABLE_NAME
--------------------------------------------------------------------------------
UPD$_TOP_FG_EVENTS
UPD$_EVENT_HISTOGRAM
UPD$_OSSTAT
UPD$_PARAMETER
UPD$_SESSION_EVENT
UPD$_SESSTAT
UPD$_SESS_TIME_MODEL
UPD$_SNAPSHOT
UPD$_SYSSTAT
UPD$_SYSTEM_EVENT
UPD$_SYS_TIME_MODEL
UPD$_THREAD

12 rows selected.
```


An upgrade is always resumable and restartable

- All scripts are idempotent, including `catalog.sql` and `catproc.sql`

```
INSERT INTO sys.registry$upg_resume(version,phaseno,errorcnt,starttime,endtime)
VALUES ('23.0.0.0.0', 57,-1, sysdate, sysdate);

...

UPDATE sys.registry$upg_resume SET endtime=sysdate WHERE phaseno=57;
```


```
$ java -jar autoupgrade.jar ... -mode deploy

Previous execution found loading latest data
Total jobs recovered: 1
+--------------------------------+
| Starting AutoUpgrade execution |
+--------------------------------+
Type 'help' to list console commands
upg>
```


```
upg> resume -job 100 -catctl_options=-p 0
```

Replay Upgrade

The New Kid on the Block

```
--The database automatically starts an upgrade
--when you plug in a lower-release PDB

SQL> alter pluggable database pdb1 open;

Pluggable database altered.

Elapsed: 00:14:01.95
```

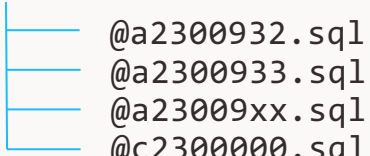
```
SQL> select property_name, property_value
     from   database_properties
     where  property_name like '%OPEN%';

PROPERTY_NAME           PROPERTY_VALUE
----------------------- --------------------
CONVERT_NONCDB_ON_OPEN  true
UPGRADE_PDB_ON_OPEN     true
```

Classic Upgrade

Phase 1
Phase 2
Phase 3
Phase 4
Phase 5
Phase 6
Phase 7
Phase 8
...
Phase *nnn*

Classic Upgrade

Phase 1
Phase 2
Phase 3
Phase 4
Phase 5



Phase 6
Phase 7
Phase 8
...
Phase *nnn*

Classic Upgrade

@a2300932.sql

```
VARIABLE initfile VARCHAR2(32)
COLUMN :initfile NEW_VALUE init_file NOPRINT;

Rem ================================================================
Rem SQLJTYPE
Rem ================================================================

BEGIN
  IF sys.dbms_registry.is_loaded('JAVAVM',sys.dbms_registry.release_version) = 1 THEN
     :initfile := 'initsjty.sql';
  ELSE
     :initfile := 'nothing.sql';
  END IF;
END;
/
SELECT :initfile FROM DUAL;
@@&init_file
```

Classic Upgrade

```
@@&init_file
```

```
.
.
[more PL/SQL code]
.
.
CREATE TABLE SYS.T1 ...
CREATE INDEX SYS.T1I1 ...
.
.
[more PL/SQL code]
.
.
```

Comparison

Classic

Phase 1
Phase 2
Phase 3
Phase 4
Phase 5
Phase 6
Phase 7
Phase 8
...
Phase *nnn*

Replay

```
DROP INDEX SYSTEM.IDX$FLOW ...
CREATE OR REPLACE ...
ALTER TYPE ...
CREATE FUNCTION ...
CREATE TABLE SYS.T1 ...
CREATE INDEX SYS.T1I1 ...
DROP INDEX MDSYS.IDX$IK ...
DROP TABLE MDSYS.TBL$TT ...
CREATE OR REPLACE ...
ALTER TYPE ...
GRANT SELECT ON ...
CREATE VIEW ...
```

```
select sqlstmt from pdb_sync$;


ALTER SESSION SET "_oracle_script_counter"=7
alter pluggable database application app$cdb$pdbonly$ncdbtopdb begin install '1.0.upgmode'
alter session set "_enable_view_pdb"=false
alter session set NLS_LENGTH_SEMANTICS=BYTE
INSERT INTO sys.utl_recomp_skip_list select obj# from obj$ where BITAND(flags, 4194304)=0 ...
create or replace view sys.cdb$common_root_objects sharing=object as
select u.name owner, o.name object_name, o.type# object_type, o.namespace nsp,
       o.subname object_subname, o.signature object_sig,
       decode(bitand(o.flags, (65536+131072+4294967296)),
       4294967296+65536, 'EDL', 131072, 'DL', 'MDL') sharing
  from sys.obj$ o, sys.user$ u
 where o.owner#=u.user# and bitand(o.flags, (65536+131072+4294967296)) <> 0
   and bitand(o.flags,0)=0


(output truncated)
```


Classic

- Triggered by AutoUpgrade
- Runs `catalog.sql` / `catproc.sql`
- Many `CREATE OR REPLACE` statements for objects that didn't change
- Customizable
- Used by AutoUpgrade

Replay

- Triggered by `OPEN` command
- Runs the captured statements
- Only statements that actually do some change
- Automated

Classic

```
Stages
   SETUP              <1 min
   PREUPGRADE         <1 min
   PRECHECKS          <1 min
   PREFIXUPS          <1 min
   DRAIN              <1 min
   DBUPGRADE          19 min
   DISPATCH           <1 min
   UNPLUGWORK         <1 min
   POSTCHECKS         <1 min
   POSTFIXUPS         10 min
   POSTUPGRADE        <1 min
   SYSUPDATES         <1 min
```

Replay

```
Stages
   SETUP              <1 min
   PREUPGRADE         <1 min
   PRECHECKS          <1 min
   PREFIXUPS          <1 min
   DRAIN              <1 min
   DBUPGRADE          17 min
   DISPATCH           <1 min
   UNPLUGWORK         <1 min
   POSTCHECKS         <1 min
   POSTFIXUPS         10 min
   POSTUPGRADE        <1 min
   SYSUPDATES         <1 min
```


Replay upgrade doesn't handle pre- and post-upgrade tasks

- You must run these manually
- Or use Replay Upgrade through AutoUpgrade

```
SQL> alter pluggable database pdb1 open;
alter pluggable database pdb1 open
*
ERROR at line 1:
ORA-60510: encountered an error during Replay Upgrade
```

If Replay Upgrade fails

- Check for errors:
  - `SELECT * FROM dba_replay_upgrade_errors`
  - `SELECT * FROM dba_app_errors`
  - `SELECT * FROM dba_applications WHERE app_name='APP$CDB$CATALOG';`
  - Check alert log
  - Trace files
- Revert to classic upgrade
  - Use AutoUpgrade (`upg1.replay=no`)

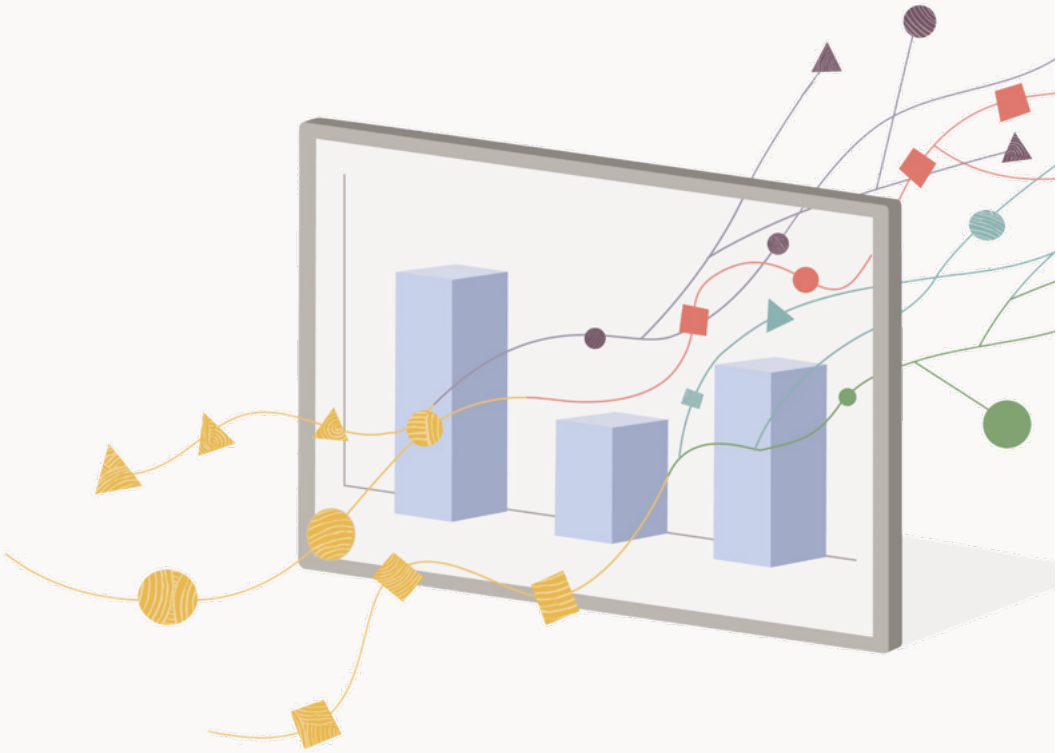
```
--To disable replay upgrade
ALTER DATABASE UPGRADE SYNC OFF;

--Or
ALTER DATABASE PROPERTY SET UPGRADE_PDB_ON_OPEN='false';

--To disable convert on open
ALTER DATABASE PROPERTY SET CONVERT_NONCDB_ON_OPEN='false';
```

Distributed Upgrades

Faster upgrades on Oracle RAC Database

41%

faster



If you are upgrading a cluster database ...

... you must manually separate the database instance from the cluster. Set the CLUSTER_DATABASE initialization parameter to false.

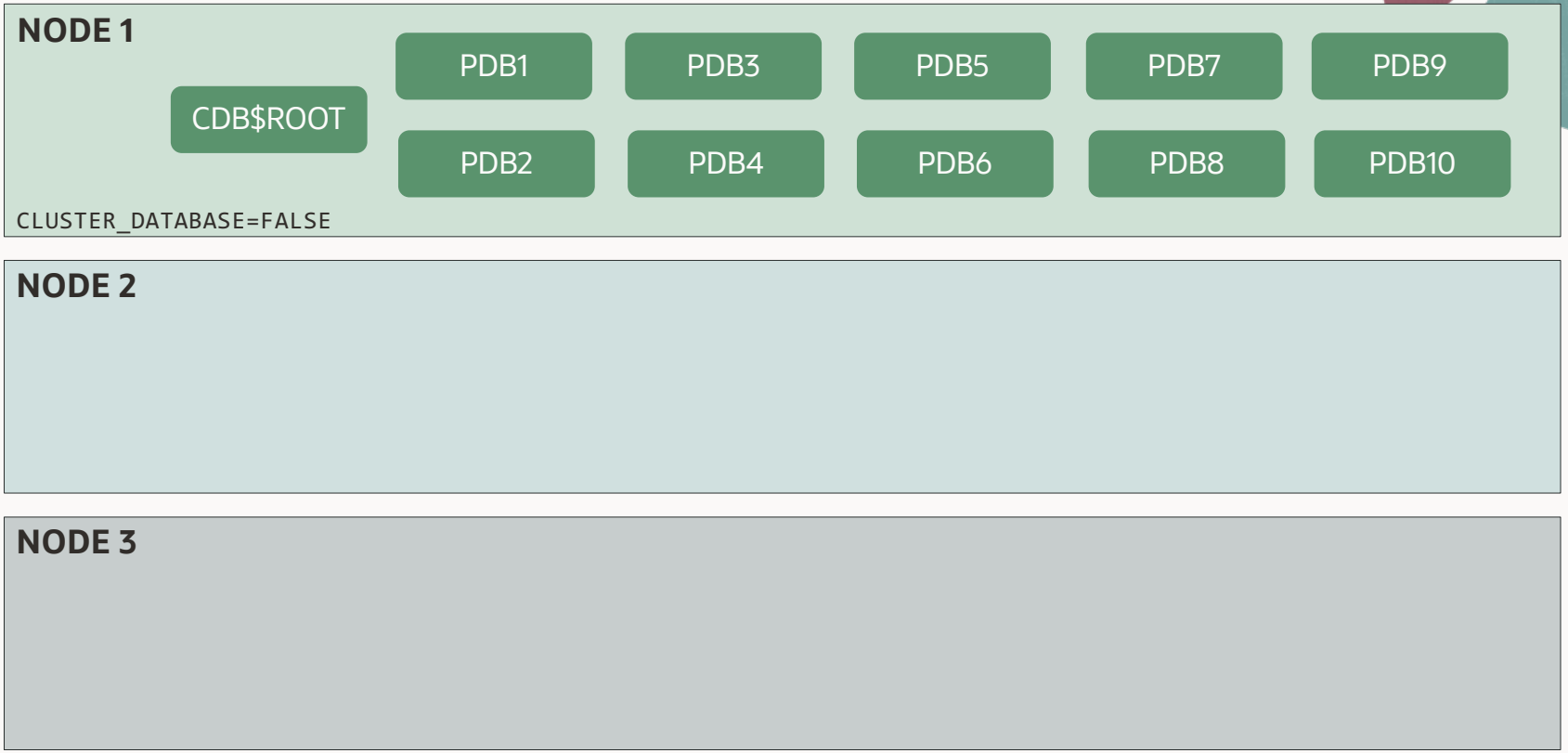
Regular RAC Database Upgrade

- `CLUSTER_DATABASE = FALSE`
- Upgrade root and all PDBs
- One instance active
- All other nodes are idle

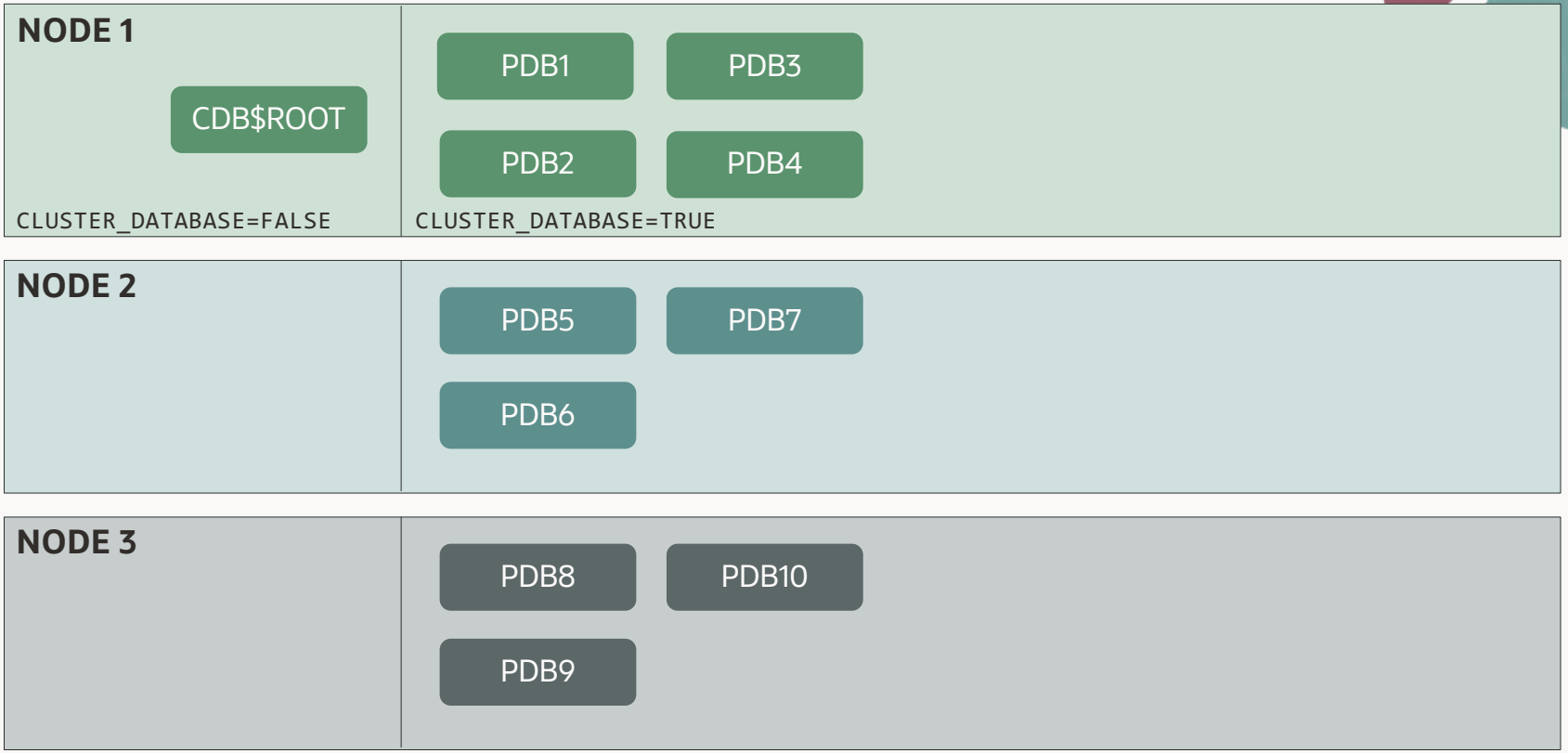
Distributed Upgrade

- `CLUSTER_DATABASE = FALSE`
- Upgrade root only
- `CLUSTER_DATABASE = TRUE`
- Upgrade all PDBs
- All instances active

Distributed Upgrade



Distributed Upgrade



```
cat RACDB.cfg

global.autoupg_log_dir=/u01/app/oracle/autoupgrade
upg1.source_home=/u01/app/oracle/product/19
upg1.target_home=/u01/app/oracle/product/26
upg1.sid=RACDB
upg1.tune_setting=distributed_upgrade=true
```


```
java -jar autoupgrade.jar -config RACDB.cfg -mode deploy
```

```
cat RACDB.cffg

global.autoupg_log_dir=/u01/app/oracle/autoupgrade
upg1.source_home=/u01/app/oracle/product/19
upg1.target_home=/u01/app/oracle/product/26
upg1.sid=RACDB
upg1.tune_setting=distributed_upgrade=true

java -jar autoupgrade.jar -config RACDB.cfg -mode deploy
```

```
batch1.source_home=/u01/app/oracle/product/19
batch1.target_home=/u01/app/oracle/product/26
batch1.sid=RACDB1
batch1.upgrade_node=NODE1
batch1.pdbs=PDB1,PDB2,PDB3,PDB4
batch2.source_home=/u01/app/oracle/product/19
batch2.target_home=/u01/app/oracle/product/26
batch2.sid=RACDB2
batch2.upgrade_node=NODE2
batch2.pdbs=PDB5,PDB6,PDB7
batch3.source_home=/u01/app/oracle/product/19
batch3.target_home=/u01/app/oracle/product/26
batch3.sid=RACDB3
batch3.upgrade_node=NODE3
batch3.pdbs=PDB8,PDB9,PDB10
```


```
cat RACDB.cffg

global.autoupg_log_dir=/u01/app/oracle/autoupgrade
upg1.source_home=/u01/app/oracle/product/19
upg1.target_home=/u01/app/oracle/product/26
upg1.sid=RACDB
upg1.tune_setting=distributed_upgrade=true

java -jar autoupgrade.jar -config RACDB.cfg -mode deploy
```

```
batch1.source_home=/u01/app/oracle/product/19
batch1.target_home=/u01/app/oracle/product/26
batch1.sid=RACDB1
batch1.upgrade_node=NODE1
batch1.pdbs=PDB1,PDB2,PDB3,PDB4
batch2.source_home=/u01/app/oracle/product/19
batch2.target_home=/u01/app/oracle/product/26
batch2.sid=RACDB2
batch2.upgrade_node=NODE2
batch2.pdbs=PDB5,PDB6,PDB7
batch3.source_home=/u01/app/oracle/product/19
batch3.target_home=/u01/app/oracle/product/26
batch3.sid=RACDB3
batch3.upgrade_node=NODE3
batch3.pdbs=PDB8,PDB9,PDB10
```


```
cat RACDB.cffg

global.autoupg_log_dir=/u01/app/oracle/autoupgrade
upg1.source_home=/u01/app/oracle/product/19
upg1.target_home=/u01/app/oracle/product/26
upg1.sid=RACDB
upg1.tune_setting=distributed_upgrade=true

java -jar autoupgrade.jar -config RACDB.cfg -mode deploy
```

```
batch1.source_home=/u01/app/oracle/product/19
batch1.target_home=/u01/app/oracle/product/26
batch1.sid=RACDB1
batch1.upgrade_node=NODE1
batch1.pdbs=PDB1,PDB2,PDB3,PDB4
batch2.source_home=/u01/app/oracle/product/19
batch2.target_home=/u01/app/oracle/product/26
batch2.sid=RACDB2
batch2.upgrade_node=NODE2
batch2.pdbs=PDB5,PDB6,PDB7
batch3.source_home=/u01/app/oracle/product/19
batch3.target_home=/u01/app/oracle/product/26
batch3.sid=RACDB3
batch3.upgrade_node=NODE3
batch3.pdbs=PDB8,PDB9,PDB10
```

```
java -jar autoupgrade.jar -config RACDB.cfg -mode deploy
```

```
Node 1    java -jar autoupgrade.jar -config mod.conf -mode upgrade -follower
Node 2    java -jar autoupgrade.jar -config mod.conf -mode upgrade -follower
Node 3    java -jar autoupgrade.jar -config mod.conf -mode upgrade -follower
```

```
java -jar autoupgrade.jar -config RACDB.cfg -mode deploy
```

```
Node 1    java -jar autoupgrade.jar -config mod.conf -mode upgrade -follower
Node 2    java -jar autoupgrade.jar -config mod.conf -mode upgrade -follower
Node 3    java -jar autoupgrade.jar -config mod.conf -mode upgrade -follower
```

```
java -jar autoupgrade.jar -config RACDB.cfg -mode deploy
```

```
Node 1      java -jar autoupgrade.jar -config mod.conf -mode upgrade -follower
Node 2      java -jar autoupgrade.jar -config mod.conf -mode upgrade -follower
Node 3      java -jar autoupgrade.jar -config mod.conf -mode upgrade -follower
```

```
upg> status -job 101

    +--------+---------+--------+-------------+
    |Database|    Stage|Progress|         Node|
    +--------+---------+--------+-------------+
    |PDB$SEED|DBUPGRADE|    10 %|dragon-dbvui1|
    |  PDB001|DBUPGRADE|     6 %|dragon-dbvui2|
    |  PDB002|DBUPGRADE|    10 %|dragon-dbvui1|
    |  PDB003|DBUPGRADE|     6 %|dragon-dbvui2|

    (output truncated)

    |  PDB027|DBUPGRADE|     6 %|dragon-dbvui2|
    |  PDB028|DBUPGRADE|    10 %|dragon-dbvui1|
    |  PDB029|DBUPGRADE|     6 %|dragon-dbvui2|
    |  PDB030|DBUPGRADE|    10 %|dragon-dbvui1|
    +--------+---------+--------+-------------+
```

```
# Control the number of nodes used, default 2

upg1.tune_setting=distributed_upgrade=true,active_nodes_limit=n
```


Downgrade

Just In Case...

A downgraded data dictionary is different, but compatible

- A downgraded database is not identical to the pre-upgraded database

```
SQL> select count(*) from dba_tables where owner='SYS';

COUNT_BEFORE
------------
        1549


 COUNT_AFTER
------------
        1805
```

Downgrade

1

Actions in 26ai Oracle home

- Apply latest Release Update
- Open in downgrade mode
- Execute `catdwgrd.sql`
  - Rolls back one-off patches
  - Prepares for old release

2

Actions in 19c Oracle home

- Apply latest Release Update
- Open in upgrade mode
- Execute `catrelod.sql`
  - Invalidates all PL/SQL
  - Executes catalog/catproc
  - Reloads all components
  - Gather stats on bootstrap tables
- Open in normal mode
- Recompile all PL/SQL
- Gather dictionary/fixed objects stats

```
$ grep -i "truncate *table" catdwgrd0.log
...
11:56:38 SQL> truncate table wri$_optstat_tab_model;
11:56:38   3          execute immediate 'truncate table hcs_av_fact_column$';
11:56:38 SQL> Rem truncate table to store service conflicts stats persistently
11:56:38 SQL> truncate table svcobj_conflict$;
11:56:38 SQL> truncate table data_guard_site$;
11:56:38 SQL> truncate table dgpdb_site$;
11:56:39   4          execute immediate 'truncate table hcs_tbl$';
11:56:39 SQL> truncate table rsrc_cnt_history$;
```

```
$ grep -i "truncate *table" catdwgrd0.log
...
11:56:38 SQL> truncate table wri$_optstat_tab_model;
11:56:38   3          execute immediate 'truncate table hcs_av_fact_column$';
11:56:38 SQL> Rem truncate table to store service conflicts stats persistently
11:56:38 SQL> truncate table svcobj_conflict$;
11:56:38 SQL> truncate table data_guard_site$;
11:56:38 SQL> truncate table dgpdb_site$;
11:56:39   4          execute immediate 'truncate table hcs_tbl$';
11:56:39 SQL> truncate table rsrc_cnt_history$;

$ grep -i "truncate *table" catdwgrd0.log | wc -l
311
```

```
SQL> desc pdb_sync$

          Name       Null?              Type
_____________ ___________ ________________
SCNWRP        NOT NULL    NUMBER
SCNBAS        NOT NULL    NUMBER
CTIME         NOT NULL    DATE
SQLSTMT                   VARCHAR2(4000)
NAME          NOT NULL    VARCHAR2(128)
AUXNAME1                  VARCHAR2(128)
AUXNAME2                  VARCHAR2(128)
OPCODE        NOT NULL    NUMBER
FLAGS                     NUMBER
LONGSQLTXT                CLOB
REPLAY#       NOT NULL    NUMBER
CREATION#                 NUMBER
SPARE3                    VARCHAR2(128)
SPARE4                    VARCHAR2(128)
SPARE5                    VARCHAR2(4000)
SPARE6                    NUMBER
SPARE7                    NUMBER
SPARE8                    NUMBER
SQLID                     VARCHAR2(13)
APPID#                    NUMBER
VER#                      NUMBER
PATCH#                    NUMBER
APP_STATUS                NUMBER
SESSSERIAL#               NUMBER
```

```
SQL> desc pdb_sync$

          Name       Null?              Type
_____________ ___________ ________________
SCNWRP        NOT NULL    NUMBER
SCNBAS        NOT NULL    NUMBER
CTIME         NOT NULL    DATE
SQLSTMT                   VARCHAR2(4000)
NAME          NOT NULL    VARCHAR2(128)
AUXNAME1                  VARCHAR2(128)
AUXNAME2                  VARCHAR2(128)
OPCODE        NOT NULL    NUMBER
FLAGS                     NUMBER
LONGSQLTXT                CLOB
REPLAY#       NOT NULL    NUMBER
CREATION#                 NUMBER
SPARE3                    VARCHAR2(128)
SPARE4                    VARCHAR2(128)
SPARE5                    VARCHAR2(4000)
SPARE6                    NUMBER
SPARE7                    NUMBER
SPARE8                    NUMBER
SQLID                     VARCHAR2(13)
APPID#                    NUMBER
VER#                      NUMBER
PATCH#                    NUMBER
APP_STATUS                NUMBER
SESSSERIAL#               NUMBER
OWNER                     VARCHAR2(128)
OBJNAME                   VARCHAR2(128)
EDNAME                    VARCHAR2(128)
NAMESPACE                 NUMBER
SIGNATURE                 RAW(16 BYTE)
MODULE                    VARCHAR2(64)
ACTION                    VARCHAR2(64)
```

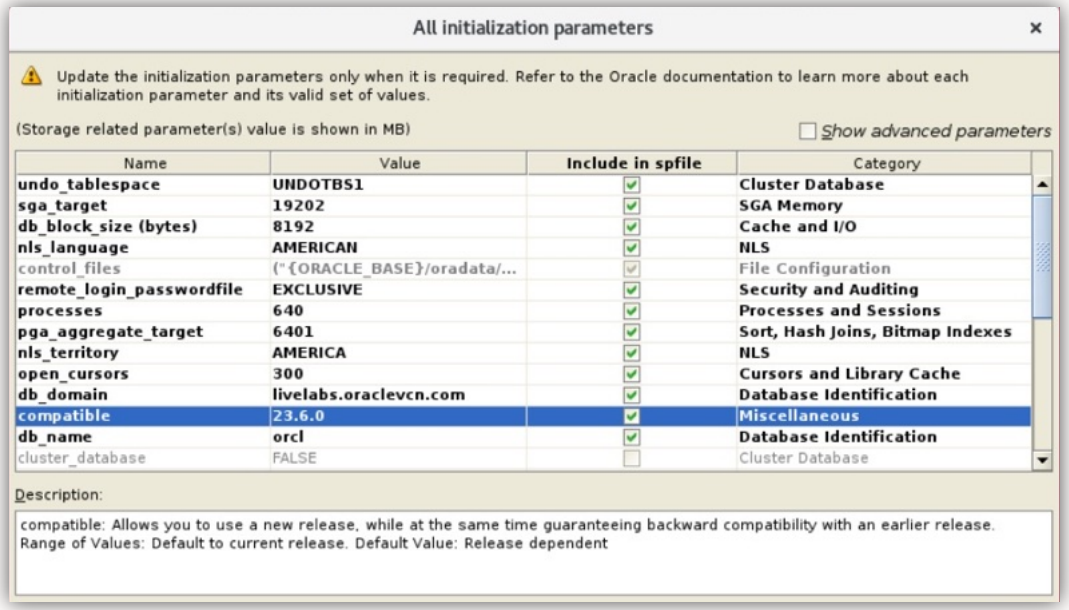
```
SQL> desc REGISTRY$LOG

        Name       Null?              Type
____________ ____________ __________________
CID                       VARCHAR2(30)
NAMESPACE                 VARCHAR2(30)
OPERATION    NOT NULL     NUMBER
OPTIME                    TIMESTAMP(6)
ERRMSG                    VARCHAR2(1000)
```

```
SQL> desc REGISTRY$LOG

        Name       Null?                              Type
____________ ____________ ___________________________________
CID                       VARCHAR2(30)
NAMESPACE                 VARCHAR2(30)
OPERATION    NOT NULL     NUMBER
OPTIME                    TIMESTAMP(6) WITH TIME ZONE
ERRMSG                    VARCHAR2(1000)
```

Downgrade, Guidelines

- Generally, dropping is avoided
- Some types of statistics are marked stale
- Complete rebuild of certain components, such as Data Pump

Don't change `COMPATIBLE` if you want the option of downgrading



You can downgrade days, months, or even years after upgrading

- No data loss

Wrapping Up

The final words

Use ORAdiff to learn about data dictionary differences

- oradiff.oracle.com

How can I make my upgrades faster?

Things that

Matter a lot

- Number of components
- Dictionary size
- Dictionary complexity
- Some feature/version combinations

Things that

Matter a lot

- Number of components
- Dictionary size
- Dictionary complexity
- Some feature/version combinations

Matter a little

- CPU speed
- I/O capabilities
- Memory
- SGA/PGA

Things that

Matter a lot

- Number of components
- Dictionary size
- Dictionary complexity
- Some feature/version combinations

Matter a little

- CPU speed
- I/O capabilities
- Memory
- SGA/PGA

Don't matter

(usually)

- Physical size
- Amount of user data

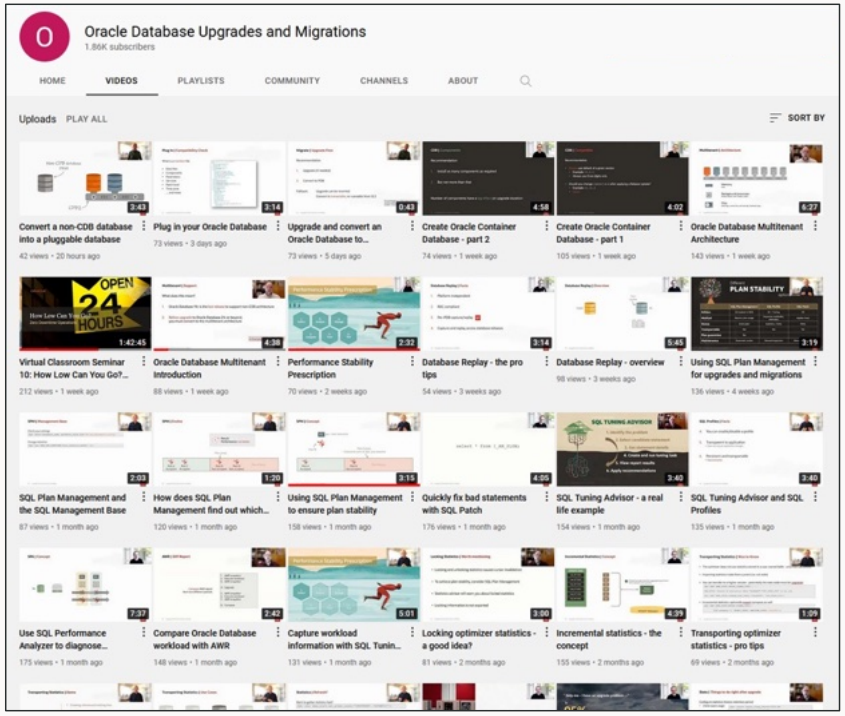
Unplug-plug upgrades are faster than upgrading an entire CDB

- Consider your rollback options
- Use refreshable clone PDBs

Key Learnings

1. Use AutoUpgrade
2. Scale by upgrading more PDBs in parallel
3. Use Distributed Upgrade

YouTube Channel



https://www.youtube.com/@upgradenow

- 600+ videos
- New videos every week
- No marketing
- No buzzword
- All tech

SCAN ME

Find Slides and Much More on Our Blogs

MikeDietrichDE.com

Mike.Dietrich@oracle.com

dohdatabase.com

Daniel.Overby.Hansen@oracle.com

DBArj.com.br

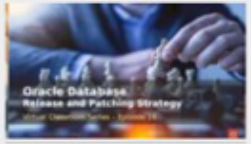
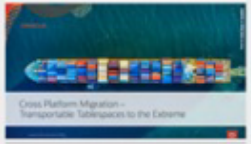
Rodrigo.R.Jorge@oracle.com

AlexZaballa.com

Alex.Zaballa@oracle.com

axdiaz.com

jorge.a.diaz@oracle.com

Virtual Classroom Seminars	Slides
Episode 16 (replaces Episode 1 from Feb 2021) Oracle Database Release and Patching Strategy for 19c and 23c *115 minutes – May 10, 2023*	
Episode 17 From SR to Patch – Insights into the Oracle Database Development process *55 minutes – June 22, 2023*	
Episode 18 Cross Platform Migration – Transportable Tablespaces to the Extreme *145 min – February 22, 2024*	
Episode 19 Move to Oracle Database 23ai – Everything you need to know about Multitenant PART 1 *145 min – May 16, 2024*	
Episode 20 Move to Oracle Database 23ai – Everything you need to know about Multitenant PART 2 *100 min – June 28, 2024*	
Episode 21 One-Button Patching with AutoUpgrade – Easing every DBA's life *55 min – October 24, 2024*	

Recorded Web Seminars

https://MikeDietrichDE.com/videos

More than 45 hours of technical content
On-demand, anytime, anywhere

Upgrade Internals

- Blog post: Remove and Clean Up Components from Oracle Database 11.2 – 19c
- Blog post: Upgrading Oracle Database Releases Using Replay Upgrade


ORACLE