Upgrade / Migrate / Consolidate

Oracle Database 19c

Updated: Nov 9, 2023

Roy F. Swonger

Vice President

Database Upgrade, Utilities & Patching

@RoyFSwonger

Mike Dietrich

Distinguished Product Manager

Database Upgrade
and Migrations

https://MikeDietrichDE.com

@MikeDietrichDE

mikedietrich

Daniel Overby Hansen

Senior Principal Product Manager
Cloud Migration

https://dohdatabase.com
@dohdatabase
dohdatabase

William Beauregard

Senior Principal Product Manager
Data Pump and Utilities

in william-beauregard-3053791

Rodrigo Jorge

Senior Principal Product Manager
Database Patching and Upgrade

https://dbarj.com.br/en
@rodrigojorgedba
rodrigoaraujorge

Roy Swonger

Daniel Overby Hansen

William Beauregard

Mike Dietrich

Rodrigo Jorge

ROY SWONGER
Vice President
Database Upgrade, Utilities & Patching

royfswonger

@royfswonger

WILLIAM BEAUREGARD

Senior Principal Product Manager
Data Pump and SQL*Loader

william-beauregard-3053791

RODRIGO JORGE
Senior Principal Product Manager
Database Patching and Upgrade

in rodrigoarauje

@rodrigojorgedba

https://dbarj.com.br/en

MIKE DIETRICH
Senior Director
Database Upgrade and Migrations

in mikedietrich

@mikedietrichde

https://mikedietrichde.com

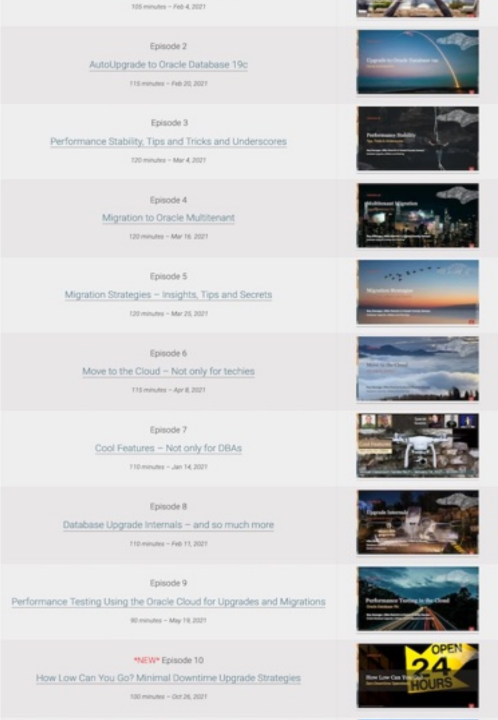
DANIEL OVERBY HANSEN

Senior Principal Product Manager
Cloud Migration

in dohdatabase

@dohdatabase

https://dohdatabase.com



105 minutes – Feb 4, 2021

Episode 2

AutoUpgrade to Oracle Database 19c

115 minutes – Feb 20, 2021

Episode 3

Performance Stability, Tips and Tricks and Underscores

120 minutes – Mar 4, 2021

Episode 4

Migration to Oracle Multitenant

120 minutes – Mar 16, 2021

Episode 5

Migration Strategies – Insights, Tips and Secrets

120 minutes – Mar 25, 2021

Episode 6

Move to the Cloud – Not only for techies

115 minutes – Apr 8, 2021

Episode 7

Cool Features – Not only for DBAs

110 minutes – Jan 14, 2021

Episode 8

Database Upgrade Internals – and so much more

110 minutes – Feb 11, 2021

Episode 9

Performance Testing Using the Oracle Cloud for Upgrades and Migrations

90 minutes – May 19, 2021

NEW Episode 10

How Low Can You Go? Minimal Downtime Upgrade Strategies

100 minutes – Oct 26, 2021

Recorded Web Seminars

https://MikeDietrichDE.com/videos

Photo by SpaceX on Unsplash

Chapter 2

Upgrade to Oracle Database 19c

your key to

Successful Database Upgrades

Step 1
Download and install Oracle 19c
eDelivery.oracle.com

Step 2
Download and install newest RU
MOS Note: 2118136.2

Step 3
Download and use AutoUpgrade
MOS Note: 2485457.1

Step 4
Performance Stability with SPM, STA and RAT

Upgrade using

AutoUpgrade

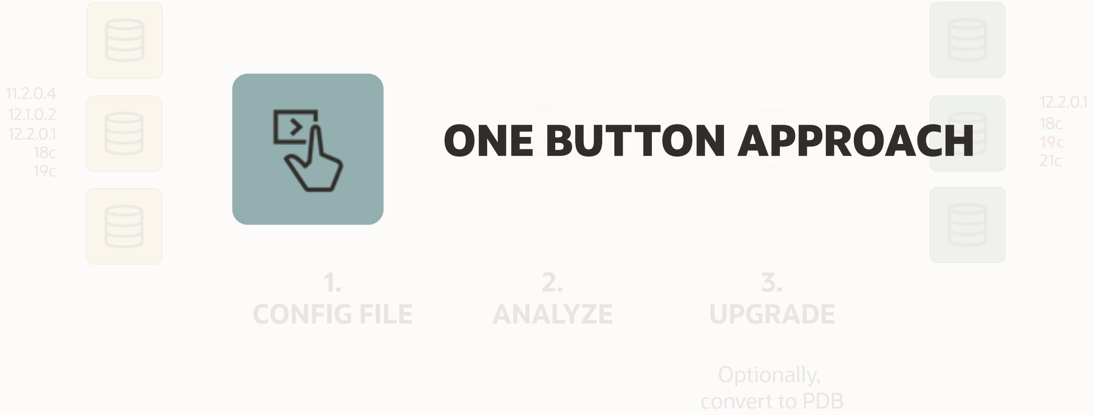
complete
AUTOMATION

rich on
FEATURES

fleet
SCALE

highly
RECOMMENDED

AutoUpgrade | Overview



11.2.0.4
12.1.0.2
12.2.0.1
18c
19c

ONE BUTTON APPROACH

1.
CONFIG FILE

2.
ANALYZE

3.
UPGRADE

Optionally,
convert to PDB

12.2.0.1
18c
19c
21c

Get started | Quick Start Guide

Simple overview

Read it, try it

Download from oracle.com

Oracle Database Upgrade: Quick Start Guide

A quick reference to a successful Oracle Database upgrade

Feb 20, 2020 | Version 1.0

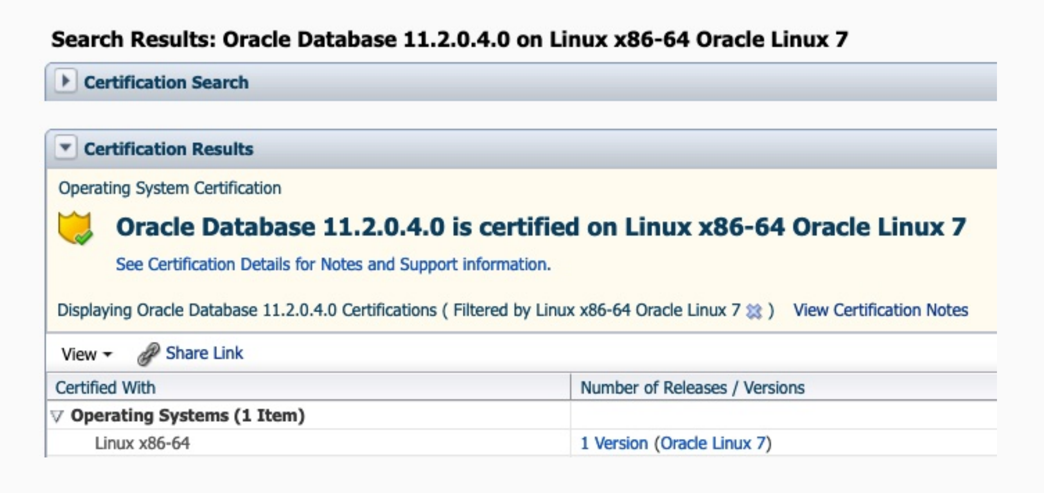
Public

Photo by Pablo Heimplatz on Unsplash

Check

Before Upgrade

Supportability | OS Certification



Search Results: Oracle Database 11.2.0.4.0 on Linux x86-64 Oracle Linux 7

Operating System Certification

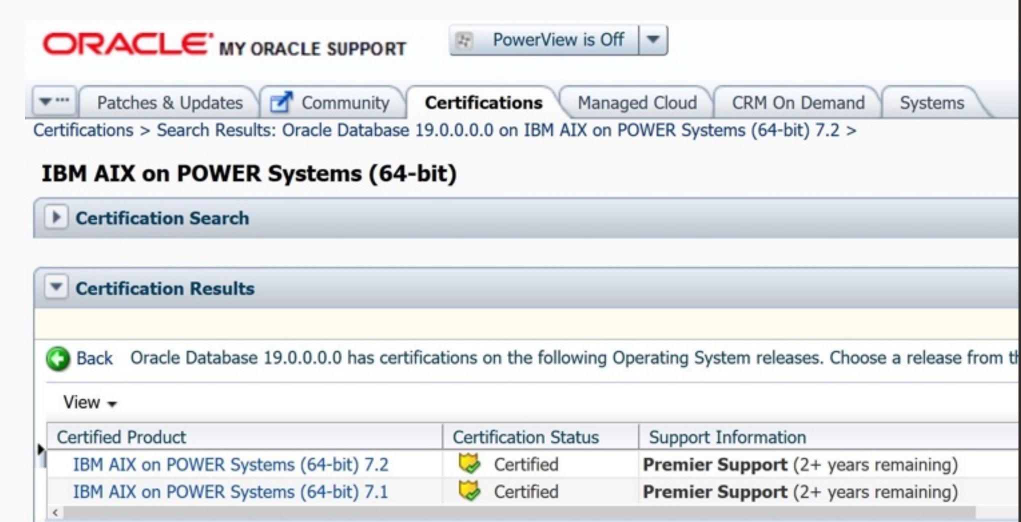
Oracle Database 11.2.0.4.0 is certified on Linux x86-64 Oracle Linux 7

See Certification Details for Notes and Support information.

Displaying Oracle Database 11.2.0.4.0 Certifications (Filtered by Linux x86-64 Oracle Linux 7) View Certification Notes

Certified With	Number of Releases / Versions
▽ **Operating Systems (1 Item)**	
Linux x86-64	1 Version (Oracle Linux 7)

Supportability | OS Certification



Certifications > Search Results: Oracle Database 19.0.0.0.0 on IBM AIX on POWER Systems (64-bit) 7.2 >

IBM AIX on POWER Systems (64-bit)

▸ Certification Search

▾ Certification Results

Back Oracle Database 19.0.0.0.0 has certifications on the following Operating System releases. Choose a release from th

View ▾

Certified Product	Certification Status	Support Information
IBM AIX on POWER Systems (64-bit) 7.2	Certified	**Premier Support** (2+ years remaining)
IBM AIX on POWER Systems (64-bit) 7.1	Certified	**Premier Support** (2+ years remaining)

Platform Certification | Linux x86-64

	Database 11.2.0.4	Database 12.1.0.2	Database 19c
Oracle Linux 4	Certified		
Oracle Linux 5	Certified	Certified	
Oracle Linux 6	Certified	Certified	
Oracle Linux 7	Certified	Certified	Certified
Oracle Linux 8		Certified	Certified
Red Hat 4	Certified		
Red Hat 5	Certified	Certified	
Red Hat 6	Certified	Certified	
Red Hat 7	Certified	Certified	Certified
Red Hat 8		Certified	Certified

Platform Certification | Windows Server

	Database 11.2.0.4	Database 12.1.0.2	Database 19c
Microsoft Windows x64 2003	Certified		
Microsoft Windows x64 2003 R2	Certified		
Microsoft Windows x64 2008	Certified	Certified	
Microsoft Windows x64 2008 R2	Certified	Certified	
Microsoft Windows x64 2012	Certified	Certified	
Microsoft Windows x64 2012 R2	Certified	Certified	Certified
Microsoft Windows x64 2016			Certified
Microsoft Windows x64 2019			Certified

Platform Certification | AIX

	Database 11.2.0.4	Database 12.1.0.2	Database 19c
POWER Systems (64-bit) 5.3	Certified		
POWER Systems (64-bit) 6.1	Certified	Certified	
POWER Systems (64-bit) 7.1	Certified	Certified	Certified
POWER Systems (64-bit) 7.2	Certified	Certified	Certified

More Links on the Upgrade Blog

Oracle Clusterware Certification on OL8/RHEL8
https://mikedietrichde.com/2020/05/14/of-course-oracle-clusterware-is-certified-on-ol8-rhel8-as-well/

Oracle Database 19c Certification on OL8/RHEL8
https://mikedietrichde.com/2020/05/11/oracle-database-19c-is-certified-on-ol8-and-rhel8/

Of course, Oracle Clusterware is certified on OL8/RHEL8 as well

Posted on May 14, 2020 by Mike.Dietrich Operating System

To be very honest, when I posted a few days ago, Oracle Database 19c is certified on OL8/RHEL8 I didn't check for the Oracle **Clusterware (OCW) certification**. I blindly assumed that this is the case. But from comments and discussions on twitter, I realized it may be necessary to point this out explicitly to avoid confusion. Even though my fellow mate, RAC Product Manager Anil Nair has pointed it out already many times: **Of course, Oracle Clusterware is certified on OL8/RHEL8 as well**.

Photo by Angela Hobbs on Unsplash

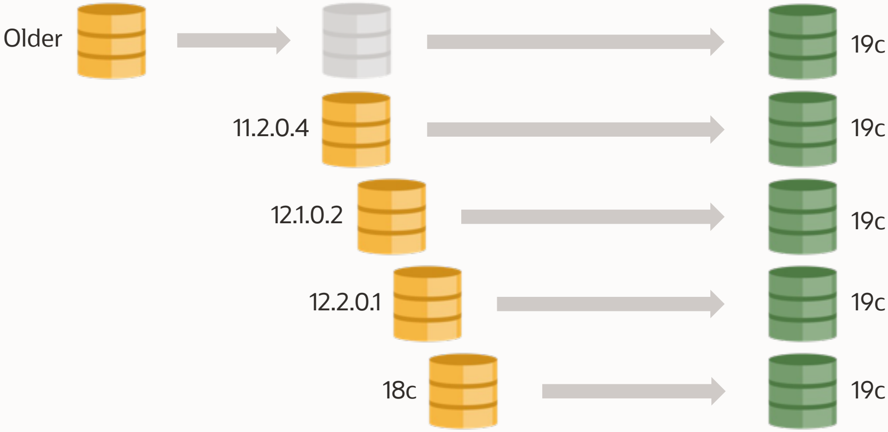
Oracle Database 19c is certified on OL8 and RHEL8

Posted on May 11, 2020 by Mike.Dietrich Operating System

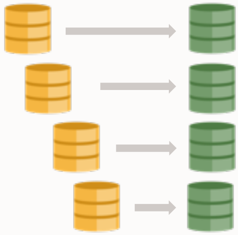
Oh … I'd say this was the question I've got asked the second most often in the past months (after a potential extension for Oracle 12.2.0.1 support): ***When will Oracle certify OL8 and RHEL8?*** And I read this question internally even more often. Now thanks to my team mate Daniel Overby Hansen who spotted it on Twitter on the weekend, we can tell you that … finally … drum roll … **Oracle Database 19c is certified on OL8 and RHEL8**.

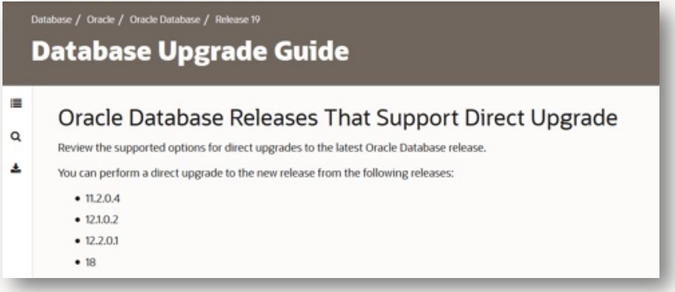
Photo by Derek Oyen on Unsplash

Database Upgrade | Supported Releases



Database Upgrade | Supported Releases





Database / Oracle / Oracle Database / Release 19

Database Upgrade Guide

Oracle Database Releases That Support Direct Upgrade

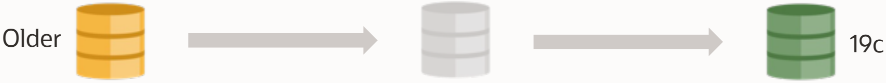
Review the supported options for direct upgrades to the latest Oracle Database release.

You can perform a direct upgrade to the new release from the following releases:

- 11.2.0.4
- 12.1.0.2
- 12.2.0.1
- 18

Database Upgrade Guide

Database Upgrade | Intermediate Upgrades

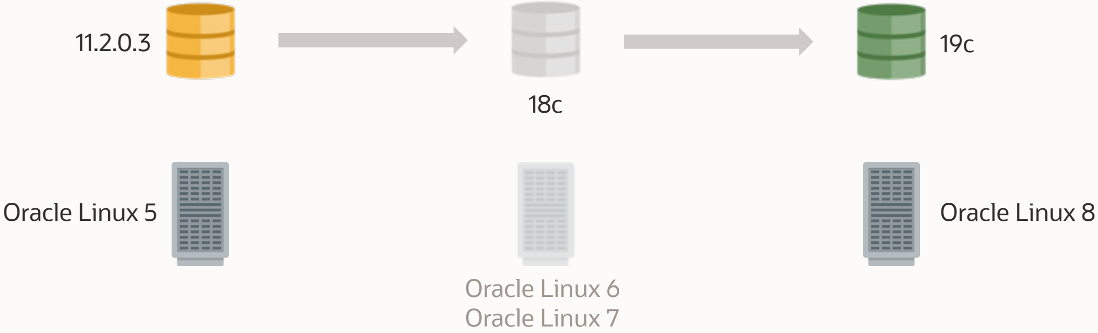


- Ideally - upgrade as far as possible in the intermediate upgrade

- But often there is a platform upgrades as well

- So - it depends

Database Upgrade | Intermediate Upgrades



11.2.0.3

18c

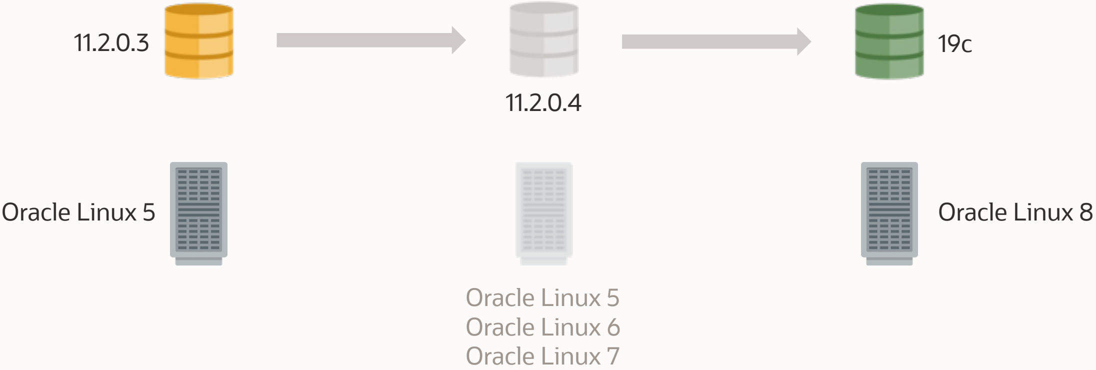
19c

Oracle Linux 5

Oracle Linux 6
Oracle Linux 7

Oracle Linux 8

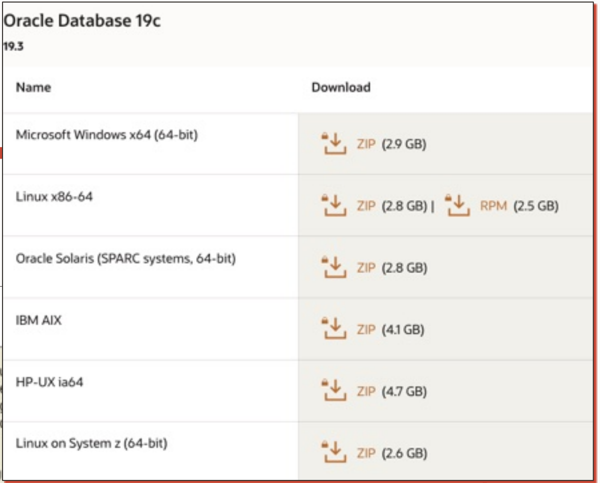
Database Upgrade | Intermediate Upgrades



Oracle 19c | Installation

Gold Image

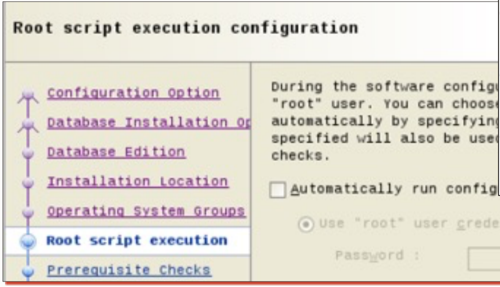
1. Create `ORACLE_HOME` directory
2. Download image
3. Unpack into `ORACLE_HOME`
4. `./runInstaller`
5. `root.sh`



Oracle Database 19c

19.3

Name	Download
Microsoft Windows x64 (64-bit)	ZIP (2.9 GB)
Linux x86-64	ZIP (2.8 GB) | RPM (2.5 GB)
Oracle Solaris (SPARC systems, 64-bit)	ZIP (2.8 GB)
IBM AIX	ZIP (4.1 GB)
HP-UX ia64	ZIP (4.7 GB)
Linux on System z (64-bit)	ZIP (2.6 GB)



Root script execution configuration

During the software configu
"root" user. You can choose
automatically by specifying
specified will also be used
checks.

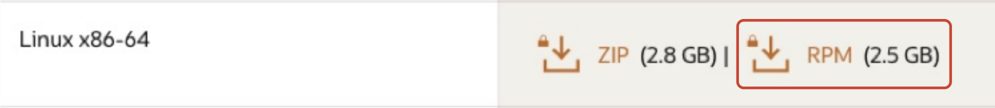
□ Automatically run config

○ Use "root" user credential

Password :

Oracle 19c | RPM Installation

RPM



Linux x86-64

ZIP (2.8 GB) | RPM (2.5 GB)

- `yum install -y oracle-database-preinstall-19c`
- `yum -y localinstall oracle-database-ee-19c-1.0-1.x86_64.rpm`
- rpm installs always into:
  `/opt/oracle/product/19c/dbhome_1`
  - Not practical for typical environments

APEX | Upgrade APEX upfront

APEX upgrade

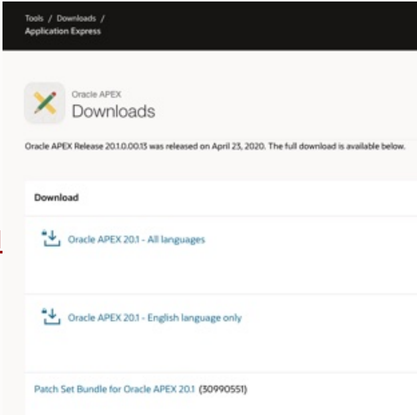
- Not part of the database upgrade
- MOS Note: 1088970.1 - Master Note APEX Upgrades

APEX certification

- Minimum APEX Version for Oracle 19c: APEX 18.2
- MOS Note: 1344948.1 - APEX Database and Web Server Certification

Download the newest APEX

- https://www.oracle.com/tools/downloads/apex-v191-downloads.html



Tools / Downloads /
Application Express

Oracle APEX
Downloads

Oracle APEX Release 20.1.0.00.13 was released on April 23, 2020. The full download is available below.

Download

Oracle APEX 20.1 - All languages

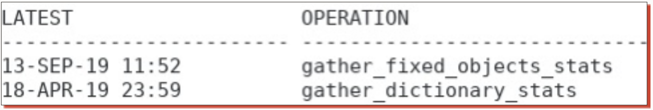
Oracle APEX 20.1 - English language only

Patch Set Bundle for Oracle APEX 20.1 (30990551)

Upgrade 19c | Speed it up

Check when dictionary stats have been gathered the last time

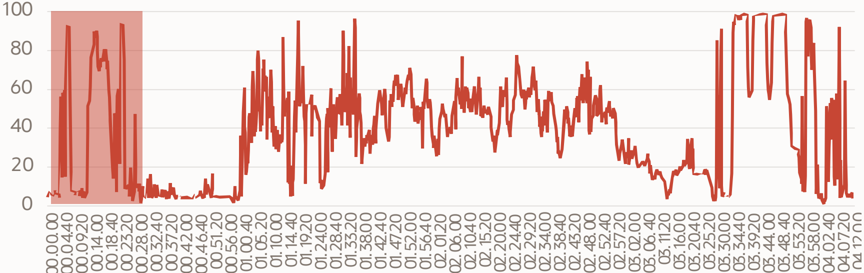
```
SELECT
   to_char(max(end_time),'dd-mon-yy hh24:mi') latest, operation
FROM
   dba_optstat_operations
WHERE
   operation in ('gather_dictionary_stats', 'gather_fixed_objects_stats')
GROUP BY
   operation;
```

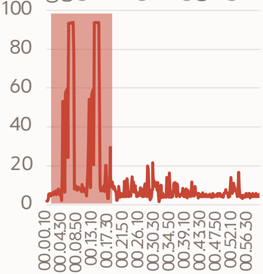


```
LATEST                   OPERATION
------------------------ ------------------------------
13-SEP-19 11:52          gather_fixed_objects_stats
18-APR-19 23:59          gather_dictionary_stats
```

Refresh stats a day before the upgrade

Upgrade 19c | Gather Stats In Advance

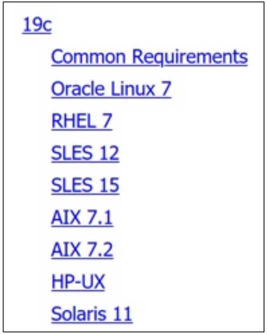




Gathering stats in advance saves 12 minutes

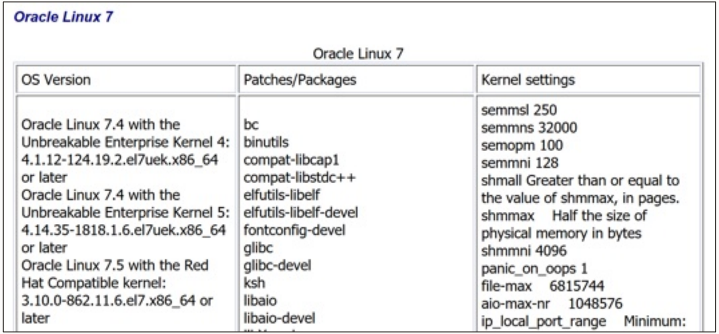
Operating System | Recommendations

Oracle Database (RDBMS) on Unix AIX,HP-UX,Linux,Solaris and MS Windows Operating Systems Installation and Configuration Requirements Quick Reference (12.1/12.2/18c/19c) (Doc ID 1587357.1)



19c

- Common Requirements
- Oracle Linux 7
- RHEL 7
- SLES 12
- SLES 15
- AIX 7.1
- AIX 7.2
- HP-UX
- Solaris 11



Oracle Linux 7

Oracle Linux 7

OS Version	Patches/Packages	Kernel settings
Oracle Linux 7.4 with the Unbreakable Enterprise Kernel 4: 4.1.12-124.19.2.el7uek.x86_64 or later Oracle Linux 7.4 with the Unbreakable Enterprise Kernel 5: 4.14.35-1818.1.6.el7uek.x86_64 or later Oracle Linux 7.5 with the Red Hat Compatible kernel: 3.10.0-862.11.6.el7.x86_64 or later	bc binutils compat-libcap1 compat-libstdc++ elfutils-libelf elfutils-libelf-devel fontconfig-devel glibc glibc-devel ksh libaio libaio-devel	semmsl 250 semmns 32000 semopm 100 semmni 128 shmall Greater than or equal to the value of shmmax, in pages. shmmax Half the size of physical memory in bytes shmmni 4096 panic_on_oops 1 file-max 6815744 aio-max-nr 1048576 ip_local_port_range Minimum:

For important databases, execute **database health checks** before upgrade

Health Checks

Health check script

- Download from MOS Note: 136697.1
- In Multitenant, it must be run in each PDB separately

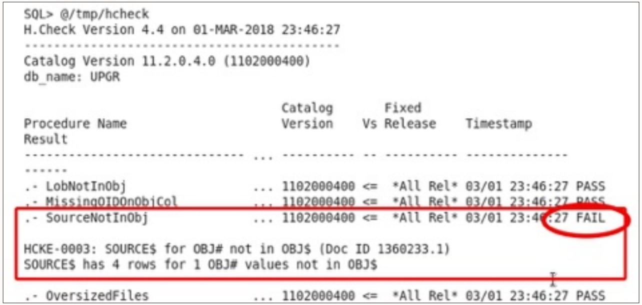
ORAchk Upgrade Readiness Assessment

- Part of Autonomous Health Framework (AHF)
- Download from MOS Note: 1457357.1
- Upgrade Readiness Check – MOS Note: 2550798.1

Health Check | hcheck.sql

If your database is highly important, do a health check

- Lightweight, non-intrusive script
- Checks consistency of selected dictionary relationships
- hcheck.sql - Script to Check for Known Problems (Doc ID 136697.1)



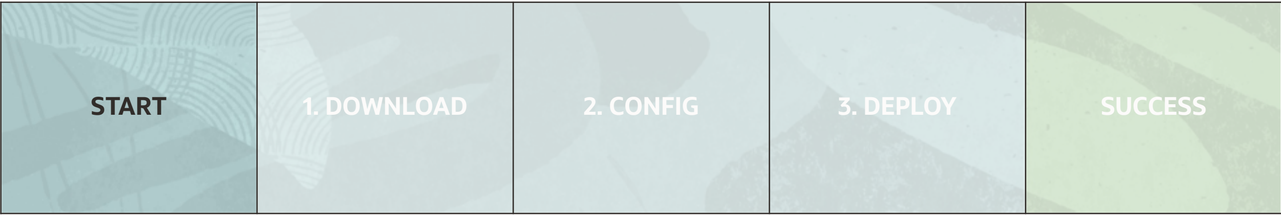
```
SQL> @/tmp/hcheck
H.Check Version 4.4 on 01-MAR-2018 23:46:27
----------------------------------------------
Catalog Version 11.2.0.4.0 (1102000400)
db_name: UPGR

                                    Catalog       Fixed
Procedure Name                      Version    Vs Release    Timestamp
Result
------------------------------ ... ---------- -- ---------- --------------
------
.- LobNotInObj                 ... 1102000400 <=  *All Rel* 03/01 23:46:27 PASS
.- MissingOIDOnObjCol          ... 1102000400 <=  *All Rel* 03/01 23:46:27 PASS
.- SourceNotInObj              ... 1102000400 <=  *All Rel* 03/01 23:46:27 FAIL

HCKE-0003: SOURCE$ for OBJ# not in OBJ$ (Doc ID 1360233.1)
SOURCE$ has 4 rows for 1 OBJ# values not in OBJ$

.- OversizedFiles              ... 1102000400 <=  *All Rel* 03/01 23:46:27 PASS
```

Photo by Joshua Sortino on Unsplash

AutoUpgrade

The ONLY recommended way to upgrade databases

Upgrade | AutoUpgrade - As Easy As 1-2-3



START

1. DOWNLOAD

2. CONFIG

3. DEPLOY

SUCCESS

Supported source releases

- 11.2.0.4
- 12.1.0.2
- 12.2.0.1
- 18
- 19

All architectures (CDB and non-CDB)
All supported operating systems
All editions (SE2, EE)
All types (single instance and RAC)

Upgrade | AutoUpgrade - As Easy As 1-2-3

START

1. DOWNLOAD

2. CONFIG

3. DEPLOY

SUCCESS

Download from My Oracle Support ID 2485457.1

Upgrade | AutoUpgrade - As Easy As 1-2-3

START

1. DOWNLOAD

2. CONFIG

3. DEPLOY

SUCCESS

Simple text file

```
upg1.source_home=/u01/app/oracle/product/12.2.0.1
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=CDB1
```

Upgrade | AutoUpgrade - As Easy As 1-2-3

START

1. DOWNLOAD

2. CONFIG

3. DEPLOY

SUCCESS

One command

```
$ java -jar autoupgrade.jar -config cdb1.cfg -mode deploy
```

Advanced monitoring and logging

Upgrade | AutoUpgrade - As Easy As 1-2-3

START

1. DOWNLOAD

2. CONFIG

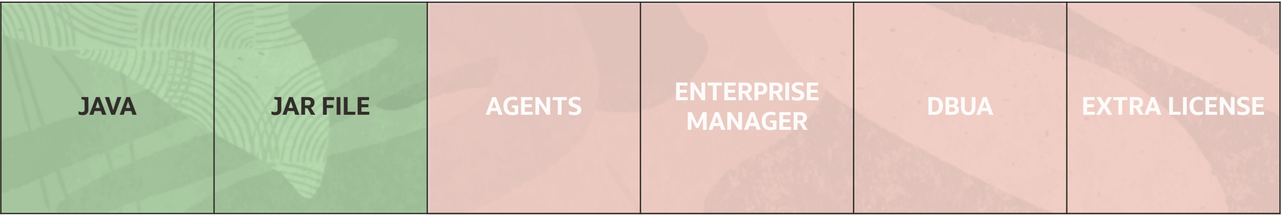
3. DEPLOY

SUCCESS

Supported target releases

- 12.2.0.1
- 18
- 19
- 21
- Any future release

AutoUpgrade | Need And Don't Need



JAVA

JAR FILE

AGENTS

ENTERPRISE MANAGER

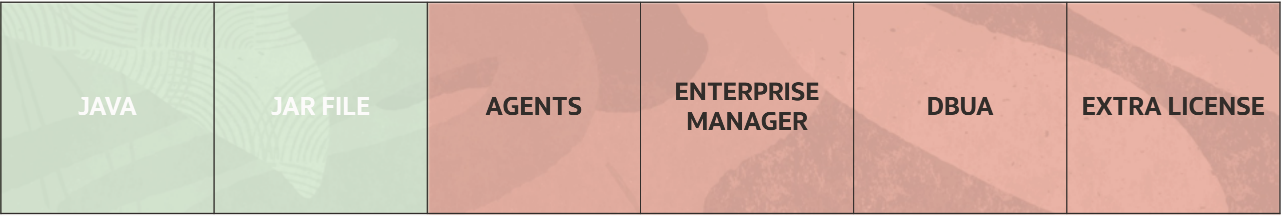
DBUA

EXTRA LICENSE

- Java 8 required
  - Part of Oracle Home since 12.1.0.2

- 3 MB jar file

AutoUpgrade | Need And Don't Need



JAVA

JAR FILE

AGENTS

ENTERPRISE MANAGER

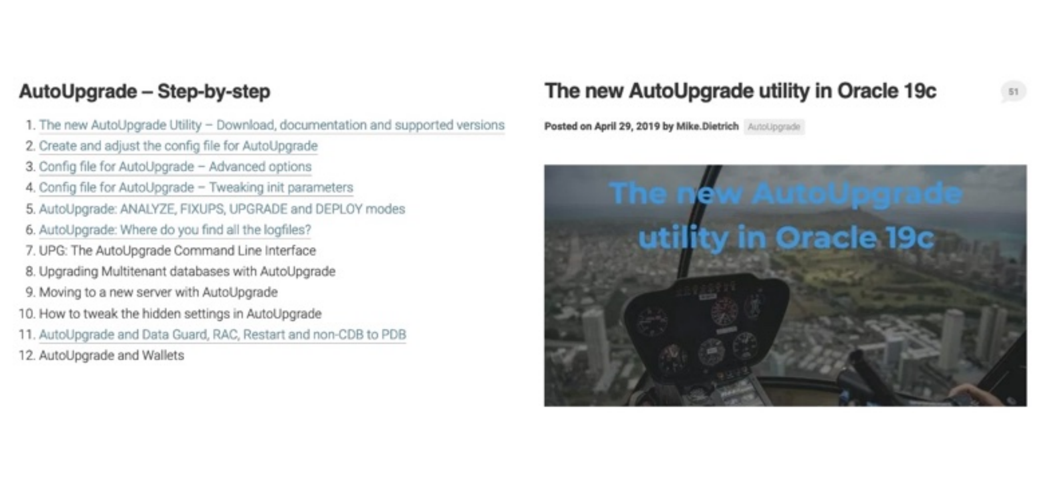
DBUA

EXTRA LICENSE

- No agents to install
- Enterprise Manager not needed
- AutoUpgrade offers superior functionality
- No extra license

AutoUpgrade | Blog Post Series

https://mikedietrichde.com/2019/04/29/the-new-autoupgrade-utility-in-oracle-19c/



AutoUpgrade – Step-by-step

1. The new AutoUpgrade Utility – Download, documentation and supported versions
2. Create and adjust the config file for AutoUpgrade
3. Config file for AutoUpgrade – Advanced options
4. Config file for AutoUpgrade – Tweaking init parameters
5. AutoUpgrade: ANALYZE, FIXUPS, UPGRADE and DEPLOY modes
6. AutoUpgrade: Where do you find all the logfiles?
7. UPG: The AutoUpgrade Command Line Interface
8. Upgrading Multitenant databases with AutoUpgrade
9. Moving to a new server with AutoUpgrade
10. How to tweak the hidden settings in AutoUpgrade
11. AutoUpgrade and Data Guard, RAC, Restart and non-CDB to PDB
12. AutoUpgrade and Wallets

The new AutoUpgrade utility in Oracle 19c

51

Posted on April 29, 2019 by Mike.Dietrich AutoUpgrade

AutoUpgrade Essentials

Photo by Daniel Cheung on Unsplash

AutoUpgrade | Essentials

Download

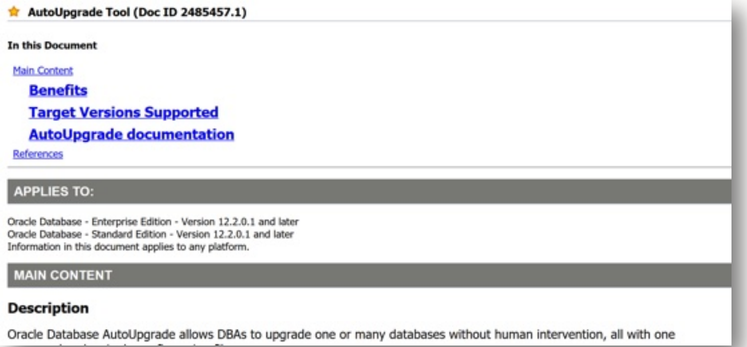
Configure

Analyze

Check

Upgrade

Always download latest version from MOS



☆ **AutoUpgrade Tool (Doc ID 2485457.1)**

In this Document

Main Content

Benefits

Target Versions Supported

AutoUpgrade documentation

References

APPLIES TO:

Oracle Database - Enterprise Edition - Version 12.2.0.1 and later
Oracle Database - Standard Edition - Version 12.2.0.1 and later
Information in this document applies to any platform.

MAIN CONTENT

Description

Oracle Database AutoUpgrade allows DBAs to upgrade one or many databases without human intervention, all with one

AutoUpgrade | Essentials

Download

Configure

Analyze

Check

Upgrade

Check your version

```
$ java -jar autoupgrade.jar -version

build.version 22.5.221011
build.date 2022/10/11 14:23:59 -0400
build.hash e9428661
build.hash_date 2022/10/11 12:55:45 -0400
build.supported_target_versions 12.2,18,19,21
build.type production
```

AutoUpgrade | Essentials

Download

Configure

Analyze

Check

Upgrade

AutoUpgrade handles older releases as well

```
$ java -jar autoupgrade.jar -version

build.version 22.5.221011
build.date 2022/10/11 14:23:59 -0400
build.hash e9428661
build.hash_date 2022/10/11 12:55:45 -0400
build.supported_target_versions 12.2,18,19,21
build.type production
```

AutoUpgrade | Essentials

- Download
- **Configure**
- Analyze
- Check
- Upgrade

Shortest possible config file version

```
upg1.source_home=/u01/app/oracle/product/12.2.0.1
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=CDB1
```

Or, generate a sample config file

```
$ java -jar autoupgrade.jar -create_sample_file config

Created sample configuration file /home/oracle/sample_config.cfg
```

Pro tip: `upg1` is a prefix that you decide.
Use it to define multiple databases

AutoUpgrade | Essentials

Download

Configure

Analyze

Check

Upgrade

Analyze your database

```
$ java -jar autoupgrade.jar -config CDB1.cfg -mode analyze

...

upg> Job 100 completed

Please check the summary report at:
/u01/app/oracle/cfgtoollogs/autoupgrade/cfgtoollogs/upgrade/auto/status/status.html
/u01/app/oracle/cfgtoollogs/autoupgrade/cfgtoollogs/upgrade/auto/status/status.log
```

Pro tip: Analyze is similar to running `preupgrade.jar`

AutoUpgrade | Essentials

Download

Configure

Analyze

Check

Upgrade

Summary report - text

```
==========================================
          Autoupgrade Summary Report
==========================================
[Date]           Tue Jan 12 10:26:19 CET 2021
[Number of Jobs] 1
==========================================
[Job ID] 100
==========================================
[DB Name]                CDB1
[Version Before Upgrade] 12.2.0.1.0
[Version After Upgrade]  19.9.0.0.0
------------------------------------------
[Stage Name]    PRECHECKS
[Status]        SUCCESS
[Start Time]    2021-01-12 10:25:58
[Duration]      0:00:20
[Log Directory] /u01/app/oracle/upg/CDB1/100/prechecks
[Detail]        /u01/app/oracle/upg/CDB1/100/prechecks/cdb1_preupgrade.log
                Precheck passed and no manual intervention needed
------------------------------------------
```

AutoUpgrade | Essentials

Download

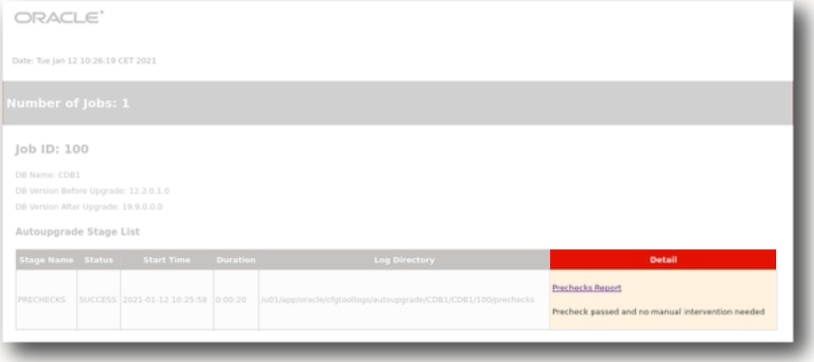
Configure

Analyze

Check

Upgrade

Summary report - HTML



ORACLE

Date: Tue Jan 12 10:26:19 CET 2021

Number of Jobs: 1

Job ID: 100

DB Name: CDB1

DB Version Before Upgrade: 12.2.0.1.0

DB Version After Upgrade: 19.9.0.0.0

Autoupgrade Stage List

Stage Name	Status	Start Time	Duration	Log Directory	Detail
PRECHECKS	SUCCESS	2021-01-12 10:25:58	0:00:20	/u01/app/oracle/cfgtoollogs/autoupgrade/CDB1/CDB1/100/prechecks	Prechecks Report Precheck passed and no manual intervention needed

AutoUpgrade | Essentials

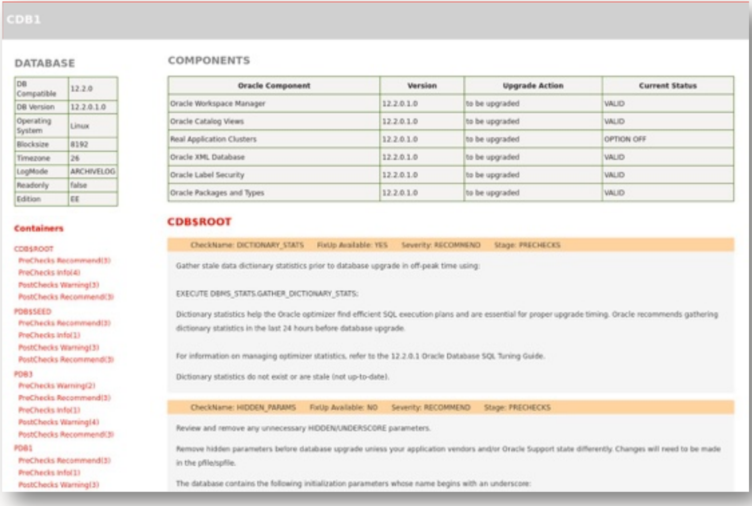
Download

Configure

Analyze

Check

Upgrade



CDB1

DATABASE

DB Compatible	12.2.0
DB Version	12.2.0.1.0
Operating System	Linux
Blocksize	8192
Timezone	26
LogMode	ARCHIVELOG
Readonly	false
Edition	EE

Containers

CDB$ROOT
- PreChecks Recommend(3)
- PreChecks Info(4)
- PostChecks Warning(3)
- PostChecks Recommend(3)

PDB$SEED
- PreChecks Recommend(3)
- PreChecks Info(1)
- PostChecks Warning(3)
- PostChecks Recommend(3)

PDB3
- PreChecks Warning(2)
- PreChecks Recommend(3)
- PreChecks Info(1)
- PostChecks Warning(4)
- PostChecks Recommend(3)

PDB1
- PreChecks Recommend(3)
- PreChecks Info(1)
- PostChecks Warning(3)

COMPONENTS

Oracle Component	Version	Upgrade Action	Current Status
Oracle Workspace Manager	12.2.0.1.0	to be upgraded	VALID
Oracle Catalog Views	12.2.0.1.0	to be upgraded	VALID
Real Application Clusters	12.2.0.1.0	to be upgraded	OPTION OFF
Oracle XML Database	12.2.0.1.0	to be upgraded	VALID
Oracle Label Security	12.2.0.1.0	to be upgraded	VALID
Oracle Packages and Types	12.2.0.1.0	to be upgraded	VALID

CDB$ROOT

CheckName: DICTIONARY_STATS FixUp Available: YES Severity: RECOMMEND Stage: PRECHECKS

Gather stale data dictionary statistics prior to database upgrade in off-peak time using:

EXECUTE DBMS_STATS.GATHER_DICTIONARY_STATS;

Dictionary statistics help the Oracle optimizer find efficient SQL execution plans and are essential for proper upgrade timing. Oracle recommends gathering dictionary statistics in the last 24 hours before database upgrade.

For information on managing optimizer statistics, refer to the 12.2.0.1 Oracle Database SQL Tuning Guide.

Dictionary statistics do not exist or are stale (not up-to-date).

CheckName: HIDDEN_PARAMS FixUp Available: NO Severity: RECOMMEND Stage: PRECHECKS

Review and remove any unnecessary HIDDEN/UNDERSCORE parameters.

Remove hidden parameters before database upgrade unless your application vendors and/or Oracle Support state differently. Changes will need to be made in the pfile/spfile.

The database contains the following initialization parameters whose name begins with an underscore:

AutoUpgrade | Essentials

Download
Configure
Analyze
Check
Upgrade

Preupgrade report comes in:

- HTML
- Text
- JSON

AutoUpgrade | Essentials

Download

Configure

Analyze

Check

Upgrade

Upgrade

```
$ java -jar autoupgrade.jar -config CDB1.cfg -mode deploy
```

Have a cup of coffee and wait, or ...

AutoUpgrade | Essentials

Download

Configure

Analyze

Check

Upgrade

Monitor

```
upg> lsj

+----+-------+---------+---------+-------+--------------+--------+---------------+
|Job#|DB_NAME|    STAGE|OPERATION| STATUS|    START_TIME| UPDATED|        MESSAGE|
+----+-------+---------+---------+-------+--------------+--------+---------------+
| 101|   CDB1|PREFIXUPS|EXECUTING|RUNNING|20/11/24 13:38|13:39:26|Remaining 12/13|
+----+-------+---------+---------+-------+--------------+--------+---------------+
```

AutoUpgrade | Essentials

Download

Configure

Analyze

Check

Upgrade

All the details

```
upg> status -job 101

Progress
-----------------------------------
Start time:      20/11/24 13:38
Elapsed (min):   13
Last update:     2020-11-24T13:48:52.139
Stage:           DBUPGRADE
Operation:       EXECUTING
Status:          RUNNING
Stage summary:
    SETUP                <1 min
    GRP                  <1 min
    PREUPGRADE           <1 min
    PRECHECKS            <1 min
    PREFIXUPS            8 min
    DRAIN                <1 min
    DBUPGRADE            3 min (IN PROGRESS)

Job Logs Locations
-----------------------------------
Logs Base:    /home/oracle/autoupg_default/CDB1/CDB1
Job logs:     /home/oracle/autoupg_default/CDB1/CDB1/101
Stage logs:   /home/oracle/autoupg_default/CDB1/CDB1/101/dbupgrade
TimeZone:     /home/oracle/autoupg_default/CDB1/CDB1/temp
```

AutoUpgrade | Essentials

Download
Configure
Analyze
Check
Upgrade

All the details - continued

```
...

Additional information
-----------------------------------
Details:
[Upgrading] is [0%] completed for [cdb1-cdb$root]
                 +---------+---------------+
                 |CONTAINER|     PERCENTAGE|
                 +---------+---------------+
                 | CDB$ROOT|   UPGRADE[12%]|
                 | PDB$SEED|UPGRADE PENDING|
                 |     PDB3|UPGRADE PENDING|
                 +---------+---------------+

Error Details:
None
```

AutoUpgrade | Essentials

Download

Configure

Analyze

Check

Upgrade

Success

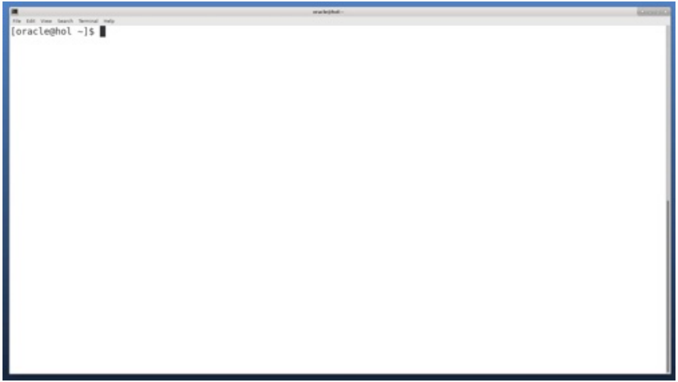
```
upg> Job 101 completed
------------------- Final Summary --------------------
Number of databases            [ 1 ]

Jobs finished successfully     [1]
Jobs failed                    [0]
Jobs pending                   [0]
------------- JOBS FINISHED SUCCESSFULLY -------------
Job 101 for CDB1

---- Drop GRP at your convenience once you consider it is no longer needed ----
Drop GRP from CDB1: drop restore point AUTOUPGRADE_9212_CDB1122010
```

And it includes:

- Recompilation (utlrp.sql)
- Time zone file upgrade
- Postupgrade fixups
- ... and so much more

AutoUpgrade | Essentials

Download
Configure
Analyze
Check
Upgrade



```
[oracle@hol ~]$ 
```

Watch on YouTube

AutoUpgrade | Essentials

One-liner using environment variables

```
export ORACLE_SID=CDB1
export ORACLE_HOME=/u01/app/oracle/product/12.2.0.1
export ORACLE_TARGET_HOME=/u01/app/oracle/product/19

java -jar autoupgrade.jar -config_values -mode analyze
```

One-liner using `config_values`

```
java -jar autoupgrade.jar \
   -config_values "sid=CDB1,source_home=/u01/app/oracle/product/12.2.0.1,target_home=/u01/app/oracle/product/19" \
   -mode analyze
```

AutoUpgrade Advanced Options

Photo by Ciprian Boiciuc on Unsplash

AutoUpgrade | Advanced Options

Many Databases

Different Servers

PFILE

Shell Scripts

Restore Point

Underscores

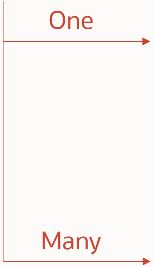
Recompilation

Time Zone

Parallel

Monitoring

Upgrade one or many databases



One

```
upg1.source_home=/u01/app/oracle/product/12.2.0.1
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=CDB1
```

Many

```
upg1.source_home=/u01/app/oracle/product/12.2.0.1
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=CDB1

upg2.source_home=/u01/app/oracle/product/11.2.0.4
upg2.target_home=/u01/app/oracle/product/19
upg2.sid=DB11204

...

upgn.source_home=/u01/app/oracle/product/12.1.0.2
upgn.target_home=/u01/app/oracle/product/19
upgn.sid=HR
```

Pro tip: You can also start multiple instances of AutoUpgrade at the same time

AutoUpgrade | Advanced Options

- Many Databases
- **Different Servers**
- PFILE
- Shell Scripts
- Restore Point
- Underscores
- Recompilation
- Time Zone
- Parallel
- Monitoring

Upgrade only when `upgrade_node` matches hostname

```
upg1.source_home=/u01/app/oracle/product/12.2.0.1
upg1.target_home=/u01/app/oracle/product/19
upg1.upgrade_node=test_server01.mycorp.net
upg1.sid=CDB1

upg2.source_home=/u01/app/oracle/product/12.2.0.1
upg2.target_home=/u01/app/oracle/product/19
upg2.upgrade_node=prod_server01.mycorp.net
upg2.sid=CDB2
```

- Database `upg1` will only be upgraded when AutoUpgrade gets executed on server `test_server01.mycorp.net`

AutoUpgrade | Advanced Options

- Many Databases
- Different Servers
- **PFILE**
- Shell Scripts
- Restore Point
- Underscores
- Recompilation
- Time Zone
- Parallel
- Monitoring

Update initialization parameters as part of the upgrade

You can:

- Add or remove parameters
- Before, during or after upgrade
- For a single or every database

AutoUpgrade | Advanced Options

- Many Databases
- Different Servers
- **PFILE**
- Shell Scripts
- Restore Point
- Underscores
- Recompilation
- Time Zone
- Parallel
- Monitoring

Remove a parameter during a specific upgrade

```
upg1.del_during_upgrade_pfile=/home/oracle/global_del_during.ora
```

```
# Example: global_del_during.ora

optimizer_features_enable
```

Add parameters to all databases after upgrade

```
global.add_after_upgrade_pfile=/home/oracle/global_add_after.ora
```

```
# Example: global_add_after.ora

deferred_segment_creation=false
_cursor_obsolete_threshold=1024
_sql_plan_directive_mgmt_control=0
_use_single_log_writer=true
```

AutoUpgrade | Advanced Options

- Many Databases
- Different Servers
- **PFILE**
- Shell Scripts
- Restore Point
- Underscores
- Recompilation
- Time Zone
- Parallel
- Monitoring

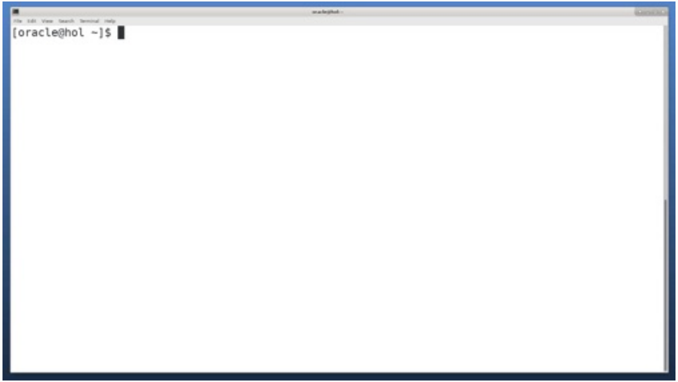
Batch-update parameters

```
global.del_during_upgrade_pfile=/home/oracle/global_del_during.ora
global.add_during_upgrade_pfile=/home/oracle/global_add_during.ora
global.del_after_upgrade_pfile=/home/oracle/global_del_during.ora
global.add_after_upgrade_pfile=/home/oracle/global_add_after.ora

upg1.source_home=/u01/app/oracle/product/12.2.0.1
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=CDB1
upg1.add_after_upgrade_pfile=/home/oracle/upg1_add_after.ora

upg2.source_home=/u01/app/oracle/product/12.2.0.1
upg2.target_home=/u01/app/oracle/product/19
upg2.sid=CDB2
upg2.add_after_upgrade_pfile=/home/oracle/upg2_add_after.ora
```

```
# Example: global_add_after.ora

deferred_segment_creation=false
_cursor_obsolete_threshold=1024
_sql_plan_directive_mgmt_control=0
_use_single_log_writer=true
```

AutoUpgrade | Advanced Options

Many Databases
Different Servers
PFILE
Shell Scripts
Restore Point
Underscores
Recompilation
Time Zone
Parallel
Monitoring



```
[oracle@hol ~]$
```

Watch on YouTube

AutoUpgrade | Advanced Options

Many Databases
Different Servers
PFILE
Shell Scripts
Restore Point
Underscores
Recompilation
Time Zone
Parallel
Monitoring

Execute your own scripts as part of the upgrade

You can:

- Before and after upgrade
- Halt or continue on error
- For a single or every database

Ideas:

- Enterprise Manager configuration
- Backup configuration
- Interact with apps using the database

AutoUpgrade | Advanced Options

Many Databases
Different Servers
PFILE
Shell Scripts
Restore Point
Underscores
Recompilation
Time Zone
Parallel
Monitoring

Shell script execution

```
global.before_action=/database/scripts/set_blackout.sh

upg1.source_home=/u01/app/oracle/product/12.2.0.1
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=CDB1
upg1.after_action=/database/scripts/start_level0.sh
```

- Permitted extension options:
  - Unix shell (.sh)
  - Microsoft Windows batch (.bat, .cmd)
  - Microsoft Windows PowerShell (.ps1)

Pro tip: If you want script execution for all upgrades use `global.before_action` and `global.after_action`

AutoUpgrade | Advanced Options

Many Databases
Different Servers
PFILE
Shell Scripts
Restore Point
Underscores
Recompilation
Time Zone
Parallel
Monitoring

Shell script execution

Default - AutoUpgrade does not react on return code

```
upg1.before_action=/database/scripts/run_this_on_UPG1_before.sh
```

Optionally - AutoUpgrade halts on non-zero return code

```
upg1.before_action=/database/scripts/run_this_on_UPG1_before.sh Y
```

Pro tip: Script output is captured and stored in *preupgrade* and *postupgrade* directory

AutoUpgrade | Advanced Options

Many Databases
Different Servers
PFILE
Shell Scripts
Restore Point
Underscores
Recompilation
Time Zone
Parallel
Monitoring

Environment variables:

- ORACLE_SID
- ORACLE_UNQNAME
- ORACLE_BASE
- ORACLE_HOME
- TNS_ADMIN

AutoUpgrade | Advanced Options

- Many Databases
- Different Servers
- PFILE
- Shell Scripts
- **Restore Point**
- Underscores
- Recompilation
- Time Zone
- Parallel
- Monitoring

Guaranteed Restore Points

```
upg1.source_home=/u01/app/oracle/product/12.2.0.1
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=CDB1
upg1.restoration=no

upg2.source_home=/u01/app/oracle/product/12.2.0.1
upg2.target_home=/u01/app/oracle/product/19
upg2.sid=CDB2
upg2.drop_grp_after_upgrade=yes
```

- Default behavior:
  - AutoUpgrade creates GRP except for
    - Standard Edition 2
    - `restoration=no`
  - GRP will be kept
  - GRP needs to be removed manually except for
    - `drop_grp_after_upgrade=yes` will only remove it when upgrade completed successfully

AutoUpgrade | Advanced Options

Many Databases
Different Servers
PFILE
Shell Scripts
Restore Point
Underscores
Recompilation
Time Zone
Parallel
Monitoring

Underscore parameters and events

```
upg1.source_home=/u01/app/oracle/product/12.2.0.1
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=CDB1
upg1.remove_underscore_parameters=yes
```

- Default behavior:
  - Underscores and events will be kept

AutoUpgrade | Advanced Options

Many Databases
Different Servers
PFILE
Shell Scripts
Restore Point
Underscores
Recompilation
Time Zone
Parallel
Monitoring

AutoUpgrade recompiles invalid Oracle-maintained objects after the upgrade

- To postpone the recompilation:

```
upg1.source_home=/u01/app/oracle/product/12.2.0.1
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=CDB1
upg1.run_utlrp=no
```

AutoUpgrade | Advanced Options

Many Databases
Different Servers
PFILE
Shell Scripts
Restore Point
Underscores
Recompilation
Time Zone
Parallel
Monitoring

During multitenant upgrades AutoUpgrade:

- Recompiles in many PDBs at the same time (`CPU_COUNT`/3)
- Recompilation in a PDB runs with three threads
- Recompilation is very CPU intensive

AutoUpgrade | Advanced Options

- Many Databases
- Different Servers
- PFILE
- Shell Scripts
- Restore Point
- Underscores
- **Recompilation**
- Time Zone
- Parallel
- Monitoring

```
$ sar -u 10 10

01:08:34 PM     CPU     %user     %nice   %system   %iowait    %steal     %idle
01:08:44 PM     all     95.09      0.00      2.18      0.01      0.00      2.72
01:08:54 PM     all     96.62      0.00      2.14      0.01      0.00      1.23
01:09:04 PM     all     96.75      0.00      2.30      0.03      0.00      0.92
01:09:14 PM     all     96.31      0.00      3.14      0.00      0.00      0.55
01:09:24 PM     all     95.72      0.03      4.07      0.00      0.00      0.18
01:09:34 PM     all     97.84      0.00      1.87      0.00      0.00      0.28
01:09:44 PM     all     97.12      0.00      2.06      0.01      0.00      0.81
01:09:54 PM     all     95.67      0.00      1.85      0.01      0.00      2.47
01:10:04 PM     all     95.39      0.00      2.95      0.01      0.00      1.65
01:10:14 PM     all     95.23      0.00      2.46      0.00      0.00      2.31
Average:        all     96.17      0.00      2.50      0.01      0.00      1.31
```

AutoUpgrade | Advanced Options

Many Databases
Different Servers
PFILE
Shell Scripts
Restore Point
Underscores
Recompilation
Time Zone
Parallel
Monitoring

Two new *tune settings* to control recompilation

- `utlrp_pdb_in_parallel`
- `utlrp_threads_per_pdb`

Example:

```
upg1.tune_setting=utlrp_pdb_in_parallel=3,utlrp_threads_per_pdb=4
```

AutoUpgrade will recompile:

- Three PDBs at a time
- Use four threads per PDB

CPU consumption will use a maximum of 12 cores

AutoUpgrade | Advanced Options

- Many Databases
- Different Servers
- PFILE
- Shell Scripts
- Restore Point
- Underscores
- Recompilation
- **Time Zone**
- Parallel
- Monitoring

Skip time zone upgrade

```
upg1.source_home=/u01/app/oracle/product/12.2.0.1
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=CDB1
upg1.timezone_upg=no
```

- Default behavior:
  - Time zone adjustment happens post upgrade
  - Database will be restarted several times
  - Important when you use "Downgrade" as fallback strategy as time zone can't be downgraded

AutoUpgrade | Advanced Options

- Many Databases
- Different Servers
- PFILE
- Shell Scripts
- Restore Point
- Underscores
- Recompilation
- Time Zone
- **Parallel**
- Monitoring

CDB

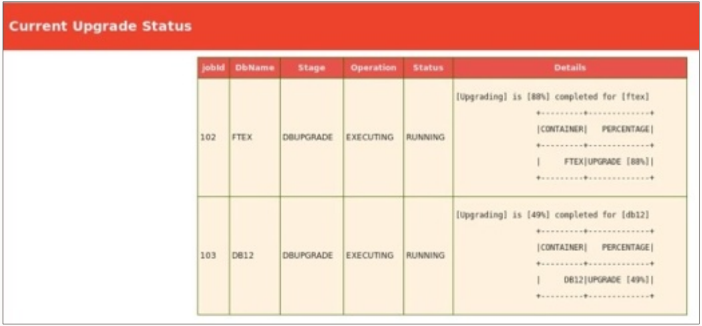
```
upg1.catctl_options=-n 64 -N 8
```

-n	Total number of parallel processes (min 4, max unlimited, default `CPU_COUNT`)
-N	Number of parallel processes per PDB (min 1, max 8, default 2)

Concurrent PDB upgrades: n / N

AutoUpgrade | Advanced Options

- Many Databases
- Different Servers
- PFILE
- Shell Scripts
- Restore Point
- Underscores
- Recompilation
- Time Zone
- Parallel
- **Monitoring**



Current Upgrade Status

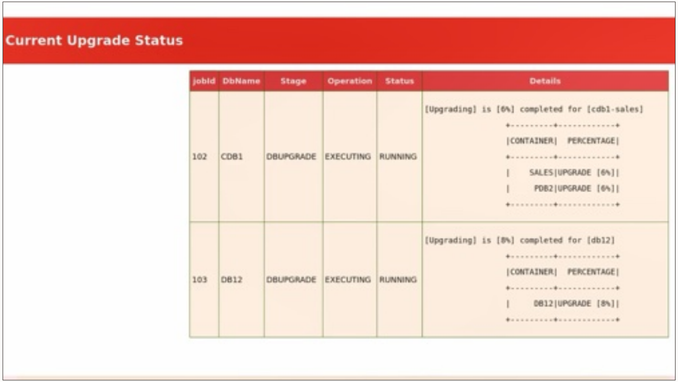
<table>
<tr><th>jobId</th><th>DbName</th><th>Stage</th><th>Operation</th><th>Status</th><th>Details</th></tr>
<tr><td>102</td><td>FTEX</td><td>DBUPGRADE</td><td>EXECUTING</td><td>RUNNING</td><td>[Upgrading] is [88%] completed for [ftex]
+---------+-------------+
|CONTAINER| PERCENTAGE|
+---------+-------------+
| FTEX|UPGRADE [88%]|
+---------+-------------+</td></tr>
<tr><td>103</td><td>DB12</td><td>DBUPGRADE</td><td>EXECUTING</td><td>RUNNING</td><td>[Upgrading] is [49%] completed for [db12]
+---------+-------------+
|CONTAINER| PERCENTAGE|
+---------+-------------+
| DB12|UPGRADE [49%]|
+---------+-------------+</td></tr>
</table>

Monitor via browser:

`<au_global_log_dir>/cfgtoollogs/upgrade/auto/state.html`

Refreshes automatically

AutoUpgrade | Advanced Options – Monitoring Demo

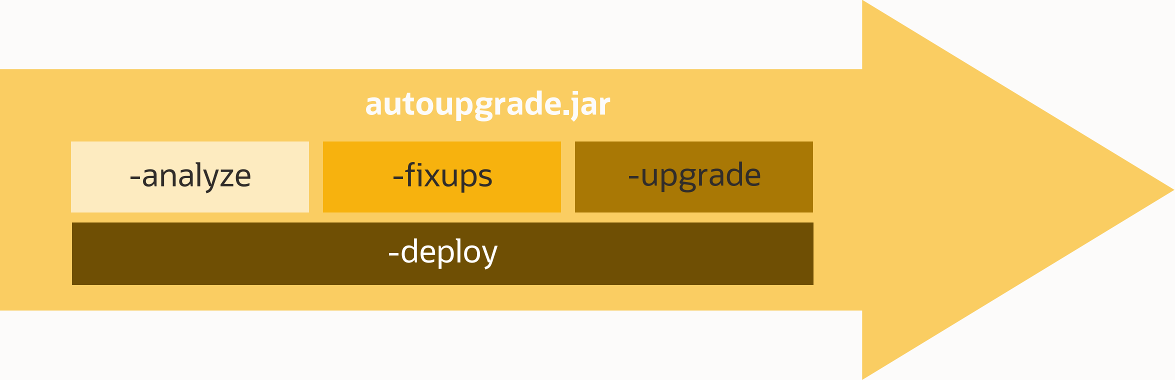


Current Upgrade Status

<table>
<tr><th>jobId</th><th>DbName</th><th>Stage</th><th>Operation</th><th>Status</th><th>Details</th></tr>
<tr><td>102</td><td>CDB1</td><td>DBUPGRADE</td><td>EXECUTING</td><td>RUNNING</td><td>[Upgrading] is [6%] completed for [cdb1-sales]
+---------+------------+
|CONTAINER| PERCENTAGE|
+---------+------------+
| SALES|UPGRADE [6%]|
| PDB2|UPGRADE [6%]|
+---------+------------+</td></tr>
<tr><td>103</td><td>DB12</td><td>DBUPGRADE</td><td>EXECUTING</td><td>RUNNING</td><td>[Upgrading] is [8%] completed for [db12]
+---------+------------+
|CONTAINER| PERCENTAGE|
+---------+------------+
| DB12|UPGRADE [8%]|
+---------+------------+</td></tr>
</table>

Watch on YouTube

AutoUpgrade | 4 Operation Modes



autoupgrade.jar

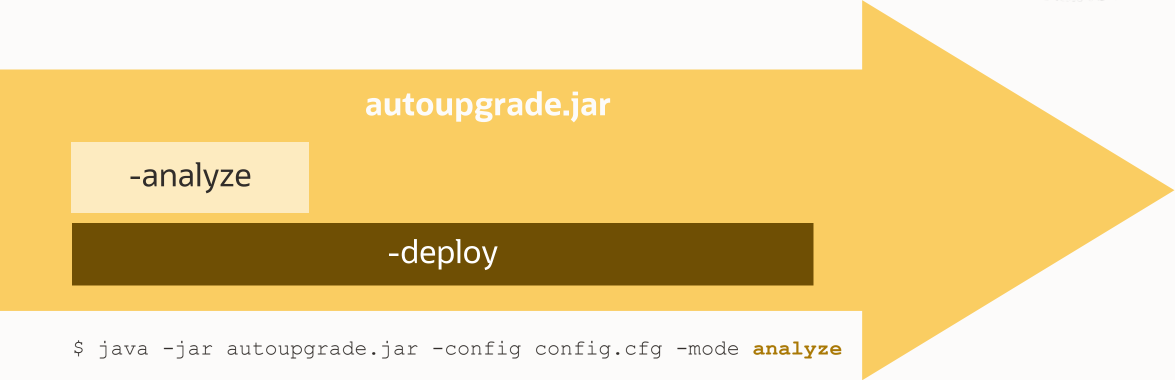
-analyze

-fixups

-upgrade

-deploy

AutoUpgrade | Best Practice



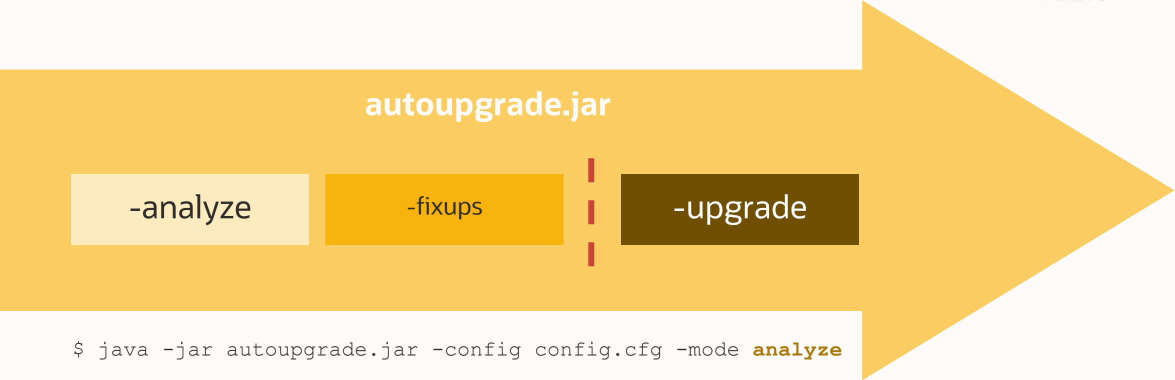
autoupgrade.jar

-analyze

-deploy

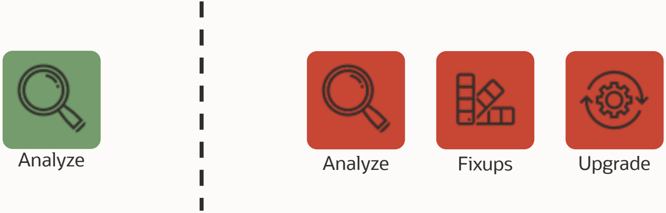
```
$ java -jar autoupgrade.jar -config config.cfg -mode analyze

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

AutoUpgrade | Fast Deploy



```
$ java -jar autoupgrade.jar -config config.cfg -mode analyze
$ java -jar autoupgrade.jar -config config.cfg -mode fixups
$ java -jar autoupgrade.jar -config config.cfg -mode upgrade
```

AutoUpgrade | Traditional



Analyze

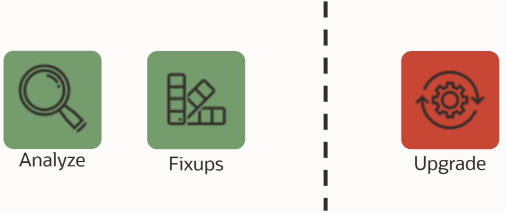
Analyze

Fixups

Upgrade

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

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

AutoUpgrade | Fast Deploy



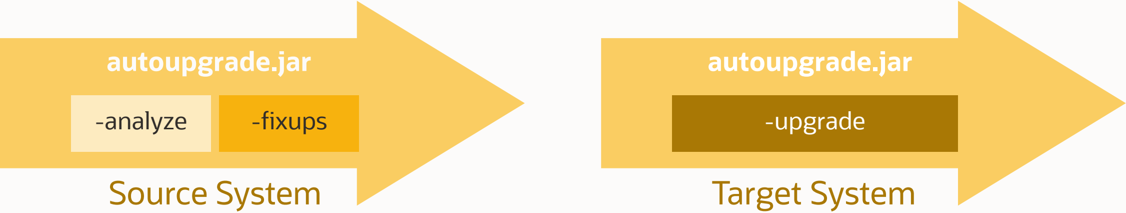
```
$ java -jar autoupgrade.jar -mode analyze
$ java -jar autoupgrade.jar -mode fixups

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

AutoUpgrade | Fast Deploy

⚠ Between fixups and downtime there is a risk that new, undetected issues are introduced

Pro tip: Blog post with more details

AutoUpgrade | Move to new hardware



```
$ java -jar autoupgrade.jar -config config.cfg -mode analyze

$ java -jar autoupgrade.jar -config config.cfg -mode fixups

$ java -jar autoupgrade.jar -config config.cfg -mode upgrade
```

AutoUpgrade | Job Console

- Most important commands

```
upg>

lsj                      // List jobs
resume  -job <number>    // Restarts a job
status  -job <number>    // Show job status
restore -job <number>    // Restores database from GRP
abort   -job <number>    // Aborts the specified job
```

AutoUpgrade | Log File Structure

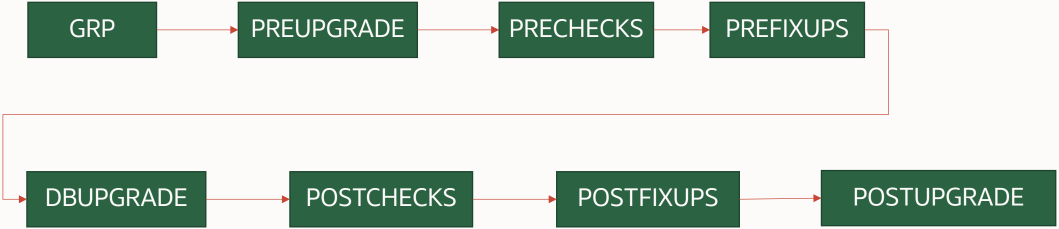
- Logs written in TEXT and JSON format
  - `/cfgtoollogs`
    - `./upgrade/auto` ◀ Status Log
  - `/database_1`
    - `./job_number`
      - `./prechecks` ◀ HTML Report
      - `./preupgrade`
      - `./prefixups`
      - `./drain`
      - `./dbupgrade` ◀ Upgrade Logs
      - `./postupgrade`
    - `./temp`
  - `/database_2`
    - ...

Proactive Fixups result in
faster upgrades of CDBs with many PDBs

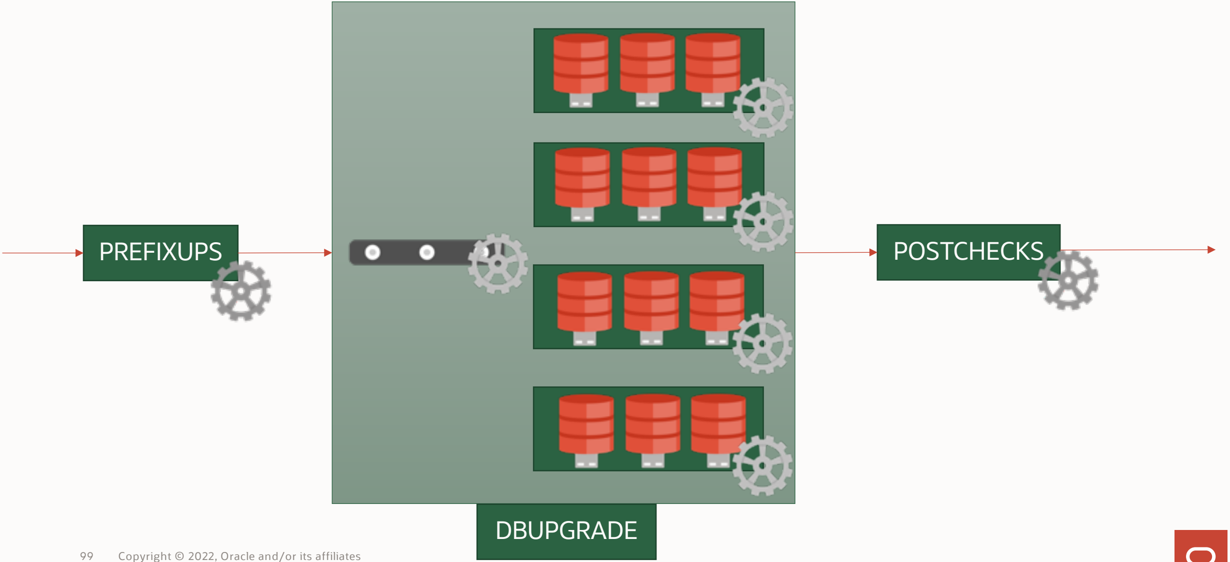
Proactive Fixups | What is it?

- Performance feature
- Changes only the order of the tasks of AutoUpgrade workflow
- Isolates errors in PDBs
- Valid for CDB upgrades only

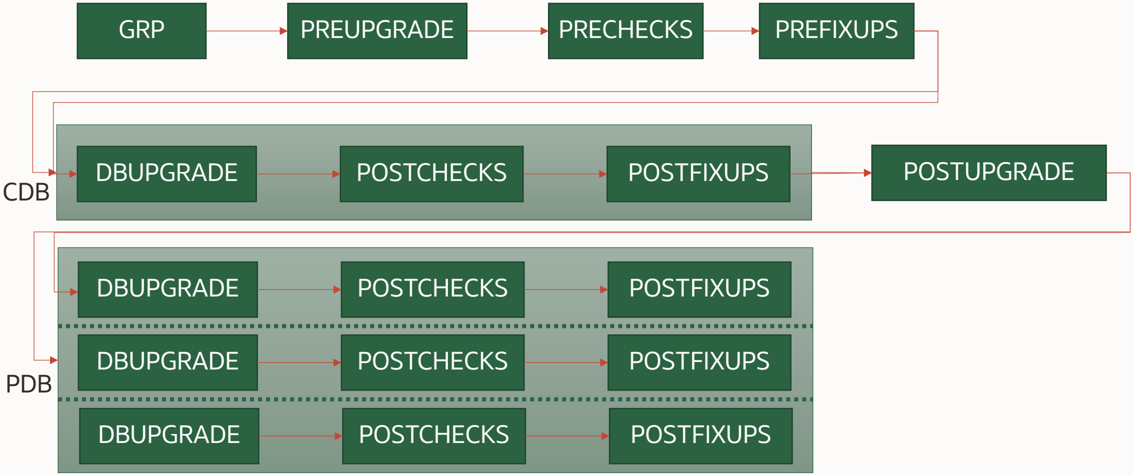
Proactive Fixups | Classic Flow



Proactive Fixups | Classic Flow

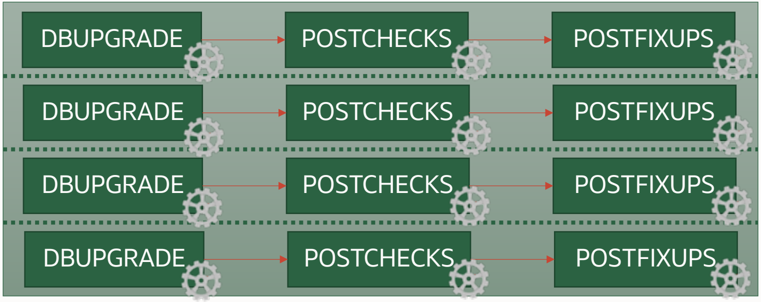


Proactive Fixups | New Flow



Proactive Fixups | New Flow

PDBSUPG STAGE



Proactive Fixups | New Flow

PDBSUPG STAGE

```
Stage-Progress Per Container

        +--------+---------+--------+
        |Database|    Stage|Progress|
        +--------+---------+--------+
        |PDB$SEED| DBUPGRADE|   91 %|
        |   PDB01|POSTFIXUPS|    0 %|
        |   PDB02| DBUPGRADE|   20 %|
        |   PDB03|POSTFIXUPS|   25 %|
        |   PDB04|POSTFIXUPS|   75 %|
        |   PDB05|POSTFIXUPS|   10 %|
        |   PDB06| DBUPGRADE|    6 %|
        |   PDB07| DBUPGRADE|   91 %|
        |   PDB08| DBUPGRADE|   91 %|
        |   PDB09| DBUPGRADE|   91 %|
        +--------+---------+--------+
```

Proactive Fixups | Gain

4 PDBs + ROOT | 4 Cores

Default **Proactive Fixups**

```
INFO      PREUPGRADE        <1 min     | INFO      PREUPGRADE        <1 min
INFO      PRECHECKS          1 min     | INFO      PRECHECKS          1 min
INFO      PREFIXUPS          8 min     | INFO      PREFIXUPS          7 min
INFO      DRAIN             <1 min     | INFO      DRAIN             <1 min
INFO      DBUPGRADE        143 min     | INFO      DBUPGRADE        130 min
INFO      POSTCHECKS         2 min     | INFO      POSTCHECKS        <1 min
INFO      POSTFIXUPS        34 min     | INFO      POSTFIXUPS        <1 min
INFO      POSTUPGRADE        1 min     | INFO      POSTUPGRADE        1 min
                                       |
                     TOTAL  179 min    |                     TOTAL  130 min
```

Proactive Fixups | Gain

16 PDBs + ROOT | 8 Cores | Defaults

Default

Proactive Fixups

```
INFO     PREUPGRADE        <1 min     | INFO     PREUPGRADE        <1 min
INFO     PRECHECKS         <1 min     | INFO     PRECHECKS         <1 min
INFO     PREFIXUPS         <1 min     | INFO     PREFIXUPS         14 min
INFO     DRAIN              2 min     | INFO     DRAIN              2 min
INFO     DBUPGRADE        210 min     | INFO     DBUPGRADE        195 min
INFO     POSTCHECKS         3 min     | INFO     POSTCHECKS        <1 min
INFO     POSTFIXUPS        46 min     | INFO     POSTFIXUPS        <1 min
INFO     POSTUPGRADE       <1 min     | INFO     POSTUPGRADE        1 min

                    TOTAL 259 min     |                    TOTAL 195 min
```

👍 The more PDBs, the greater the benefit

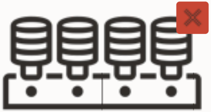
👍 Control the order of the PDBs using config file entry `pdbs`

👍

Proactive Fixups isolates each PDB
Errors in a PDB does not affect others

Proactive Fixups | Isolation

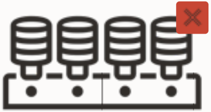
DEFAULT



Error in a PDB upgrade:

- Entire job halts
- Job can't complete

PROACTIVE FIXUPS



Error in a PDB upgrade:

- Other upgrades continue
- Job completes

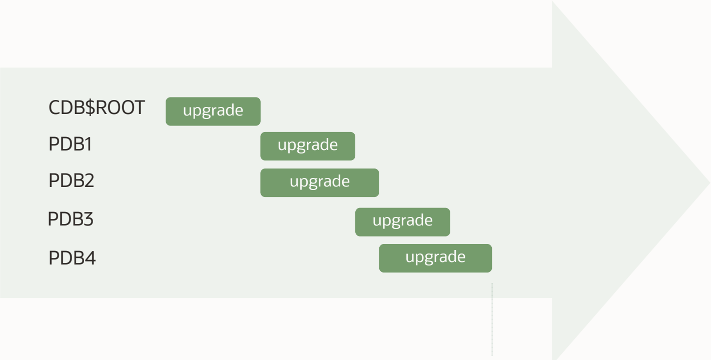
⚠

Restore point protects on CDB level only
Only entire CDB can be flashed back

Proactive Fixups | Availability

DEFAULT

`make_pdbs_available=false`

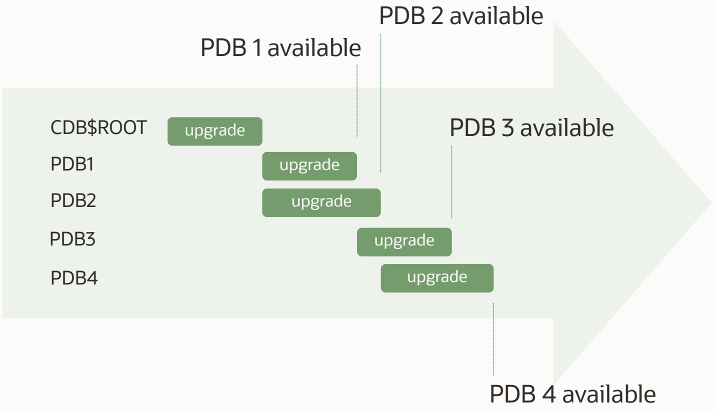


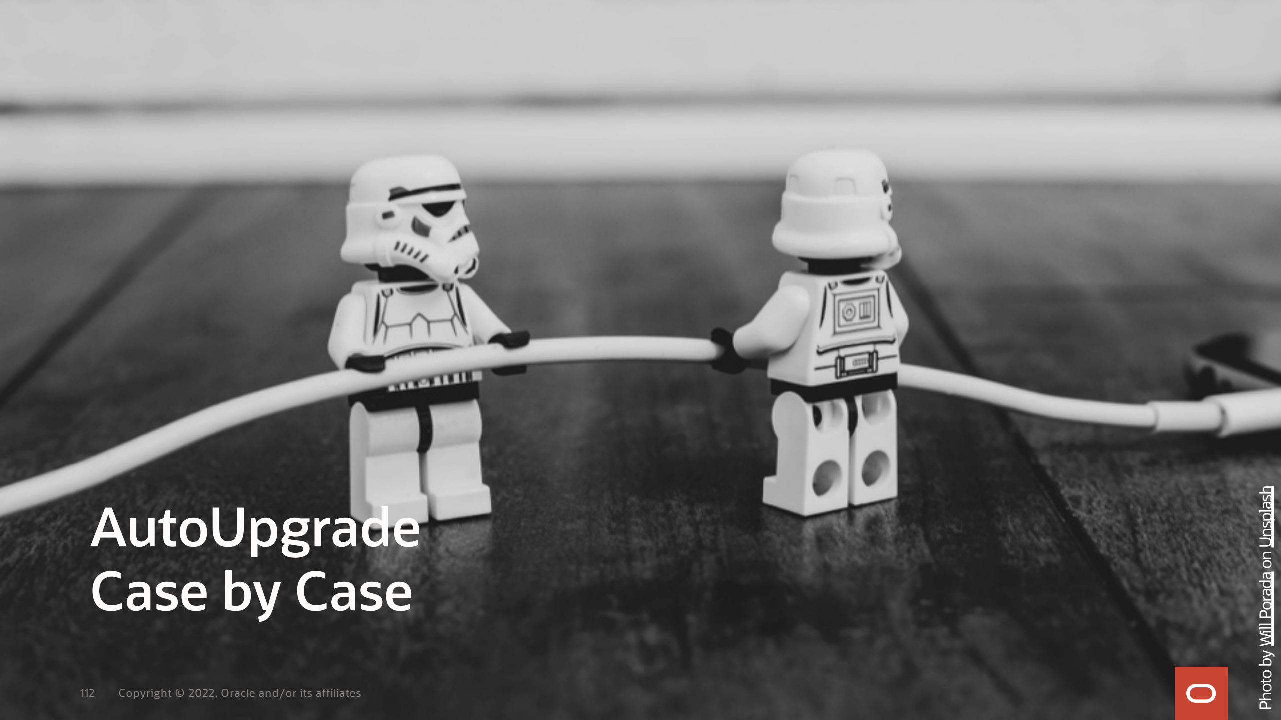
All PDBs available

Proactive Fixups | Availability

IMMEDIATELY AVAILABLE

`make_pdbs_available=true`





AutoUpgrade Case by Case

Photo by Will Porada on Unsplash

Photo by Katarzyna Pe on Unsplash

Patching

We made upgrading easy.
Now we make patching just as easy.

AutoUpgrade functionality extended to patching

1

Install Oracle Home including Release Update and additional patches (MOS Doc ID 555.1)

2

Create a simple configuration file

3

Start AutoUpgrade in deploy mode

```
$ cat DB19.cfg

patch1.source_home=/u01/app/oracle/product/19.0.0.0/dbhome_19_15_0
patch1.target_home=/u01/app/oracle/product/19.0.0.0/dbhome_19_16_0
patch1.sid=DB19
```

```
$ java -jar autoupgrade.jar -config DB19.cfg -mode deploy
```

USE

Familiar interface

Console

Logging

ANALYZE

Prechecks

Summary report

PROTECT

Resumable

Restoration

Restore point

Fallback

AUTOMATE

`srvctl`

`/etc/oratab`

Files

Datapatch

Patching ❤

Encryption

Hot clone

Refreshable clone

RAC

Proactive fixups

Distributed upgrade

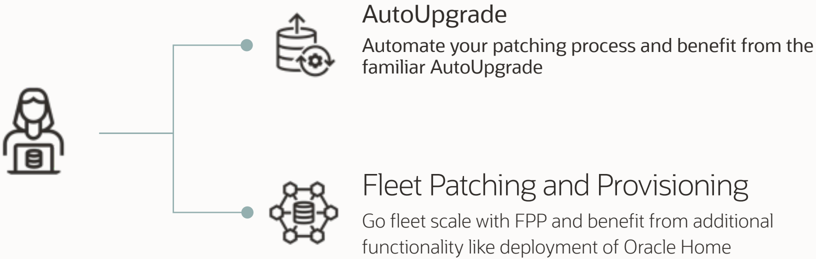
...

What's missing

Windows

RAC rolling

Data Guard standby-first



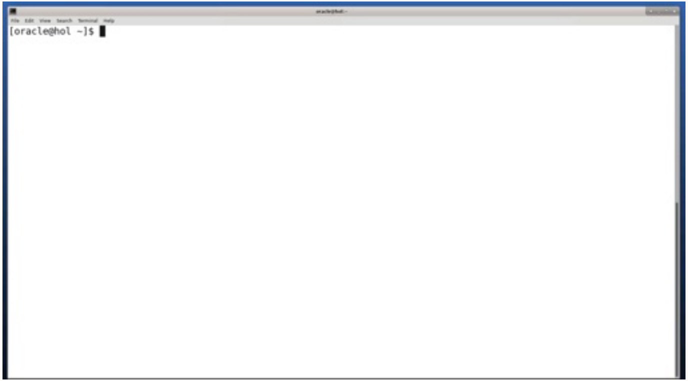
AutoUpgrade

Automate your patching process and benefit from the familiar AutoUpgrade

Fleet Patching and Provisioning

Go fleet scale with FPP and benefit from additional functionality like deployment of Oracle Home

Patching from 19.14.0 to 19.16.0



```
[oracle@hol ~]$
```

Watch on YouTube

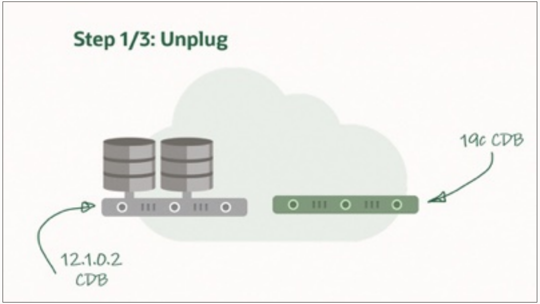
Photo by Katarzyna Pe on Unsplash

Unplug / Plug / Upgrade

AutoUpgrade | Unplug-plug Upgrade

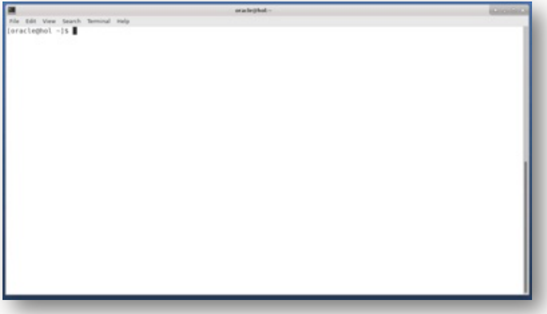
Upgrade a single PDB

- Faster
- More flexible
- Requires compatible target CDB
- Not compatible with Flashback Database
  - Consider using Refreshable PDBs
  - Copy data files (`target_pdb_copy_option`)



AutoUpgrade | Unplug-plug Upgrade

```
upg1.sid=CDB12102
upg1.target_cdb=CDB19
upg1.pdbs=pdb1
upg1.source_home=/u01/app/oracle/product/12102
upg1.target_home=/u01/app/oracle/product/19
```



```
[oracle@hol ~]$
```

Watch on YouTube

AutoUpgrade | Unplug-plug Upgrade

Upgrade several PDBs

```
upg1.pdbs=pdb1,pdb2,pdb3
```

Rename a PDB

```
upg1.pdbs=pdb1
upg1.target_pdb_name.pdb1=sales
```

Copy data files on plug-in

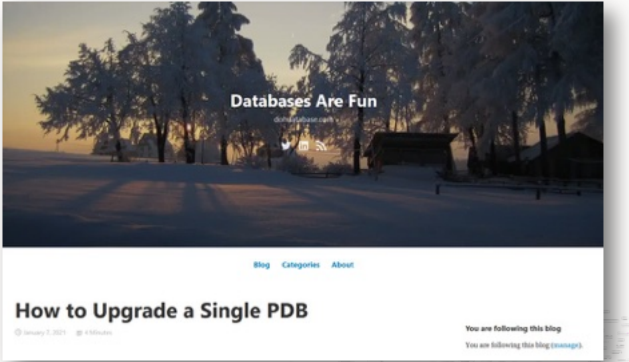
```
upg1.pdbs=pdb1
upg1.target_pdb_copy_option.pdb1=file_name_convert=('pdb1','sales')
```

AutoUpgrade | Unplug-plug Upgrade

Current limitations:

- Does not support Data Guard
- Does not support TDE Tablespace Encryption

AutoUpgrade | Unplug-plug Upgrade



Databases Are Fun

dohdatabase.com

How to Upgrade a Single PDB

January 7, 2021 4 Minutes

You are following this blog

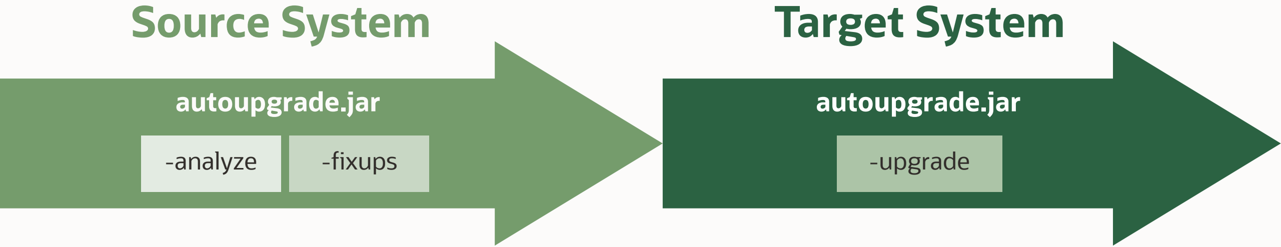
You are following this blog (manage).

https://dohdatabase.com/how-to-upgrade-a-single-pdb

Photo by Hello I'm Nik 🎞 on Unsplash

AutoUpgrade to a New Server

Upgrade to a new server | Overview



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

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

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

Upgrade to a new server | Details

Source Server

```
upg1.source_home=/u01/app/oracle/product/12
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=DB12
```

- `-mode analyze`
- `-mode fixups`
- `shutdown immediate`

Copy database including redo logs, control files, SPFILE, password file

Target Server

- Update `/etc/oratab`
- Prepare `?/network/admin` files

- `STARTUP UPGRADE`

```
upg1.source_home=/tmp
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=DB12
```

- `-mode upgrade`

Pro tip: Find more details in blog post
[Oracle AutoUpgrade between two servers]

Upgrade to a new server | Demo

Watch on YouTube

Photo by Danilo Alvesd on Unsplash

Plugin-Only with AutoUpgrade

AutoUpgrade | Plugin AFTER Upgrade

Plug in non-CDB into an existing CDB without upgrading

- Non-CDB is upgraded already
- Config file

```
upg1.source_home=/u01/app/oracle/product/19
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=DB12
upg1.target_cdb=CDB2
```

- AutoUpgrade

```
java -jar autoupgrade.jar -config DB12NEW.cfg -mode deploy
```

- Oracle AutoUpgrade between two servers – and Plugin?

PDB Convert | Re-use data files

Fully automated plug-in, re-use data files

```
upg1.source_home=/u01/app/oracle/product/19
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=DB19
upg1.target_cdb=CDB2
```

Command

```
java -jar autoupgrade.jar -config DB19.cfg -mode deploy
```

PDB Convert | Copy data files

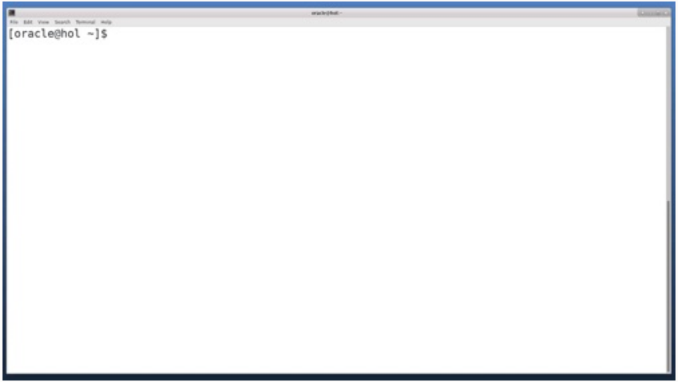
Fully automated plug-in, copy data files

```
upg1.source_home=/u01/app/oracle/product/19
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=DB19
upg1.target_cdb=CDB2
upg1.target_pdb_name=SALES
#Copy files and perform search/replace on file names
upg1.target_pdb_copy_option=file_name_convert=('DB19','SALES')
#Copy files and generate new OMF file names
upg1.target_pdb_copy_option=file_name_convert=none
```

Command

```
Java -jar autoupgrade.jar -config DB19.cfg -mode deploy
```

PDB Convert | Demo



```
[oracle@hol ~]$
```

Watch on YouTube

AutoUpgrade with Data Guard

AutoUpgrade supports upgrading databases that are part of a Data Guard configuration

AutoUpgrade | Data Guard

- Detected by AutoUpgrade automatically
- Works for broker-managed and manual Data Guard environments
- Primary database handled by AutoUpgrade
  Standby database handled manually

Pro tip: Get more details in this blog post series

Redo is applied on standby databases continuously during upgrade

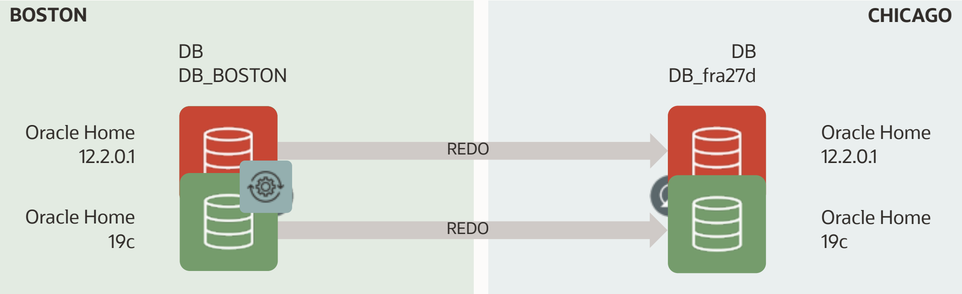
- Complies with MAA recommendations

Data Guard | Comparison

BEFORE `defer_standby_log_shipping=yes`	AFTER `defer_standby_log_shipping=no`
Maximum protection	Minimum downtime
Upgrade team recommendation	MAA recommendation
Redo log transport deferred	Redo log transport enabled
Redo apply stopped	Redo apply active
Protected by disconnected standby and guaranteed restore point	Protected by guaranteed restore point
	DEFAULT

👍 When upgrading to Oracle Database 19c keep Data Guard broker running

Data Guard | Concept

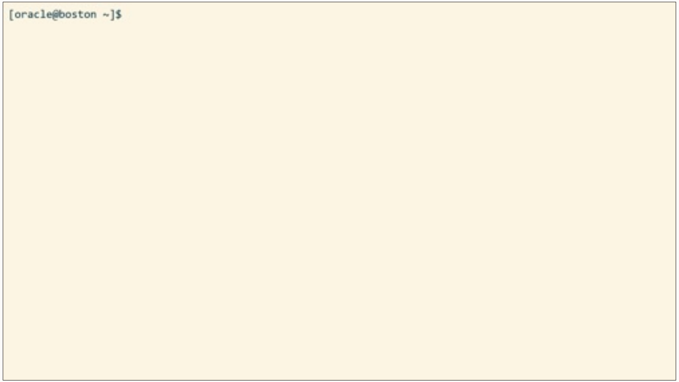


```
$ java -jar autoupgrade.jar ... -mode deploy
```

- Restore point
- Start in new Oracle Home
- Upgrade

```
$ srvctl stop database -d $ORACLE_UNQNAME

$ #switch to new Oracle Home
$ srvctl upgrade database -d $ORACLE_UNQNAME
$ srvctl start database -d $ORACLE_UNQNAME \
      -startoption mount
```

Data Guard | Demo



```
[oracle@boston ~]$
```

Watch on YouTube

Data Guard | Deferring Log Transport

⚠

A word of advice:
If `defer_standby_log_shipping=yes`,
all remote log archive destinations are deferred

A log archive destination can be used for:

- Standby databases
- GoldenGate downstream capture
- Per PDB Data Guard
- ZDLRA real-time redo transport

⚠ When a CDB with Data Guard receives a new PDB, special attention is needed

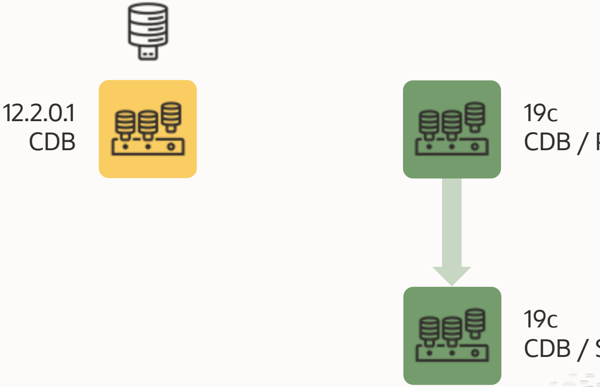
⚠ PDB data files must be in exact same location on primary and standby database, otherwise, MRP process will crash

👍

New AutoUpgrade config file parameter `manage_standbys_clause` defaults to `NONE`

AutoUpgrade will create PDBs using `STANDBYS=NONE` clause

Data Guard | Plug-in on standby



```
$ cat PDB1.cfg

upg1.pdbs=PDB1
upg1.sid=CDB122
upg1.target_sic
...

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

```
SQL> show pdbs

CON_NAME     OPEN MODE
PDB1         READ WRITE
```

```
SQL> CREATE PLU
     PDB1 USINC
     STANDBYS=N
```

```
RMAN> restore pluggable database
      pdb1 from service .... ;

SQL> alter pluggable database
     enable recovery;
SQL> alter database datafile
     ... online;
```

Data Guard | Plug-in on standby

For:

- Non-CDB to PDB conversion
- Unplug-plug upgrade

PDB is available on primary database only

- For a period, PDB is not protected by Data Guard
- Restore and recover data files to standby database
- Making Use Deferred PDB Recovery and the STANDBYS=NONE Feature with Oracle Multitenant (Doc ID 1916648.1)

You can re-use the PDB data files on the standby database, but special attention is needed

- Use AutoUpgrade config file parameter `manage_standbys_clause=all`

Data Guard | Re-use data files

To re-use data files and keep standby database intact

- Including making PDB available on standby database immediately

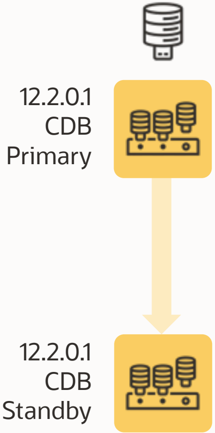
Special care is required

- Data files on standby database must be in exact same location as on primary database

For ASM and OMF:

- Reusing the Source Standby Database Files When Plugging a non-CDB as a PDB into the Primary Database of a Data Guard Configuration (Doc ID 2273304.1)

Data Guard | Re-use data files



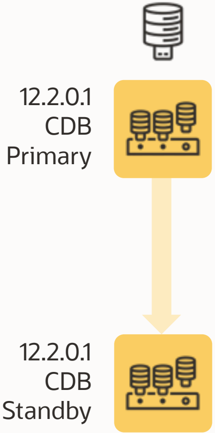
12.2.0.1
CDB
Primary

12.2.0.1
CDB
Standby

```
SQL> select name from v$datafile where con_id=3;

NAME
--------------------------------------------------------------------------------
+DATA/DB_BOSTON/DD934E8207292138E053E801000A8351/DATAFILE/system.269.1103046537
+DATA/DB_BOSTON/DD934E8207292138E053E801000A8351/DATAFILE/sysaux.270.1103046537
+DATA/DB_BOSTON/DD934E8207292138E053E801000A8351/DATAFILE/undotbs1.268.1103046537
+DATA/DB_BOSTON/DD934E8207292138E053E801000A8351/DATAFILE/users.273.1103046827
```

```
SQL> select name from v$datafile where con_id=3;

NAME
--------------------------------------------------------------------------------
+DATA/DB_FRA27D/DD934E8207292138E053E801000A8351/DATAFILE/system.265.1103050007
+DATA/DB_FRA27D/DD934E8207292138E053E801000A8351/DATAFILE/sysaux.266.1103050007
+DATA/DB_FRA27D/DD934E8207292138E053E801000A8351/DATAFILE/undotbs1.267.1103050009
+DATA/DB_FRA27D/DD934E8207292138E053E801000A8351/DATAFILE/users.269.1103050009
```

Data Guard | Re-use data files



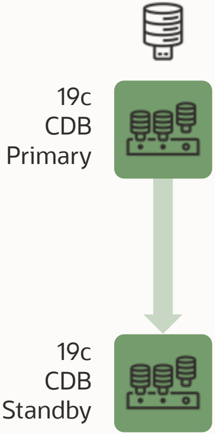
```
SQL> alter pluggable database PDB1 unplug into '/tmp/manifest_PDB1.xml';
```

The manifest file contains

- File path on primary database only
- Not standby database

```
<?xml version="1.0" encoding="UTF-8"?>
<PDB>
  <xmlversion>1</xmlversion>
  <pdbname>PDB1</pdbname>
  ...
  <guid>DDB49CFEFD8ED4FCE053E801000A078C</guid>
  ...
  <tablespace>
    <name>USERS</name>
    ...
    <file>

<path>+DATA/DB_BOSTON/DD934E8207292138E053E801000A8351/DATAFILE/users.273.1
103046827</path>
```

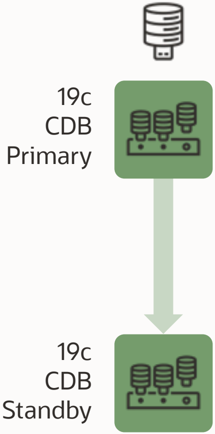
Data Guard | Re-use data files



```
SQL> create pluggable database PDB1 using '/tmp/manifest_PDB1.xml' ... ;
```

- Manifest file lists the location of data files on primary
- No information about standby databases
- Standby database expect data files at the same location as on primary

Data Guard | Re-use data files

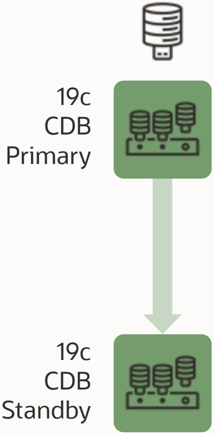


I will just move the files in ASM!

```
ASMCMD> cp users.269.1103050009 +DATA/DB_BOSTON/.../users.273.1103046827

ASMCMD-8016: copy source '+DATA/DB_FRA27D/.../users.269.1103050009' and
target '+DATA/DB_BOSTON/.../users.273.1103046827' failed
ORA-15056: additional error message
ORA-15046: ASM file name 'users.273.1103046827' is not in single-file
creation form
ORA-06512: at "SYS.X$DBMS_DISKGROUP", line 617
ORA-06512: at line 3 (DBD ERROR: OCIStmtExecute)
```

Only a database can produce files with ASM/OMF data file names

Data Guard | Re-use data files



19c
CDB
Primary

19c
CDB
Standby

ASM alias to the rescue!

- On standby, create aliases for the primary data files

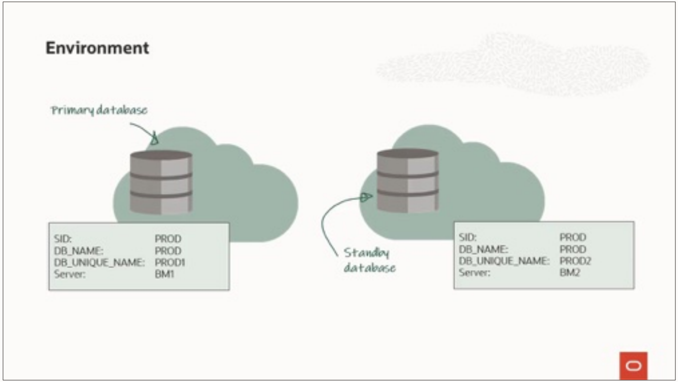
```
ASMCMD> alter diskgroup data add alias '...' for '...' ;
```

- Plug in PDB, standby will find aliases and find the real file locations
  From alert log

```
Recovery scanning directory +DATA/DB_BOSTON/... for any matching files
Deleted Oracle managed file +DATA/DB_BOSTON/...
Successfully added datafile 37 to media recovery
Datafile #37: +DATA/DB_FRA27D/.../DATAFILE/users.269.1103050009'
```

⚠ Don't jeopardize your Data Guard!
Test the procedure and
verify your environment afterwards

AutoUpgrade | Data Guard



Environment

SID:	PROD
DB_NAME:	PROD
DB_UNIQUE_NAME:	PROD1
Server:	BM1

SID:	PROD
DB_NAME:	PROD
DB_UNIQUE_NAME:	PROD2
Server:	BM2

Watch on YouTube

Photo by Philipp Katzenberger on Unsplash

AutoUpgrade and RAC

UPGRADE
RAC DATABASE

1

UPGRADE GRID INFRASTRUCTURE

- Not covered by AutoUpgrade
- Recommended to upgrade one week in advance

2

UPGRADE DATABASE

- Upgrade with AutoUpgrade
- Everything handled by AutoUpgrade

AutoUpgrade | RAC

WHAT IS REQUIRED?

- Linux or Unix-based system
- Registered and managed through `srvctl`
- SPFile in ASM

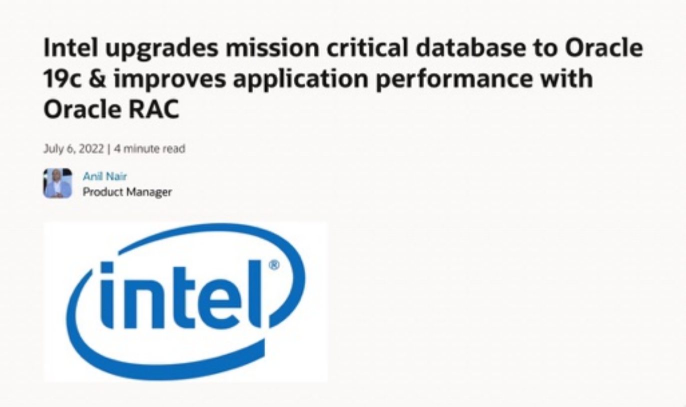
WHAT DO YOU GET?

- Fully mananged upgrade
- Supports RAC and RAC One Node
- No extra configuration
- Just connect to one node and AutoUpgrade takes care of the rest
  - `CLUSTER_DATABASE=FALSE`
  - `srvctl` configuration

It is recommended to keep Grid Infrastructure and database patch level in sync

Pro tip: Additional details can be found in blog post

Success Story | Intel



Intel upgrades mission critical database to Oracle 19c & improves application performance with Oracle RAC

July 6, 2022 | 4 minute read

Anil Nair
Product Manager

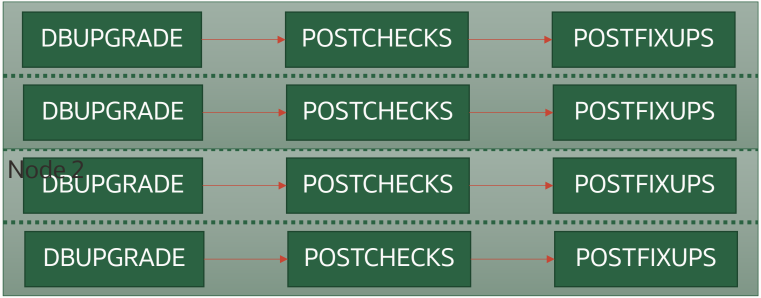
blogs.oracle.com

Distributed upgrade uses all nodes in a cluster resulting in faster upgrades of CDBs

- Applies to RAC only
- Requires Proactive Fixups

Distributed Upgrade | Concept

Node 1



Distributed Upgrade | What is it?

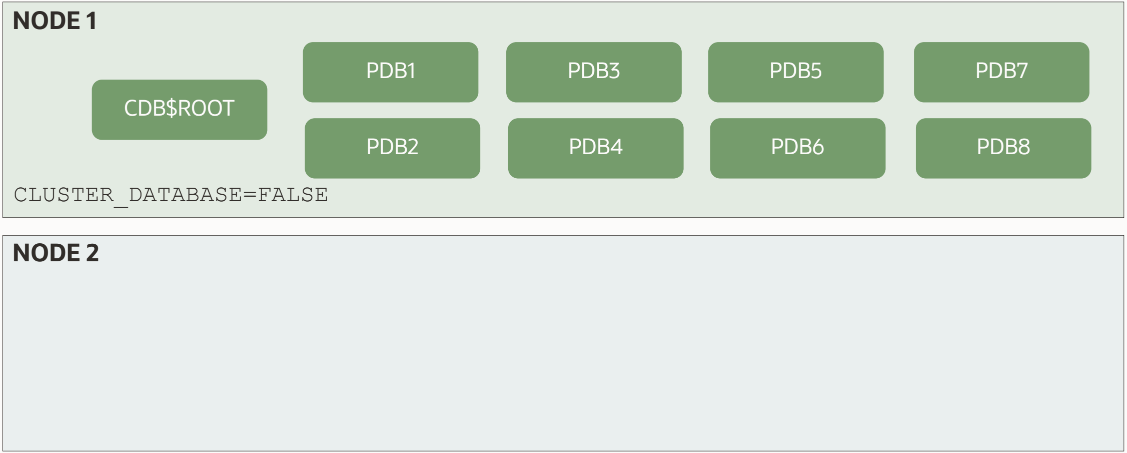
- Performance feature
- Valid for CDB upgrades on RAC only
- First, CDB$ROOT upgrades on local node

  ```
  CLUSTER_DATABASE=FALSE
  ```

- Then, leverage resources on all nodes to upgrade PDBs

  ```
  CLUSTER_DATABASE=TRUE
  ```

Distributed Upgrade | Before



NODE 1

CDB$ROOT

PDB1

PDB3

PDB5

PDB7

PDB2

PDB4

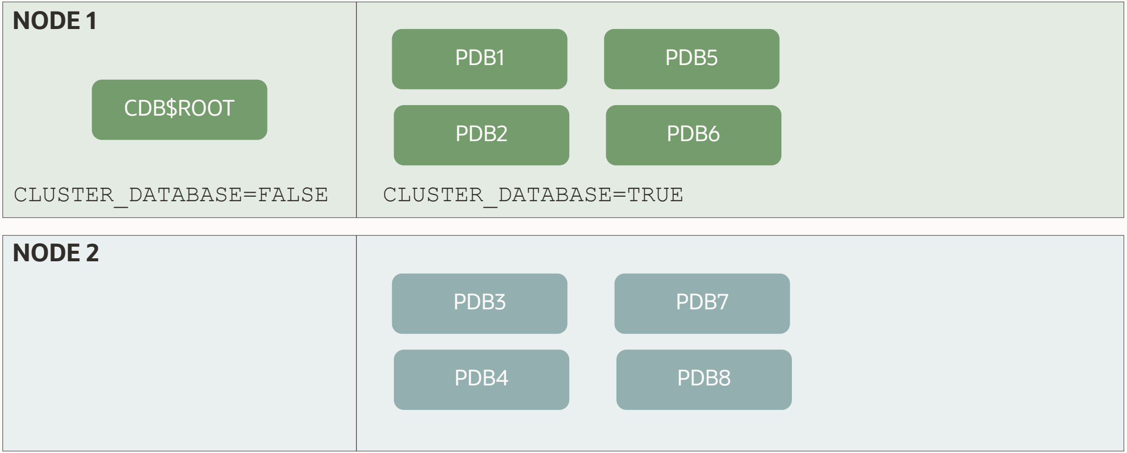
PDB6

PDB8

CLUSTER_DATABASE=FALSE

NODE 2

Distributed Upgrade | After



Distributed Upgrade | Console Message

```
Stage-Progress Per Container

        +--------+----------+--------+----+
        |Database|     Stage|Progress|Node|
        +--------+----------+--------+----+
        |PDB$SEED| DBUPGRADE|    91 %| au1|
        |   PDB01|POSTFIXUPS|     0 %| au1|
        |   PDB03|POSTFIXUPS|     0 %| au1|
        |   PDB04|POSTFIXUPS|     0 %| au1|
        |   PDB05|POSTFIXUPS|     0 %| au1|
        |   PDB02| DBUPGRADE|    91 %| au2|
        |   PDB06| DBUPGRADE|    91 %| au2|
        |   PDB07| DBUPGRADE|    91 %| au2|
        |   PDB08| DBUPGRADE|    91 %| au2|
        |   PDB09| DBUPGRADE|    91 %| au2|
        +--------+----------+--------+----+
```

Distributed Upgrade | Use

To enable distributed upgrade:

```
$ cat RACDB.cfg

global.autoupg_log_dir=/u01/app/oracle/cfgtoollogs/autoupgrade
upg1.log_dir=/u01/app/oracle/cfgtoollogs/autoupgrade/ RACDB
upg1.source_home=/u01/app/oracle/product/12.2.0.1
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=RACDB
upg1.tune_setting=distributed_upgrade=true

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

Distributed Upgrade | Use

Under the hood

1. **AutoUpgrade creates a special config file**

2. AutoUpgrade spawns itself on all nodes

```
global.autoupg_log_dir=/u01/app/oracle/cfgtoollogs/...
# Databases section
# Database Batch 1
batch1.sid=RACDB1
batch1.source_home=/u01/app/oracle/product/12.2.0.1
batch1.target_home=/u01/app/oracle/product/19
batch1.upgrade_node=boston1
batch1.pdbs=PDB$SEED,PDB01,PDB03,PDB04,PDB05
batch1.tune_setting=DISTRIBUTED_UPGRADE=true,...
# Database Batch 2
batch2.sid=RACDB2
batch2.source_home=/u01/app/oracle/product/12.2.0.1
batch2.target_home=/u01/app/oracle/product/19
batch2.upgrade_node=boston2
batch2.pdbs=PDB02,PDB06,PDB07,PDB08,PDB09
batch2.tune_setting=DISTRIBUTED_UPGRADE=true,...
```

Distributed Upgrade | Use

Under the hood

1. AutoUpgrade creates a special config file

2. **AutoUpgrade spawns itself on all nodes**

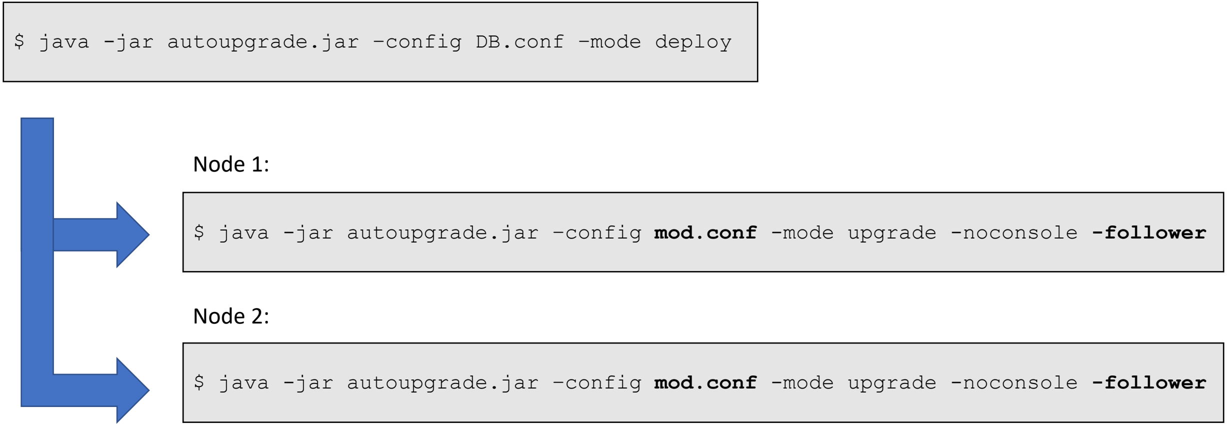
Node 1

```
$ java -jar autoupgrade.jar -config mod.conf \
     -mode upgrade -noconsole -follower
```

Node 2

```
$ java -jar autoupgrade.jar -config mod.conf \
     -mode upgrade -noconsole -follower
```

Distributed Upgrade | Architecture



```
$ java -jar autoupgrade.jar -config DB.conf -mode deploy
```

Node 1:

```
$ java -jar autoupgrade.jar -config mod.conf -mode upgrade -noconsole -follower
```

Node 2:

```
$ java -jar autoupgrade.jar -config mod.conf -mode upgrade -noconsole -follower
```

Distributed Upgrade | Architecture

Custom config file for 1 CDB + 10 PDBs upgrade

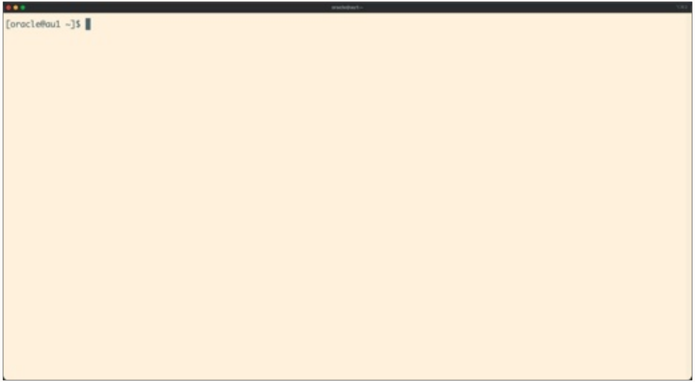
```
global.autoupg_log_dir=/u01/app/oracle/cfgtoollogs/autoupgrade/CDB12_iad1d7/au_logs
# Databases section
# Database Batch 1
batch1.sid=CDB121
batch1.source_home=/u01/app/oracle/product/12.2.0.1/dbhome_1
batch1.target_home=/u01/app/oracle/product/19/dbhome_2
batch1.upgrade_node=au1
batch1.pdbs=PDB$SEED,PDB01,PDB03,PDB04,PDB05
batch1.tune_setting=DISTRIBUTED_UPGRADE=true,PROACTIVE_FIXUPS=true
# Database Batch 2
batch2.sid=CDB122
batch2.source_home=/u01/app/oracle/product/12.2.0.1/dbhome_1
batch2.target_home=/u01/app/oracle/product/19/dbhome_2
batch2.upgrade_node=au2
batch2.pdbs=PDB02,PDB06,PDB07,PDB08,PDB09
batch2.tune_setting=DISTRIBUTED_UPGRADE=true,PROACTIVE_FIXUPS=true
```

Uses ORACLE_BASE

Node 1 (au1) receives 5 PDBs

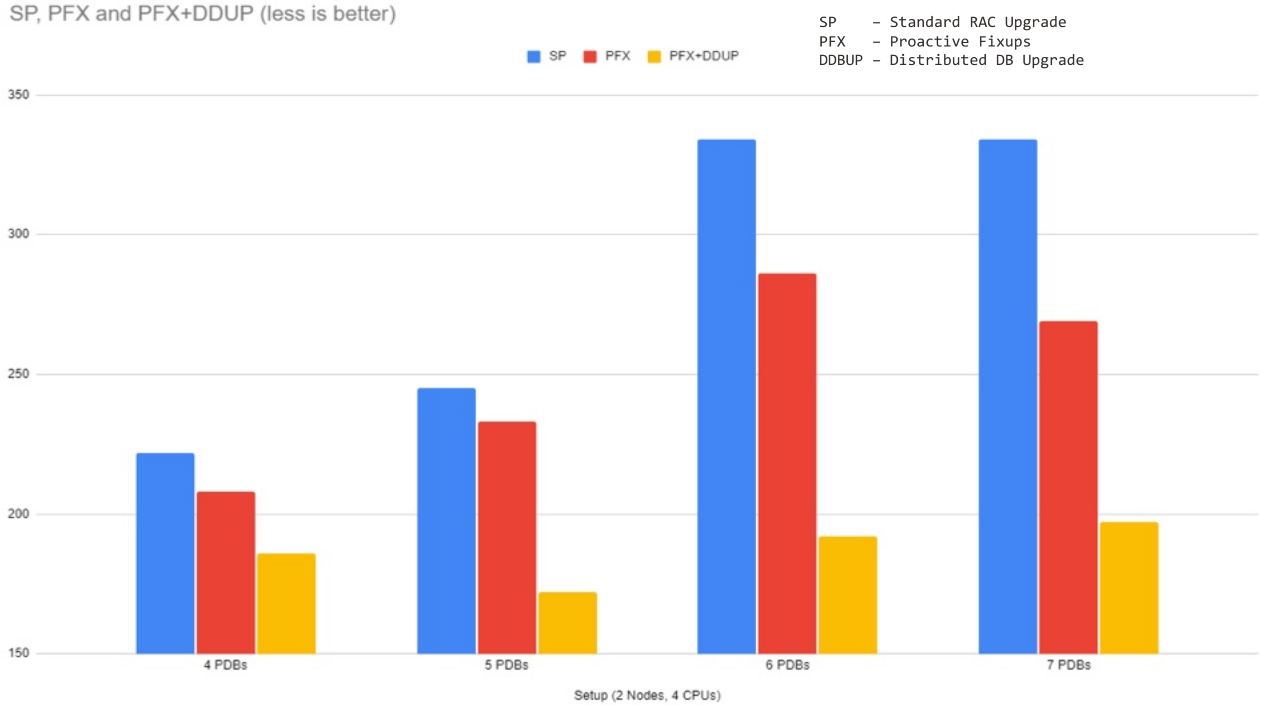
Node 2 (au2) also receives another 5 PDBs

Distributed Upgrade | Demo



```
[oracle@au1 ~]$
```

Watch on YouTube



```
SP    – Standard RAC Upgrade
PFX   – Proactive Fixups
DDBUP – Distributed DB Upgrade
```

41%

In benchmark, time saved by using distributed upgrade

- 2 node RAC database
- 4 CPUs each
- CDB with 8 PDBs

By default, AutoUpgrade uses two nodes

You can control how many nodes are being used

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

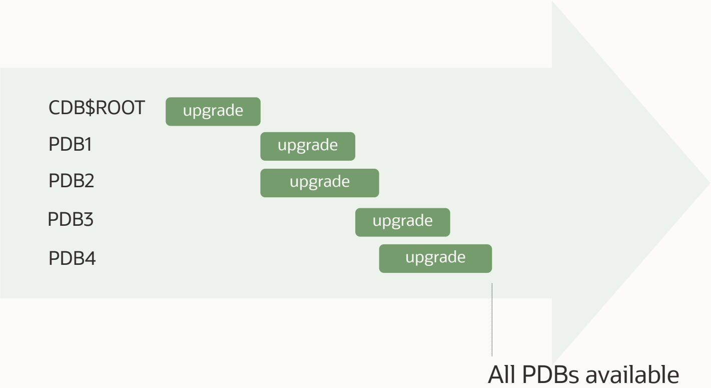
Some PDBs are more important

Control the order of the upgrade

PDB Availability

DEFAULT

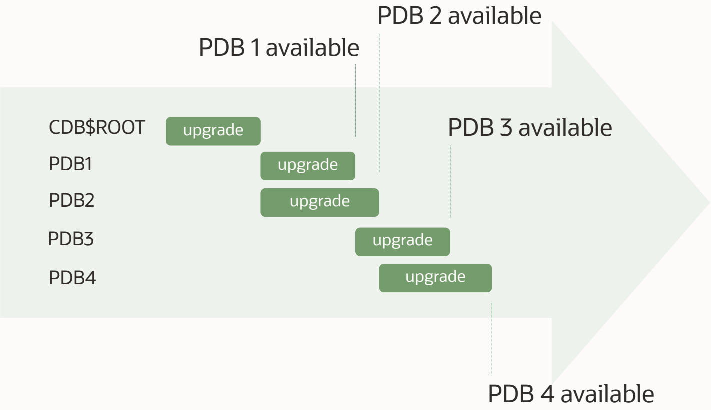
`make_pdbs_available=false`



PDB Availability

IMMEDIATELY AVAILABLE

```
make_pdbs_available=true
```

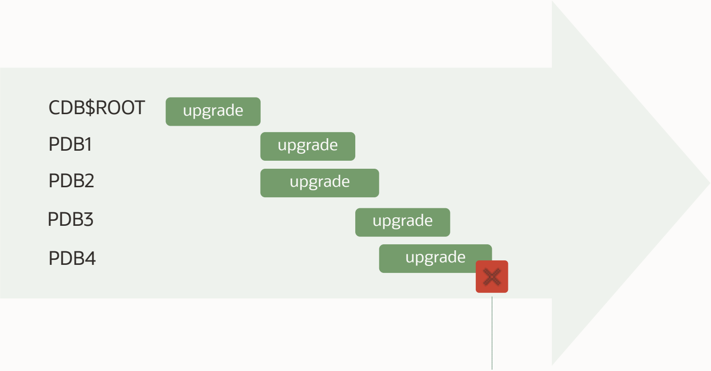


```
alter pluggable database SALESPROD priority 1;

alter pluggable database SALESDEV priority 2;

alter pluggable database SALESUAT priority 2;

alter pluggable database SALESTEST priority 3;
```

PDB Availability

IMMEDIATELY AVAILABLE

```
make_pdbs_available=true
```



PDB 4 crash ...

Flashback entire CDB?

Distributed Upgrade | Questions

- All nodes on which the database is configured (srvctl) must be up and running, otherwise, AutoUpgrade will error out

Photo by Hello I'm Nik 🎞 on Unsplash

AutoUpgrade and TDE

AutoUpgrade fully supports Transparent Data Encryption

- Isolated keystore mode coming in a later version

TDE | Keystore

- New config file parameter: `global.keystore`
- Governs directory of AutoUpgrade keystore
- Password protected software keystore
- Optionally, an auto-open keystore

TDE | Keystore

```
$ cat DB12.cfg

global.keystore=/etc/oracle/keystores/autoupgrade/DB12
...


$ ls -l /etc/oracle/keystores/autoupgrade/DB12

-rw-------. 1 oracle dba 720 Mar 28 14:56 ewallet.p12
```

TDE | Keystore

AutoUpgrade keystore contains

- Database TDE keystore passwords (user-supplied)
- Passphrases or transport secrets (auto-generated)

TDE | Keystore

```
$ java -jar autoupgrade.jar -config DB12.cfg -load_password


TDE> add DB12

Enter your secret/Password:
Re-enter your secret/Password:
```

TDE | Keystore

In the TDE console, the following commands are available:

- `add`
- `delete`
- `list`
- `save`
- `help`
- `exit`

A password protects the AutoUpgrade keystore, unless you also create an auto-login keystore

TDE | Keystore

```
$ java -jar autoupgrade.jar -config DB12.cfg -load_password

TDE> save

Convert the keystore to auto-login [YES|NO] ?




$ ls -l /etc/oracle/keystores/autoupgrade/DB12

-rw-------. 1 oracle dba 765 Mar 28 14:56 cwallet.sso
-rw-------. 1 oracle dba 720 Mar 28 14:56 ewallet.p12
```

Protect the AutoUpgrade keystore like you protect any other keystore

- Apply restrictive file system permissions
- Audit access
- Back it up

TDE | Upgrade Non-CDB or CDB

To upgrade an encrypted non-CDB or entire CDB

- An auto-login TDE keystore must be present

```
SQL> -- LOCAL_AUTOLOGIN is also usable
SQL> select wallet_type from v$encryption_wallet;

AUTOLOGIN
```

You do not need an AutoUpgrade keystore

TDE | Upgrade Non-CDB or CDB

Workaround

- If database has issues finding the right keystore, you can override TNS_ADMIN location in config file:

```
upg1.source_tns_admin_dir=/u01/app/oracle/admin/DB12/tns_admin
upg1.target_tns_admin_dir=/u01/app/oracle/admin/DB12/tns_admin
```

Defining keystore location in *sqlnet.ora* is deprecated in Oracle Database 19c

👍 Use `WALLET_ROOT` parameter to define keystore location and use new TDE functionality

TDE | Upgrade Non-CDB or CDB

Use AutoUpgrade to switch to keystore configuration using `WALLET_ROOT`

Create text file with new initialization parameters:

```
$ cat /tmp/au-pfile-tde.txt

WALLET_ROOT='/etc/oracle/keystores/$ORACLE_SID'
TDE_CONFIGURATION='KEYSTORE_CONFIGURATION=FILE'
```

TDE | Upgrade Non-CDB or CDB

Instruct AutoUpgrade to add parameters during and after upgrade:

```
upg1.add_during_upgrade_pfile=/tmp/au-pfile-tde.txt
upg1.add_after_upgrade_pfile=/tmp/au-pfile-tde.txt
```

AutoUpgrade automatically copies keystore from previous location into location defined by `WALLET_ROOT`

Pro tip: Get more details in blog post

TDE | Upgrade Encrypted Non-CDB and Convert

To upgrade an encrypted non-CDB and afterwards convert it to a PDB:

Create config file

```
global.autoupg_log_dir=/u01/app/oracle/cfgtoollogs/autoupgrade
global.keystore=/u01/app/oracle/admin/autoupgrade/keystore

upg1.log_dir=/u01/app/oracle/cfgtoollogs/autoupgrade/DB12
upg1.source_home=/u01/app/oracle/product/12.2.0.1
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=DB12
upg1.target_cdb=CDB2
```

TDE | Upgrade Encrypted Non-CDB and Convert

Analyze the non-CDB for upgrade readiness

```
$ java -jar autoupgrade.jar -config DB12.cfg -mode analyze
```

Summary report will show which keystore passwords are needed:

```
REQUIRED ACTIONS
================
       1.  Perform the specified action ...
       ORACLE_SID                        Action Required
       ------------------------------  ------------------------
       DB12                            Add TDE password
       CDB2                            Add TDE password
```

TDE | Upgrade Encrypted Non-CDB and Convert

Start TDE console to load passwords

```
$ java -jar autoupgrade.jar -config DB12.cfg -load_password
```

Add database keystore passwords

```
TDE> add DB12

TDE> add CDB2
```

Start upgrade

```
$ java -jar autoupgrade.jar -config DB12.cfg -mode deploy
```

TDE | Upgrade Encrypted PDB

To upgrade an encrypted PDB using unplug-plug:

Create config file

```
global.autoupg_log_dir=/u01/app/oracle/cfgtoollogs/autoupgrade
global.keystore=/u01/app/oracle/admin/autoupgrade/keystore

upg1.log_dir=/u01/app/oracle/cfgtoollogs/autoupgrade/PDB1
upg1.source_home=/u01/app/oracle/product/12.2.0.1
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=CDB1
upg1.target_cdb=CDB2
upg1.pdbs=PDB1
```

TDE | Upgrade Encrypted PDB

Analyze the PDB for upgrade readiness

```
$ java -jar autoupgrade.jar -config PDB1.cfg -mode analyze
```

Summary report will show which keystore passwords are needed:

```
REQUIRED ACTIONS
================
       1.  Perform the specified action ...
       ORACLE_SID                        Action Required
       ------------------------------    ------------------------
       CDB1                              Add TDE password
       CDB2                              Add TDE password
```

TDE | Upgrade Encrypted PDB

Start TDE console to load passwords

```
$ java -jar autoupgrade.jar -config PDB1.cfg -load_password
```

Add database keystore passwords

```
TDE> add CDB1

TDE> add CDB2
```

Start upgrade

```
$ java -jar autoupgrade.jar -config PDB1.cfg -mode deploy
```

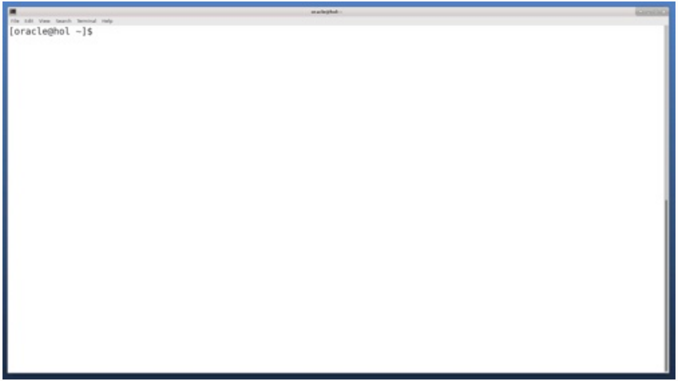
👍 AutoUpgrade also supports converting an encrypted non-CDB to PDB

AutoUpgrade is compatible with Secure External Password Store

- Supported from Oracle Database 12.2

Pro tip: Get more details in blog post

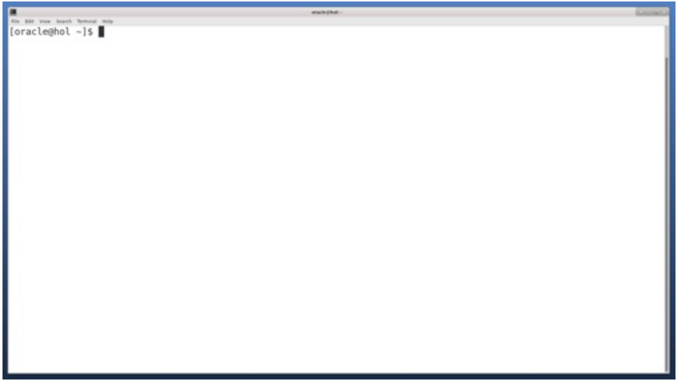
TDE | Demo - Upgrading encrypted PDB



```
[oracle@hol ~]$
```

Watch on YouTube

TDE | Demo - Upgrading and converting to PDB



```
[oracle@hol ~]$
```

Watch on YouTube

TDE | Additional Information

- Blog post series
- Configuring an External Store for a Keystore Password

Photo by Hello I'm Nik 🎞 on Unsplash

AutoUpgrade on Exadata

AutoUpgrade is the recommended utility to upgrade Oracle Databases to 19c

Source:
19c Grid Infrastructure and Database Upgrade steps for Exadata Database Machine running on Oracle Linux (Doc ID 2542082.1)

AutoUpgrade | Exadata

Follow elaborate procedure in MOS note:

19c Grid Infrastructure and Database Upgrade steps
for Exadata Database Machine running on Oracle Linux (Doc ID 2542082.1)

AutoUpgrade version 21.1.3 or higher is required

- but always use the latest version of AutoUpgrade

What about ExaCC and ExaCS?

AutoUpgrade | ExaCC + ExaCS

Follow elaborate procedure in MOS note:

- Upgrading to 19c Oracle Database on Exadata Cloud Service (ExaCS) and Exadata Cloud at Customer Gen2 (ExaCC) (Doc ID 2628228.1)

Above mentioned procedure is the only support method

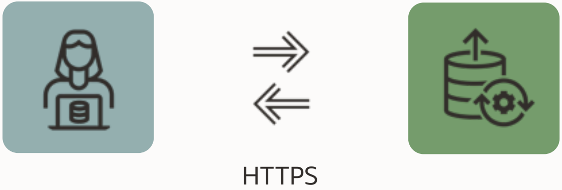
Currently, DBUA is utilized by cloud tooling

Photo by Hello I'm Nik 🎞️ on Unsplash

AutoUpgrade REST API

REST API | Why use it?

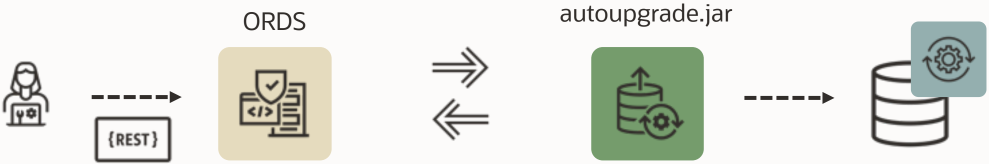
- Well-known API
- Flexibility
- Simplicity
- Upgrade-on-demand



HTTPS

REST API | How it works

AutoUpgrade uses ORDS to handle the Java calls



Requirement:

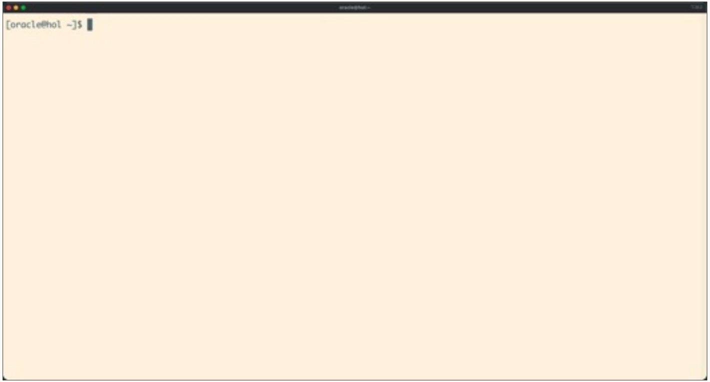
- Oracle REST Data Services (ORDS) 22.1.0 or later

REST API | Enable

```
$ #Enable the AutoUpgrade API
$ java -jar ords.war set-property autoupgrade.api.enabled true

$ #Set the location of AutoUpgrade log files
$ java -jar ords.war set-property autoupgrade.api.loglocation /u01/autoupgrade_logs

$ #Which AutoUpgrade.jar to use
$ java -jar ords.war set-property autoupgrade.api.aulocation /u01/autoupgrade.jar

$ #Which Java to use
$ java -jar ords.war set-property autoupgrade.api.jvmlocation /bin/java
```

ℹ Always use a separate instance of ORDS for AutoUpgrade APIs ℹ Always protect the REST APIs endpoints with a firewall

REST API | Demo - Install



```
[oracle@hol ~]$ 
```

Watch on YouTube

REST API | Config file vs. JSON

Config File

```
global.autoupg_log_dir=/home/oracle/logs
upg1.source_home=/u01/app/product/11
upg1.target_home=/u01/app/product/19
upg1.sid=UPGR
upg1.log_dir=/home/oracle/logs
upg1.restoration=no
```

JSON for REST API

```
{
  "global": {
    "autoupg_log_dir": "/home/oracle/logs"
  },
  "jobs": [{
    "source_home": "/u01/app/product/11",
    "target_home": "/u01/app/product/19",
    "sid": "UPGR",
    "log_dir": "/home/oracle/logs",
    "restoration": "no"
  }]
}
```

REST API | Config file vs. JSON

Config File

```
global.autoupg_log_dir=/home/oracle/logs
upg1.source_home=/u01/app/product/11
upg1.target_home=/u01/app/product/19
upg1.sid=UPGR
upg1.log_dir=/home/oracle/logs
upg1.restoration=no
```

JSON for REST API

```
{
  "global": {
    "autoupg_log_dir": "/home/oracle/logs"
  },
  "jobs": [{
    "source_home": "/u01/app/product/11",
    "target_home": "/u01/app/product/19",
    "sid": "UPGR",
    "log_dir": "/home/oracle/logs",
    "restoration": "no"
  }]
}
```

REST API | Config file vs. JSON

Config File

```
global.autoupg_log_dir=/home/oracle/logs
upg1.source_home=/u01/app/product/11
upg1.target_home=/u01/app/product/19
upg1.sid=UPGR
upg1.log_dir=/home/oracle/logs
upg1.restoration=no
```

JSON for REST API

```
{
  "global": {
    "autoupg_log_dir": "/home/oracle/logs"
  },
  "jobs": [{
    "source_home": "/u01/app/product/11",
    "target_home": "/u01/app/product/19",
    "sid": "UPGR",
    "log_dir": "/home/oracle/logs",
    "restoration": "no"
  }]
}
```

REST API | Config file vs. JSON

Config File

```
global.autoupg_log_dir=/home/oracle/logs
upg1.source_home=/u01/app/product/11
upg1.target_home=/u01/app/product/19
upg1.sid=UPGR
upg1.log_dir=/home/oracle/logs
upg1.restoration=no
```

JSON for REST API

```
{
  "global": {
    "autoupg_log_dir": "/home/oracle/logs"
  },
  "jobs": [{
    "source_home": "/u01/app/product/11",
    "target_home": "/u01/app/product/19",
    "sid": "UPGR",
    "log_dir": "/home/oracle/logs",
    "restoration": "no"
  }]
}
```

REST API | Config file vs. JSON

Config File

```
global.autoupg_log_dir=/home/oracle/logs
upg1.source_home=/u01/app/product/11
upg1.target_home=/u01/app/product/19
upg1.sid=UPGR
upg1.log_dir=/home/oracle/logs
upg1.restoration=no
```

JSON for REST API

```
{
  "global": {
    "autoupg_log_dir": "/home/oracle/logs"
  },
  "jobs": [{
    "source_home": "/u01/app/product/11",
    "target_home": "/u01/app/product/19",
    "sid": "UPGR",
    "log_dir": "/home/oracle/logs",
    "restoration": "no"
  }]
}
```

REST API | Methods

Methods available in the REST API

- **task** (GET / POST)
- **tasks** (GET)
- **status** (GET)
- **progress** (GET)
- **console** (GET)
- **log** (GET)

Only API with POST method

Pro tip: Read more about REST APIs

REST API | Create a New AutoUpgrade Task

```
$ curl -k --data-binary "@UPGR.json" -X POST --header "Content-Type:application/json"
'https://localhost:8443/ords/autoupgrade/task?mode=analyze'
```

Ignore verification of the certificate

JSON file

HTTP method

Equal

Endpoint API

AutoUpgrade configuration

Input file type

```
{
    "taskid": "job_2022_04_27_05_17:24.146_0",
    "status": "submitted",
    "message": "",
    "link": "https://localhost:8443/ords/autoupgrade/task?taskid=job_2022_04_27_05.17.24.146_0",
    "config": {
        "global": {
```

```
$ java -jar autoupgrade.jar -config UPGR.cfg -mode analyze
```

```
                "source_home": "/u01/app/oracle/product/11.2.0.4",
                "target_home": "/u01/app/oracle/product/19",
                "sid": "UPGR",
                "log_dir": "/home/oracle/logs",
                "restoration": "no"
            }
        ]
    }
}
```

REST API | List all Tasks

```
$ curl -k https://localhost:8443/ords/autoupgrade/tasks
```

```
{
    "total_tasks": 1,
    "tasks": [
        {
            "mode": "analyze",
            "taskid": "job_2022_04_27_05.17.24.146_0",
            "config": {
                "jobs": [
                    {
                        "source_home": "/u01/app/oracle/product/11.2.0.4",
                        "sid": "UPGR"
                    }
                ]
            },
            "link": "https://localhost:8443/ords/autoupgrade/task?taskid=job_2022_04_27_05.17.24.146_0"
        }
    ]
}
```

Task Identifier

REST API | Get Specific Task

```
$ curl -k 'https://localhost:8443/ords/autoupgrade/task?taskid=job_2022_04_27_05.17.24.146_0'
```

Task Identifier

```
{
    "taskid": "job_2022_04_27_05.17.24.146_0",
    "status": "finished",
    "message": "",
    "link": "https://localhost:8443/ords/autoupgrade/task?taskid=job_2022_04_27_05.17.24.146_0",
    "config": {
        "global": {
            "autoupg_log_dir": "/home/oracle/logs"
        },
        "jobs": [
            {
                "source_home": "/u01/app/oracle/product/11.2.0.4",
                "target_home": "/u01/app/oracle/product/19",
                "sid": "UPGR",
                "log_dir": "/home/oracle/logs",
                "restoration": "no"
            }
        ]
    }
}
```

Task Status

REST API | Get console output for Job

```
$ curl -k 'https://localhost:8443/ords/autoupgrade/console?taskid=job_2022_04_27_05.17.24.146_0'
```

```
AutoUpgrade is not fully tested on OpenJDK 64-Bit Server VM, Oracle recommends to use Java HotSpot(TM)
AutoUpgrade 22.2.220324 launched with default internal options
Processing config file ...
+--------------------------------+
| Starting AutoUpgrade execution |
+--------------------------------+
1 Non-CDB(s) will be analyzed
Job 100 database upgr
Job 100 completed
------------------- Final Summary --------------------
Number of databases            [ 1 ]

Jobs finished                  [1]
Jobs failed                    [0]

Please check the summary report at:
/u01/AU_REST/autoupgrade_logs/job_2022_04_27_05.17.24.146_0/cfgtoollogs/upgrade/auto/status/status.html
/u01/AU_REST/autoupgrade_logs/job_2022_04_27_05.17.24.146_0/cfgtoollogs/upgrade/auto/status/status.log
```

REST API | Resubmit in deploy mode

```
$ curl -k -X POST 'https://localhost:8443/ords/autoupgrade/task?taskid=job_2022_04_27_05.17.24.146_0&mode=deploy'
```

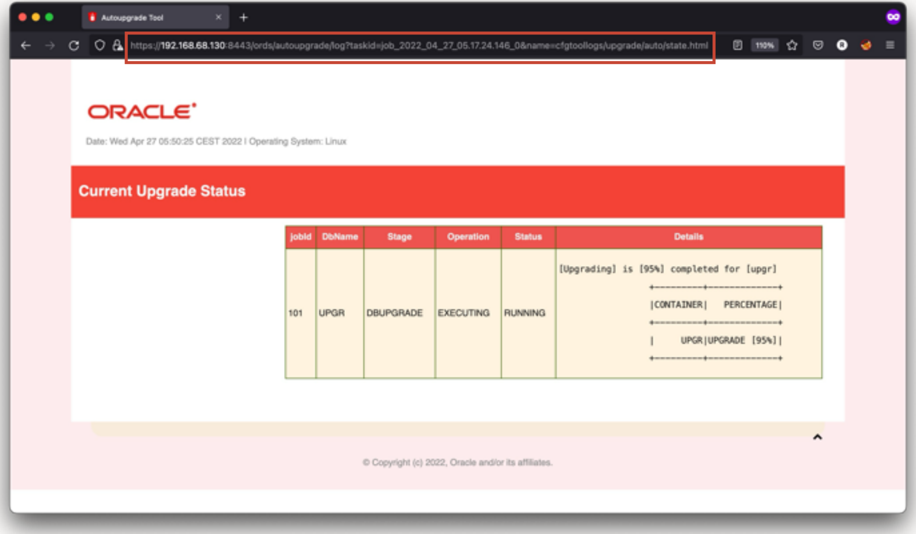
```
{
    "taskid": "job_2022_04_27_05.17.24.146_0",
    "status": "submitted",
    "message": "",
    "link": "https://localhost:8443/ords/autoupgrade/task?taskid=job_2022_04_27_05.17.24.146_0",
    "config": {
        "global": {
            "autoupg_log_dir": "/home/oracle/logs"
        },
        "jobs": [
            {
                "source_home": "/u01/app/oracle/product/11.2.0.4",
                "target_home": "/u01/app/oracle/product/19",
                "sid": "UPGR",
                "log_dir": "/home/oracle/logs",
                "restoration": "no"
            }
        ]
    }
}
```

REST API | List all files created by task

```
$ curl -k 'https://localhost:8443/ords/autoupgrade/log?taskid=job_2022_04_27_05.17.24.146_0'
```

```
{
    "logs": [
        ...,
        {
            "filename": "cfgtoollogs/upgrade/auto/status/status.html",
            "link":
"https://localhost:8443/ords/autoupgrade/log?taskid=job_2022_04_27_05.17.24.146_0&name=cfgtoollogs/upgrade/auto/status/status.html"
        },
        {
            "filename": "cfgtoollogs/upgrade/auto/status/status.log",
            "link":
"https://localhost:8443/ords/autoupgrade/log?taskid=job_2022_04_27_05.17.24.146_0&name=cfgtoollogs/upgrade/auto/status/status.log"
        },
        {
            "filename": "cfgtoollogs/upgrade/auto/status/progress.json",
            "link":
"https://localhost:8443/ords/autoupgrade/log?taskid=job_2022_04_27_05.17.24.146_0&name=cfgtoollogs/upgrade/auto/status/progress.json"
        }
    ]
}
```

REST API | State.html



https://192.168.68.130:8443/ords/autoupgrade/log?taskid=job_2022_04_27_05.17.24.146_0&name=cfgtoollogs/upgrade/auto/state.html

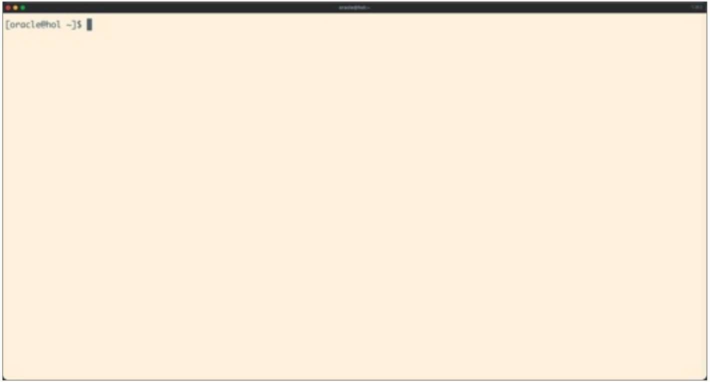
Date: Wed Apr 27 05:50:25 CEST 2022 | Operating System: Linux

Current Upgrade Status

<table>
<tr><th>jobId</th><th>DbName</th><th>Stage</th><th>Operation</th><th>Status</th><th>Details</th></tr>
<tr><td>101</td><td>UPGR</td><td>DBUPGRADE</td><td>EXECUTING</td><td>RUNNING</td><td>[Upgrading] is [95%] completed for [upgr]
+---------+-------------+
|CONTAINER| PERCENTAGE|
+---------+-------------+
| UPGR|UPGRADE [95%]|
+---------+-------------+</td></tr>
</table>

© Copyright (c) 2022, Oracle and/or its affiliates.

REST API | Demo - Upgrade



```
[oracle@hol ~]$
```

[Watch on YouTube](#)

REST API | More details

- Official Documentation
- Blog post

Photo by Joshua Fernandez on Unsplash

Compatible

Compatible | Recommendation

When should you change `COMPATIBLE`?

A week or two after the upgrade - requires a database restart

Caution: When you change `COMPATIBLE` you can't:

- Flashback to restore point
- Downgrade

Compatible | Recommendation

Which value should you use for `COMPATIBLE`?

- The default of the database release
- 11.2.0
- 12.1.0
- 12.2.0
- 18.0.0
- 19.0.0

Should you change `COMPATIBLE` when patching?

- NEVER!
  - Except for …

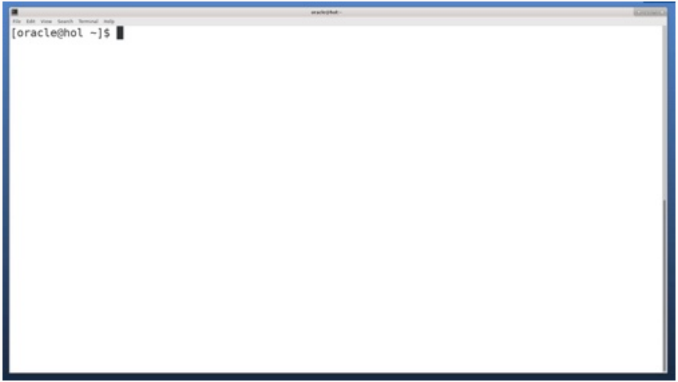
Compatible | AutoUpgrade

AutoUpgrade does not change `COMPATIBLE`

Unless you want it

```
upg1.drop_grp_after_upgrade=yes
upg1.raise_compatible=yes
```

Compatible | Demo



```
[oracle@hol ~]$ 
```

Watch on YouTube

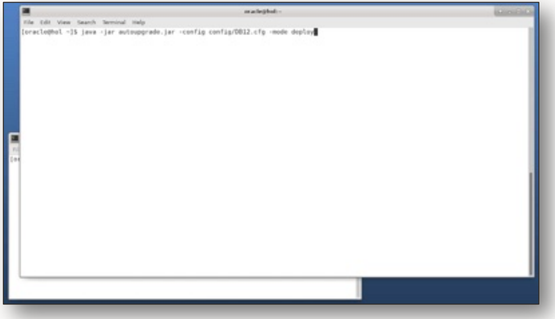
Photo by Chris Briggs on Unsplash

What if …

Tips and Tricks and Workarounds

AutoUpgrade | What if … your session is lost

- AutoUpgrade is fully resumable
- Restart using the same command line
- Previous work is preserved
  - upgrade restarts from where it left



```
[oracle@hol ~]$ java -jar autoupgrade.jar -config config/DB12.cfg -mode deploy
```

Watch on YouTube

AutoUpgrade | What if … your session is lost

Or better, use `nohup`

```
nohup java -jar autoupgrade.jar -config db.cfg -mode deploy -noconsole &
```

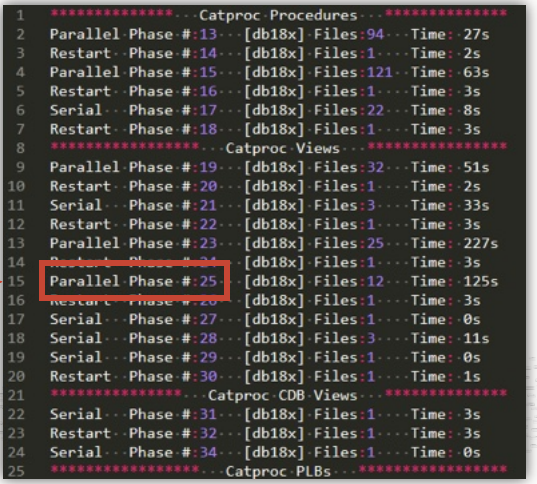
Or even better, use a terminal multiplexer

- `tmux`
- `screen`

Upgrade | Resumable

`dbupgrade` is also fully resumable

```
# Resume from failed phase
$ dbupgrade -R

# Resume from specific phase
$ dbupgrade -p 25
```



```
***************   Catproc Procedures   ***************
Parallel Phase #:13   [db18x] Files:94   Time: 27s
Restart  Phase #:14   [db18x] Files:1    Time: 2s
Parallel Phase #:15   [db18x] Files:121  Time: 63s
Restart  Phase #:16   [db18x] Files:1    Time: 3s
Serial   Phase #:17   [db18x] Files:22   Time: 8s
Restart  Phase #:18   [db18x] Files:1    Time: 3s
******************   Catproc Views   ****************
Parallel Phase #:19   [db18x] Files:32   Time: 51s
Restart  Phase #:20   [db18x] Files:1    Time: 2s
Serial   Phase #:21   [db18x] Files:3    Time: 33s
Restart  Phase #:22   [db18x] Files:1    Time: 3s
Parallel Phase #:23   [db18x] Files:25   Time: 227s
Restart  Phase #:24   [db18x] Files:1    Time: 3s
Parallel Phase #:25   [db18x] Files:12   Time: 125s
Restart  Phase #:26   [db18x] Files:1    Time: 3s
Serial   Phase #:27   [db18x] Files:1    Time: 0s
Serial   Phase #:28   [db18x] Files:3    Time: 11s
Serial   Phase #:29   [db18x] Files:1    Time: 0s
Restart  Phase #:30   [db18x] Files:1    Time: 1s
***************   Catproc CDB Views   **************
Serial   Phase #:31   [db18x] Files:1    Time: 3s
Restart  Phase #:32   [db18x] Files:1    Time: 3s
Serial   Phase #:34   [db18x] Files:1    Time: 0s
******************   Catproc PLBs   *****************
```

DBUA is not!

AutoUpgrade | What if … a fixup fails?

To override a fixup

1. **ANALYZE**
2. Edit checklist
3. Edit config file
4. Deploy

Run analyze

```
java -jar autoupgrade.jar -config DB.cfg -mode analyze
```

It will produce a checklist:

```
../prechecks/<sid>_checklist.cfg
```

AutoUpgrade | What if ... a fixup fails?

To override a fixup

1. Analyze
2. **EDIT CHECKLIST**
3. Edit config file
4. Deploy

Edit checklist and find the failing fixup

```
----------------------------------------------------
[checkname]          DICTIONARY_STATS
[stage]              PRECHECKS
[fixup_available]    YES
[runfix]             NO
[severity]           RECOMMEND
----------------------------------------------------
```

Pro tip: Set `runfix` to `_SKIP` to prevent the check and fixup from running

AutoUpgrade | What if ... a fixup fails?

To override a fixup

1. Analyze
2. Edit checklist
3. **EDIT CONFIG FILE**
4. Deploy

Edit config file and specify your customer checklist

```
upg1.checklist=../prechecks/<sid>_checklist.cfg
```

AutoUpgrade | What if ... a fixup fails?

To override a fixup

1. Analyze
2. Edit checklist
3. Edit config file
4. **DEPLOY**

Upgrade

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

AutoUpgrade | What if … a fixup fails?

`fxlist` displays the fixups

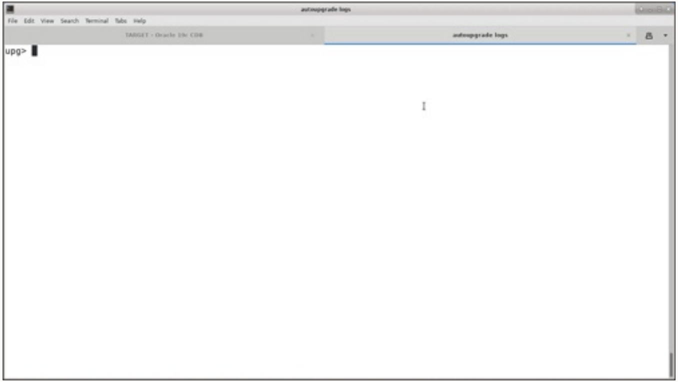
```
upg> fxlist -job 100

...

PostFixUps of Job 100

        Database DB12
                 +--------------------+---------+--------+
                 |          FixUp Name| Severity|Run Fix?|
                 +--------------------+---------+--------+
                 |OLD_TIME_ZONES_EXIST|  WARNING|     YES|
                 |     POST_DICTIONARY|RECOMMEND|     YES|
                 |          POST_UTLRP|RECOMMEND|     YES|
                 |  TIMESTAMP_MISMATCH|  WARNING|     YES|
                 +--------------------+---------+--------+
```

AutoUpgrade | What if … a fixup fails?

Change fixup execution using `fxlist` (`yes`, `no`, `skip`)

```
upg> fxlist -job 100 -c DB12 alter OLD_TIME_ZONES_EXIST run no

...

PostFixUps of Job 100

        Database DB12
                 +--------------------+---------+--------+
                 |          FixUp Name| Severity|Run Fix?|
                 +--------------------+---------+--------+
                 |OLD_TIME_ZONES_EXIST|  WARNING|      NO|
                 |     POST_DICTIONARY|RECOMMEND|     YES|
                 |          POST_UTLRP|RECOMMEND|     YES|
                 |  TIMESTAMP_MISMATCH|  WARNING|     YES|
                 +--------------------+---------+--------+
```

AutoUpgrade | What if … a fixup fails?



```
upg>
```

Watch on YouTube

AutoUpgrade | What if ... you need to restart?

Use AutoUpgrade to:

- Flashback the database
- Revert a plug-in operation (only when data files are copied)
- Revert a non-CDB to PDB conversion (only when data files are copied)

```
java -jar autoupgrade.jar -restore -jobs n
```

If you revert or restore in any other way, you must tell AutoUpgrade

AutoUpgrade | What if ... you need to flash back?

Pre Upgrade Environment	Post Upgrade Environment
CREATE RESTORE POINT **grpt** GUARANTEE FLASHBACK DATABASE;	
UPGRADE	
	SHUTDOWN IMMEDIATE
	STARTUP MOUNT;
	FLASHBACK DATABASE TO RESTORE POINT **grpt**;
	SHUTDOWN IMMEDIATE
STARTUP MOUNT;	
ALTER DATABASE OPEN RESETLOGS;	
DROP RESTORE POINT **grpt**;	

AutoUpgrade | What if … you need to flash back?

Guaranteed Restore Points

```
upg1.source_home=/u01/app/oracle/product/12.2.0.1
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=CDB1
upg1.restoration=yes
upg1.drop_grp_after_upgrade=no
```

- Default behavior:
  - AutoUpgrade creates GRP except for
    - Standard Edition 2
    - `restoration=no`
  - GRP will be kept
  - GRP needs to be removed manually except for
    - `drop_grp_after_upgrade=yes` will only remove it when upgrade completed successfully

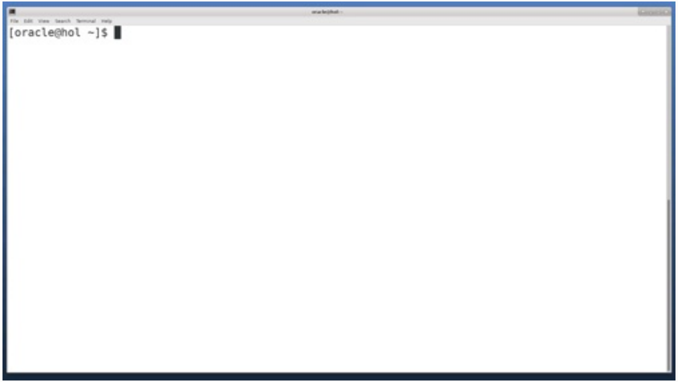
AutoUpgrade | What if ... you need to flash back?

AutoUpgrades handles everything, including

- `/etc/oratab`
- Clusterware registration
- Moving files
  - PFile
  - SPFile
  - Password file
  - Etc.

Pro tip: If restoring a primary database, you must manually handle the standby database

AutoUpgrade | What if ... you need to flash back?



```
[oracle@hol ~]$
```

Watch on YouTube

AutoUpgrade | What if ... you need to restart?

If you revert or restore in any other way, you must tell AutoUpgrade

1. Clear recovery data for a specific job

```
java -jar autoupgrade.jar -config DB.cfg -clear_recovery_data -job n
```

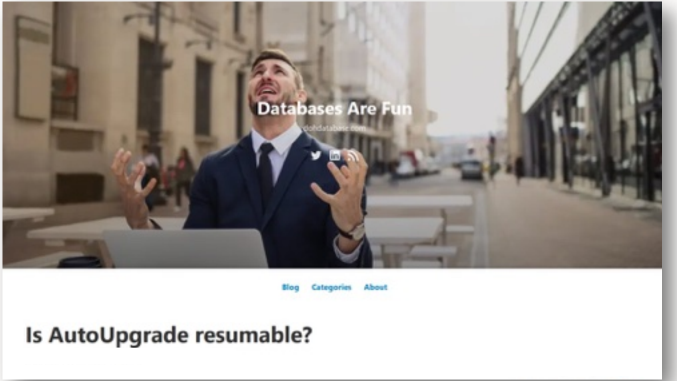
2. Clear all recovery data

```
java -jar autoupgrade.jar -config DB.cfg -clear_recovery_data
```

3. Remove AutoUpgrade log directories - use with caution

```
global.autoupg_log_dir=/u01/app/oracle/cfgtoollogs/autoupgrade
upg1.log_dir=/u01/app/oracle/admin/DB1/upglogs

rm -rf /u01/app/oracle/cfgtoollogs/autoupgrade
rm -rf /u01/app/oracle/admin/DB1/upglogs
```

AutoUpgrade | What if … you need to restart?



Databases Are Fun

dohdatabase.com

Is AutoUpgrade resumable?

https://dohdatabase.com/is-autoupgrade-resumable/

AutoUpgrade | What if … you need many

Can you start many instances of AutoUpgrade? Yes!

Each AutoUpgrade instance must have its own logging directory

First AutoUpgrade instance

```
$ more CDB1.cfg
global.autoupg_log_dir=/tmp/AutoUpgrade1
...

$ java -jar autoupgrade.jar \
   -config CDB1.cfg \
   -mode analyze
```

Second AutoUpgrade instance

```
$ more CDB2.cfg
global.autoupg_log_dir=/tmp/AutoUpgrade2
...

$ java -jar autoupgrade.jar \
   -config CDB2.cfg \
   -mode analyze
```

AutoUpgrade | What if ... you are using Fail Safe?

Like any other upgrade

1. Remove database from all cluster groups
2. Upgrade with AutoUpgrade
3. Add back to cluster groups

AutoUpgrade | What if ... Database Vault

MOS notes

Requirement for Upgrading Database with Database Vault (Doc ID 2757126.1)

AutoUpgrade | What if ... AutoUpgrade fails

1. Create zip file

```
$ java -jar autoupgrade.jar -config config.cfg -zip
```

2. Optionally, add `opatch lsinventory`

```
$ $ORACLE_HOME/OPatch/opatch lsinventory > opatch.txt
$ zip -r AUPG_210419_0735_461.zip opatch.txt
```

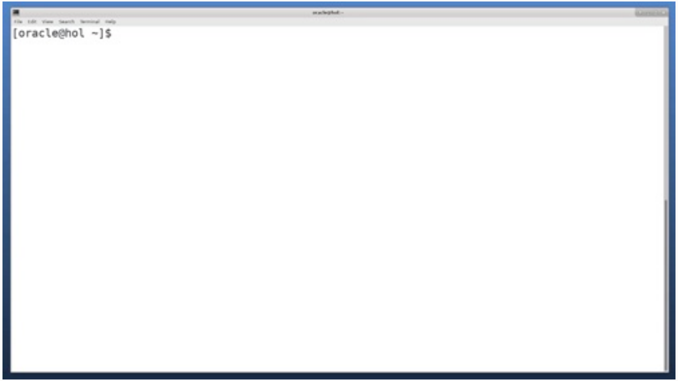
3. Upload it to My Oracle Support

My Oracle Support

Welcome

Search our Knowledge base and engage Communities to learn about products, services, and to find help resolving issues

AutoUpgrade | What if ... AutoUpgrade fails



```
[oracle@hol ~]$
```

Watch on YouTube

AutoUpgrade | What if ... AutoUpgrade fails

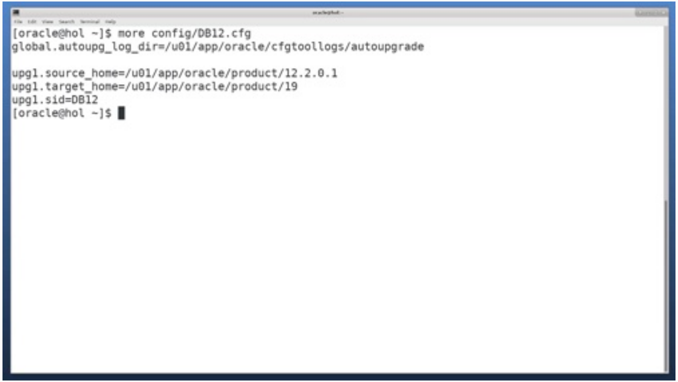
What does an error mean?

```
$ java -jar autoupgrade.jar -error_code UPG-1400

ERROR1400.ERROR  = UPG-1400
ERROR1400.CAUSE  = Database upgrade failed with errors
```

Omit the error code and get a list of all error codes

```
$ java -jar autoupgrade.jar -error_code

ERROR1000.ERROR  = UPG-1000
ERROR1000.CAUSE  = It was not possible to create the data file where the jobsTable is being written or there was a
problem during the writing, it might be thrown due to a permission error or a busy resource scenario

ERROR1001.ERROR  = UPG-1001
ERROR1001.CAUSE  = There was a problem reading the state file perhaps there was corruption writing the file and in
the next write it might be fixed
.
.
```

AutoUpgrade | What if … AutoUpgrade fails



```
[oracle@hol ~]$ more config/DB12.cfg
global.autoupg_log_dir=/u01/app/oracle/cfgtoollogs/autoupgrade

upg1.source_home=/u01/app/oracle/product/12.2.0.1
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=DB12
[oracle@hol ~]$
```

Watch on YouTube

Daniel Overby Hansen

Lead Developer
SimCorp A/S - Denmark

"After qualifying the new AutoUpgrade tool on a representative portion of our database landscape we found that tool was doing a great job and is production-ready. In our automation tool we have removed a lot of "home-grown" code and replaced it with AutoUpgrade functionality.

Since August 2019 all upgrades at SimCorp have been executed using the AutoUpgrade tool."

Photo by Ales Krivec on Unsplash

After Upgrade

Important things to
do after the upgrade completed
successfully

Things to do right after upgrade

Configure statistics history retention period

- Check space usage:

```
SQL> select space_usage_kbytes/1024 mb
from v$sysaux_occupants where
occupant_name='SM/OPTSTAT';
```

- Check retention
  - Default: 31 days

```
SQL> select dbms_stats.get_stats_history_retention from dual;
```

- Adjust setting
  - Example: 10 days

```
SQL> exec dbms_stats.alter_stats_history_retention(10);
```

Time Zone | Facts

Upgrade and potential adjustment of `TIMESTAMP WITH TIME ZONE` data type

- Use `utltz*` scripts or DBMS_DST directly

Time zone can't be downgraded

Time zone upgrade may take long

Files are in `$ORACLE_HOME/oracore/zoneinfo`

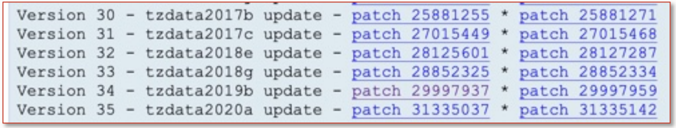
Default time zone file

Database Release	Default TZ Version
11.2.0.2 - 11.2.0.4	DST V14
12.1.0.1, 12.1.0.2	DST V18
12.2.0.1	DST V26
18c	DST V31
19c	DST V32
21c	DST V35

Time Zone | Upgrade

Recommendation

- Apply most recent time zone patch before upgrade
  - MOS Note:412160.1
  - Not RAC rolling!



```
Version 30 - tzdata2017b update - patch 25881255 * patch 25881271
Version 31 - tzdata2017c update - patch 27015449 * patch 27015468
Version 32 - tzdata2018e update - patch 28125601 * patch 28127287
Version 33 - tzdata2018g update - patch 28852325 * patch 28852334
Version 34 - tzdata2019b update - patch 29997937 * patch 29997959
Version 35 - tzdata2020a update - patch 31335037 * patch 31335142
```

- AutoUpgrade adjusts time zone
  - Default: `upg1.timezone_upg=yes`
  - If *DST-source* > *DST-target*, AutoUpgrade copies necessary files

- Or upgrade manually with scripts
  - `?/rdbms/admin/utltz_countstar.sql`
  - `?/rdbms/admin/utltz_upg_check.sql`
  - `?/rdbms/admin/utltz_upg_apply.sql`

Time Zone | Upgrade Multitenant

Check CDB$ROOT and PDB$SEED

```
perl catcon.pl -n 1 -s -l /home/oracle -b utltz_upg_check_ROOT_SEED
               -d /u01/app/oracle/product/19/rdbms/admin utltz_upg_check.sql
```

Then check PDBs

```
perl catcon.pl -n 1 -S -l /home/oracle -b utltz_upg_check_PDBs
               -d /u01/app/oracle/product/19/rdbms/admin utltz_upg_check.sql
```

Finally, upgrade time zone

```
perl catcon.pl -n 1 -s -l /home/oracle -b utltz_upg_apply_ROOT_SEED
               -d /u01/app/oracle/product/19/rdbms/admin utltz_upg_apply.sql
```

```
SQL> alter pluggable database all open;
```

```
perl catcon.pl -n 1 -S -l /home/oracle -b utltz_upg_apply_PDBs
               -d /u01/app/oracle/product/19/rdbms/admin utltz_upg_apply.sql
```

- How to patch all my PDBs with a new time zone file

Time Zone | Improve Performance

Adjust parallel degree of tables

- `ALTER TABLE TAB1 PARALLEL 16;`
- `PARALLEL_DEGREE_POLICY=AUTO`

Use DBMS_DST directly

- You can control table/MV upgrade
- You can parallelize the table/MV upgrades

Purge statistics history

Purge scheduler logs

Time Zone | Issues and Workaround

`DBMS_DST.FIND_AFFECTED_TABLES` doesn't run parallel

- Bug 30681085
- Fixed in 21c and in 19.11.0
- Manual workaround:
  - Edit `?/rdbms/admin/utltz_upg_check.sql`
  - Add:

```
BEGIN
    DBMS_DST.FIND_AFFECTED_TABLES (AFFECTED_TABLES => 'SYS.DST$AFFECTED_TABLES',
                                   LOG_ERRORS => TRUE,
                                   LOG_ERRORS_TABLE => 'SYS.DST$ERROR_TABLE',
                                   PARALLEL => TRUE);
  EXCEPTION ...
```

Time Zone | Issues and Workaround

Applying time zone changes is slow

- Bug 31940092
- Fixed in 21c and in 19.11.0
- Manual workaround:
  - Edit `?/rdbms/admin/utltz_upg_apply.sql`
  - Remove or comment out:

```
-- Alter sessions to avoid (performance) issues
ALTER SESSION SET nls_sort               = 'BINARY';
-- ALTER SESSION SET "_with_subquery"       = 'MATERIALIZE';
ALTER SESSION SET "_simple_view_merging" = TRUE;
```

Time Zone | Oracle 21: "Online" Upgrade of TZ Data

New init.ora Parameter in Oracle 21c

- `TIMEZONE_VERSION_UPGRADE_ONLINE=TRUE`

No or minimal locks

Still requires 1 restart of the database

- But not several restarts anymore
- No `STARTUP UPGRADE` required

When `DBMS_DST.UPGRADE_DATABASE` is used:

- TSTZ operation will be done online whenever possible
- Otherwise, an exclusive DML lock will be acquired

Before Oracle 21c:

```
SQL> EXEC DBMS_DST.BEGIN_UPGRADE(35);
BEGIN DBMS_DST.BEGIN_UPGRADE(35); END;

*
ERROR at line 1:
ORA-56926: database must be in UPGRADE mode in order
           to start an upgrade window
ORA-06512: at "SYS.DBMS_SYS_ERROR", line 79
ORA-06512: at "SYS.DBMS_DST", line 1240
ORA-06512: at line 1
```

Documentation: Oracle 21c Database Globalization Support Guide, Chapter 4.7.1

NEW IN 21c

Post Upgrade | Unified Audit Trail

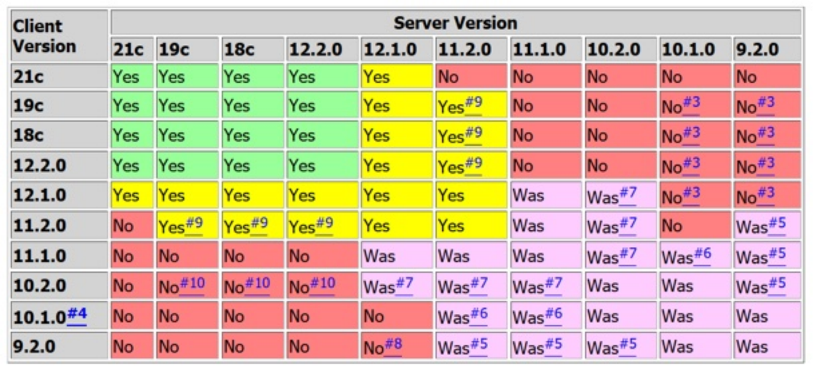
- Internal structure changed to improve query performance of `UNIFIED_AUDIT_TRAIL` view
  - Old records must be converted to new format
  - To convert:

```
SQL> EXEC DBMS_AUDIT_MGMT.TRANSFER_UNIFIED_AUDIT_RECORDS;
```

  - Only applicable if upgrading from Oracle Database 12.1

More Information | Client Connectivity

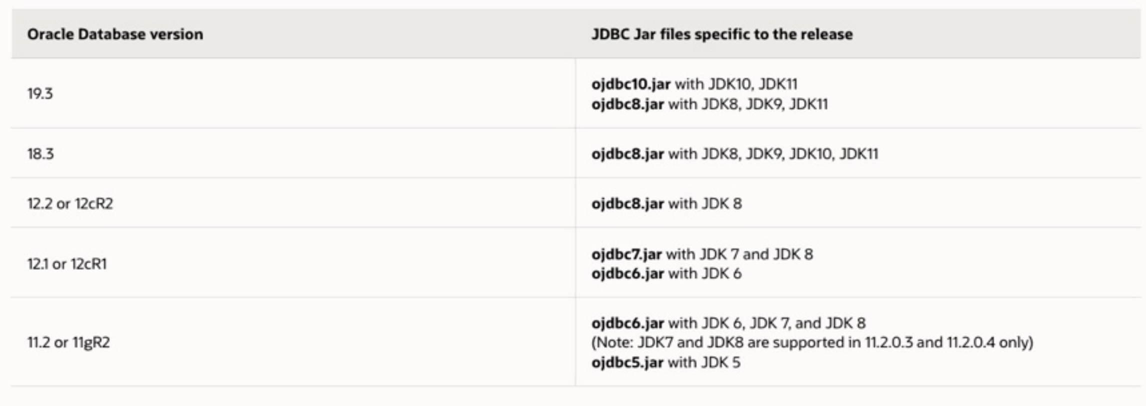
MOS Note: 2073031
Client / Server Interoperability Support Matrix for Different Oracle Versions



Client Version	Server Version									
	21c	19c	18c	12.2.0	12.1.0	11.2.0	11.1.0	10.2.0	10.1.0	9.2.0
21c	Yes	Yes	Yes	Yes	Yes	No	No	No	No	No
19c	Yes	Yes	Yes	Yes	Yes	Yes[#9]	No	No	No[#3]	No[#3]
18c	Yes	Yes	Yes	Yes	Yes	Yes[#9]	No	No	No[#3]	No[#3]
12.2.0	Yes	Yes	Yes	Yes	Yes	Yes[#9]	No	No	No[#3]	No[#3]
12.1.0	Yes	Yes	Yes	Yes	Yes	Yes	Was	Was[#7]	No[#3]	No[#3]
11.2.0	No	Yes[#9]	Yes[#9]	Yes[#9]	Yes	Yes	Was	Was[#7]	No	Was[#5]
11.1.0	No	No	No	No	Was	Was	Was	Was[#7]	Was[#6]	Was[#5]
10.2.0	No	No[#10]	No[#10]	No[#10]	Was[#7]	Was[#7]	Was[#7]	Was	Was	Was[#5]
10.1.0[#4]	No	No	No	No	No	Was[#6]	Was[#6]	Was	Was	Was
9.2.0	No	No	No	No	No[#8]	Was[#5]	Was[#5]	Was[#5]	Was	Was

More Information | JDBC/JDK Connectivity

Oracle JDBC FAQ
https://www.oracle.com/technetwork/database/enterprise-edition/jdbc-faq-090281.html



Oracle Database version	JDBC Jar files specific to the release
19.3	**ojdbc10.jar** with JDK10, JDK11 **ojdbc8.jar** with JDK8, JDK9, JDK11
18.3	**ojdbc8.jar** with JDK8, JDK9, JDK10, JDK11
12.2 or 12cR2	**ojdbc8.jar** with JDK 8
12.1 or 12cR1	**ojdbc7.jar** with JDK 7 and JDK 8 **ojdbc6.jar** with JDK 6
11.2 or 11gR2	**ojdbc6.jar** with JDK 6, JDK 7, and JDK 8 (Note: JDK7 and JDK8 are supported in 11.2.0.3 and 11.2.0.4 only) **ojdbc5.jar** with JDK 5

Things to know

Best Practices and things to take care on

Oracle 19c | Multimedia Removal

> *Oracle Multimedia is desupported in Oracle Database 19c, and the implementation is removed.*

Database 19c Upgrade Guide

- API is removed, component (ORDIM) still exist
- If not in use, recommended to remove before upgrade
- Oracle Locator still exists and works
- Blog post: Simple migration from Oracle multimedia to secure-file blob data type

Oracle 19c | Multimedia Removal

More information

- https://mikedietrichde.com/2019/02/18/oracle-multimedia-will-be-removed-in-oracle-database-19c/
- MOS Note: 2347372.1 – Oracle Multimedia Statement of Direction
- MOS Note: 2375644.1 – How To Migrate Data From Oracle Multimedia Data Types to BLOB columns

Oracle 19c | Streams Removal

> *Starting in Oracle Database 19c (19.1), Oracle Streams is desupported. Oracle GoldenGate is the replication solution for Oracle Database.*

Database 19c Upgrade Guide

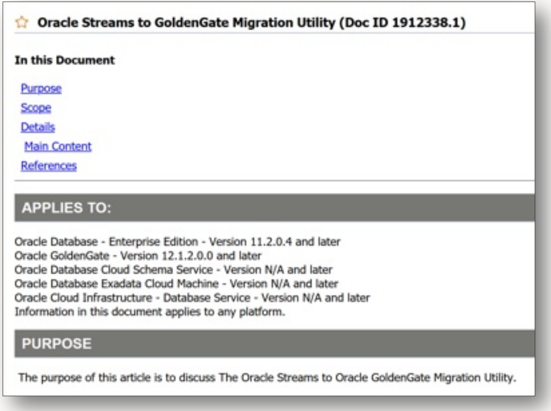
- Oracle Database Advanced Queuing is not deprecated
- Fully supported in Oracle Database 19c

Oracle 19c | Streams Removal

How to migrate from Streams to GoldenGate

[Oracle Streams to Oracle GoldenGate Conversion (Doc ID 1383303.1)]

[Oracle Streams to GoldenGate Migration Utility (Doc ID 1912338.1)]



☆ **Oracle Streams to GoldenGate Migration Utility (Doc ID 1912338.1)**

In this Document

- Purpose
- Scope
- Details
  - Main Content
- References

APPLIES TO:

Oracle Database - Enterprise Edition - Version 11.2.0.4 and later
Oracle GoldenGate - Version 12.1.2.0.0 and later
Oracle Database Cloud Schema Service - Version N/A and later
Oracle Database Exadata Cloud Machine - Version N/A and later
Oracle Cloud Infrastructure - Database Service - Version N/A and later
Information in this document applies to any platform.

PURPOSE

The purpose of this article is to discuss The Oracle Streams to Oracle GoldenGate Migration Utility.

Oracle 19c | DBMS_JOB Behavior Change

> *Oracle continues to support the DBMS_JOB package. However, you must grant the CREATE JOB privilege to the database schemas that submit DBMS_JOB jobs.*

> *Oracle Scheduler replaces the DBMS_JOB package. Although DBMS_JOB is still supported for backward compatibility, Oracle strongly recommends that you switch from DBMS_JOB to Oracle Scheduler.*

Database 19c Upgrade Guide

- During and after upgrade, for each job in `DBMS_JOB` a corresponding entry will be created with `DBMS_SCHEDULER`

Oracle 19c | DBMS_JOB Behavior Change

> *Oracle continues to support the DBMS_JOB package. However, you must grant the CREATE JOB privilege to the database schemas that submit DBMS_JOB jobs.*

> *Oracle Scheduler replaces the DBMS_JOB package. Although DBMS_JOB is still supported for backward compatibility, Oracle strongly recommends that you switch from DBMS_JOB to Oracle Scheduler.*

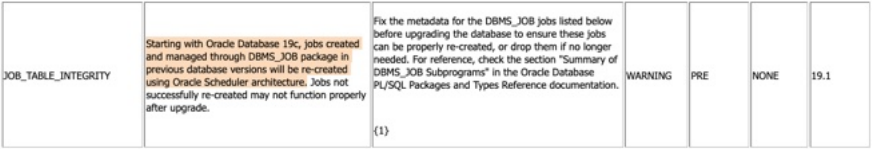
Database 19c Upgrade Guide

- During and after upgrade, for each job in `DBMS_JOB` a corresponding entry will be created with `DBMS_SCHEDULER`

Oracle 19c | DBMS_JOB Behavior Change

More information

- The check (JOB_TABLE_INTEGRITY) in `preupgrade.jar` is only checking for inconsistencies or any issues



JOB_TABLE_INTEGRITY	Starting with Oracle Database 19c, jobs created and managed through DBMS_JOB package in previous database versions will be re-created using Oracle Scheduler architecture. Jobs not successfully re-created may not function properly after upgrade.	Fix the metadata for the DBMS_JOB jobs listed below before upgrading the database to ensure these jobs can be properly re-created, or drop them if no longer needed. For reference, check the section "Summary of DBMS_JOB Subprograms" in the Oracle Database PL/SQL Packages and Types Reference documentation. {1}	WARNING	PRE	NONE	19.1

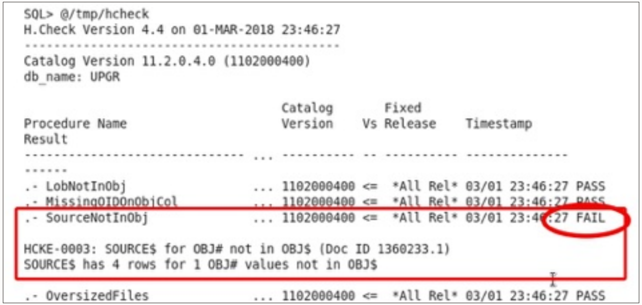
MOS Note: 2380601.1 - Database Preupgrade tool check list

- https://mikedietrichde.com/2019/05/24/dbms_job-behavior-change-in-oracle-19c-during-upgrade/

Health Check | hcheck.sql

If your database is highly important, do a health check

- Lightweight, non-intrusive script
- Checks consistency of selected dictionary relationships
- hcheck.sql - Script to Check for Known Problems (Doc ID 136697.1)



```
SQL> @/tmp/hcheck
H.Check Version 4.4 on 01-MAR-2018 23:46:27
-------------------------------------------
Catalog Version 11.2.0.4.0 (1102000400)
db_name: UPGR

                                   Catalog       Fixed
Procedure Name                     Version    Vs Release    Timestamp
Result
------------------------------ ... ---------- -- ---------- --------------
------
.- LobNotInObj                 ... 1102000400 <=  *All Rel* 03/01 23:46:27 PASS
.- MissingOIDOnObjCol          ... 1102000400 <=  *All Rel* 03/01 23:46:27 PASS
.- SourceNotInObj              ... 1102000400 <=  *All Rel* 03/01 23:46:27 FAIL

HCKE-0003: SOURCE$ for OBJ# not in OBJ$ (Doc ID 1360233.1)
SOURCE$ has 4 rows for 1 OBJ# values not in OBJ$

.- OversizedFiles              ... 1102000400 <=  *All Rel* 03/01 23:46:27 PASS
```

AutoUpgrade | Gathering Fixed Objects Stats

> *After an upgrade, or after other database configuration changes, Oracle strongly recommends that you regather fixed object statistics after you have run representative workloads on Oracle Database.*

Database 19c Upgrade Guide, chapter 7

Never run it right after upgrade

AutoUpgrade | Gathering Fixed Objects Stats

Ask yourself: Do you remember this?

If not, DBMS_SCHEDULER to the rescue

AutoUpgrade | Gathering Fixed Objects Stats

1. Create a .sql script

```
BEGIN
   DBMS_SCHEDULER.CREATE_JOB (
      job_name => '"SYS"."GATHER_FIXED_OBJECTS_STATS_ONE_TIME"',
           job_type => 'PLSQL_BLOCK',
           job_action => 'BEGIN DBMS_STATS.GATHER_FIXED_OBJECTS_STATS; END;',
           start_date => SYSDATE+7,
           auto_drop => TRUE,
           comments => 'Gather fixed objects stats after upgrade - one time'
   );
   DBMS_SCHEDULER.ENABLE (
      name => '"SYS"."GATHER_FIXED_OBJECTS_STATS_ONE_TIME"'
   );
END;
/
```

AutoUpgrade | Gathering Fixed Objects Stats

2. Create a .sh script

```
$ORACLE_HOME/perl/bin/perl $ORACLE_HOME/rdbms/admin/catcon.pl \
   -n 4 -e \
   -C 'PDB$SEED' \
   -b sched_gfos -d /home/oracle/sched_gfos/ sched_gfos.sql
```

3. Execute .sh script after upgrade

```
upg1.after_action=/home/oracle/sched_gfos/sched_gfos.sh
```

Further information and non-CDB example in blog post

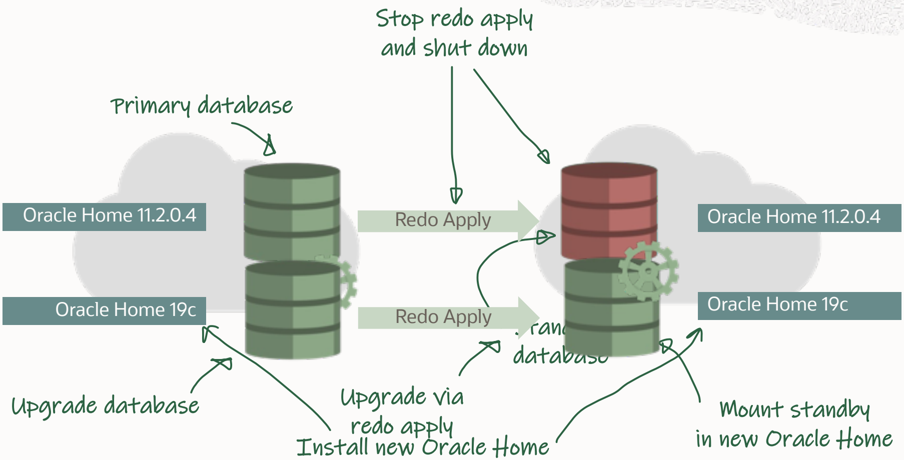
AutoUpgrade | Gathering Fixed Objects Stats

Fixed Objects Statistics (GATHER_FIXED_OBJECTS_STATS) Considerations (Doc ID 798257.1)

Photo by Adam Muise on Unsplash

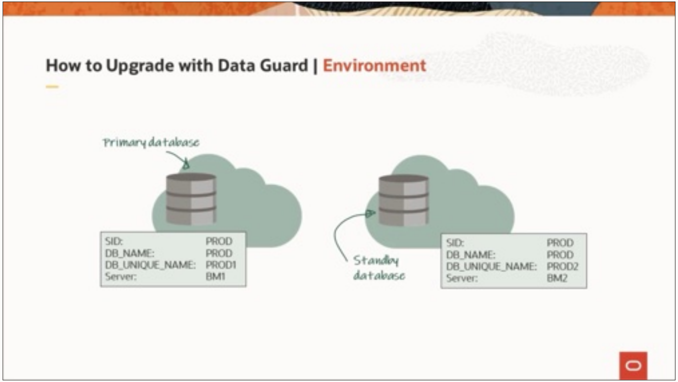
Upgrade with Physical Standbys in place

Upgrade with Data Guard | Concept



Remember use latest
Release Update

Upgrade with Data Guard | Demo



How to Upgrade with Data Guard | Environment

Primary database

SID:	PROD
DB_NAME:	PROD
DB_UNIQUE_NAME:	PROD1
Server:	BM1

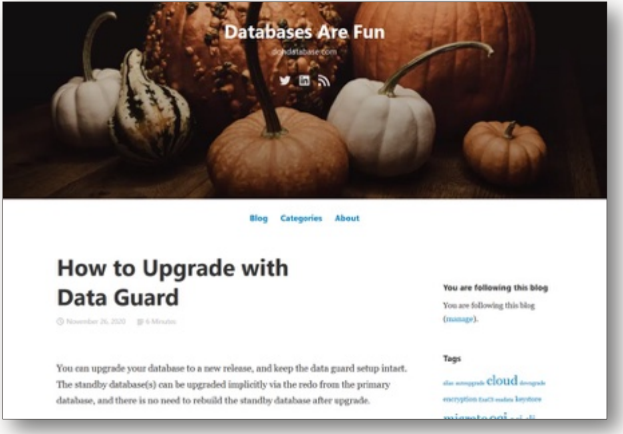
Standby database

SID:	PROD
DB_NAME:	PROD
DB_UNIQUE_NAME:	PROD2
Server:	BM2

Upgrade with Data Guard | Redo Apply

Keep standby database in MOUNT mode
until the redo from the upgrade has been applied

Upgrade with Data Guard | Blog Post



Databases Are Fun

dohdatabase.com

How to Upgrade with Data Guard

November 26, 2020 6 Minutes

You can upgrade your database to a new release, and keep the data guard setup intact. The standby database(s) can be upgraded implicitly via the redo from the primary database, and there is no need to rebuild the standby database after upgrade.

You are following this blog

You are following this blog (manage).

Tags

[illegible] autoupgrade cloud downgrade encryption ExaCS exadata keystore migrate to oci [illegible]

https://dohdatabase.com/2020/11/26/how-to-upgrade-with-data-guard/

Photo by Tim Mossholder on Unsplash

Alternative Upgrade Options

Any other ways to upgrade databases?

Alternative Upgrade Options | Command Line

`preupgrade.jar`

- Download the newest from MOS Note: 884522.1
- You need to do everything by yourself
- Not supported anymore from 21c
  - `java -jar autoupgrade.jar -preupgrade "target_version=21"`

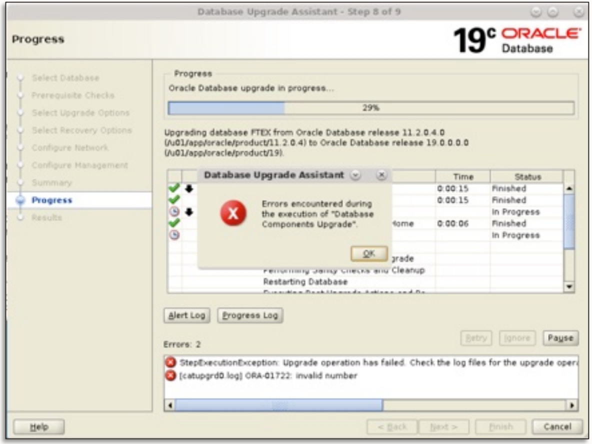
`dbupgrade`

- Can be used without parameters
- Wrapper for `catctl.pl` with default settings

Alternative Upgrade Options | GUI

Database Upgrade Assistant

- Not our choice
- Not fully resumable
  - See Blog Post



Progress

Oracle Database upgrade in progress...

Upgrading database FTEX from Oracle Database release 11.2.0.4.0 (/u01/app/oracle/product/11.2.0.4) to Oracle Database release 19.0.0.0.0 (/u01/app/oracle/product/19).

	Time	Status
	0:00:15	Finished
	0:00:15	Finished
		In Progress
...Home	0:00:06	Finished
		In Progress
...grade		
Performing Sanity Checks and Cleanup		
Restarting Database		

Database Upgrade Assistant

Errors encountered during the execution of "Database Components Upgrade".

Errors: 2

```
StepExecutionException: Upgrade operation has failed. Check the log files for the upgrade oper
[catupgrd0.log] ORA-01722: invalid number
```

Photo by David Maunsell on Unsplash

Fallback Strategies

For Database Upgrades

Fallback | Database Downgrade

Works also days after the upgrade without losing any changes

- `COMPATIBLE` must not be changed

- If timezone file was upgraded, same timezone file must be present in old Oracle Home

Fallback | Database Downgrade

In new, higher release Oracle Home (e.g., 19c)

```
SQL> startup downgrade
SQL> @?/rdbms/admin/catdwgrd.sql
SQL> shutdown immediate
```

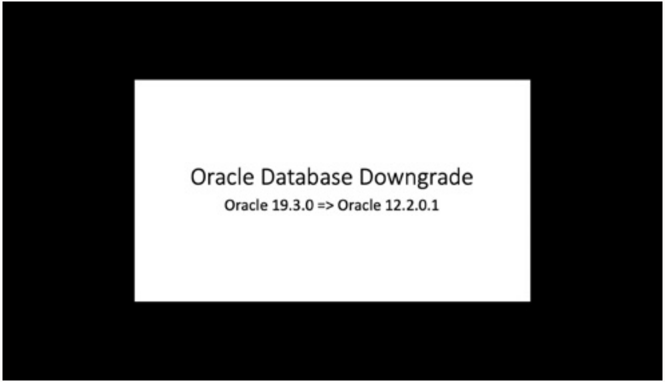
Start database in old, lower release Oracle Home (e.g., 11.2.0.4)

```
SQL> startup upgrade
SQL> @?/rdbms/admin/catrelod.sql
```

In addition,

- Recompile object (`utlrp`)
- Run `datapatch`
- Gather stats (dictionary, fixed objects and optimizer)
- Grid Infrastructure and /etc/oratab

Fallback | Database Downgrade


Oracle Database Downgrade

Oracle 19.3.0 => Oracle 12.2.0.1

Watch on YouTube

Fallback | Database Downgrade

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

The data dictionary will be different - but compatible

Examples:

- New table is not dropped, but truncated
- New index is not dropped
- Generally, dropping is avoid

Fallback | Flashback Database

Puts the database back into the state it had before the upgrade

- Fast - typically an upgrade is reverted in 5-10 minutes
- Standard technique in AutoUpgrade
- Requires Enterprise Edition
- Requires ARCHIVELOG mode
- 10-20 GB of Flashback logs
- `COMPATIBLE` must not be changed

Fallback | Flashback Database

Pre Upgrade Environment	Post Upgrade Environment
CREATE RESTORE POINT **grpt** GUARANTEE FLASHBACK DATABASE;	
UPGRADE	
	SHUTDOWN IMMEDIATE
	STARTUP MOUNT;
	FLASHBACK DATABASE TO RESTORE POINT **grpt**;
	SHUTDOWN IMMEDIATE
STARTUP MOUNT;	
ALTER DATABASE OPEN RESETLOGS;	
DROP RESTORE POINT **grpt**;	

Fallback | Flashback Database

Guaranteed Restore Points

```
upg1.source_home=/u01/app/oracle/product/12.2.0.1
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=CDB1
upg1.restoration=yes
upg1.drop_grp_after_upgrade=no
```

- Default behavior:
  - AutoUpgrade creates GRP except for
    - Standard Edition 2
    - `restoration=no`
  - GRP will be kept
  - GRP needs to be removed manually except for
    - `drop_grp_after_upgrade=yes` will only remove it when upgrade completed successfully

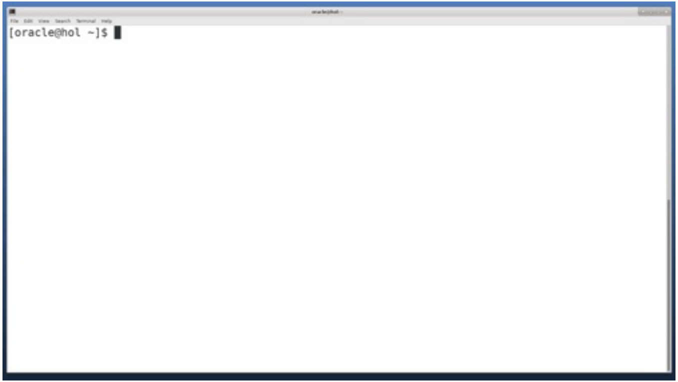
Fallback | Flashback Database

Use AutoUpgrade to:

- Flashback the database
- Revert a plug-in operation (only when data files are copied)
- Revert a non-CDB to PDB conversion (only when data files are copied)

```
java -jar autoupgrade.jar -restore -jobs n
```

Fallback | Flashback Database



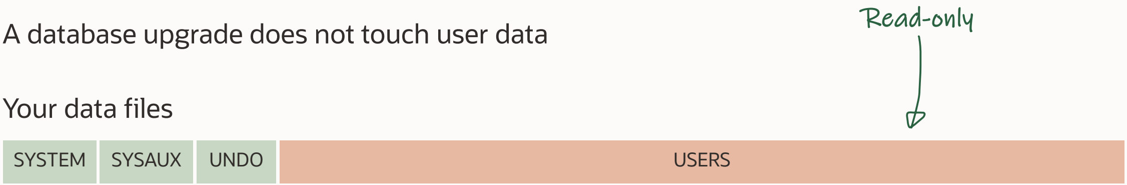
```
[oracle@hol ~]$ 
```

Watch on YouTube

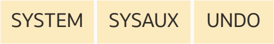
Fallback | Partial Offline Backup

A database upgrade does not touch user data

Your data files



Partial offline backup (plus redo log and control files)



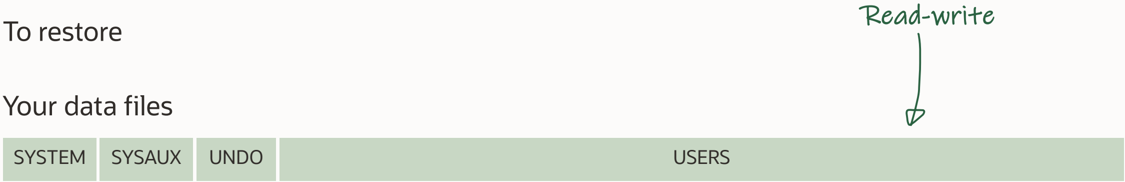
Start upgrade

Pro tip: Works for SE2 and databases in NOARCHIVELOG mode

Fallback | Partial Offline Backup

To restore

Your data files



Your backup (plus redo log and control files)

SYSTEM | SYSAUX | UNDO

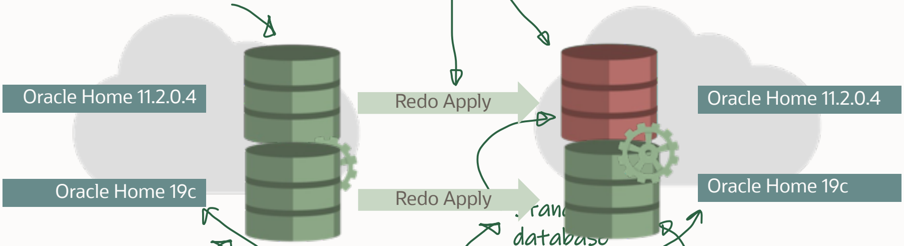
Fallback | Partial Offline Backup

Oracle Database Downgrade with Partial Offline Backup

Oracle 18.3.0 => Oracle 11.2.0.4

Watch on YouTube

Fallback | **Data Guard**



Stop redo apply
and shut down

Primary database

Standby database

Upgrade database

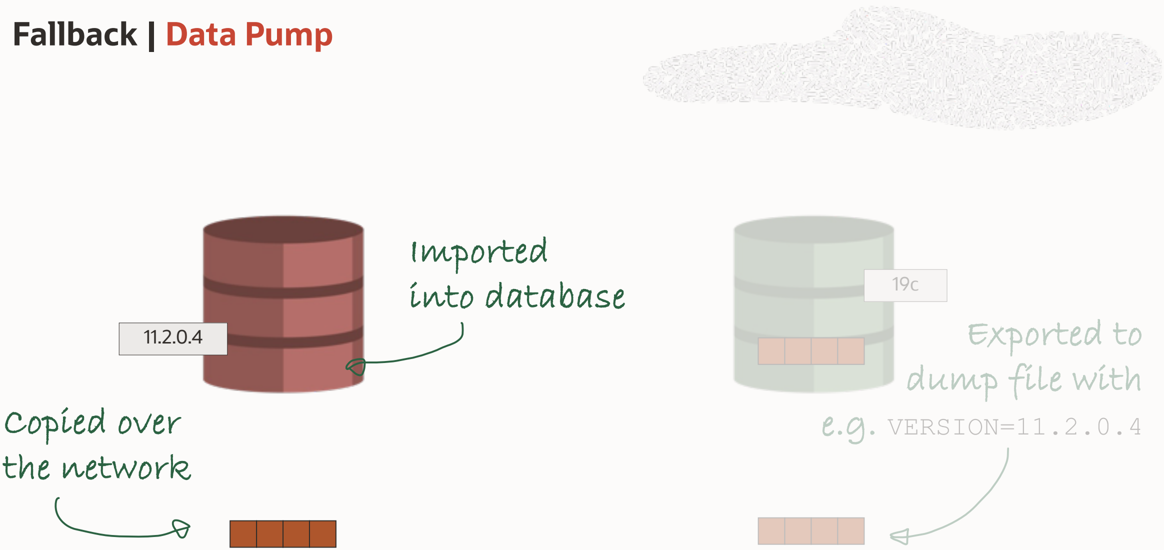
Upgrade via
redo apply

Install new Oracle Home

Mount standby
in new Oracle Home

Remember use latest
Release Update

Fallback | Data Pump



Fallback | Data Pump

To create a dump file compatible with a lower release

```
version=11.2.0.4
```

Other options are

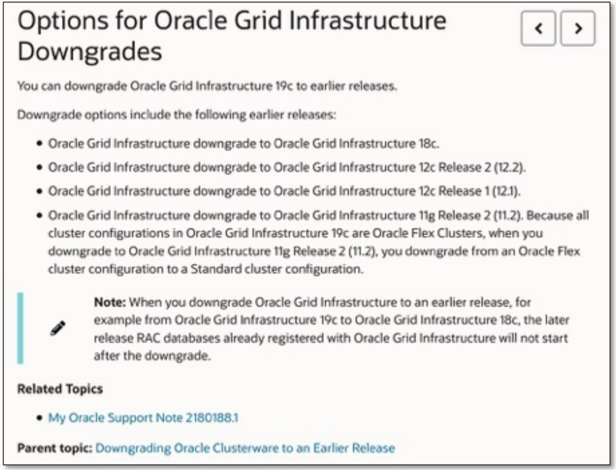
- `COMPATIBLE`
- `LATEST`

[Export/Import DataPump Parameter VERSION - Compatibility of Data Pump Between Different Oracle Versions (Doc ID 553337.1)]

Pro tip: Read more about VERSION in the documentation

Fallback | Grid Infrastructure Downgrade

- Documentation



Options for Oracle Grid Infrastructure Downgrades

You can downgrade Oracle Grid Infrastructure 19c to earlier releases.

Downgrade options include the following earlier releases:

- Oracle Grid Infrastructure downgrade to Oracle Grid Infrastructure 18c.
- Oracle Grid Infrastructure downgrade to Oracle Grid Infrastructure 12c Release 2 (12.2).
- Oracle Grid Infrastructure downgrade to Oracle Grid Infrastructure 12c Release 1 (12.1).
- Oracle Grid Infrastructure downgrade to Oracle Grid Infrastructure 11g Release 2 (11.2). Because all cluster configurations in Oracle Grid Infrastructure 19c are Oracle Flex Clusters, when you downgrade to Oracle Grid Infrastructure 11g Release 2 (11.2), you downgrade from an Oracle Flex cluster configuration to a Standard cluster configuration.

> **Note:** When you downgrade Oracle Grid Infrastructure to an earlier release, for example from Oracle Grid Infrastructure 19c to Oracle Grid Infrastructure 18c, the later release RAC databases already registered with Oracle Grid Infrastructure will not start after the downgrade.

Related Topics

- My Oracle Support Note 2180188.1

Parent topic: Downgrading Oracle Clusterware to an Earlier Release

Photo by Dušan veverkolog on Unsplash

Want to Know More?

Webinar: AutoUpgrade to Oracle Database 19c

Recording

Slides

Chapter 3
Upgrade Internals

Photo by Rod Long on Unsplash

Database Upgrade Basics

Photo by Daniel Cheung on Unsplash

Photo by Nick Fewings on Unsplash

Database Upgrade

Why is Upgrade so different?

Why Upgrade is Different | Short or Long?

Things that matter a lot

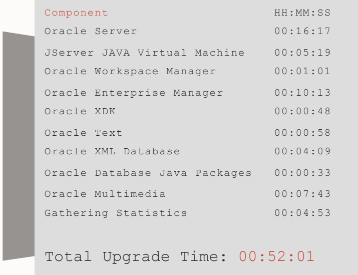
- Number of installed components ⟷ `select COMP_ID, STATUS from CDB_REGISTRY;`
- Size & complexity of dictionary ⟷ `select count(*) from OBJ$;`
- Some feature/version combinations

Things that matter a little

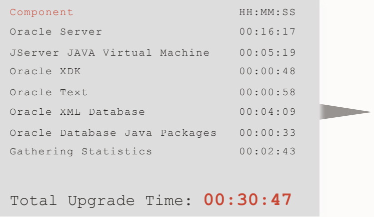
- CPU and disk speed
- SGA/PGA

Things that don't matter

- Amount of user data



```
Component                                   HH:MM:SS
Oracle Server                               00:16:17
JServer JAVA Virtual Machine                00:05:19
Oracle Workspace Manager                    00:01:01
Oracle Enterprise Manager                   00:10:13
Oracle XDK                                  00:00:48
Oracle Text                                 00:00:58
Oracle XML Database                         00:04:09
Oracle Database Java Packages               00:00:33
Oracle Multimedia                           00:07:43
Gathering Statistics                        00:04:53

Total Upgrade Time: 00:52:01
```

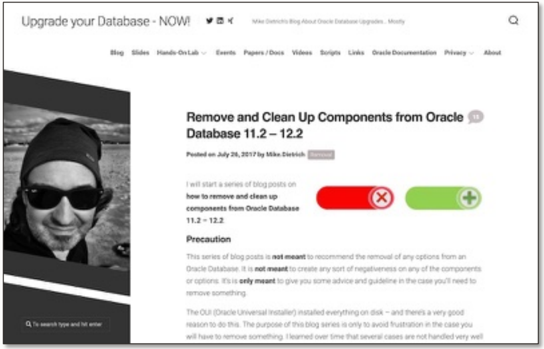


```
Component                                   HH:MM:SS
Oracle Server                               00:16:17
JServer JAVA Virtual Machine                00:05:19
Oracle XDK                                  00:00:48
Oracle Text                                 00:00:58
Oracle XML Database                         00:04:09
Oracle Database Java Packages               00:00:33
Gathering Statistics                        00:02:43

Total Upgrade Time: 00:30:47
```

Why Upgrade is Different | Component Invalid or Removal?

Remove and Clean Up Components

- If you plan to do this, do it BEFORE the upgrade
- Especially components which don't exist in the new release should be removed beforehand



Upgrade your Database - NOW!

Mike Dietrich's Blog About Oracle Database Upgrades... Mostly

Remove and Clean Up Components from Oracle Database 11.2 – 12.2

Posted on July 26, 2017 by Mike.Dietrich Removal

I will start a series of blog posts on **how to remove and clean up components from Oracle Database 11.2 – 12.2**

Precaution

This series of blog posts is **not meant** to recommend the removal of any options from an Oracle Database. It is **not meant** to create any sort of negativeness on any of the components or options. It's is **only meant** to give you some advice and guideline in the case you'll need to remove something.

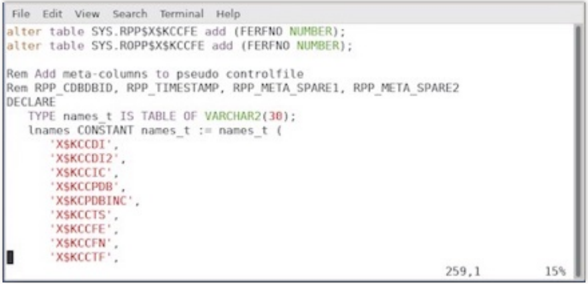
The OUI (Oracle Universal Installer) installed everything on disk – and there's a very good reason to do this. The purpose of this blog series is only to avoid frustration in the case you will have to remove something. I learned over time that several cases are not handled very well

Why Upgrade is Different | DDL Workload

Heavy DDL workload

- A trivial upgrade from 11.2.0.4 to 19c includes approximately
  - 2,500 `ALTER`s (mostly tables, also types and users)
  - 16,500 `CREATE OR REPLACE`s (views, types, synonyms, procedures, packages, functions)
  - 10,200 `GRANT` statements
  - 1150 `CREATE TABLE` statements
  - 360 `CREATE INDEX` statements
  - And more…

Almost exclusively in SYSTEM,
partially also in SYSAUX tablespaces



```
alter table SYS.RPP$X$KCCFE add (FERFNO NUMBER);
alter table SYS.ROPP$X$KCCFE add (FERFNO NUMBER);

Rem Add meta-columns to pseudo controlfile
Rem RPP_CDBDBID, RPP_TIMESTAMP, RPP_META_SPARE1, RPP_META_SPARE2
DECLARE
   TYPE names_t IS TABLE OF VARCHAR2(30);
   lnames CONSTANT names_t := names_t (
      'X$KCCDI',
      'X$KCCDI2',
      'X$KCCIC',
      'X$KCCPDB',
      'X$KCPDBINC',
      'X$KCCTS',
      'X$KCCFE',
      'X$KCCFN',
      'X$KCCTF',
                                                         259,1        15%
```

Why Upgrade is Different | Upgrade Mode

`STARTUP UPGRADE` limits many aspects of the database

- Requires SYSDBA privilege
- Suppresses expected errors
  - Based on object type and error code
    - Example: ORA-955 during CREATE TABLE ("table or view does exist")
- Enforces exclusive access for the upgrade process
  - `CLUSTER_DATABASE=FALSE`
  - No system triggers
  - No AQ
  - No resource manager
  - No AWR
  - etc.

```
ALTER SYSTEM SET _system_trig_enabled=FALSE SCOPE=MEMORY;
Autotune of undo retention is turned off.
ALTER SYSTEM SET _undo_autotune=FALSE SCOPE=MEMORY;
ALTER SYSTEM SET undo_retention=900 SCOPE=MEMORY;
ALTER SYSTEM SET aq_tm_processes=0 SCOPE=MEMORY;
ALTER SYSTEM SET enable_ddl_logging=FALSE SCOPE=MEMORY;
Resource Manager disabled during database migration: plan '' not set
ALTER SYSTEM SET resource_manager_plan='' SCOPE=MEMORY;
ALTER SYSTEM SET recyclebin='OFF' DEFERRED SCOPE=MEMORY;
Resource Manager disabled during database migration
```

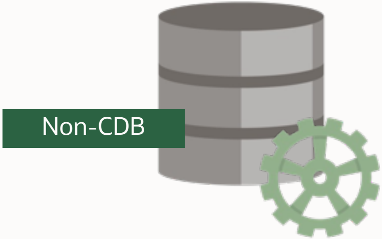
Note:This is an excerpt from the alert.log – these parameters will be set implicitly during a STARTUP UPGRADE

Why Upgrade is Different | Upgrade Mode

What happens during an upgrade?

- Upgrade scripts
    - Example: `c18.sql`, `i18.sql`
- Install scripts
    - Example: `catalog.sql`, `catproc.sql`, `a18.sql`

Parallel Upgrade | Non-CDB

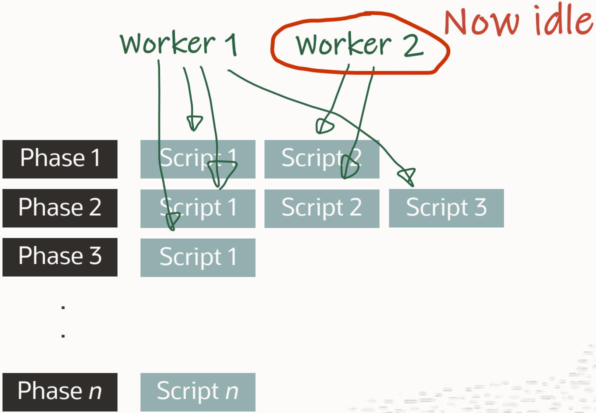


A number of parallel processes

- Minimum 1
- Maximum 8
- Default 4

```
$ dbupgrade -n 2
```

Parallel Upgrade | Non-CDB



Phase 1	Script 1	Script 2	
Phase 2	Script 1	Script 2	Script 3
Phase 3	Script 1		
.			
.			
Phase *n*	Script *n*		

Parallel Upgrade | Non-CDB

Does not scale linear

Contention

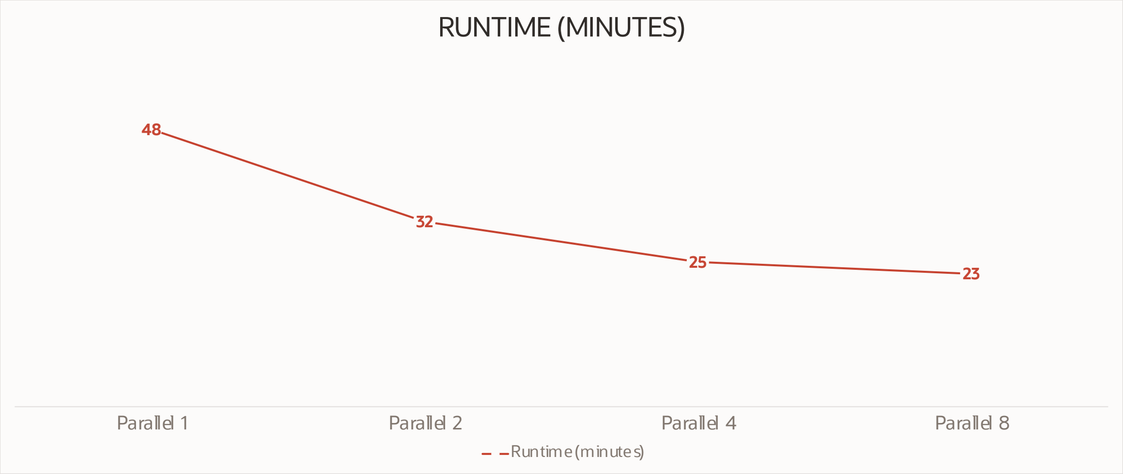
Parallel Upgrade | Benchmark

Bare Metal DB System
36 OCPUs
768 GB memory
NVMe disks

11.2.0.4
8 GB SGA
2 GB PGA
8 CPU_COUNT

AMD
APS
CATALOG
CATJAVA
CATPROC
CONTEXT
EXF
JAVAVM
ORDIM
OWB
OWM
RUL
SDO
XDB
XML
XOQ

Parallel Upgrade | Benchmark



Parallel Upgrade | Container Database

A number of processors are assigned

- Minimum 4
- Maximum unlimited
- Default CPU count

```
$ dbupgrade -n 4
```

Parallel Upgrade | Container Database

Each PDB gets a number of parallel processes

- Minimum 1
- Maximum 8
- Default 2

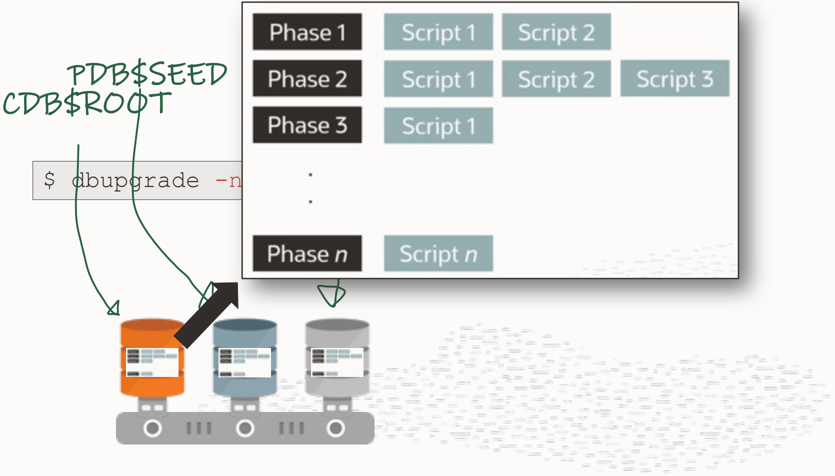
```
$ dbupgrade -N 2
```

Parallel Upgrade | Container Database

But - there is another limit

$$\frac{\text{Total number of processors (n)}}{\text{Processor per PDB (N)}} = \text{PDBs upgraded simultaneously}$$

Parallel Upgrade | Single Tenant



```
$ dbupgrade -n
```

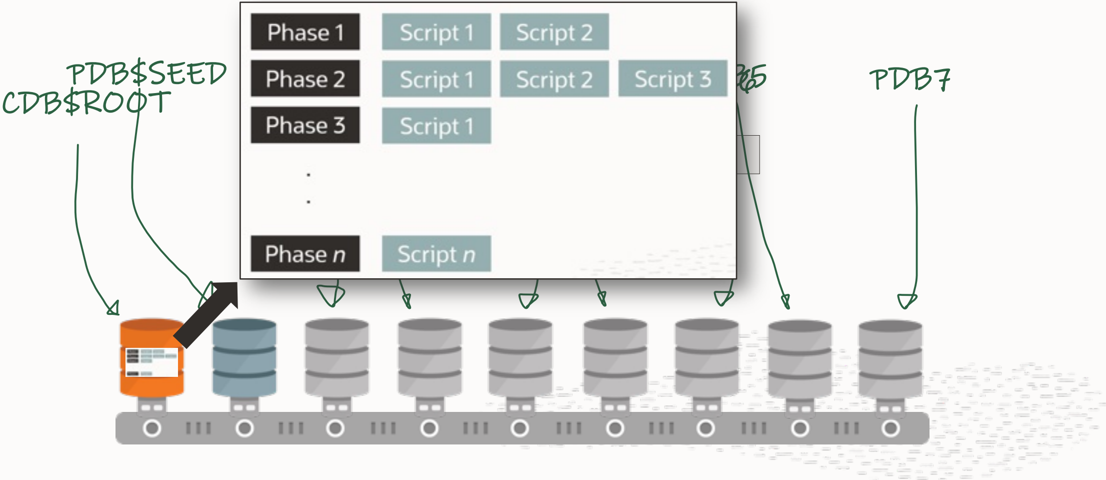
Phase 1	Script 1	Script 2	
Phase 2	Script 1	Script 2	Script 3
Phase 3	Script 1		
.			
.			
Phase *n*	Script *n*		

Parallel Upgrade | Non-CDB vs. Single Tenant

Non-CDB is always faster

than single tenant

Parallel Upgrade | Multitenant

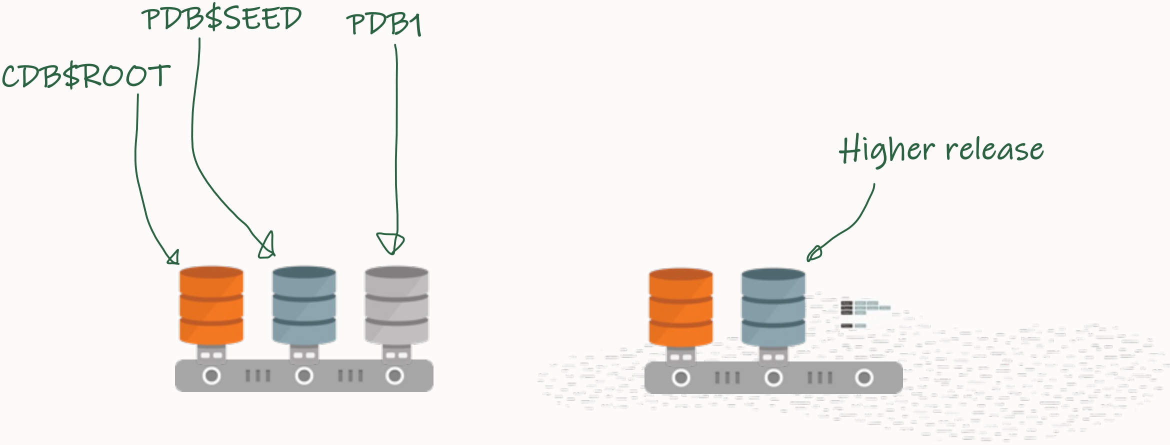


Phase 1	Script 1	Script 2	
Phase 2	Script 1	Script 2	Script 3
Phase 3	Script 1		
.			
.			
Phase *n*	Script *n*		

Parallel Upgrade | Multitenant

Scale by upgrading more PDBs simultaneously

Parallel Upgrade | Unplug-plug Upgrade



Parallel Upgrade | Unplug-plug

Unplug-plug always faster than

Non-CDB
Single Tenant
Multitenant

Parallel Upgrade | How Does It Work

Few PDBs

More processors per PDB

Increase parameter N

Parallel Upgrade | How Does It Work

Many PDBs

Less processors per PDB

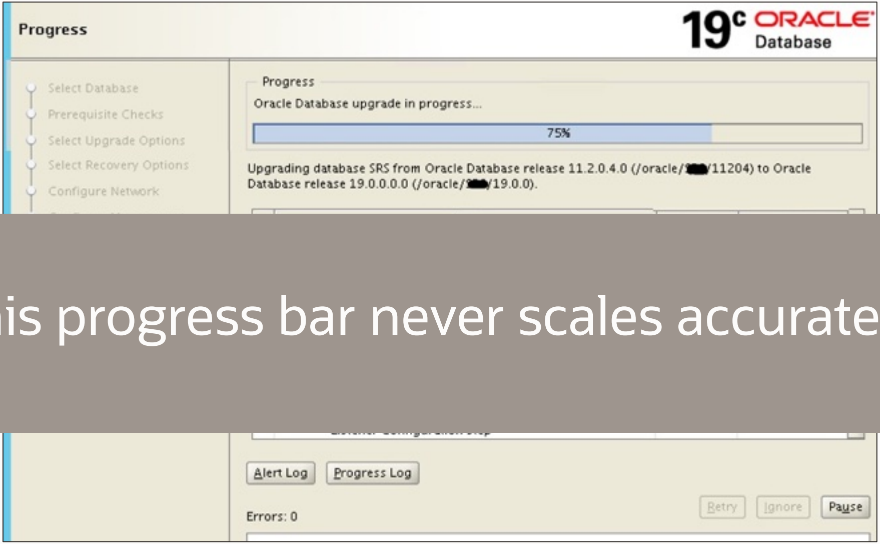
Decrease parameter `N`

Parallel Upgrade | How Does It Work

But as always

It depends

? How long will my upgrade take?



Oracle Database upgrade in progress...

Upgrading database SRS from Oracle Database release 11.2.0.4.0 (/oracle/[redacted]/11204) to Oracle Database release 19.0.0.0.0 (/oracle/[redacted]/19.0.0).

Errors: 0

This progress bar never scales accurately!

⚠ We can't tell you upfront

Basic Facts | How long does an upgrade take?

Things that matter a lot

- Number of installed components
- Size & complexity of data dictionary

Things that matter a little

- CPU and disk speed
- SGA/PGA

Things that don't matter (usually)

- Amount of user data

👍 We can't change the size or complexity of the data dictionary,
but we can check components

Remove desupported components
before upgrade

Oracle 19c | Multimedia Removal

> *Oracle Multimedia is desupported in Oracle Database 19c, and the implementation is removed.*

Database 19c Upgrade Guide

- API is removed, component (ORDIM) still exist
- If not in use, recommended to remove before upgrade
- Oracle Locator still exists and works
- Blog post: Simple migration from Oracle multimedia to secure-file blob data type

Most components can be removed online

Components | All vs. Minimum

Oracle Database 19c by default has 15 components in CDB_REGISTRY

- One set in CDB$ROOT
- One set in PDB$SEED
- One set in each PDB

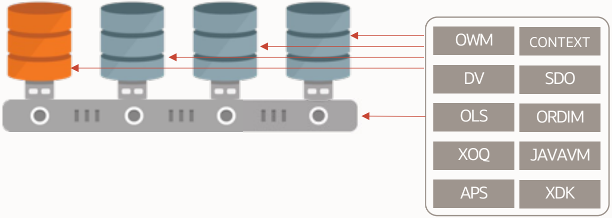
Each component runs upgrade scripts

- Most components upgrade serially, one after another
- Potential contention when many PDB upgrades happen in parallel

Components | Experiment

Container database

- 3 user created PDBs
- PDB$SEED
- Default set of components



Remove component after component

Compare upgrade timings

- 8 CPU cores
- Classic upgrade enforced for parallel processing of PDBs

Components | Experiment

Starting point →

- Oracle Database 19.12.0
- 15 components in each container
- 75 entries in REGISTRY$
- CDB$ROOT gets upgraded always at first
- 4 PDBs including PDB$SEED
  - 4 PDBs upgraded in parallel
  - 2 workers each PDB

End point

- Oracle Database 21.3.0
- 4 components in each container
- 20 entries in REGISTRY$

Components | Experiment

Starting point

```
COMP_ID    COMP_NAME                             STATUS
---------  ------------------------------------  ---------
APS        OLAP Analytic Workspace               VALID
CATALOG    Oracle Database Catalog Views         VALID
CATJAVA    Oracle Database Java Packages         VALID
CATPROC    Oracle Database Packages and Types    VALID
CONTEXT    Oracle Text                           VALID
DV         Oracle Database Vault                 VALID
JAVAVM     JServer JAVA Virtual Machine          VALID
OLS        Oracle Label Security                 VALID
ORDIM      Oracle Multimedia                     VALID
OWM        Oracle Workspace Manager              VALID
RAC        Oracle Real Application Clusters      OPTION OFF
SDO        Spatial                               VALID
XDB        Oracle XML Database                   VALID
XML        Oracle XDK                            VALID
XOQ        Oracle OLAP API                       VALID
```

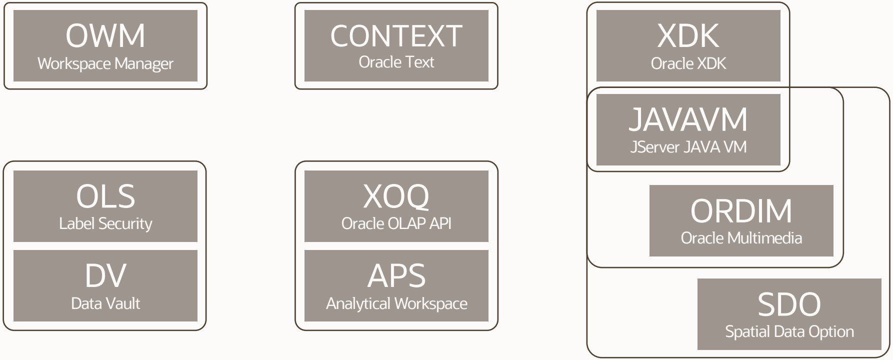
(only one container shown)

End point

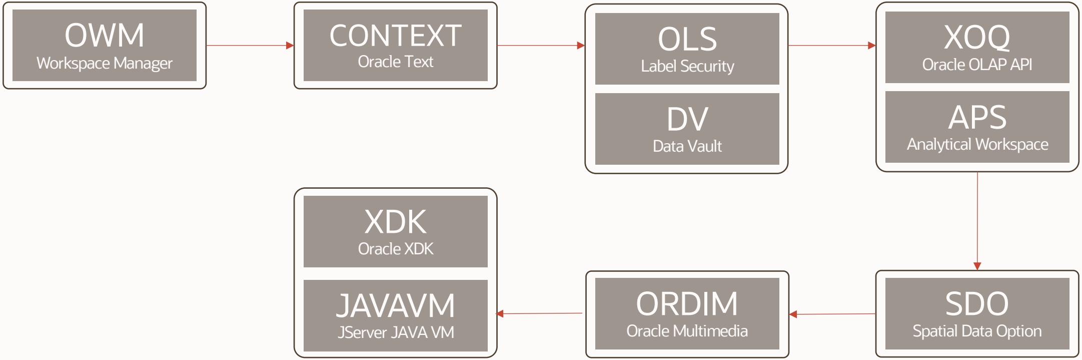
```
COMP_ID    COMP_NAME                             STATUS
---------  ------------------------------------  ----------
CATALOG    Oracle Database Catalog Views         VALID
CATPROC    Oracle Database Packages and Types    VALID
RAC        Oracle Real Application Clusters      OPTION OFF
XDB        Oracle XML Database                   VALID
```

(only one container shown)

Components | Dependencies



Components | Removal Order



OWM
Workspace Manager

CONTEXT
Oracle Text

OLS
Label Security

DV
Data Vault

XOQ
Oracle OLAP API

APS
Analytical Workspace

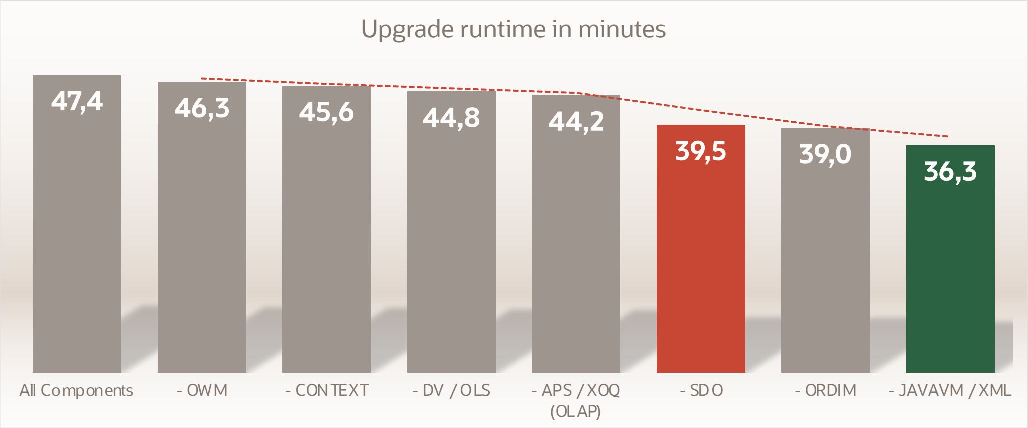
SDO
Spatial Data Option

ORDIM
Oracle Multimedia

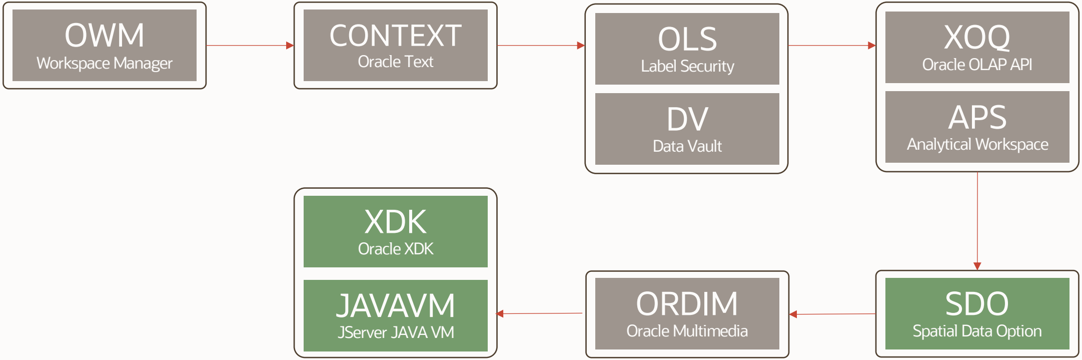
XDK
Oracle XDK

JAVAVM
JServer JAVA VM

Components | Result



Components | Highest Impact



OWM
Workspace Manager

CONTEXT
Oracle Text

OLS
Label Security

DV
Data Vault

XOQ
Oracle OLAP API

APS
Analytical Workspace

SDO
Spatial Data Option

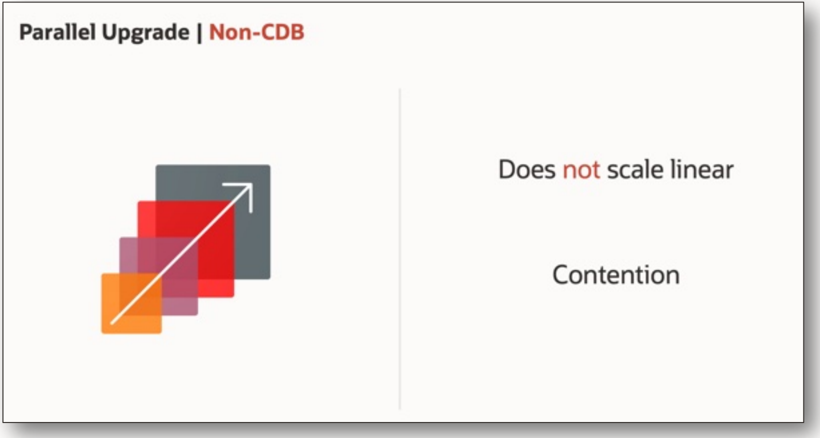
ORDIM
Oracle Multimedia

XDK
Oracle XDK

JAVAVM
JServer JAVA VM

Components | CDB$ROOT vs PDB

Do you remember this slide?



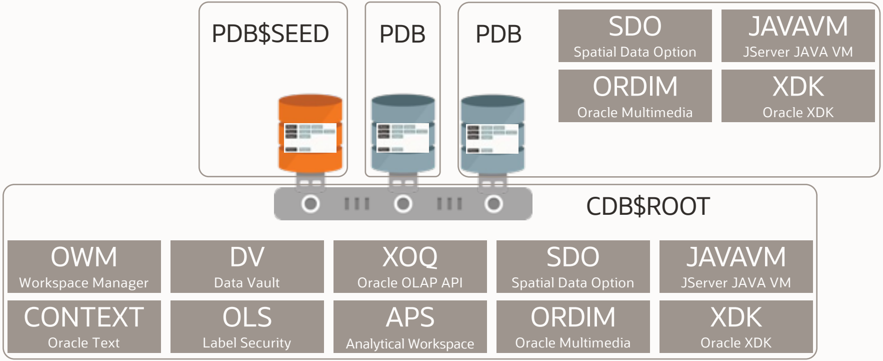
Parallel Upgrade | Non-CDB

Does not scale linear

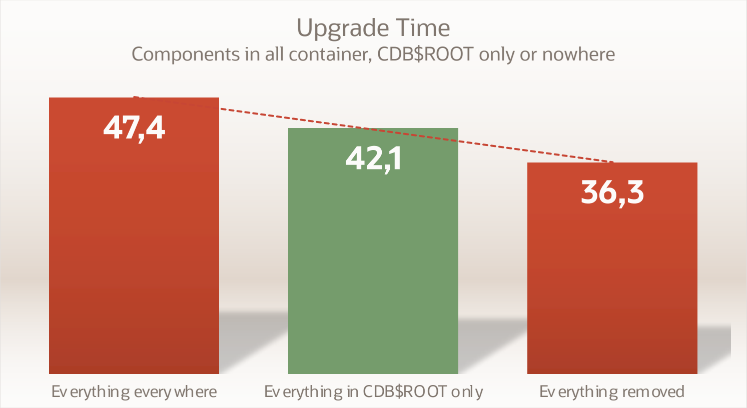
Contention

Components | CDB$ROOT vs PDB

This may be a solution



Components | Compromise



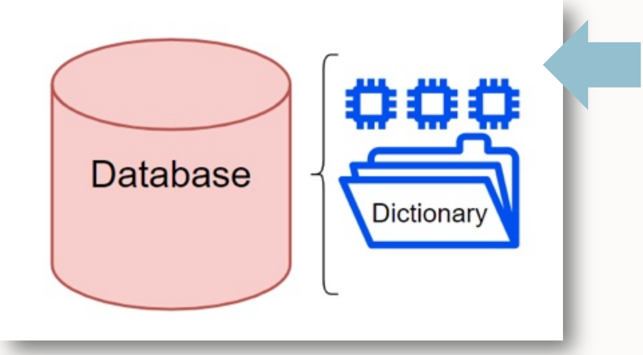
Components | Compromise

ⓘ Find the right balance between functionality and complexity

Database Upgrade Internals

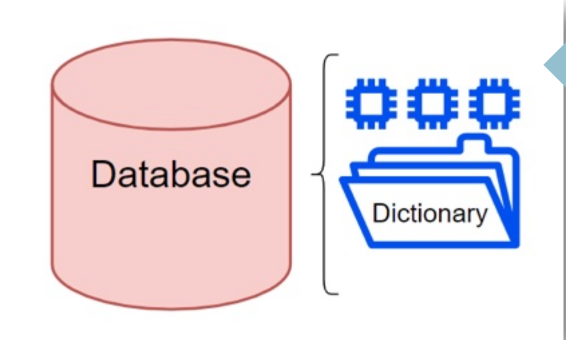
Photo by Daniel Cheung on Unsplash

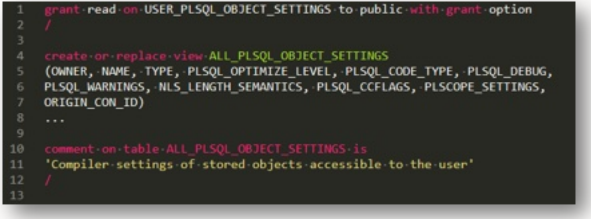
Upgrade Internals | How does the upgrade process work?



Components

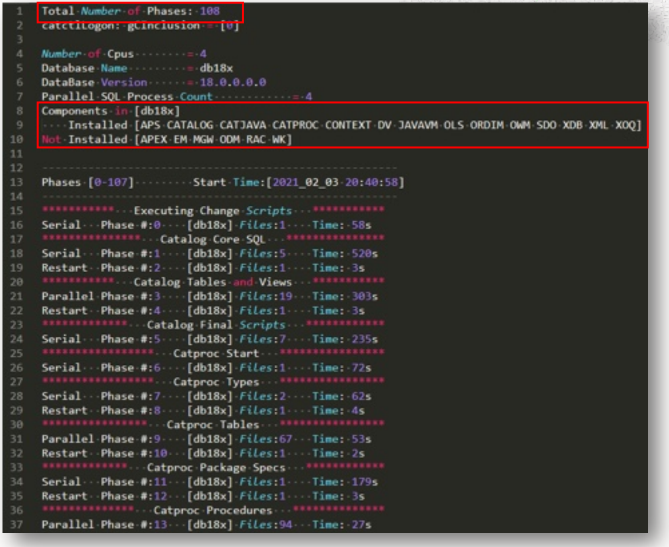
Upgrade Internals | How does the upgrade process work?





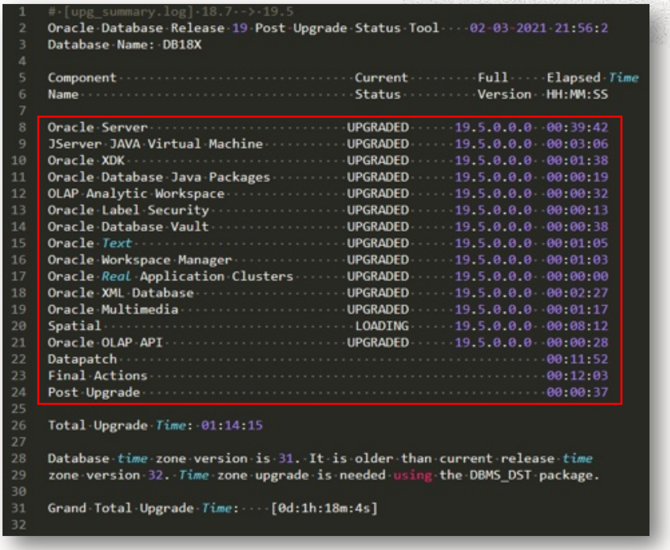
```
grant read on USER_PLSQL_OBJECT_SETTINGS to public with grant option
/

create or replace view ALL_PLSQL_OBJECT_SETTINGS
(OWNER, NAME, TYPE, PLSQL_OPTIMIZE_LEVEL, PLSQL_CODE_TYPE, PLSQL_DEBUG,
PLSQL_WARNINGS, NLS_LENGTH_SEMANTICS, PLSQL_CCFLAGS, PLSCOPE_SETTINGS,
ORIGIN_CON_ID)
...

comment on table ALL_PLSQL_OBJECT_SETTINGS is
'Compiler settings of stored objects accessible to the user'
/

```

Upgrade Internals | How does the upgrade process work?



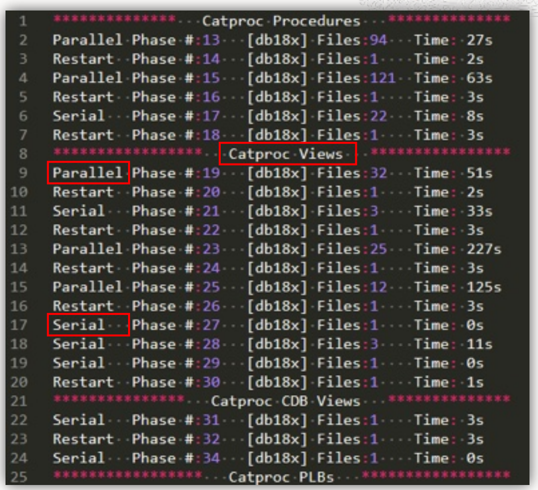
```
Total Number of Phases: 108
catctlLogon: gCInclusion = [0]

Number of Cpus         = 4
Database Name          = db18x
DataBase Version       = 18.0.0.0.0
Parallel SQL Process Count            = 4
Components in [db18x]
    Installed [APS CATALOG CATJAVA CATPROC CONTEXT DV JAVAVM OLS ORDIM OWM SDO XDB XML XOQ]
Not Installed [APEX EM MGW ODM RAC WK]

------------------------------------------------------
Phases [0-107]         Start Time:[2021_02_03 20:40:58]
------------------------------------------------------
***********   Executing Change Scripts   ***********
Serial   Phase #:0    [db18x] Files:1    Time: 58s
***************   Catalog Core SQL   ***************
Serial   Phase #:1    [db18x] Files:5    Time: 520s
Restart  Phase #:2    [db18x] Files:1    Time: 3s
***********   Catalog Tables and Views   ***********
Parallel Phase #:3    [db18x] Files:19   Time: 303s
Restart  Phase #:4    [db18x] Files:1    Time: 3s
*************   Catalog Final Scripts   ************
Serial   Phase #:5    [db18x] Files:7    Time: 235s
*****************   Catproc Start   ****************
Serial   Phase #:6    [db18x] Files:1    Time: 72s
*****************   Catproc Types   ****************
Serial   Phase #:7    [db18x] Files:2    Time: 62s
Restart  Phase #:8    [db18x] Files:1    Time: 4s
****************   Catproc Tables   ****************
Parallel Phase #:9    [db18x] Files:67   Time: 53s
Restart  Phase #:10   [db18x] Files:1    Time: 2s
*************   Catproc Package Specs   ************
Serial   Phase #:11   [db18x] Files:1    Time: 179s
Restart  Phase #:12   [db18x] Files:1    Time: 3s
**************   Catproc Procedures   **************
Parallel Phase #:13   [db18x] Files:94   Time: 27s
```

Upgrade Internals | How does the upgrade process work?



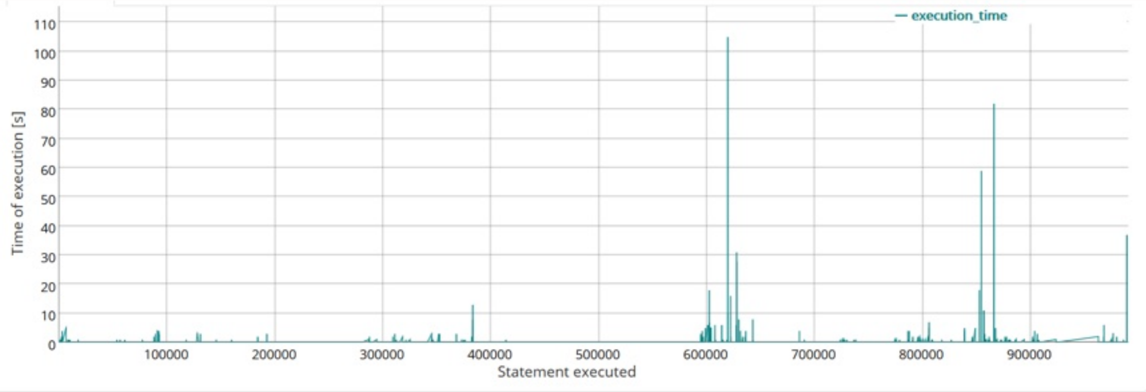
```
# [upg_summary.log] 18.7 -> 19.5
Oracle Database Release 19 Post Upgrade Status Tool    02-03-2021 21:56:2
Database Name: DB18X

Component                               Current         Full     Elapsed Time
Name                                    Status          Version  HH:MM:SS

Oracle Server                          UPGRADED      19.5.0.0.0  00:39:42
JServer JAVA Virtual Machine           UPGRADED      19.5.0.0.0  00:03:06
Oracle XDK                             UPGRADED      19.5.0.0.0  00:01:38
Oracle Database Java Packages          UPGRADED      19.5.0.0.0  00:00:19
OLAP Analytic Workspace                UPGRADED      19.5.0.0.0  00:00:32
Oracle Label Security                  UPGRADED      19.5.0.0.0  00:00:13
Oracle Database Vault                  UPGRADED      19.5.0.0.0  00:00:38
Oracle Text                            UPGRADED      19.5.0.0.0  00:01:05
Oracle Workspace Manager               UPGRADED      19.5.0.0.0  00:01:03
Oracle Real Application Clusters       UPGRADED      19.5.0.0.0  00:00:00
Oracle XML Database                    UPGRADED      19.5.0.0.0  00:02:27
Oracle Multimedia                      UPGRADED      19.5.0.0.0  00:01:17
Spatial                                 LOADING      19.5.0.0.0  00:08:12
Oracle OLAP API                        UPGRADED      19.5.0.0.0  00:00:28
Datapatch                                                        00:11:52
Final Actions                                                    00:12:03
Post Upgrade                                                     00:00:37

Total Upgrade Time: 01:14:15

Database time zone version is 31. It is older than current release time
zone version 32. Time zone upgrade is needed using the DBMS_DST package.

Grand Total Upgrade Time:    [0d:1h:18m:4s]

```

Upgrade Internals | How does the upgrade process work?

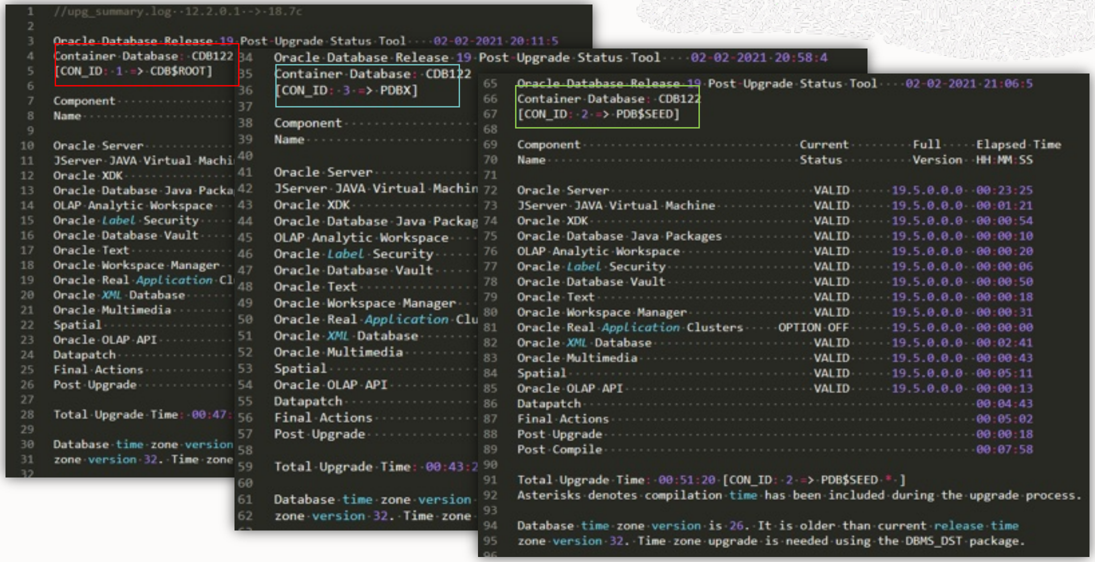


```
**************   Catproc Procedures   **************
Parallel Phase #:13   [db18x] Files:94   Time: 27s
Restart  Phase #:14   [db18x] Files:1    Time: 2s
Parallel Phase #:15   [db18x] Files:121  Time: 63s
Restart  Phase #:16   [db18x] Files:1    Time: 3s
Serial   Phase #:17   [db18x] Files:22   Time: 8s
Restart  Phase #:18   [db18x] Files:1    Time: 3s
*****************   Catproc Views   ****************
Parallel Phase #:19   [db18x] Files:32   Time: 51s
Restart  Phase #:20   [db18x] Files:1    Time: 2s
Serial   Phase #:21   [db18x] Files:3    Time: 33s
Restart  Phase #:22   [db18x] Files:1    Time: 3s
Parallel Phase #:23   [db18x] Files:25   Time: 227s
Restart  Phase #:24   [db18x] Files:1    Time: 3s
Parallel Phase #:25   [db18x] Files:12   Time: 125s
Restart  Phase #:26   [db18x] Files:1    Time: 3s
Serial   Phase #:27   [db18x] Files:1    Time: 0s
Serial   Phase #:28   [db18x] Files:3    Time: 11s
Serial   Phase #:29   [db18x] Files:1    Time: 0s
Restart  Phase #:30   [db18x] Files:1    Time: 1s
***************   Catproc CDB Views   **************
Serial   Phase #:31   [db18x] Files:1    Time: 3s
Restart  Phase #:32   [db18x] Files:1    Time: 3s
Serial   Phase #:34   [db18x] Files:1    Time: 0s
*****************   Catproc PLBs   *****************
```

Upgrade Internals | Timing



Upgrade Internals | CDB Upgrade

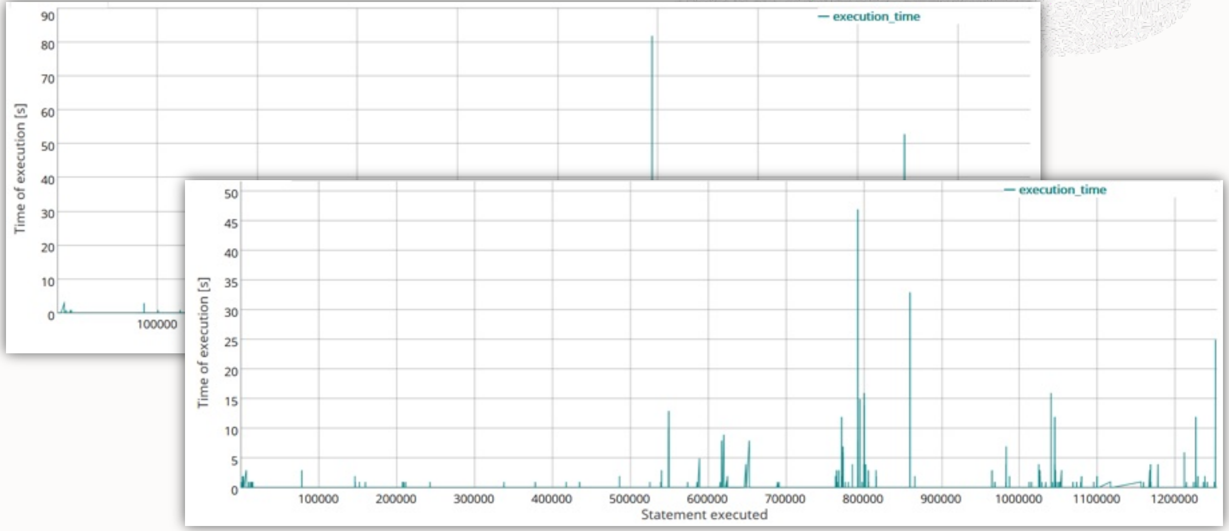


```
//upg_summary.log  12.2.0.1 -> 18.7c

Oracle Database Release 19 Post Upgrade Status Tool    02-02-2021 20:11:5
Container Database: CDB122
[CON_ID: 1 => CDB$ROOT]

Component ........................
Name .............................

Oracle Server ....................
JServer JAVA Virtual Machi
Oracle XDK .......................
Oracle Database Java Packa
OLAP Analytic Workspace ..
Oracle Label Security ....
Oracle Database Vault ....
Oracle Text ......................
Oracle Workspace Manager .
Oracle Real Application Cl
Oracle XML Database ......
Oracle Multimedia ........
Spatial ..........................
Oracle OLAP API ..........
Datapatch ........................
Final Actions ....................
Post Upgrade .....................

Total Upgrade Time: 00:47:

Database time zone version
zone version 32. Time zone
```

```
Oracle Database Release 19 Post Upgrade Status Tool    02-02-2021 20:58:4
Container Database: CDB122
[CON_ID: 3 => PDBX]

Component ........................
Name .............................

Oracle Server ....................
JServer JAVA Virtual Machin
Oracle XDK .......................
Oracle Database Java Packag
OLAP Analytic Workspace ...
Oracle Label Security .....
Oracle Database Vault .....
Oracle Text ......................
Oracle Workspace Manager ..
Oracle Real Application Clu
Oracle XML Database .......
Oracle Multimedia .........
Spatial ..........................
Oracle OLAP API ..........
Datapatch ........................
Final Actions ....................
Post Upgrade .....................

Total Upgrade Time: 00:43:2

Database time zone version
zone version 32. Time zone
```

```
Oracle Database Release 19 Post Upgrade Status Tool    02-02-2021 21:06:5
Container Database: CDB122
[CON_ID: 2 => PDB$SEED]

Component .............................. Current ......... Full     Elapsed Time
Name ................................... Status .......... Version  HH:MM:SS

Oracle Server .......................... VALID ...... 19.5.0.0.0  00:23:25
JServer JAVA Virtual Machine ........... VALID ...... 19.5.0.0.0  00:01:21
Oracle XDK ............................. VALID ...... 19.5.0.0.0  00:00:54
Oracle Database Java Packages .......... VALID ...... 19.5.0.0.0  00:00:10
OLAP Analytic Workspace ................ VALID ...... 19.5.0.0.0  00:00:20
Oracle Label Security .................. VALID ...... 19.5.0.0.0  00:00:06
Oracle Database Vault .................. VALID ...... 19.5.0.0.0  00:00:50
Oracle Text ............................ VALID ...... 19.5.0.0.0  00:00:18
Oracle Workspace Manager ............... VALID ...... 19.5.0.0.0  00:00:31
Oracle Real Application Clusters .... OPTION OFF .... 19.5.0.0.0  00:00:00
Oracle XML Database .................... VALID ...... 19.5.0.0.0  00:02:41
Oracle Multimedia ...................... VALID ...... 19.5.0.0.0  00:00:43
Spatial ................................ VALID ...... 19.5.0.0.0  00:05:11
Oracle OLAP API ........................ VALID ...... 19.5.0.0.0  00:00:13
Datapatch ............................................................ 00:04:43
Final Actions ........................................................ 00:05:02
Post Upgrade ......................................................... 00:00:18
Post Compile ......................................................... 00:07:58

Total Upgrade Time: 00:51:20 [CON_ID: 2 => PDB$SEED * ]
Asterisks denotes compilation time has been included during the upgrade process.

Database time zone version is 26. It is older than current release time
zone version 32. Time zone upgrade is needed using the DBMS_DST package.
```

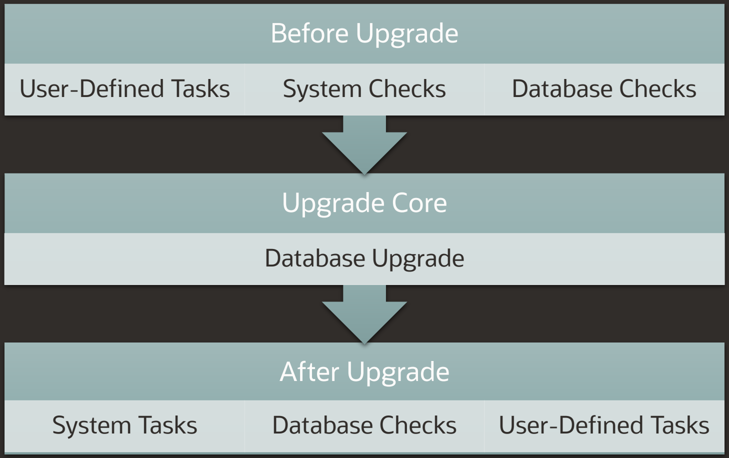
Upgrade Internals | CDB Upgrade Timings



Database AutoUpgrade Diagnosability

Photo by James Pond on Unsplash

AutoUpgrade | Flow-Overview



Before Upgrade

User-Defined Tasks

System Checks

Database Checks

Upgrade Core

Database Upgrade

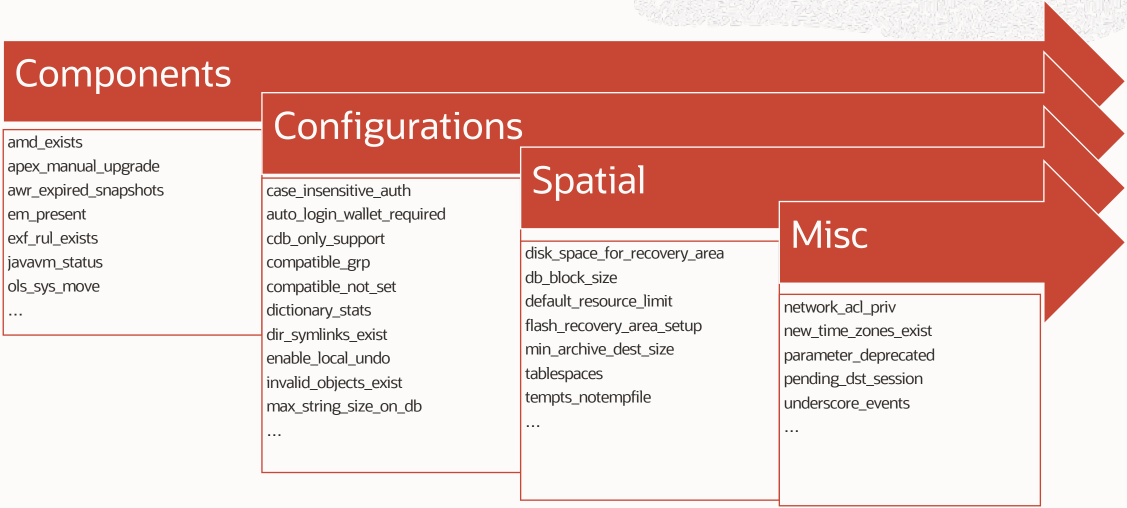
After Upgrade

System Tasks

Database Checks

User-Defined Tasks

AutoUpgrade | Checks Overview



Components

- amd_exists
- apex_manual_upgrade
- awr_expired_snapshots
- em_present
- exf_rul_exists
- javavm_status
- ols_sys_move
- …

Configurations

- case_insensitive_auth
- auto_login_wallet_required
- cdb_only_support
- compatible_grp
- compatible_not_set
- dictionary_stats
- dir_symlinks_exist
- enable_local_undo
- invalid_objects_exist
- max_string_size_on_db
- …

Spatial

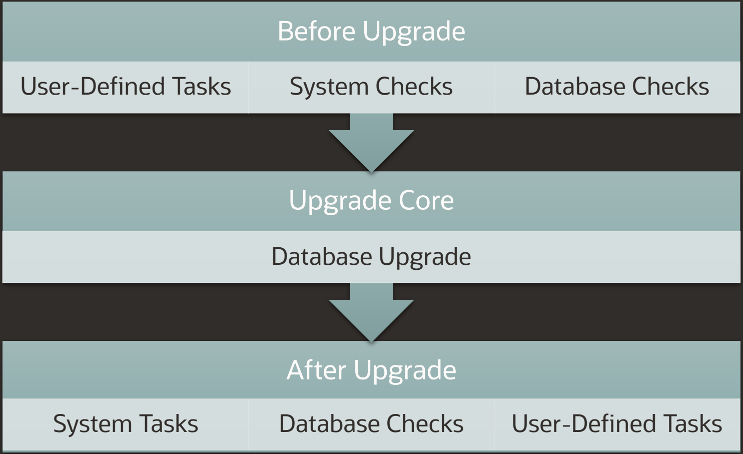
- disk_space_for_recovery_area
- db_block_size
- default_resource_limit
- flash_recovery_area_setup
- min_archive_dest_size
- tablespaces
- tempts_notempfile
- …

Misc

- network_acl_priv
- new_time_zones_exist
- parameter_deprecated
- pending_dst_session
- underscore_events
- …

AutoUpgrade | Flow-Overview

115+ Checks
14+ System



Before Upgrade

User-Defined Tasks

System Checks

Database Checks

Upgrade Core

Database Upgrade

After Upgrade

System Tasks

Database Checks

User-Defined Tasks

19+ Checks
10+ System

AutoUpgrade | Job Diagnose

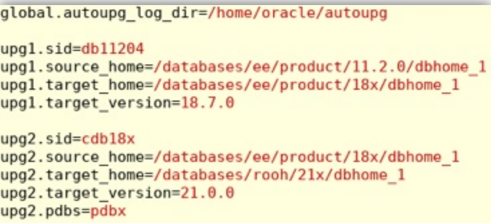
- Commands toolbox

```
upg>

lsj                            // List jobs
resume  -job <number>          // Restarts a job
status  -job <number>          // Show job status
restore -job <number>          // Restores database from GRP
abort   -job <number>          // Aborts the specified job
```

AutoUpgrade

Understanding the directory structure

AutoUpgrade | Understanding the directory structure



```
global.autoupg_log_dir=/home/oracle/autoupg

upg1.sid=db11204
upg1.source_home=/databases/ee/product/11.2.0/dbhome_1
upg1.target_home=/databases/ee/product/18x/dbhome_1
upg1.target_version=18.7.0

upg2.sid=cdb18x
upg2.source_home=/databases/ee/product/18x/dbhome_1
upg2.target_home=/databases/rooh/21x/dbhome_1
upg2.target_version=21.0.0
upg2.pdbs=pdbx
```

One job

Second job

AutoUpgrade | Log File Structure

- /cfgtoollogs
  - ./upgrade/auto ◀ General Logs + State Files
- /database_1
  - ./job_number
    - ./prechecks ◀ HTML Report
    - ./preupgrade
    - ./prefixups
    - ./drain
    - ./dbupgrade ◀ Upgrade Logs
    - ./postupgrade
  - ./temp ◀ AU Temp files
- /database_2
  - …

AutoUpgrade | Different Stages

- ANALYZE

  [PRECHECKS]

- FIXUPS

  [PRECHECKS,PREFIXUPS]

- DEPLOY

  [GRP, PREUPGRADE, PRECHECKS, PREFIXUPS, DRAIN, DBUPGRADE, POSTCHECKS, POSTFIXUPS, POSTUPGRADE, NONCDBTOPDB, SYSUPDATES]

- UPGRADE

  [DBUPGRADE,POSTCHECKS,POSTFIXUPS,SYSUPDATES]

- POSTFIXUPS

  [POSTFIXUPS]

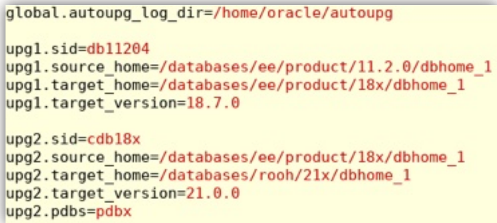
AutoUpgrade | Log File Structure

- /cfgtoollogs
  ./upgrade/auto
- /database_1
  ./job_number
    ./prechecks
    ./preupgrade
    ./prefixups
    ./drain
    ./dbupgrade
    ./postupgrade
  ./temp
- /database_2
  ./job_number
  …

Depends on the execution mode

AutoUpgrade | Log File Structure

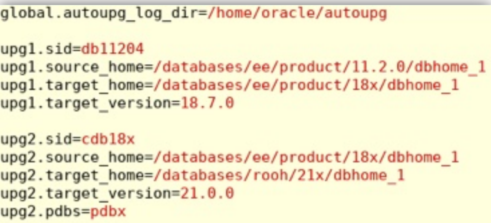
analyze



```
global.autoupg_log_dir=/home/oracle/autoupg

upg1.sid=db11204
upg1.source_home=/databases/ee/product/11.2.0/dbhome_1
upg1.target_home=/databases/ee/product/18x/dbhome_1
upg1.target_version=18.7.0

upg2.sid=cdb18x
upg2.source_home=/databases/ee/product/18x/dbhome_1
upg2.target_home=/databases/rooh/21x/dbhome_1
upg2.target_version=21.0.0
upg2.pdbs=pdbx
```

AutoUpgrade | Log File Structure

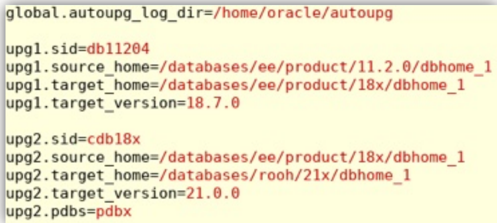
analyze => fixups



```
global.autoupg_log_dir=/home/oracle/autoupg

upg1.sid=db11204
upg1.source_home=/databases/ee/product/11.2.0/dbhome_1
upg1.target_home=/databases/ee/product/18x/dbhome_1
upg1.target_version=18.7.0

upg2.sid=cdb18x
upg2.source_home=/databases/ee/product/18x/dbhome_1
upg2.target_home=/databases/rooh/21x/dbhome_1
upg2.target_version=21.0.0
upg2.pdbs=pdbx
```

AutoUpgrade | Log File Structure

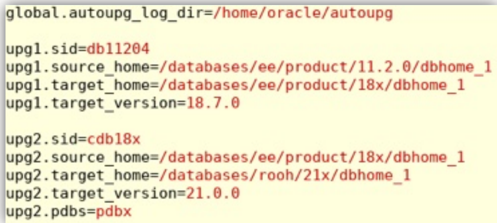
analyze => fixups => deploy



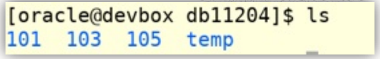
```
global.autoupg_log_dir=/home/oracle/autoupg

upg1.sid=db11204
upg1.source_home=/databases/ee/product/11.2.0/dbhome_1
upg1.target_home=/databases/ee/product/18x/dbhome_1
upg1.target_version=18.7.0

upg2.sid=cdb18x
upg2.source_home=/databases/ee/product/18x/dbhome_1
upg2.target_home=/databases/rooh/21x/dbhome_1
upg2.target_version=21.0.0
upg2.pdbs=pdbx
```

AutoUpgrade | Log File Structure

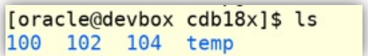
analyze => fixups => deploy



```
global.autoupg_log_dir=/home/oracle/autoupg

upg1.sid=db11204
upg1.source_home=/databases/ee/product/11.2.0/dbhome_1
upg1.target_home=/databases/ee/product/18x/dbhome_1
upg1.target_version=18.7.0

upg2.sid=cdb18x
upg2.source_home=/databases/ee/product/18x/dbhome_1
upg2.target_home=/databases/rooh/21x/dbhome_1
upg2.target_version=21.0.0
upg2.pdbs=pdbx
```



```
[oracle@devbox db11204]$ ls
101  103  105  temp
```

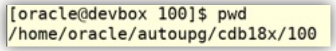


```
[oracle@devbox cdb18x]$ ls
100  102  104  temp
```

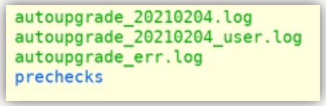
AutoUpgrade | Pre-Checks Folder

../prechecks

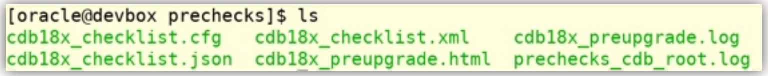
[cdb18x] analyze[100] > fixups[102] > deploy[104]



```
[oracle@devbox 100]$ pwd
/home/oracle/autoupg/cdb18x/100
```



```
autoupgrade_20210204.log
autoupgrade_20210204_user.log
autoupgrade_err.log
prechecks
```



```
[oracle@devbox prechecks]$ ls
cdb18x_checklist.cfg    cdb18x_checklist.xml     cdb18x_preupgrade.log
cdb18x_checklist.json   cdb18x_preupgrade.html   prechecks_cdb_root.log
```

AutoUpgrade | Pre-Checks Folder

../prechecks

```
cdb18x_preupgrade.html
cdb18x_preupgrade.log
```

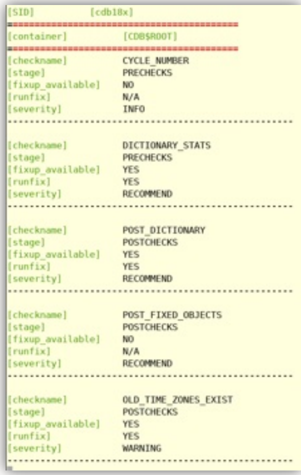
Reports

```
cdb18x_checklist.cfg
```

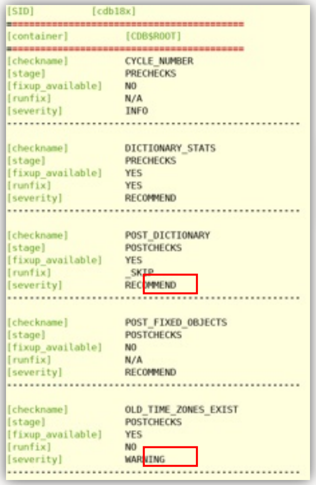
Customize fixups execution

AutoUpgrade | Pre-Checks Folder

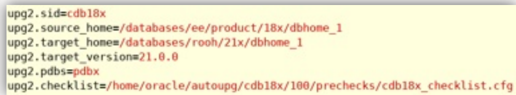
cdb18x_checklist.cfg



```
[SID]           [cdb18x]
==============================================
[container]             [CDB$ROOT]
==============================================
[checkname]             CYCLE_NUMBER
[stage]                 PRECHECKS
[fixup_available]       NO
[runfix]                N/A
[severity]              INFO
----------------------------------------------------
[checkname]             DICTIONARY_STATS
[stage]                 PRECHECKS
[fixup_available]       YES
[runfix]                YES
[severity]              RECOMMEND
----------------------------------------------------
[checkname]             POST_DICTIONARY
[stage]                 POSTCHECKS
[fixup_available]       YES
[runfix]                YES
[severity]              RECOMMEND
----------------------------------------------------
[checkname]             POST_FIXED_OBJECTS
[stage]                 POSTCHECKS
[fixup_available]       NO
[runfix]                N/A
[severity]              RECOMMEND
----------------------------------------------------
[checkname]             OLD_TIME_ZONES_EXIST
[stage]                 POSTCHECKS
[fixup_available]       YES
[runfix]                YES
[severity]              WARNING
----------------------------------------------------
```



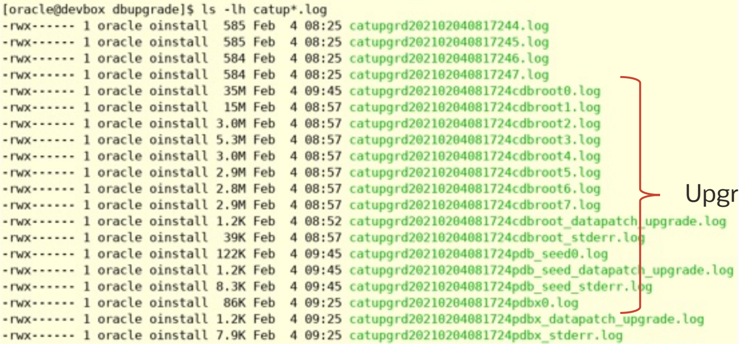
```
[SID]           [cdb18x]
==============================================
[container]             [CDB$ROOT]
==============================================
[checkname]             CYCLE_NUMBER
[stage]                 PRECHECKS
[fixup_available]       NO
[runfix]                N/A
[severity]              INFO
----------------------------------------------------
[checkname]             DICTIONARY_STATS
[stage]                 PRECHECKS
[fixup_available]       YES
[runfix]                YES
[severity]              RECOMMEND
----------------------------------------------------
[checkname]             POST_DICTIONARY
[stage]                 POSTCHECKS
[fixup_available]       YES
[runfix]                SKIP
[severity]              RECOMMEND
----------------------------------------------------
[checkname]             POST_FIXED_OBJECTS
[stage]                 POSTCHECKS
[fixup_available]       NO
[runfix]                N/A
[severity]              RECOMMEND
----------------------------------------------------
[checkname]             OLD_TIME_ZONES_EXIST
[stage]                 POSTCHECKS
[fixup_available]       YES
[runfix]                NO
[severity]              WARNING
----------------------------------------------------
```



```
upg2.sid=cdb18x
upg2.source_home=/databases/ee/product/18x/dbhome_1
upg2.target_home=/databases/rooh/21x/dbhome_1
upg2.target_version=21.0.0
upg2.pdbs=pdbx
upg2.checklist=/home/oracle/autoupg/cdb18x/100/prechecks/cdb18x_checklist.cfg
```

AutoUpgrade | dbupgrade Folder

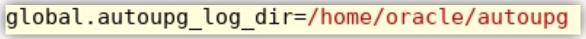
[cdb18x] analyze[100] > fixups[102] > deploy[104]



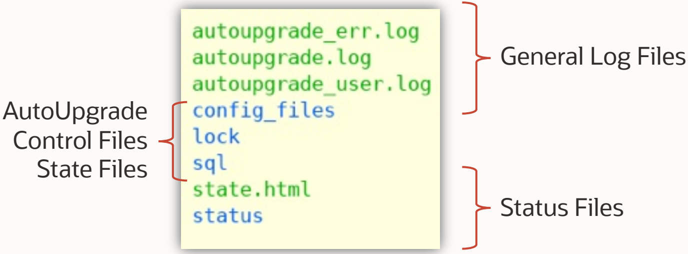
```
[oracle@devbox dbupgrade]$ ls -lh catup*.log
-rwx------ 1 oracle oinstall  585 Feb  4 08:25 catupgrd202102040817244.log
-rwx------ 1 oracle oinstall  585 Feb  4 08:25 catupgrd202102040817245.log
-rwx------ 1 oracle oinstall  584 Feb  4 08:25 catupgrd202102040817246.log
-rwx------ 1 oracle oinstall  584 Feb  4 08:25 catupgrd202102040817247.log
-rwx------ 1 oracle oinstall  35M Feb  4 09:45 catupgrd20210204081724cdbroot0.log
-rwx------ 1 oracle oinstall  15M Feb  4 08:57 catupgrd20210204081724cdbroot1.log
-rwx------ 1 oracle oinstall 3.0M Feb  4 08:57 catupgrd20210204081724cdbroot2.log
-rwx------ 1 oracle oinstall 5.3M Feb  4 08:57 catupgrd20210204081724cdbroot3.log
-rwx------ 1 oracle oinstall 3.0M Feb  4 08:57 catupgrd20210204081724cdbroot4.log
-rwx------ 1 oracle oinstall 2.9M Feb  4 08:57 catupgrd20210204081724cdbroot5.log
-rwx------ 1 oracle oinstall 2.8M Feb  4 08:57 catupgrd20210204081724cdbroot6.log
-rwx------ 1 oracle oinstall 2.9M Feb  4 08:57 catupgrd20210204081724cdbroot7.log
-rwx------ 1 oracle oinstall 1.2K Feb  4 08:52 catupgrd20210204081724cdbroot_datapatch_upgrade.log
-rwx------ 1 oracle oinstall  39K Feb  4 08:57 catupgrd20210204081724cdbroot_stderr.log
-rwx------ 1 oracle oinstall 122K Feb  4 09:45 catupgrd20210204081724pdb_seed0.log
-rwx------ 1 oracle oinstall 1.2K Feb  4 09:45 catupgrd20210204081724pdb_seed_datapatch_upgrade.log
-rwx------ 1 oracle oinstall 8.3K Feb  4 09:45 catupgrd20210204081724pdb_seed_stderr.log
-rwx------ 1 oracle oinstall  86K Feb  4 09:25 catupgrd20210204081724pdbx0.log
-rwx------ 1 oracle oinstall 1.2K Feb  4 09:25 catupgrd20210204081724pdbx_datapatch_upgrade.log
-rwx------ 1 oracle oinstall 7.9K Feb  4 09:25 catupgrd20210204081724pdbx_stderr.log
```

Upgrade logs

AutoUpgrade | General Progress Folder



```
global.autoupg_log_dir=/home/oracle/autoupg
```



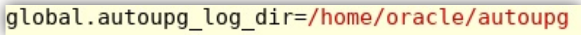
```
autoupgrade_err.log
autoupgrade.log
autoupgrade_user.log
config_files
lock
sql
state.html
status
```

General Log Files

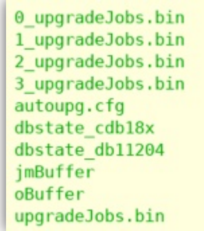
AutoUpgrade
Control Files
State Files

Status Files

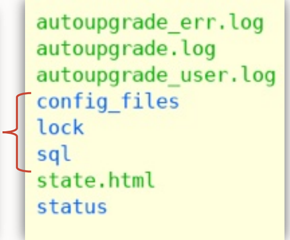
AutoUpgrade | General Progress Folder



```
global.autoupg_log_dir=/home/oracle/autoupg
```



```
0_upgradeJobs.bin
1_upgradeJobs.bin
2_upgradeJobs.bin
3_upgradeJobs.bin
autoupg.cfg
dbstate_cdb18x
dbstate_db11204
jmBuffer
oBuffer
upgradeJobs.bin
```

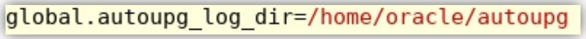


```
autoupgrade_err.log
autoupgrade.log
autoupgrade_user.log
config_files
lock
sql
state.html
status
```

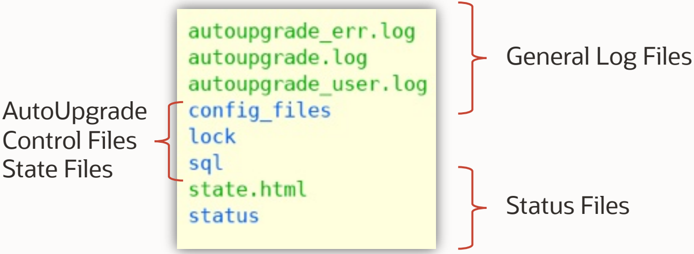
General Log Files

Status Files

AutoUpgrade | General Progress Folder



```
global.autoupg_log_dir=/home/oracle/autoupg
```



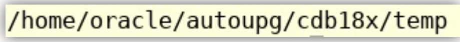
```
autoupgrade_err.log
autoupgrade.log
autoupgrade_user.log
config_files
lock
sql
state.html
status
```

General Log Files

AutoUpgrade
Control Files
State Files

Status Files

AutoUpgrade | `temp` Directory



```
/home/oracle/autoupg/cdb18x/temp
```

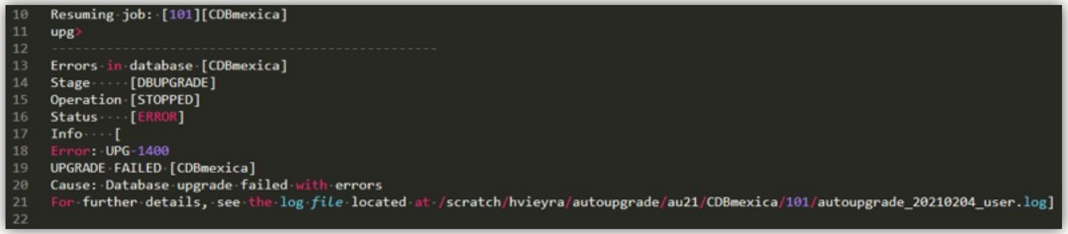
- Before/During/After AutoUpgrade-Generated `pfile`s
- Recompilation scripts
- Restart scripts
- Network Files backups
  - `tnsnames.ora`
  - `listener.ora`
  - `sqlnet.ora`
- TimeZone Upg Logs
- Context Files

AutoUpgrade

How do we know if an issue occurred?

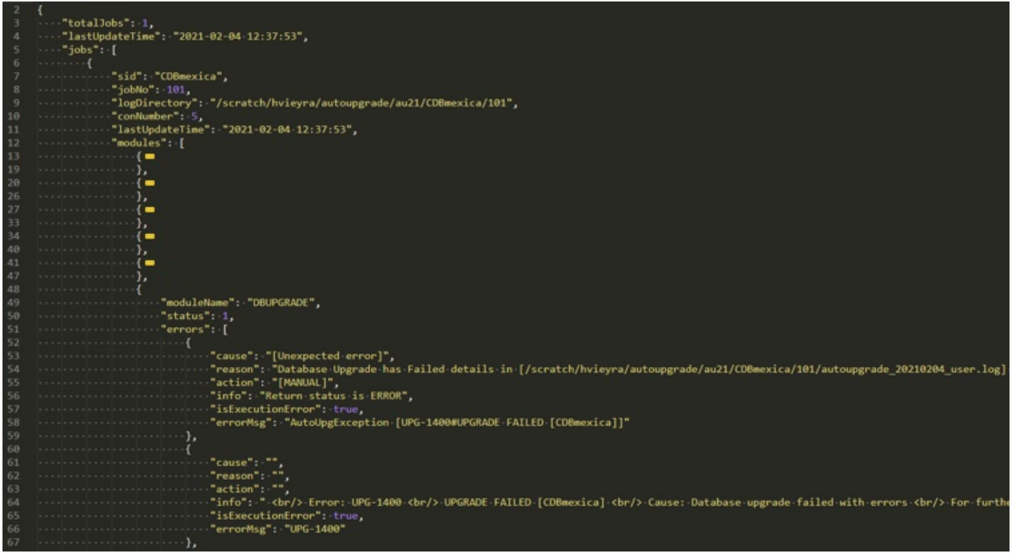
AutoUpgrade console | status files | log files

AutoUpgrade | Console



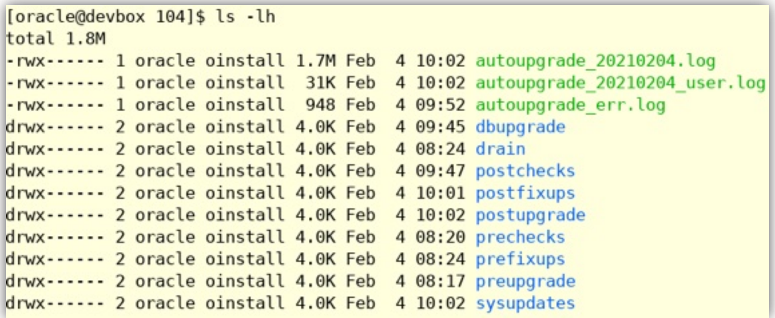
```
Resuming job: [101][CDBmexica]
upg>
-------------------------------------------------
Errors in database [CDBmexica]
Stage     [DBUPGRADE]
Operation [STOPPED]
Status    [ERROR]
Info    [
Error: UPG-1400
UPGRADE FAILED [CDBmexica]
Cause: Database upgrade failed with errors
For further details, see the log file located at /scratch/hvieyra/autoupgrade/au21/CDBmexica/101/autoupgrade_20210204_user.log]

```

AutoUpgrade | Status [status.json & progress.json]



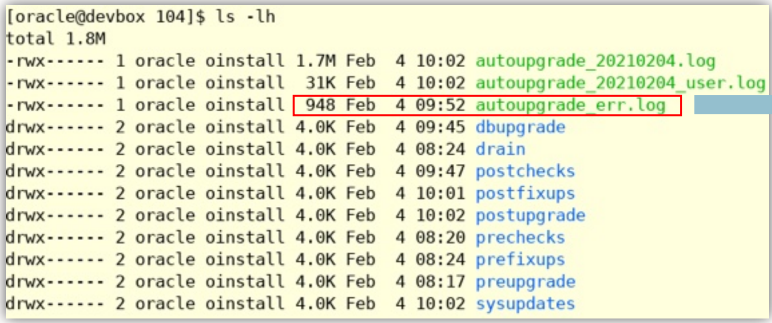
```
{
    "totalJobs": 1,
    "lastUpdateTime": "2021-02-04 12:37:53",
    "jobs": [
        {
            "sid": "CDBmexica",
            "jobNo": 101,
            "logDirectory": "/scratch/hvieyra/autoupgrade/au21/CDBmexica/101",
            "conNumber": 5,
            "lastUpdateTime": "2021-02-04 12:37:53",
            "modules": [
                {…
                },
                {…
                },
                {…
                },
                {…
                },
                {…
                },
                {
                    "moduleName": "DBUPGRADE",
                    "status": 1,
                    "errors": [
                        {
                            "cause": "[Unexpected error]",
                            "reason": "Database Upgrade has Failed details in [/scratch/hvieyra/autoupgrade/au21/CDBmexica/101/autoupgrade_20210204_user.log]
                            "action": "[MANUAL]",
                            "info": "Return status is ERROR",
                            "isExecutionError": true,
                            "errorMsg": "AutoUpgException [UPG-1400#UPGRADE FAILED [CDBmexica]]"
                        },
                        {
                            "cause": "",
                            "reason": "",
                            "action": "",
                            "info": " <br/> Error: UPG-1400 <br/> UPGRADE FAILED [CDBmexica] <br/> Cause: Database upgrade failed with errors <br/> For furthe
                            "isExecutionError": true,
                            "errorMsg": "UPG-1400"
                        },
```

AutoUpgrade | Log files



```
[oracle@devbox 104]$ ls -lh
total 1.8M
-rwx------ 1 oracle oinstall 1.7M Feb  4 10:02 autoupgrade_20210204.log
-rwx------ 1 oracle oinstall  31K Feb  4 10:02 autoupgrade_20210204_user.log
-rwx------ 1 oracle oinstall  948 Feb  4 09:52 autoupgrade_err.log
drwx------ 2 oracle oinstall 4.0K Feb  4 09:45 dbupgrade
drwx------ 2 oracle oinstall 4.0K Feb  4 08:24 drain
drwx------ 2 oracle oinstall 4.0K Feb  4 09:47 postchecks
drwx------ 2 oracle oinstall 4.0K Feb  4 10:01 postfixups
drwx------ 2 oracle oinstall 4.0K Feb  4 10:02 postupgrade
drwx------ 2 oracle oinstall 4.0K Feb  4 08:20 prechecks
drwx------ 2 oracle oinstall 4.0K Feb  4 08:24 prefixups
drwx------ 2 oracle oinstall 4.0K Feb  4 08:17 preupgrade
drwx------ 2 oracle oinstall 4.0K Feb  4 10:02 sysupdates
```

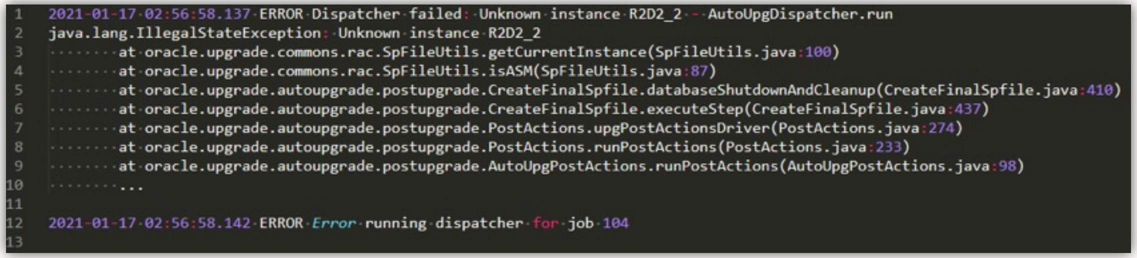
AutoUpgrade | Log files



```
[oracle@devbox 104]$ ls -lh
total 1.8M
-rwx------ 1 oracle oinstall 1.7M Feb  4 10:02 autoupgrade_20210204.log
-rwx------ 1 oracle oinstall  31K Feb  4 10:02 autoupgrade_20210204_user.log
-rwx------ 1 oracle oinstall  948 Feb  4 09:52 autoupgrade_err.log
drwx------ 2 oracle oinstall 4.0K Feb  4 09:45 dbupgrade
drwx------ 2 oracle oinstall 4.0K Feb  4 08:24 drain
drwx------ 2 oracle oinstall 4.0K Feb  4 09:47 postchecks
drwx------ 2 oracle oinstall 4.0K Feb  4 10:01 postfixups
drwx------ 2 oracle oinstall 4.0K Feb  4 10:02 postupgrade
drwx------ 2 oracle oinstall 4.0K Feb  4 08:20 prechecks
drwx------ 2 oracle oinstall 4.0K Feb  4 08:24 prefixups
drwx------ 2 oracle oinstall 4.0K Feb  4 08:17 preupgrade
drwx------ 2 oracle oinstall 4.0K Feb  4 10:02 sysupdates
```

Stage where it failed

AutoUpgrade | Log files



```
2021-01-17 02:56:58.137 ERROR Dispatcher failed: Unknown instance R2D2_2 - AutoUpgDispatcher.run
java.lang.IllegalStateException: Unknown instance R2D2_2
        at oracle.upgrade.commons.rac.SpFileUtils.getCurrentInstance(SpFileUtils.java:100)
        at oracle.upgrade.commons.rac.SpFileUtils.isASM(SpFileUtils.java:87)
        at oracle.upgrade.autoupgrade.postupgrade.CreateFinalSpfile.databaseShutdownAndCleanup(CreateFinalSpfile.java:410)
        at oracle.upgrade.autoupgrade.postupgrade.CreateFinalSpfile.executeStep(CreateFinalSpfile.java:437)
        at oracle.upgrade.autoupgrade.postupgrade.PostActions.upgPostActionsDriver(PostActions.java:274)
        at oracle.upgrade.autoupgrade.postupgrade.PostActions.runPostActions(PostActions.java:233)
        at oracle.upgrade.autoupgrade.postupgrade.AutoUpgPostActions.runPostActions(AutoUpgPostActions.java:98)
        ...

2021-01-17 02:56:58.142 ERROR Error running dispatcher for job 104

```

Recompile

How to tune the recompilation?

Recompilation | Check

Invalid objects after upgrade

```
SQL> select con_id, count(*) from CDB_OBJECTS
      where status='INVALID' group by con_id order by 1;

CON_ID    COUNT(*)
--------- ----------
        1         51
        3       6359
        4       6356
        5       6356
```

Recompilation | `utlrp.sql`

Usually, after upgrade, recompilation should happen

- `utlrp.sql`
  - Calls `utlprp.sql` with `CPU_COUNT – 1`
  - Creates `CPU_COUNT – 1` parallel jobs
  - Recompilation happens PDB after PDB
  - Attempts to compile ALL invalid objects

- `utlprp.sql`
  - Used to override the default parallel degree
  - Example

```
SQL> start ?/rdbms/admin/utlprp 32
```

Recompilation | AutoUpgrade

By default, AutoUpgrade recompiles after upgrade

- Recompilation attempts to compile everything invalid

Postpone recompilation

```
upg1.source_home=/u01/app/oracle/product/12.2.0.1
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=CDB1
upg1.run_utlrp=no
upg1.after_action=/database/scripts/compile_my_way.sh
```

- But you can't postpone PDB$SEED's recompilation
- CDB$ROOT recompiles partially already, too

Recompilation | Option 1

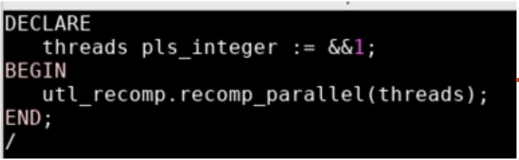
Run your own compilation script(s)

- Sub scripts
- Scheduler
- Parallel degree

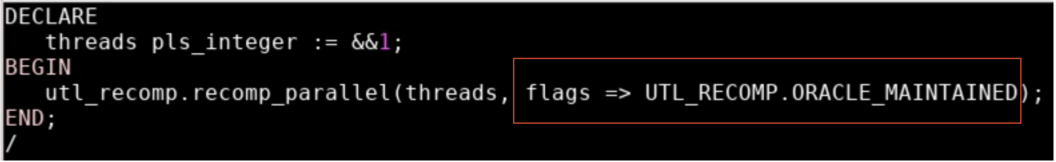
Recompilation | Option 2 *unofficial*

Modify `utlprp.sql`

- Makes sense only when you have a lot of INVALID user objects
- Force recompilation to compile ONLY oracle-maintained objects
- Backport available soon



```
DECLARE
   threads pls_integer := &&1;
BEGIN
   utl_recomp.recomp_parallel(threads);
END;
/
```



```
DECLARE
   threads pls_integer := &&1;
BEGIN
   utl_recomp.recomp_parallel(threads, flags => UTL_RECOMP.ORACLE_MAINTAINED);
END;
/
```

Only attempt tuning the recompilation where it takes UNUSUALLY long!

Upgrade | Diagnose Performance Issue

```
grep -i "Elapsed" <catupgrd0 filename> | sort
```

Chapter 4
Performance Stability

Photo by Loic Leray on Unsplash

your key to

Successful Database Upgrades

Step 1
Download and install Oracle 19c
eDelivery.oracle.com

Step 2
Download and install newest RU
MOS Note: 2118136.2

Step 3
Download and use AutoUpgrade
MOS Note: 2485457.1

Step 4
Performance Stability with SPM, STA and RAT

General Performance Best Practices

Photo by Yannis Zaugg on Unsplash

Photo by Alexander Andrews on Unsplash

Parameters

Parameters | General Recommendations

Default

Deprecated/desupported

Underscores/events

Applications

The fewer parameters, the better

```
SQL> select name, value
     from v$parameter
     where isdefault='FALSE';

 NAME                               VALUE
_____________________________________________________________________________
_bug27355984_xt_preproc_timeout    1000
_cursor_obsolete_threshold         1024
_exclude_seed_cdb_view             FALSE
_optimizer_aggr_groupby_elim       FALSE
_use_single_log_writer             TRUE
audit_file_dest                    /u01/app/oracle/admin/CDB2/adump
audit_trail                        NONE
compatible                         19.0.0
control_files                      /u02/fast_recovery_area/CDB2/control02.ctl
```

Parameters | General Recommendations

Default

Deprecated/desupported

Underscores/events

Applications

```
SQL> startup
ORA-32004: obsolete or deprecated parameter(s) specified for RDBMS instance
ORACLE instance started.

Total System Global Area       1577055360 bytes
Fixed Size                        9135232 bytes
Variable Size                   385875968 bytes
Database Buffers               1174405120 bytes
Redo Buffers                      7639040 bytes
Database mounted.
Database opened.
```

Pro tip: The Upgrade Guide contains a list of deprecated and desupported parameters

Parameters | General Recommendations

Default

Deprecated/desupported

Underscores/events

Applications

Use

- as few as possible
- not longer than needed

```
SQL> select name, value
     from v$parameter
     where substr(name, 0, 1) = '_' or name='event';
```

Create plan for remove it again

Pro tip: During upgrade it is recommended to remove all underscores and events

Parameters | General Recommendations

Default

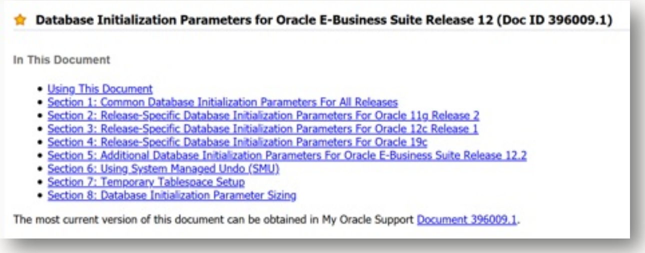
Deprecated/desupported

Underscores/events

Applications

Follow application specific recommendations

- E-Business Suite
- Siebel
- ...



Database Initialization Parameters for Oracle E-Business Suite Release 12 (Doc ID 396009.1)

In This Document

- Using This Document
- Section 1: Common Database Initialization Parameters For All Releases
- Section 2: Release-Specific Database Initialization Parameters For Oracle 11g Release 2
- Section 3: Release-Specific Database Initialization Parameters For Oracle 12c Release 1
- Section 4: Release-Specific Database Initialization Parameters For Oracle 19c
- Section 5: Additional Database Initialization Parameters For Oracle E-Business Suite Release 12.2
- Section 6: Using System Managed Undo (SMU)
- Section 7: Temporary Tablespace Setup
- Section 8: Database Initialization Parameter Sizing

The most current version of this document can be obtained in My Oracle Support Document 396009.1.

Parameters | Tracking Your Changes

Never implement a change without a comment

```
SQL> alter system set
     "_cursor_obsolete_threshold"=1024
     comment='04-03-2021 Daniel: MOS 2431353.1, evaluate after upgrade'
     scope=both;
```

Or, in your PFile

```
*._cursor_obsolete_threshold=1024#04-03-2021 Daniel: MOS 2431353.1, evaluate after upgrade
```

View your comments

```
SQL> select value, update_comment from v$parameter where name='_cursor_obsolete_threshold';

VALUE    UPDATE_COMMENT
________ ____________________________________________________________
1024     04-03-2021 Daniel: MOS 2431353.1, evaluate after upgrade
```

Parameters

COMPATIBLE vs OPTIMIZER_FEATURES_ENABLE

Fully independent from each other

- `COMPATIBLE`
  - Enables features
  - Always use the default value 19.0.0 in Oracle 19c
- `OPTIMIZER_FEATURES_ENABLE`
  - Just reverts to the parameters used in a previous release
  - Avoid using it if possible
  - This is not a Swiss Army knife!
  - You will turn off a lot of great features

> *Modifying the OPTIMIZER_FEATURES_ENABLE parameter generally is strongly discouraged and should only be used as a short term measure at the suggestion of Oracle Global Support.*

Use Caution if Changing the OPTIMIZER_FEATURES_ENABLE Parameter After an Upgrade (Doc ID 1362332.1)

Automatic Memory Management | AMM

> *You can allow the Oracle Database instance to automatically manage and tune memory for you.*

Database 19c, Database Administrator’s Guide, chapter 6

Controlled by parameters `MEMORY_TARGET` and `MEMORY_MAX_TARGET`

Sounds like a good idea.... Don't use it!

AMM | Pitfalls

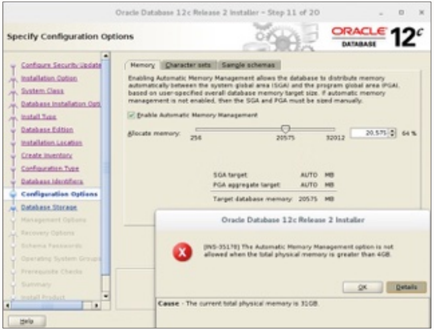
1. No support for HugePages on Linux

2. Potentially, PGA can *rob* memory from SGA

3. There is an overhead of resizing the memory areas

4. Issues

5. Only on few platforms can you use AMM
   with more than 4 GB of memory (MOS Doc ID 2244817.1)



Enabling Automatic Memory Management allows the database to distribute memory automatically between the system global area (SGA) and the program global area (PGA) based on user-specified overall database memory target size. If automatic memory management is not enabled, then the SGA and PGA must be sized manually.

☑ Enable Automatic Memory Management

Allocate memory: 256 20575 32012 20,575 64 %

SGA target: AUTO MB
PGA aggregate target: AUTO MB
Target database memory: 20575 MB

Oracle Database 12c Release 2 Installer

[INS-35178] The Automatic Memory Management option is not allowed when the total physical memory is greater than 4GB.

Cause - The current total physical memory is 31GB.

[DBT-11211] The Automatic Memory Management option is not allowed when the total physical memory is greater than 4GB.
INS-35178 : The Automated Memory management option is not allowed when the Total physical memory is greater than 4GB

AMM | If You Really Want ...

Possible use cases:

1. ASM instances
2. Small non-production databases

Always consult the Database Installation Guide of your platform

Fix Control Persistence | DBMS_OPTIM_BUNDLE

Overview

Check

Enable

Output

Result

Info and Issues

Fix Control Persistence

- `DBMS_OPTIM_BUNDLE`
- Exists since 12.1.0.2 April 2017 (and earlier on Exadata)
- Idea:
  - Enable Optimizer behavior changing fixes at will
  - Fixes are installed but disabled by default

Fix Control Persistence | DBMS_OPTIM_BUNDLE

- Overview
- **Check**
- Enable
- Output
- Result
- Info and Issues

```
SQL> set serverout on
SQL> exec dbms_optim_bundle.GetBugsForBundle;

19.10.0.0.210119DBRU:
    Bug: 29487407,  fix_controls: 29487407
    Bug: 30998035,  fix_controls: 30998035
    Bug: 30786641,  fix_controls: 30786641
    Bug: 31444353,  fix_controls: 31444353
    Bug: 30486896,  fix_controls: 30486896
    Bug: 28999046,  fix_controls: 28999046
    Bug: 30902655,  fix_controls: 30902655
    Bug: 30681521,  fix_controls: 30681521
    Bug: 29302565,  fix_controls: 29302565
    Bug: 30972817,  fix_controls: 30972817
...
```

Fix Control Persistence | DBMS_OPTIM_BUNDLE

Overview

Check

Enable

Output

Result

Info and Issues

```
begin
  dbms_optim_bundle.enable_optim_fixes(
   action                     => 'ON',
   scope                      => 'BOTH',
   current_setting_precedence => 'YES');
end;
/
```

Fix Control Persistence | DBMS_OPTIM_BUNDLE

Overview

Check

Enable

Output

Result

Info and Issues

```
1) Current _fix_control setting for spfile:
None

2) Final _fix_control setting for spfile considering current_setting_precedence
is YES
29331066:1  28965084:1  28776811:1  28498976:1  28567417:1  28558645:1
29132869:1  29450812:1  29687220:1  29304314:1  29930457:1  27261477:1
31069997:1  31077481:1  28602253:1  29653132:0  29937655:1  30347410:1
30602828:1  30896685:0  29487407:1  30998035:1  30786641:1  31444353:0
30486896:1  28999046:1  30902655:1  30681521:1  29302565:1  30972817:1
30222669:1  31668694:1  31001490:1  30198239:7  30980115:1  30616738:0
31895670:0  19138896:1  31670824:0   9876287:1  30564898:1  32075777:0
30570982:1

3) Current _fix_control setting in memory:
29331066:0  28965084:0  28776811:0  28498976:0  28567417:0  28558645:0
29132869:0  29450812:0  29687220:0  29304314:0  29930457:0  27261477:0
31069997:0  31077481:0  28602253:0  29653132:0  29937655:0  30347410:0
30602828:0  30896685:0  29487407:0  30998035:0  30786641:0  31444353:0
30486896:0  28999046:0  30902655:0  30681521:0  29302565:0  30972817:0
30222669:0  31668694:0  31001490:0  30198239:0  30980115:0  30616738:0
31895670:0  19138896:0  31670824:0   9876287:0  30564898:0  32075777:0
30570982:0
```

Fix Control Persistence | DBMS_OPTIM_BUNDLE

- Overview
- Check
- Enable
- Output
- **Result**
- Info and Issues

In the SPFILE:

```
*._fix_control='29331066:1','28965084:1','28776811:1','28498976:1','285
67417:1','28558645:1','29132869:1','29450812:1','29687220:1','29304314:
1','29930457:1','27261477:1','31069997:1','31077481:1','28602253:1','29
653132:0','29937655:1','30347410:1','30602828:1','30896685:0','29487407
:1','30998035:1','30786641:1','31444353:0','30486896:1','28999046:1','3
0902655:1','30681521:1','29302565:1','30972817:1','30222669:1','3166869
4:1','31001490:1','30198239:7','30980115:1','30616738:0','31895670:0','
19138896:1','31670824:0','9876287:1','30564898:1','32075777:0','3057098
2:1'#added through dbms_optim_bundle package
```

- Restart necessary to take effect

Fix Control Persistence | DBMS_OPTIM_BUNDLE

Overview
Check
Enable
Output
Result
Info and Issues

Changes do not propagate to standby database

- Set manually for each database
- Use ALTER SYSTEM for databases in MOUNT mode

MOS Note: 2147007.1 - Automatic Fix Control Persistence

Blog Post: DBMS_OPTIM_BUNDLE Package

Blog Post: You may need a one-off in 19.10.0

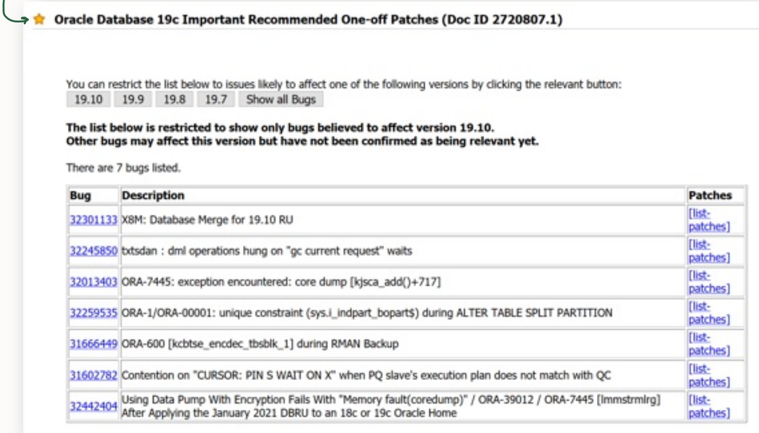
- Oracle 19.10.0 – Patch 31862593 required
- Otherwise, the 19.10 fixes can't be enabled

Blog Post: DBMS_OPTIM_BUNDLE is missing … again?!

- Occasionally the package disappeared in several RUs

Patches | Important One-Offs

Mark as favorite



★ **Oracle Database 19c Important Recommended One-off Patches (Doc ID 2720807.1)**

You can restrict the list below to issues likely to affect one of the following versions by clicking the relevant button:

The list below is restricted to show only bugs believed to affect version 19.10.
Other bugs may affect this version but have not been confirmed as being relevant yet.

There are 7 bugs listed.

Bug	Description	Patches
32301133	X8M: Database Merge for 19.10 RU	[list-patches]
32245850	txtsdan : dml operations hung on "gc current request" waits	[list-patches]
32013403	ORA-7445: exception encountered: core dump [kjsca_add()+717]	[list-patches]
32259535	ORA-1/ORA-00001: unique constraint (sys.i_indpart_bopart$) during ALTER TABLE SPLIT PARTITION	[list-patches]
31666449	ORA-600 [kcbtse_encdec_tbsblk_1] during RMAN Backup	[list-patches]
31602782	Contention on "CURSOR: PIN S WAIT ON X" when PQ slave's execution plan does not match with QC	[list-patches]
32442404	Using Data Pump With Encryption Fails With "Memory fault(coredump)" / ORA-39012 / ORA-7445 [lmmstrmlrg] After Applying the January 2021 DBRU to an 18c or 19c Oracle Home	[list-patches]

Oracle Database 19c
Important Recommended One-off Patches
(Doc ID 2720807.1)

Photo by Veri Ivanova on Unsplash

Statistics

Dictionary Statistics | Overview

Statistics on SYS and other oracle maintained schemas

Gets executed by automatic optimizer statistics gathering

If disabled, consider instead to allow it to work only of dictionary stats

```
SQL> exec dbms_stats.set_global_prefs('autostats_target','oracle');
```

Dictionary Statistics | Gather

Refresh manually:

- Before and after upgrade
- Before (source) and after (target) logical migration
- After major application upgrades

Gather manually

```
SQL> BEGIN
        DBMS_STATS.GATHER_SCHEMA_STATS('SYS');
        DBMS_STATS.GATHER_SCHEMA_STATS('SYSTEM');
     END;
/
```

```
$ORACLE_HOME/perl/bin/perl $ORACLE_HOME/rdbms/admin/catcon.pl \
  -l /tmp \
  -b gatherstats -- \
  --x"begin dbms_stats.gather_schema_stats('SYS'); dbms_stats.gather_schema_stats('SYSTEM'); end;"
```

Fixed Objects Stats | Overview

> *After an upgrade, or after other database configuration changes, Oracle strongly recommends that you regather fixed object statistics after you have run representative workloads on Oracle Database.*

Database 19c Upgrade Guide, chapter 7

Never run it right after upgrade

Fixed Objects Stats | Definition

What is it?

```
SQL> SELECT owner, table_name
     FROM dba_tab_statistics
     WHERE object_type = 'FIXED TABLE';

OWNER    TABLE_NAME

________ _______________
SYS      X$KQFTA
SYS      X$KQFVI
SYS      X$KQFVT
SYS      X$KQFDT
SYS      X$KQFCO
SYS      X$KQFOPT
SYS      X$KYWMPCTAB
...
```

Pro tip: Dynamic statistics (sampling) are not used for X$ tables

Fixed Objects Stats | After Upgrade

Ask yourself: Do you remember this?

If not, DBMS_SCHEDULER to the rescue

Fixed Objects Stats | After Upgrade

1. Create a .sql script

```
BEGIN
   DBMS_SCHEDULER.CREATE_JOB (
      job_name => '"SYS"."GATHER_FIXED_OBJECTS_STATS_ONE_TIME"',
           job_type => 'PLSQL_BLOCK',
           job_action => 'BEGIN DBMS_STATS.GATHER_FIXED_OBJECTS_STATS; END;',
           start_date => SYSDATE+7,
           auto_drop => TRUE,
           comments => 'Gather fixed objects stats after upgrade - one time'
   );
   DBMS_SCHEDULER.ENABLE (
      name => '"SYS"."GATHER_FIXED_OBJECTS_STATS_ONE_TIME"'
   );
END;
/
```

Fixed Objects Stats | After Upgrade

2. Create a .sh script

```
$ORACLE_HOME/perl/bin/perl $ORACLE_HOME/rdbms/admin/catcon.pl \
   -n 4 -e \
   -C 'PDB$SEED' \
   -b sched_gfos -d /home/oracle/sched_gfos/ sched_gfos.sql
```

3. Execute .sh script after upgrade

```
upg1.after_action=/home/oracle/sched_gfos/sched_gfos.sh
```

Fixed Objects Stats | Other situations

Also gather fixed objects stats after:

1. Major application upgrades
2. Using new functionality in the database
3. Major database configuration change

Always gather fixed objects stats when the system is warmed up - after your representative workload

Check out Best Practices for Gathering Optimizer Statistics with Oracle Database 19c
Fixed Objects Statistics (GATHER_FIXED_OBJECTS_STATS) Considerations (Doc ID 798257.1)

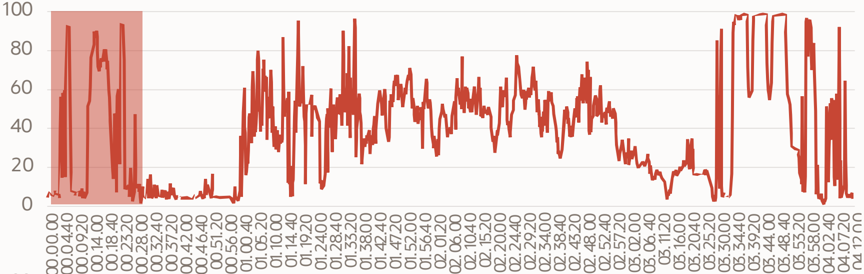
Pro tip: Automated stats gathering only gather fixed objects stats if they are completely missing

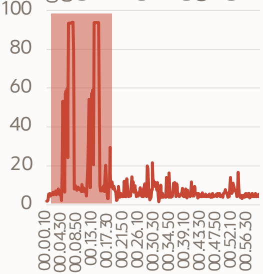
Statistics | Check

Has my stats been refreshed within the last 7 days?

```
SQL> select con_id, operation, target, end_time
from cdb_optstat_operations
where
  (   (operation = 'gather_fixed_objects_stats')
   or (operation = 'gather_dictionary_stats' and (target is null or target in ('SYS', 'SYSTEM')))
   or (operation = 'gather_schema_stats' and target in ('SYS', 'SYSTEM'))
  )
  and end_time > sysdate - 7
order by con_id, end_time;

 CON_ID   OPERATION                      TARGET    END_TIME

_________ _____________________________ _________ _________________________________________
        1 gather_schema_stats           SYS       26-FEB-21 07.00.19.182084000 AM +01:00
        1 gather_schema_stats           SYSTEM    26-FEB-21 07.00.22.351981000 AM +01:00
        1 gather_dictionary_stats                 26-FEB-21 07.05.17.931954000 AM +01:00
        1 gather_fixed_objects_stats              26-FEB-21 07.14.55.088707000 AM +01:00
        2 gather_schema_stats           SYS       26-FEB-21 07.02.40.485494000 AM +01:00
        2 gather_schema_stats           SYSTEM    26-FEB-21 07.02.46.151578000 AM +01:00
        3 gather_schema_stats           SYS       26-FEB-21 07.02.46.171862000 AM +01:00
        3 gather_schema_stats           SYSTEM    26-FEB-21 07.02.49.725878000 AM +01:00
```

Statistics | Gather Stats Before Upgrade





Gathering stats in advance saves 12 minutes

Dictionary and fixed objects

Statistics | Good Stats During Upgrade

The larger the dictionary, the bigger the effect

	DURATION	REDUCTION
No dictionary and fixed objects stats	15 min 55 sec	
Gathered dictionary and fixed objects stats	14 min 10 sec	11 %
Gathered schema and cluster index stats	13 min 41 sec	3.4 % to previous
Total downtime saved	**2 min 14 sec**	**14 % overall**

This example has been done with one of the tiny Hands-On Lab databases

Statistics | Good Stats During Upgrade

Upgrade duration for Oracle E-Business Suite

	DURATION	REDUCTION
No dictionary and fixed objects stats	10 hrs 56 min 52 sec	
Gathered dictionary and fixed objects stats	52 min 42 sec	93 %
Gathered schema and cluster index stats	52 min 25 sec	0.5 % to previous
Total downtime saved	**10 hrs 4 min 14 sec**	**93.5 % overall**

Statistics | Good Stats During Upgrade

Stale / no stats

```
   ID OPERATION            OPTIONS              OBJECT_NAME
----- -------------------- -------------------- -----------
    0 UPDATE STATEMENT
    1   UPDATE                                  DEPENDENCY$
    2     FILTER
    3       TABLE ACCESS   FULL                 DEPENDENCY$
    4       INDEX          FULL SCAN            I_OBJ2
    5       INDEX          FULL SCAN            I_OBJ2
    6       TABLE ACCESS   BY INDEX ROWID BATCHED OBJ$
    7         INDEX        RANGE SCAN           I_OBJ1
    8     TABLE ACCESS     BY INDEX ROWID BATCH OBJ$
    9       INDEX          RANGE SCAN           I_OBJ1
```

9h 59m 23s 87ms

Good stats

```
   ID OPERATION            OPTIONS              OBJECT_NAME
----- -------------------- -------------------- -----------
    0 UPDATE STATEMENT
    1   UPDATE                                  DEPENDENCY$
    2     FILTER
    3       TABLE ACCESS   FULL                 DEPENDENCY$
    4       INDEX          RANGE SCAN           I_OBJ1
    5       INDEX          RANGE SCAN           I_OBJ1
    6       TABLE ACCESS   BY INDEX ROWID BATCHED OBJ$
    7         INDEX        RANGE SCAN           I_OBJ1
    8     TABLE ACCESS     BY INDEX ROWID BATCH OBJ$
    9       INDEX          RANGE SCAN           I_OBJ1
```

2s 33ms

System Statistics | Overview

> ”
> *The system statistics describe hardware characteristics such as I/O and CPU performance and utilization.*
> *System statistics enable the query optimizer to more accurately estimate I/O and CPU costs when choosing execution plans.*

Database 19c SQL Tuning Guide, chapter 10

That sounds like a good idea

System Statistics | Recommendation

> *… in most cases you should use the defaults and not gather system statistics.*

> *Databases supporting a pure data warehouse workload on an Oracle Exadata Database Machine can benefit from system statistics gathered using the EXADATA option*
>
> *… if the workload is mixed or you are not in a position to test the effect of using EXADATA system statistics, then stick to the defaults even on this platform.*

Nigel Bayliss, Optimizer blog

System Statistics | Reference

To delete system statistics (and revert to defaults)

```
SQL> EXEC DBMS_STATS.DELETE_SYSTEM_STATS
```

References:

- Optimizer blog, Should You Gather System Statistics?
- SQL Tuning Guide, System Statistics
- SQL Tuning Guide, Guidelines for Gathering Optimizer Statistics Manually
- Database Performance Tuning Guide, Session and System Statistics

Statistics Advisor | Overview

New in Oracle Database 12.2

Give it a try, but ...

Be aware - potentially it will eat your SYSAUX tablespace

Statistics Advisor | Check

How much space is used?

```
SQL> select occupant_name,space_usage_kbytes
     from v$sysaux_occupants;

OCCUPANT_NAME                  SPACE_USAGE_KBYTES
------------------------------ ------------------
SM/ADVISOR                     5901376
...
```

```
SQL> select * from (
        select segment_name, owner, tablespace_name, bytes/1024/1024 "size(mb)", segment_type
        from dba_segments
        where tablespace_name='SYSAUX'
        order by bytes desc)
     where rownum <= 10;

SEGMENT_NAME                   OWNER      TABLESPACE  SIZE(MB)   SEGMENT_TYPE
------------------------------ ---------- ---------- ---------- ---------------
WRI$_ADV_OBJECTS               SYS        SYSAUX      3600       TABLE
WRI$_ADV_OBJECTS_IDX_01        SYS        SYSAUX      1400       INDEX
```

Statistics Advisor | Disable

If you want to disable the automatic statistics advisor job

1. In 21c, disable the auto task

```
SQL> exec dbms_stats.set_global_prefs('AUTO_STATS_ADVISOR_TASK','FALSE');
```

2. 19c, request backport of bug 26749785 and then disable

3. Or disable with workaround

```
SQL> begin
        dbms_advisor.set_task_parameter('AUTO_STATS_ADVISOR_TASK','_AUTO_MMON_INTERVAL',2147483647);
        dbms_advisor.set_task_parameter('AUTO_STATS_ADVISOR_TASK','_AUTO_STATS_INTERVAL',2147483647);
     end;
     /
```

Pro tip: If you disable the automatic statistics advisor job, you can still do manual executions

Statistics Advisor | Purge

Refer to these two MOS notes:

1. SYSAUX Tablespace Grows Rapidly After Upgrading Database to 12.2.0.1 or Above Due To Statistics Advisor (Doc ID 2305512.1)

2. How To Purge Optimizer Statistics Advisor Old Records From 12.2 Onwards (Doc ID 2660128.1)

Statistics Advisor | References

- Mike Dietrich blog post: Oracle Optimizer Statistics Advisor in Oracle Database 12.2.0.1
- MOS note: SYSAUX Tablespace Grows Rapidly After Upgrading Database to 12.2.0.1 or Above Due To Statistics Advisor (Doc ID 2305512.1)
- MOS note: Optimizer Statistics Advisor In 12.2 (Quick Overview) (Doc ID 2259398.1)
- Oracle Database 19c SQL Tuning Guide, Analyzing Statistics Using Optimizer Statistics Advisor

Stats | Things to do right after upgrade

Configure statistics history retention period

- Check space usage:

```
SQL> select space_usage_kbytes/1024 mb
     from   v$sysaux_occupants
     where  occupant_name='SM/OPTSTAT';
```

- Check retention
  - Default: 31 days

```
SQL> select dbms_stats.get_stats_history_retention from dual;
```

- Adjust setting
  - Example: 10 days

```
SQL> exec dbms_stats.alter_stats_history_retention(10);
```

Stats | Pending Statistics

Potentially mitigate the risk for newly created stats with Pending Statistics

- Turn on Pending Stats

```
SQL> exec dbms_stats.set_global_prefs('PENDING','TRUE');
```

- Gather new stats
  - Only pending
  - Not visible to the app

```
SQL> exec dbms_stats.gather_schema_stats('SH');
```

- Verify stats

```
SQL> alter session set optimizer_use_pending_statistics=TRUE;
```

- If stats are ok, publish

```
SQL> exec dbms_stats.publish_pending_stats;
```

Testing
Best Practices

Photo by Battlecreek Coffee Roasters on Unsplash

“ Help me - I have an upgrade problem …”

In

95%

of all cases, "upgrade problem" in reality is a performance issue after upgrade. Or not database related.

There is exactly one way to mitigate the risk.

TESTING!

Photo by nikko macaspac on Unsplash

Testing | Typical Mistakes

Only 10% of real data used
Artificially created data sets
Outdated data
Tests done on a laptop
No testing tools used
No stale statistics refreshed

”

Testing?? What a waste of time!

- *Real* experts fix it *after* go-live …

Photo by Elijah O'Donnell on Unsplash

Testing | Completeness versus Costs

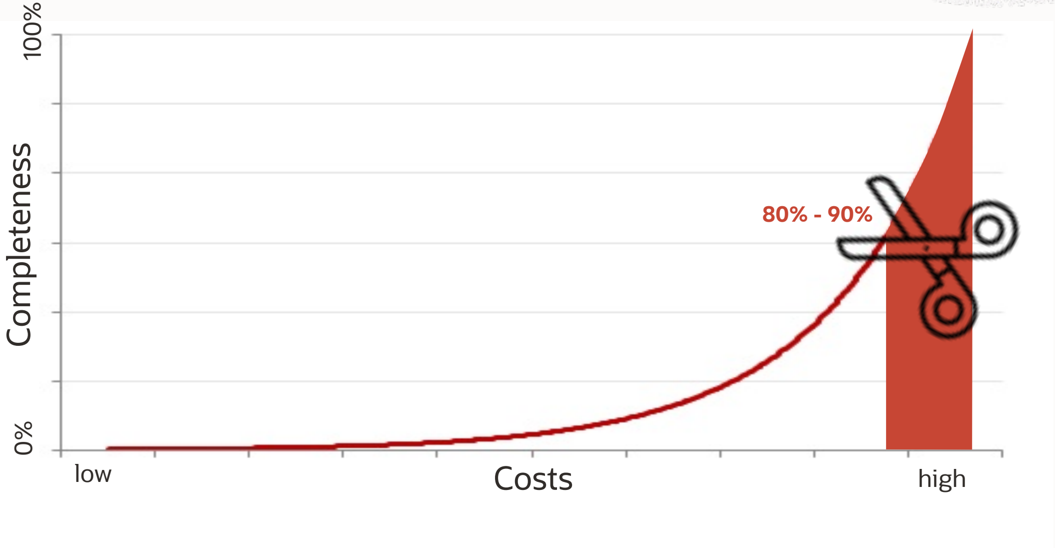


Photo by Alex Motoc on Unsplash

Testing

Setup a proper test environment

Test Environments | Ideas

Snapshot standby database

- Leverage existing standby databases
- Increase RTO a little - and gain a *free* test environment

Test Environments | Ideas

Hybrid Data Guard in Oracle Cloud Infrastructure

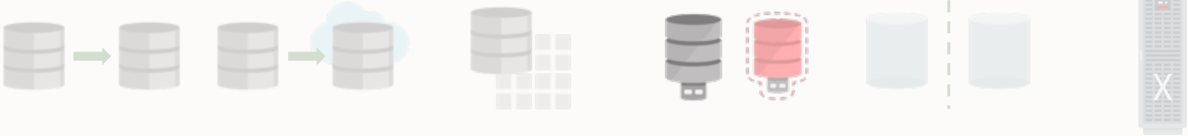
- Create as many as you like
- Pay-as-you-go

Test Environments | Ideas

CloneDB

- Copy-on-write
- Uses image copies of data files stored on NFS, delta is written locally

Test Environments | Ideas



Snapshot Copy PDBs

- Requires compatible storage system
- Or, use CloneDB functionality (requires source PDB is read-only)

Test Environments | Ideas

Split Mirror Clone PDBs

- Requires ASM and Oracle Database 18c
- A point-in-time version of a PDB

Test Environments | Ideas

Exadata Sparse Snapshots

- Space savings - fast provisioning
- Clone still has access to Exadata storage features

Pro tip: Cool blog post on PDB sparse clone

Photo by Carlos Muza on Unsplash

Testing

Take care on statistics

Statistics | Refresh?

Should you refresh object statistics when you upgrade to Oracle 19c?

- It is not required

- But especially when you upgrade from 11.2, histograms can change
  - Avoid gradual change of plans when stats become stale
  - Better regather object statistics as soon as possible

Statistics | Refresh?

Want to gather statistics fast?

```
SQL> exec dbms_stats.set_global_prefs('CONCURRENT','AUTOMATIC');
```

Even faster (if you have CPU available)?

```
SQL> exec dbms_stats.set_global_prefs('DEGREE', DBMS_STATS.AUTO_DEGREE);
```

Fastest (if you have a lot of CPU available)?

```
SQL> exec dbms_stats.set_global_prefs('CONCURRENT','AUTOMATIC');
SQL> exec dbms_stats.set_global_prefs('DEGREE', DBMS_STATS.AUTO_DEGREE);
```

Or import fresh statistics from a matching test system

Pro tip: Read Nigel Bayliss' blog on
How to Gather Optimizer Statistics Fast!

Transporting Statistics | Overview

> *When you transport optimizer statistics between databases, you must use DBMS_STATS to copy the statistics to and from a staging table, and tools to make the table contents accessible to the destination database.*

Database 19c SQL Tuning Guide, chapter 17

You can transport the following statistics

- Schema
- Table
- *Database (rare)*
- *Dictionary and fixed objects (rare)*

Pro tip: You can read more about transporting statistics in the SQL Tuning Guide

Transporting Statistics | Use Cases

Use production statistics in test environments

- No data
- Subset of data

Transporting Statistics | Use Cases

- Use production statistics for query tuning

Transporting Statistics | Use Cases

Get new statistics after upgrading without re-gathering

- Benefit from optimizer changes
- If time does not allow for complete re-gathering of statistics

Transporting Statistics | Use Cases

Get new statistics after migration without re-gathering

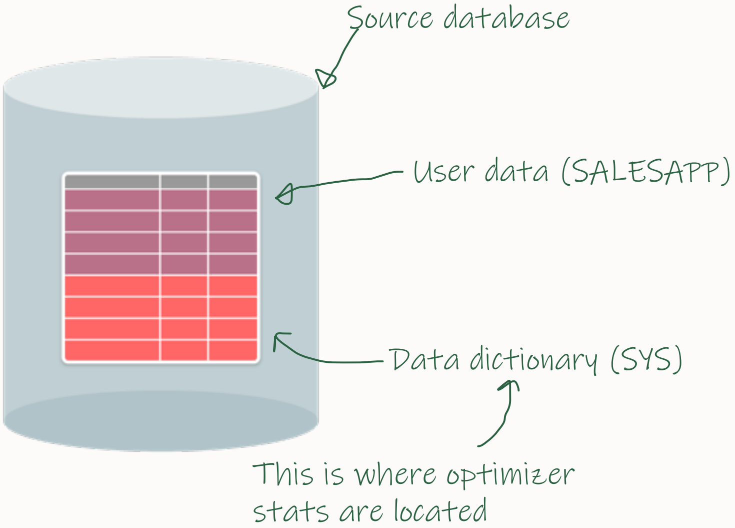
- Character set migration requires new statistics
- Logical migrations (transportable tablespaces, import)

Transporting Statistics | Use Cases

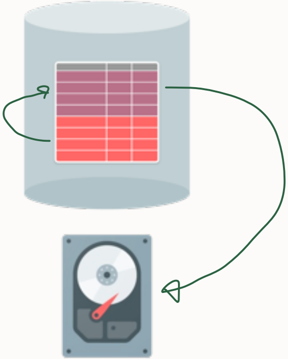
Faster than transporting statistics using Data Pump

- Replace native Data Pump statistics export with DBMS_STATS
- Superior performance

Transporting Statistics | Workflow



Transporting Statistics | Workflow



Create staging table

```
SQL> EXEC DBMS_STATS.CREATE_STAT_TABLE (
       ownname => 'SALESAPP',
       stattab => 'OPT_STATS_STG');
```

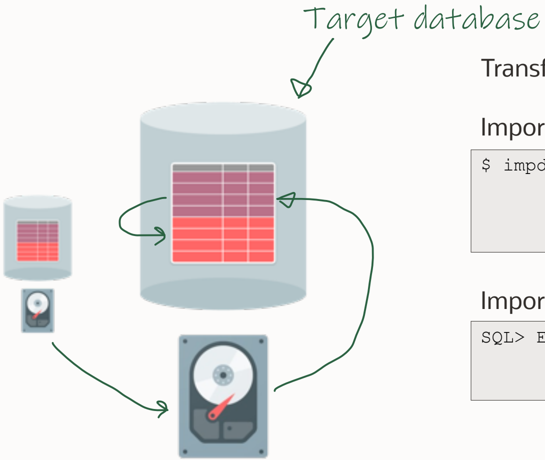
Export statistics to staging table

```
SQL> EXEC DBMS_STATS.EXPORT_SCHEMA_STATS (
       ownname => 'SALESAPP',
       stattab => 'OPT_STATS_STG');
```

Export staging table using Data Pump

```
$ expdp SALESAPP \
      DIRECTORY=mydirectory \
      DUMPFILE=opt_stats_stg.dmp \
      TABLES=OPT_STATS_STG
```

Transporting Statistics | Workflow



Transfer dump file

Import staging table using Data Pump

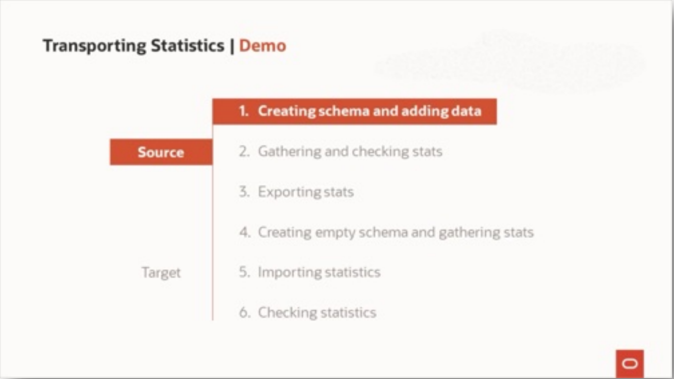
```
$ impdp SALESAPP \
      DIRECTORY=mydirectory \
      DUMPFILE=opt_stats_stg.dmp \
      TABLES=OPT_STATS_STG
```

Import statistics from staging table

```
SQL> EXEC DBMS_STATS.IMPORT_SCHEMA_STATS (
       ownname => 'SALESAPP',
       stattab => 'OPT_STATS_STG');
```

Pro tip: You can also import directly from source database using a database link

Transporting Statistics | Demo



Transporting Statistics | Demo

Source

Target

1. **Creating schema and adding data**
2. Gathering and checking stats
3. Exporting stats
4. Creating empty schema and gathering stats
5. Importing statistics
6. Checking statistics

Watch on YouTube

Transporting Statistics | Nice to Know

- The optimizer does not use statistics stored in a user-owned table - only from dictionary

- Importing statistics make them current (i.e. not stale)

- You can transfer to a higher version - potentially the stats table must be upgraded

```
SQL> EXEC DBMS_STATS.IMPORT_SCHEMA_STATS ( ...

ORA-20002: Version of statistics table "SALESAPP"."OPT_STATS_STG" is too old

SQL> EXEC DBMS_STATS.UPGRADE_STAT_TABLE ('SALESAPP', 'OPT_STATS_STG');
```

- Incremental statistics: optionally export synopses as well

```
SQL> EXEC DBMS_STATS.EXPORT_SCHEMA_STATS (
        ...
        stat_category => 'OBJECT_STATS, REALTIME_STATS, SYNOPSES');
```

Transporting Statistics | Nice to Know - 2

- Gather stats on the staging table after Data Pump import
  - Also, gather on staging table indexes
  - Before executing `DBMS_STATS.IMPORT_SCHEMA_STATS / IMPORT_TABLE_STATS`

- If enabled, imported statistics will be added as pending stats until you publish them

- Statistics preferences (`degree`, `method_opt`, `incremental` etc.) are not transported
  - Separate precedures
    `DBMS_STATS.EXPORT_TABLE_PREFS` / `DBMS_STATS.IMPORT_TABLE_PREFS`
  - Database-, schema- and table-level procedures
  - Uses the same staging table

Transporting Statistics | Nice to Know - 3

- Important bug fixes:
  - BUG 29296074 - OPT: INTERNAL MERGE STATEMENT FOR SCAN RATE IS SLOW USING GV$ TABLES

Export, Import or Deletion of Table Statistics into statistics table is slow with DBMS_STATS in 19.16 (Doc ID 2949078.1)

Transporting Statistics | Customer Feedback

> *We have adopted this method for stats. We migrated 60 TB database from AIX to Exadata using cross-platform transportable tablespace without stats.*
>
> *Gathering stats from scratch took more than 36 hours.*
> *We transported the statistics in less then 2 hours.*

Taoqir Hassan, comment on YouTube channel

Photo by Jen Theodore on Unsplash

You finally did it.

You produced the ultimate stats.

It is time to lock 'em!

Locking Statistics | Use Cases

> ”
>
> *You can lock statistics to prevent them from changing.*

Database 19c SQL Tuning Guide, chapter 15

- Certain static environments
- Highly volatile tables
- Enable use of dynamic statistics
- … and all the exceptions

Locking Statistics | Show it

Lock table statistics

```
SQL> EXEC DBMS_STATS.LOCK_TABLE_STATS(ownname=>'MYAPP', tabname=>'MY_VOLATILE_TAB1');
```

You can also lock at:

- Schema-level
- Partition-level

You can also unlock statistics

Pro tip: Locking table statistics also lock index and partition statistics

Locking Statistics | Worth mentioning

- Locking and unlocking statistics causes cursor invalidation
- To achieve plan stability, consider SQL Plan Management
- Statistics advisor will warn you about locked statistics
- Locking information is not exported

Incremental Statistics | Overview

Introduced in Oracle 11g

Improved since Oracle 12.2.0.1

Concept

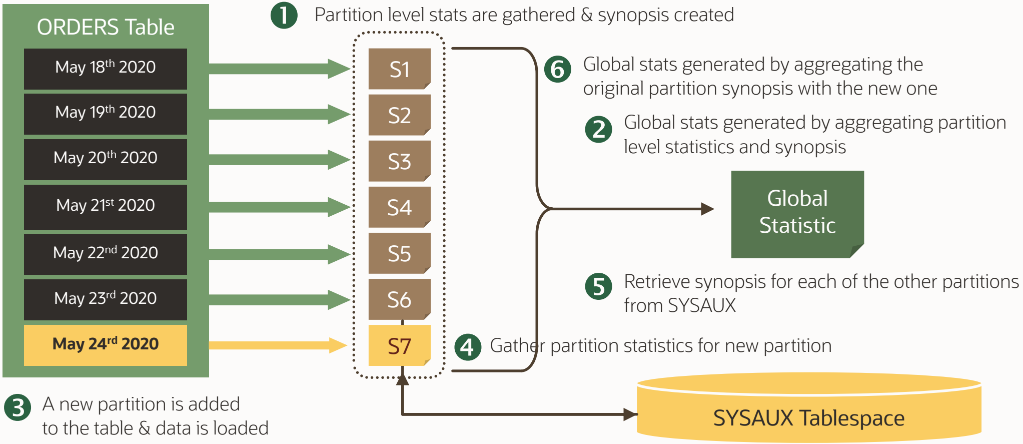
- Update global statistics for partitioned objects quickly with less overhead
- List and interval partitioning

Known pitfalls

- Original synopses on disk can easily consume hundreds of GB
- By default, transporting stats with DBMS_STATS does not include synopses

New efficient algorithm for synopses can drastically reduce space consumption

Incremental Statistics | Concept



1 Partition level stats are gathered & synopsis created

2 Global stats generated by aggregating partition level statistics and synopsis

3 A new partition is added to the table & data is loaded

4 Gather partition statistics for new partition

5 Retrieve synopsis for each of the other partitions from SYSAUX

6 Global stats generated by aggregating the original partition synopsis with the new one

Incremental Statistics | Important Configuration Steps

Enable incremental stats on a per-table basis

```
SQL> EXEC dbms_stats.set_table_prefs(null,'ORDERS','INCREMENTAL','TRUE');
```

Must-Do

- *Changed* partitions won't be eligible for new stats generation

```
SQL> exec dbms_stats.set_database_prefs('INCREMENTAL_STALENESS','USE_STALE_PERCENT');
```

Optional

- Adjust the stale percentage – default: 10%

```
SQL> exec dbms_stats.set_database_prefs('STALE_PERCENT','15');
```

Incremental Statistics | Upgrade to Oracle 19c

New very efficient HyperLogLog algorithm used

- Synopses shrink significantly by factors of 10x – 25x
- Available since Oracle 12.2.0.1

Two options

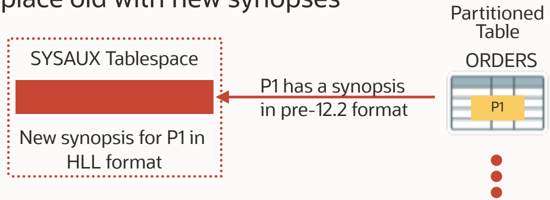
- Coexistence of old and new synopses (default)
- Regathering of synopses

Recommendation

- Use regathering as coexistence leads to issues

Incremental Statistics | Upgrade to Oracle 19c

Replace old with new synopses


Partitioned Table
ORDERS
SYSAUX Tablespace
P1 has a synopsis in pre-12.2 format
P1
New synopsis for P1 in HLL format

- Drop old synopses

```
SQL> exex dbms_stats.delete_table_stats(…, cascade_indexes=>FALSE, stat_category=>'SYNOPSES');
```

- Choose new HLL algorithm

```
SQL> exec dbms_stats.set_table_prefs('XY','ORDERS','APPROXIMATE_NDV_ALGORITHM','HYPERLOGLOG');
```

- Disallow coexistence and force regathering

```
SQL> exec dbms_stats.set_table_prefs('XY','ORDERS','INCREMENTAL_STALENESS','NULL';
```

See also: Optimizer Blog

Performance Stability Prescription

1. Collect

2. Compare

3. Analyze

4. Tune

5. Manage

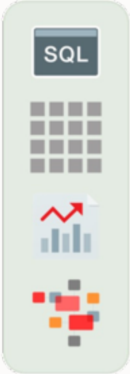
6. Test

SQL Tuning Set | Definition

> *An SQL Tuning Set (STS) enables you to group SQL statements and related metadata in a single database object, which you can use to meet your tuning goals.*
>
> *Specifically, SQL tuning sets achieve the following goals:*
>
> - *Providing input to the performance tuning advisors*
> - *Transporting SQL between databases*

Database 19c SQL Tuning Guide, chapter 23

SQL Tuning Set | Definition



SQL statement

Context

Statistics

Plans

SQL Tuning Set | Create

First, create a SQL Tuning Set

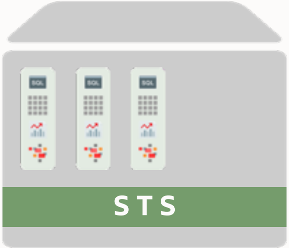
```
SQL> BEGIN
  DBMS_SQLSET.CREATE_SQLSET (
    sqlset_name  => 'UPG_STS_1',
    description  => 'For upgrade - from source'
  );
END;
/
```

Pro tip: You can also use DBMS_SQLTUNE to create a SQL Tuning Set

SQL Tuning Set | Capture

Next, capture statements from AWR

```
SQL> DECLARE
   begin_id number;
   end_id number;
   cur sys_refcursor;
BEGIN
   SELECT min(snap_id), max(snap_id) INTO begin_id, end_id
   FROM dba_hist_snapshot;

open cur for
  select value(p) from table(dbms_sqltune.select_workload_repository(
       begin_snap        => begin_id,
       end_snap          => end_id,
       basic_filter      => 'parsing_schema_name not in (''SYS'')',
       ranking_measure1 => 'elapsed_time',
       result_limit      => 5000,
       attribute_list    => 'ALL')) p;

  dbms_sqltune.load_sqlset('UPG_STS_1', cur);

close cur;

END;
/
```



Pro tip: Consider excluding other internal schemas like *DBSNMP*, *ORACLE_OCM*, *LBACSYS*, *WMSYS*, *XDB*, *SYSTEM*

SQL Tuning Set | Capture

Optionally, capture statements from cursor cache

```
SQL> BEGIN
   DBMS_SQLSET.CAPTURE_CURSOR_CACHE_SQLSET(
      sqlset_name     => 'UPG_STS_1',
      time_limit      => 900,
      repeat_interval => 60,
      capture_option  => 'MERGE',
      capture_mode    => DBMS_SQLTUNE.MODE_ACCUMULATE_STATS,
      basic_filter    => 'parsing_schema_name not in (''SYS'')',
      sqlset_owner    => NULL,
      recursive_sql   => 'HAS_RECURSIVE_SQL');
   END;
/
```

Careful - puts load on your system

Pro tip: SQL Tuning Guide shows how to load all statements from a given schema

SQL Tuning Set | Transport

Pack into staging table on source database

```
SQL> BEGIN
   DBMS_SQLTUNE.CREATE_STGTAB_SQLSET (
      table_name          => 'UPG_STGTAB_1');
   DBMS_SQLTUNE.PACK_STGTAB_SQLSET (
      sqlset_name         => 'UPG_STS_1',
      staging_table_name  => 'UPG_STGTAB_1');
END;
```

Optionally, use `DBMS_SQLTUNE.REMAP_STGTAB_SQLSET` to remap between `CON_DBID`

Export with Data Pump

```
$ expdp user \
     directory=mydirectory
     dumpfile=upg_stgtab_1.dmp
     tables=UPG_STGTAB_1
```

SQL Tuning Set | Transport

Import with Data Pump to target database

```
$ impdp user \
     directory=mydirectory
     dumpfile=upg_stgtab_1.dmp
     tables=UPG_STGTAB_1
```

Unpack staging table

```
SQL> BEGIN
   DBMS_SQLTUNE.UNPACK_STGTAB_SQLSET (
      sqlset_name        => '%',
      replace            => true,
      staging_table_name => 'UPG_STGTAB_1'
   );
END;
/
```

SQL Tuning Set | License

> *SQL Tuning Sets can also be accessed by way of database server APIs and command-line interfaces. Usage of any subprograms in the DBMS_SQLSET package to manage SQL Tuning Sets is part of the EE and EE-ES offerings.*
>
> *In addition, the following subprograms, part of the DBMS_SQLTUNE package, provide an older interface to manage SQL Tuning Sets and are also part of the EE and EE-ES offerings:*
>
> *ADD_SQLSET_REFERENCE*
> *CAPTURE_CURSOR_CACHE_SQLSET*
> *CREATE_SQLSET*
> *CREATE_STGTAB_SQLSET*
> *DELETE_SQLSET*
> *DROP_SQLSET*
> *LOAD_SQLSET*
> *PACK_STGTAB_SQLSET*
> *REMOVE_SQLSET_REFERENCE*
> *SELECT_CURSOR_CACHE*
> *SELECT_SQLSET*
> *SELECT_WORKLOAD_REPOSITORY*
> *UNPACK_STGTAB_SQLSET*
> *UPDATE_SQLSET*

Database 19c Database Licensing Information User Manual

SQL Tuning Set | Recommendation

Always capture workload data into SQL Tuning Sets

Collect at least
one full month of workload data
before any upgrade or migration

Workload Information

AWR – Automatic Workload Repository

Change the retention to a minimum of 40 days

```
exec
dbms_workload_repository.modify_snapshot_settings(
retention=>57600, interval=>30);
```

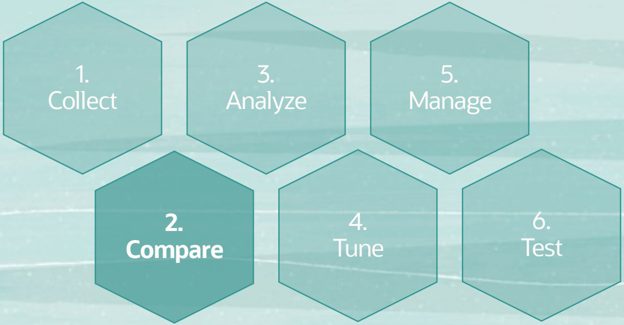
Collect SQL statements and plans

Use AWR as main source

Capture from Cursor Cache for OLTP

Collect statements, plans and stats in SQL Tuning Sets

Performance Stability Prescription



1. Collect
2. **Compare**
3. Analyze
4. Tune
5. Manage
6. Test

AWR | Diff Report

Compare AWR report
from two different periods

1. AWR snapshot
2. Execute workload
3. AWR snapshot

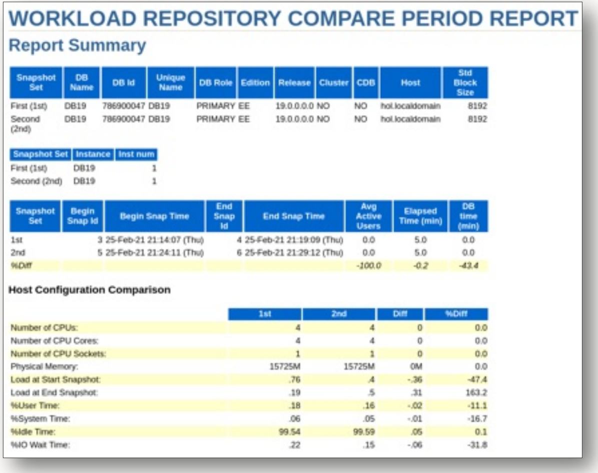
4. Upgrade

5. AWR snapshot
6. Execute workload
7. AWR snapshot

8. Compare

AWR | Diff Report

Use script `awrddrpt.sql`



WORKLOAD REPOSITORY COMPARE PERIOD REPORT

Report Summary

Snapshot Set	DB Name	DB Id	Unique Name	DB Role	Edition	Release	Cluster	CDB	Host	Std Block Size
First (1st)	DB19	786900047	DB19	PRIMARY	EE	19.0.0.0.0	NO	NO	hol.localdomain	8192
Second (2nd)	DB19	786900047	DB19	PRIMARY	EE	19.0.0.0.0	NO	NO	hol.localdomain	8192

Snapshot Set	Instance	Inst num
First (1st)	DB19	1
Second (2nd)	DB19	1

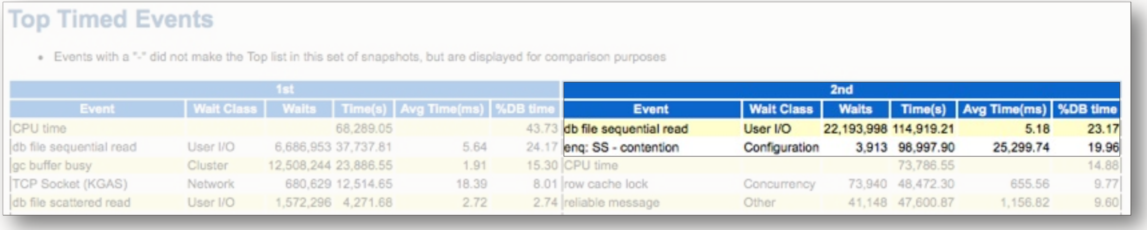
Snapshot Set	Begin Snap Id	Begin Snap Time	End Snap Id	End Snap Time	Avg Active Users	Elapsed Time (min)	DB time (min)
1st	3	25-Feb-21 21:14:07 (Thu)	4	25-Feb-21 21:19:09 (Thu)	0.0	5.0	0.0
2nd	5	25-Feb-21 21:24:11 (Thu)	6	25-Feb-21 21:29:12 (Thu)	0.0	5.0	0.0
%Diff					-100.0	-0.2	-43.4

Host Configuration Comparison

	1st	2nd	Diff	%Diff
Number of CPUs:	4	4	0	0.0
Number of CPU Cores:	4	4	0	0.0
Number of CPU Sockets:	1	1	0	0.0
Physical Memory:	15725M	15725M	0M	0.0
Load at Start Snapshot:	.76	.4	-.36	-47.4
Load at End Snapshot:	.19	.5	.31	163.2
%User Time:	.18	.16	-.02	-11.1
%System Time:	.06	.05	-.01	-16.7
%Idle Time:	99.54	99.59	.05	0.1
%IO Wait Time:	.22	.15	-.06	-31.8

AWR | Diff Report

Use script `awrddrpt.sql`



Top Timed Events

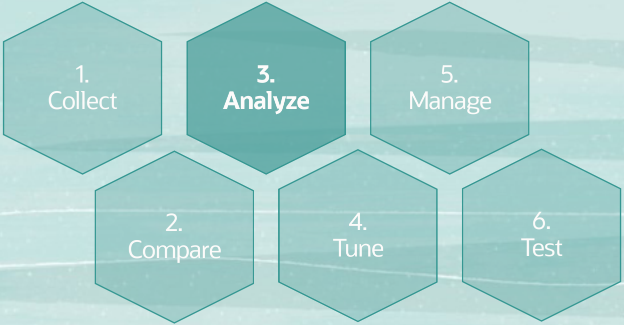
- Events with a "-" did not make the Top list in this set of snapshots, but are displayed for comparison purposes

1st						2nd					
Event	Wait Class	Waits	Time(s)	Avg Time(ms)	%DB time	Event	Wait Class	Waits	Time(s)	Avg Time(ms)	%DB time
CPU time			68,289.05		43.73	db file sequential read	User I/O	22,193,998	114,919.21	5.18	23.17
db file sequential read	User I/O	6,686,953	37,737.81	5.64	24.17	enq: SS - contention	Configuration	3,913	98,997.90	25,299.74	19.96
gc buffer busy	Cluster	12,508,244	23,886.55	1.91	15.30	CPU time			73,786.55		14.88
TCP Socket (KGAS)	Network	680,629	12,514.65	18.39	8.01	row cache lock	Concurrency	73,940	48,472.30	655.56	9.77
db file scattered read	User I/O	1,572,296	4,271.68	2.72	2.74	reliable message	Other	41,148	47,600.87	1,156.82	9.60

Requires Enterprise Edition + Diagnostic pack

Pro tip: For migrations, you can transport AWR data

Performance Stability Prescription



1. Collect

2. Compare

3. Analyze

4. Tune

5. Manage

6. Test

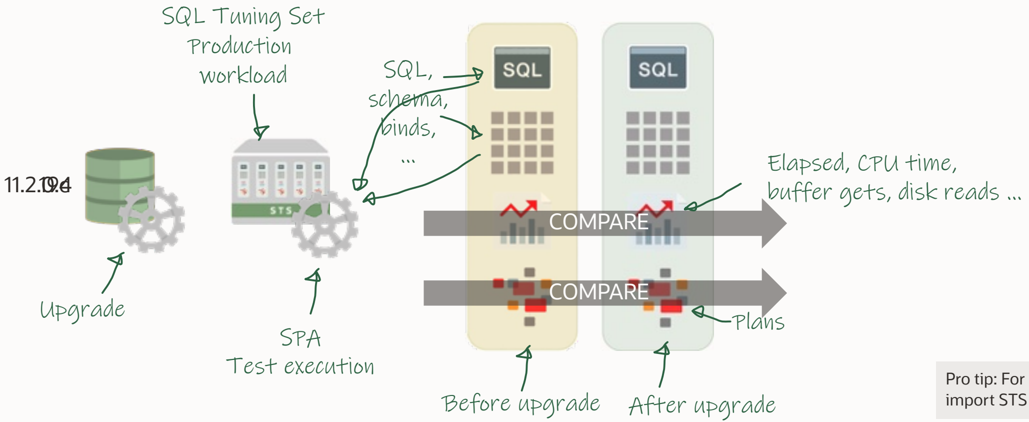
SQL Performance Analyzer | SPA

> *SPA provides fine-grained assessment of environment changes on SQL execution plans and statistics by running the SQL statements both in isolation and serially manner in before-change and after-change environments.*
>
> *SPA functionality is well integrated with existing SQL Tuning Set (STS), SQL Tuning Advisor, and SQL Plan Management functionality.*

Oracle Database Real Application Testing Data Sheet

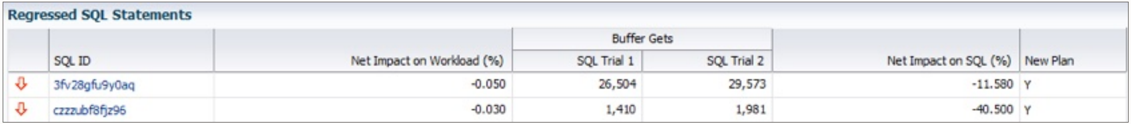
Requires Enterprise Edition + Real Application Testing

SPA | Concept



Pro tip: For migrations, import STS into target database

SPA | Regressed Report



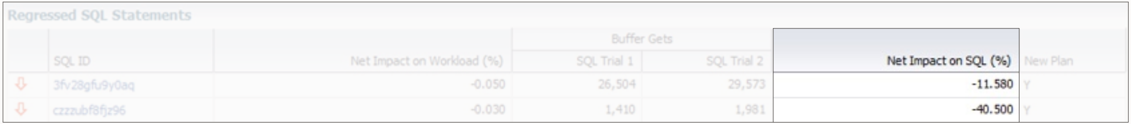
Regressed SQL Statements

			Buffer Gets			
	SQL ID	Net Impact on Workload (%)	SQL Trial 1	SQL Trial 2	Net Impact on SQL (%)	New Plan
⇩	3fv28gfu9y0aq	-0.050	26,504	29,573	-11.580	Y
⇩	czzzubf8fjz96	-0.030	1,410	1,981	-40.500	Y

From production workload

From test execution

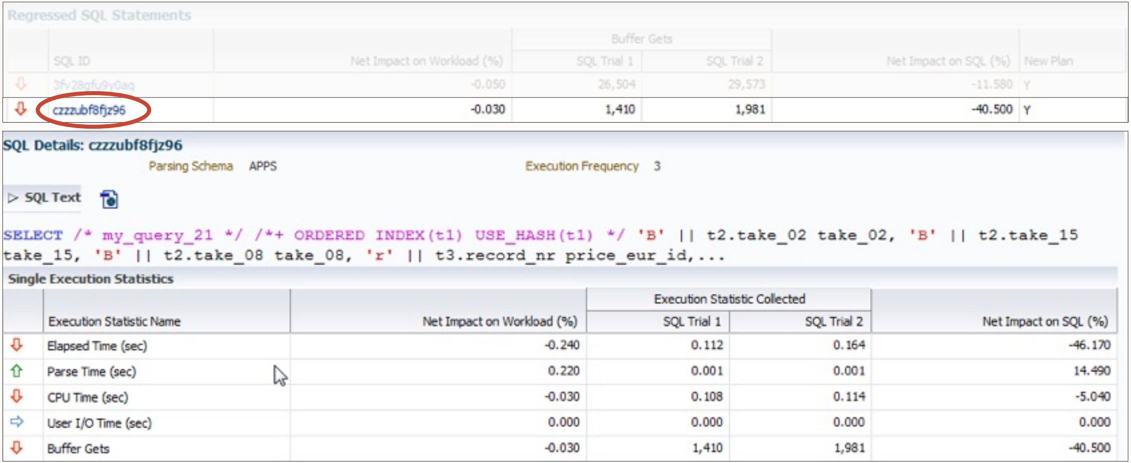
SPA | Regressed Report



Regressed SQL Statements

			Buffer Gets			
	SQL ID	Net Impact on Workload (%)	SQL Trial 1	SQL Trial 2	Net Impact on SQL (%)	New Plan
⇩	3fv28gfu9y0aq	-0.050	26,504	29,573	-11.580	Y
⇩	czzzubf8fjz96	-0.030	1,410	1,981	-40.500	Y

SPA | Regressed Report



Regressed SQL Statements

			Buffer Gets			
	SQL ID	Net Impact on Workload (%)	SQL Trial 1	SQL Trial 2	Net Impact on SQL (%)	New Plan
⇩	3fv28gfu9y0aq	-0.050	26,504	29,573	-11.580	Y
⇩	czzzubf8fjz96	-0.030	1,410	1,981	-40.500	Y

SQL Details: czzzubf8fjz96

Parsing Schema APPS

Execution Frequency 3

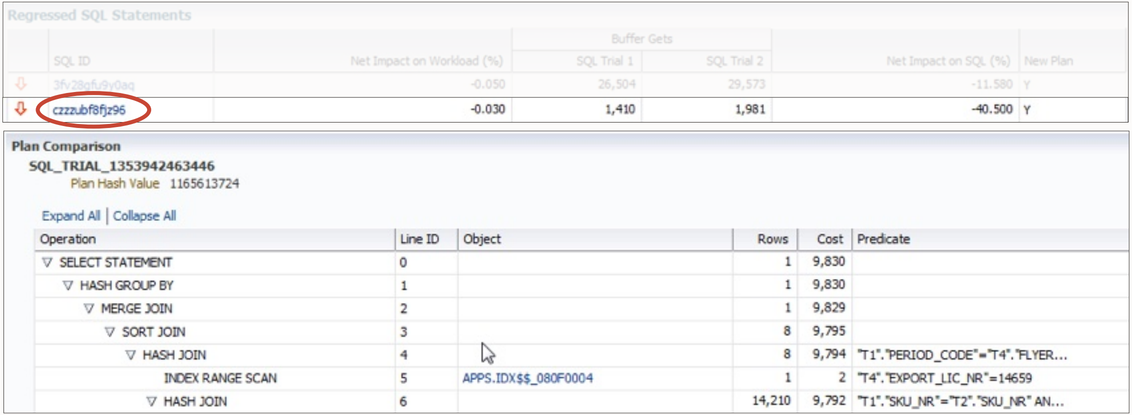
▷ SQL Text

```
SELECT /* my_query_21 */ /*+ ORDERED INDEX(t1) USE_HASH(t1) */ 'B' || t2.take_02 take_02, 'B' || t2.take_15
take_15, 'B' || t2.take_08 take_08, 'r' || t3.record_nr price_eur_id,...
```

Single Execution Statistics

			Execution Statistic Collected		
	Execution Statistic Name	Net Impact on Workload (%)	SQL Trial 1	SQL Trial 2	Net Impact on SQL (%)
⇩	Elapsed Time (sec)	-0.240	0.112	0.164	-46.170
⇧	Parse Time (sec)	0.220	0.001	0.001	14.490
⇩	CPU Time (sec)	-0.030	0.108	0.114	-5.040
⇨	User I/O Time (sec)	0.000	0.000	0.000	0.000
⇩	Buffer Gets	-0.030	1,410	1,981	-40.500

SPA | Regressed Report



Regressed SQL Statements

			Buffer Gets			
	SQL ID	Net Impact on Workload (%)	SQL Trial 1	SQL Trial 2	Net Impact on SQL (%)	New Plan
⇩	3fv28gfu9y0aq	-0.050	26,504	29,573	-11.580	Y
⇩	czzzubf8fjz96	-0.030	1,410	1,981	-40.500	Y

Plan Comparison

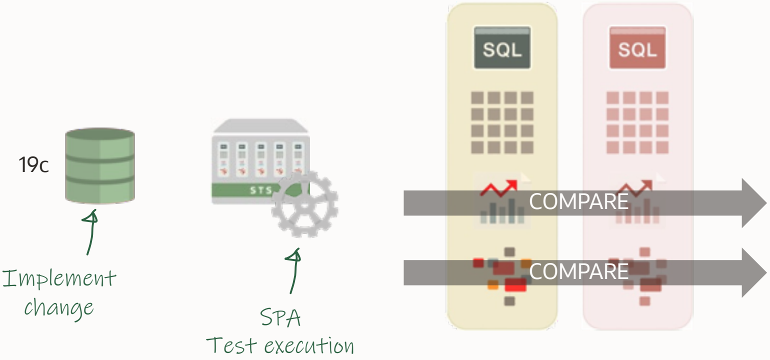
SQL_TRIAL_1353942463446

Plan Hash Value 1165613724

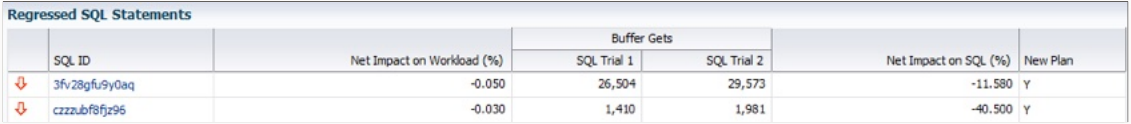
Expand All | Collapse All

Operation	Line ID	Object	Rows	Cost	Predicate
▽ SELECT STATEMENT	0		1	9,830	
▽ HASH GROUP BY	1		1	9,830	
▽ MERGE JOIN	2		1	9,829	
▽ SORT JOIN	3		8	9,795	
▽ HASH JOIN	4		8	9,794	"T1"."PERIOD_CODE"="T4"."FLYER...
INDEX RANGE SCAN	5	APPS.IDX$$_080F0004	1	2	"T4"."EXPORT_LIC_NR"=14659
▽ HASH JOIN	6		14,210	9,792	"T1"."SKU_NR"="T2"."SKU_NR" AN...

SPA | Continuous Improvement



SPA | Regressed Report



Regressed SQL Statements

	SQL ID	Net Impact on Workload (%)	Buffer Gets		Net Impact on SQL (%)	New Plan
			SQL Trial 1	SQL Trial 2		
⇩	3fv28gfu9y0aq	-0.050	26,504	29,573	-11.580	Y
⇩	czzzubf8fjz96	-0.030	1,410	1,981	-40.500	Y

SPA also compares the number of rows returned and the value of those

SPA computes a hash value of the result set for comparison

👍

Don't have a license for Real Application Testing?
Use OCI!

Check out *Performance Testing using the Oracle Cloud for Upgrades and Migrations*

Performance Stability Prescription

1. Collect

2. Compare

3. Analyze

4. Tune

5. Manage

6. Test

SQL Tuning Advisor

> *SQL Tuning Advisor is SQL diagnostic software in the Oracle Database Tuning Pack.*
>
> *...*
>
> *SQL Tuning Advisor is a mechanism for resolving problems related to suboptimally performing SQL statements.*

Database 19c SQL Tuning Guide, chapter 24

SQL Tuning Advisor | Findings

Types of findings:

1. Collection of object statistics
2. Creation of indexes
3. Rewriting SQL statements
4. Creation of SQL profiles

.... and more

Pro tip: SQL Developer has a good interface to SQL Tuning Advisor

SQL Profiles

> *A SQL profile is a database object that contains auxiliary statistics specific to a SQL statement.*
>
> *...*
>
> *The corrected statistics in a SQL profile can improve optimizer cardinality estimates, which in turn leads the optimizer to select better plans.*

Database 19c SQL Tuning Guide, chapter 26

SQL Profiles | Facts

1. Part of Tuning Pack
   - Included in some cloud offerings

2. Stores a set of hints that causes the optimizer to select a plan

3. Affects one statement only

SQL Profiles | Facts

4. You can enable/disable a profile

5. Transparent to application
   - Does not require application changes

6. Persistent and transportable
   - Documentation

7. Useful with literals using `FORCE_MATCH=TRUE`

SQL Profile | Testing

1. Enable profile for selected environments only

```
SQL> exec :p_name := dbms_sqltune.accept_sql_profile(
   task_name=>'TASK_21944',
   name=>'XT_PROFILE',
   category=>'TEST_ENV');
```

2. Verify the profile – it doesn't get used by the optimizer in the live environment

```
SQL> alter session set sqltune_category='TEST_ENV';
```

3. Accept and make visible to all sessions (`'DEFAULT'`)

```
SQL> exec dbms_sqltune.alter_sql_profile(
   name=>'XT_PROFILE',
   attribute_name=>'CATEGORY',
   value=>'DEFAULT');
```

Photo by Fabio Santaniello Bruun on Unsplash

Transporting SQL Profiles

Transporting | SQL Profiles

Prepare

Extract

Transfer

Load

SQL Profiles are stored in data dictionary

To transfer - profiles must be converted to a transportable format and stored in a staging table

```
SQL> BEGIN
   DBMS_SQLTUNE.CREATE_STGTAB_SQLPROF (
      table_name    => 'STAGING',
      table_owner   => 'SQLPROFILES');
   END;
/
```

Pro tip: Create the staging table in a schema and/or tablespace that you are migrating

Transporting | SQL Profiles

Prepare

Extract

Transfer

Load

Select the profiles that you want to transfer

To extract all profiles from `DEFAULT` category

```
SQL> BEGIN
   DBMS_SQLTUNE.PACK_STGTAB_SQLPROF (
      staging_table_name   => 'STAGING',
      staging_schema_owner => 'SQLPROFILES');
   END;
/
```

Pro tip: You can filter on `profile_name` and `profile_category` as well

Transporting | SQL Profiles

Prepare

Extract

Transfer

Load

If the staging table is migrated together with the user data, you can skip this step

Use Data Pump to transfer that single table

```
SQL> CREATE DATABASE LINK src_link ... ;

$ impdp system network_link=src_link \
      tables=SQLPROFILES.STAGING ...
```

Pro tip: You can also import from dump file if there is no network connectivity to source database

Transporting | SQL Profiles

Prepare

Extract

Transfer

Load

Finally, load the profiles from the staging table into the data dictionary

```
SQL> BEGIN
   DBMS_SQLTUNE.UNPACK_STGTAB_SQLPROF (
      staging_table_name    => 'STAGING',
      staging_schema_owner => 'SQLPROFILES',
      replace              => TRUE);
   END;
/
```

Pro tip: You can load a SQL profile into the same or higher release

Transporting | SQL Profiles

Additional resources

Documentation

Database 19c, SQL Tuning Guide, Transporting a SQL Profile

How to Move SQL Profiles from One Database to Another (Including to Higher Versions) (Doc ID 457531.1)

Photo by August Phlieger on Unsplash

Real World Example

SQL Tuning Advisor in Action

6 simple steps with

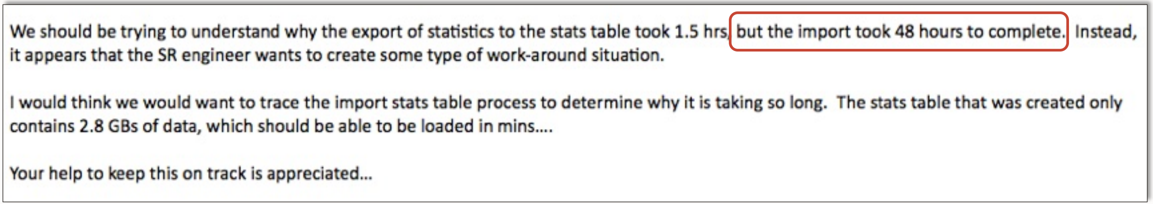
SQL TUNING ADVISOR

can make a huge difference

1. Identify the problem
2. Select candidate statement
3. Get statement details
4. Execute tuning task
5. View report results
6. Apply recommendations

Real World Example | SQL Tuning Advisor in Action

1. Identify problem to be solved



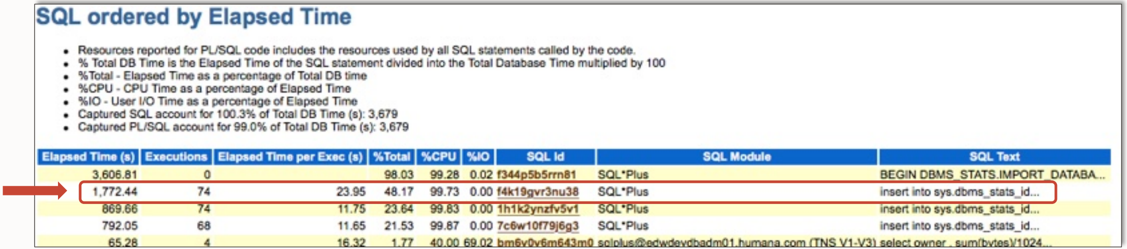
We should be trying to understand why the export of statistics to the stats table took 1.5 hrs, but the import took 48 hours to complete. Instead, it appears that the SR engineer wants to create some type of work-around situation.

I would think we would want to trace the import stats table process to determine why it is taking so long. The stats table that was created only contains 2.8 GBs of data, which should be able to be loaded in mins....

Your help to keep this on track is appreciated...

Real World Example | SQL Tuning Advisor in Action

2. Select candidate statement to tune
   - Generate an AWR Report



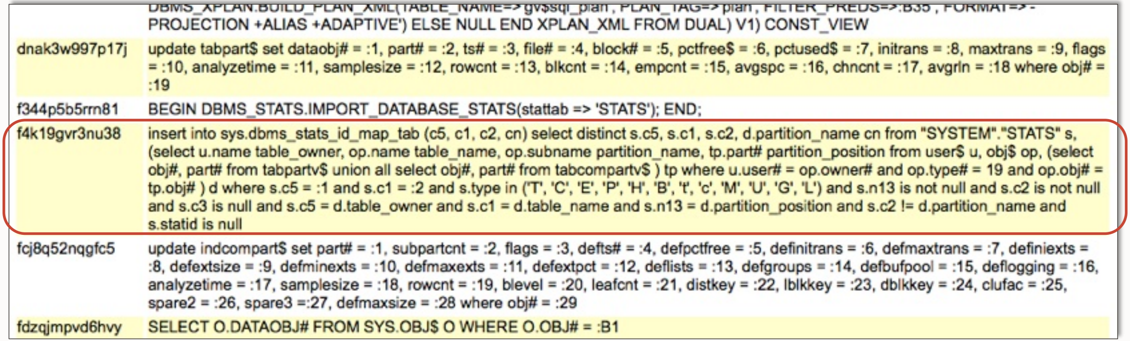
SQL ordered by Elapsed Time

- Resources reported for PL/SQL code includes the resources used by all SQL statements called by the code.
- % Total DB Time is the Elapsed Time of the SQL statement divided into the Total Database Time multiplied by 100
- %Total - Elapsed Time as a percentage of Total DB time
- %CPU - CPU Time as a percentage of Elapsed Time
- %IO - User I/O Time as a percentage of Elapsed Time
- Captured SQL account for 100.3% of Total DB Time (s): 3,679
- Captured PL/SQL account for 99.0% of Total DB Time (s): 3,679

Elapsed Time (s)	Executions	Elapsed Time per Exec (s)	%Total	%CPU	%IO	SQL Id	SQL Module	SQL Text
3,606.81	0		98.03	99.28	0.02	f344p5b5rrn81	SQL*Plus	BEGIN DBMS_STATS.IMPORT_DATABA...
1,772.44	74	23.95	48.17	99.73	0.00	f4k19gvr3nu38	SQL*Plus	insert into sys.dbms_stats_id...
869.66	74	11.75	23.64	99.83	0.00	1h1k2ynzfv5v1	SQL*Plus	insert into sys.dbms_stats_id...
792.05	68	11.65	21.53	99.87	0.00	7c6w10f79j6g3	SQL*Plus	insert into sys.dbms_stats_id...
65.28	4	16.32	1.77	40.00	69.02	bm6v0v6m643m0	sqlplus@edwdevdbadm01.humana.com (TNS V1-V3)	select owner , sum(bytes)/1024...

Real World Example | SQL Tuning Advisor in Action

3. Get SQL Statement Details



	DBMS_XPLAN.BUILD_PLAN_XML(TABLE_NAME=>'gv$sql_plan', PLAN_TAG=>'plan', FILTER_PREDS=>:B35 , FORMAT=>'-PROJECTION +ALIAS +ADAPTIVE') ELSE NULL END XPLAN_XML FROM DUAL) V1) CONST_VIEW
dnak3w997p17j	update tabpart$ set dataobj# = :1, part# = :2, ts# = :3, file# = :4, block# = :5, pctfree$ = :6, pctused$ = :7, initrans = :8, maxtrans = :9, flags = :10, analyzetime = :11, samplesize = :12, rowcnt = :13, blkcnt = :14, empcnt = :15, avgspc = :16, chncnt = :17, avgrln = :18 where obj# = :19
f344p5b5rrn81	BEGIN DBMS_STATS.IMPORT_DATABASE_STATS(stattab => 'STATS'); END;
f4k19gvr3nu38	insert into sys.dbms_stats_id_map_tab (c5, c1, c2, cn) select distinct s.c5, s.c1, s.c2, d.partition_name cn from "SYSTEM"."STATS" s, (select u.name table_owner, op.name table_name, op.subname partition_name, tp.part# partition_position from user$ u, obj$ op, (select obj#, part# from tabpartv$ union all select obj#, part# from tabcompartv$) tp where u.user# = op.owner# and op.type# = 19 and op.obj# = tp.obj#) d where s.c5 = :1 and s.c1 = :2 and s.type in ('T', 'C', 'E', 'P', 'H', 'B', 't', 'c', 'M', 'U', 'G', 'L') and s.n13 is not null and s.c2 is not null and s.c3 is null and s.c5 = d.table_owner and s.c1 = d.table_name and s.n13 = d.partition_position and s.c2 != d.partition_name and s.statid is null
fcj8q52nqgfc5	update indcompart$ set part# = :1, subpartcnt = :2, flags = :3, defts# = :4, defpctfree = :5, definitrans = :6, defmaxtrans = :7, definiexts = :8, defextsize = :9, defminexts = :10, defmaxexts = :11, defextpct = :12, deflists = :13, defgroups = :14, defbufpool = :15, deflogging = :16, analyzetime = :17, samplesize = :18, rowcnt = :19, blevel = :20, leafcnt = :21, distkey = :22, lblkkey = :23, dblkkey = :24, clufac = :25, spare2 = :26, spare3 =:27, defmaxsize = :28 where obj# = :29
fdzqjmpvd6hvy	SELECT O.DATAOBJ# FROM SYS.OBJ$ O WHERE O.OBJ# = :B1

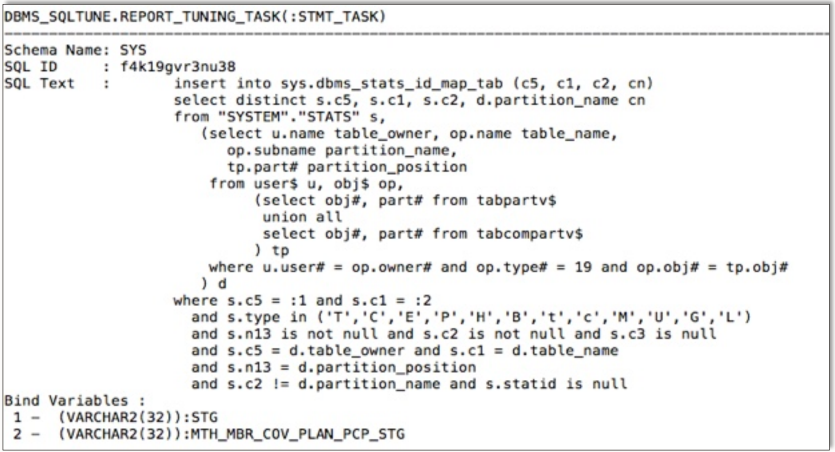
Real World Example | SQL Tuning Advisor in Action

4. Create and run SQL Tuning task

```
VARIABLE stmt_task VARCHAR2(64);

EXEC :stmt_task := DBMS_SQLTUNE.CREATE_TUNING_TASK(sql_id => 'f4k19gvr3nu38');
EXEC DBMS_SQLTUNE.EXECUTE_TUNING_TASK(:stmt_task);

SET LONG 10000
SET LONGCHUNKSIZE 10000
SET LINESIZE 100

SPOOL sql_tune_f4k19gvr3nu38.txt

SELECT dbms_sqltune.report_tuning_task( :stmt_task )FROM DUAL;

SPOOL OFF;
```

Real World Example | SQL Tuning Advisor in Action

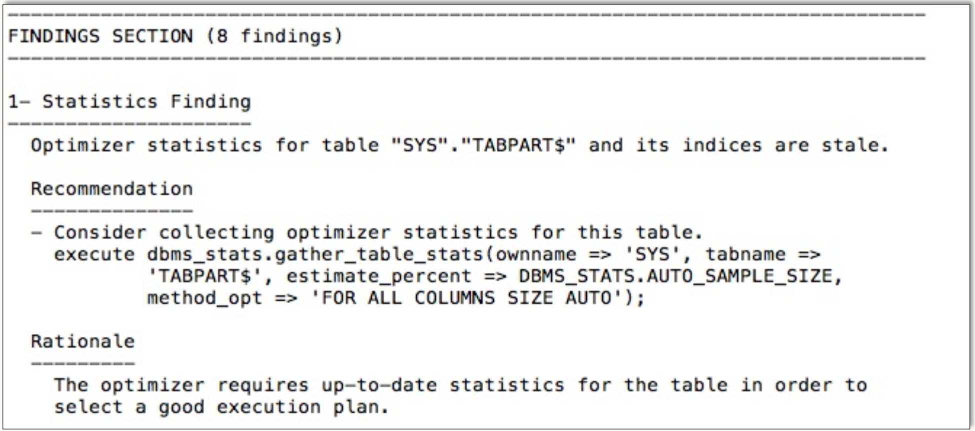
5. View report results



```
DBMS_SQLTUNE.REPORT_TUNING_TASK(:STMT_TASK)
--------------------------------------------------------------------------------------------
Schema Name: SYS
SQL ID     : f4k19gvr3nu38
SQL Text   :       insert into sys.dbms_stats_id_map_tab (c5, c1, c2, cn)
                   select distinct s.c5, s.c1, s.c2, d.partition_name cn
                   from "SYSTEM"."STATS" s,
                      (select u.name table_owner, op.name table_name,
                         op.subname partition_name,
                         tp.part# partition_position
                       from user$ u, obj$ op,
                            (select obj#, part# from tabpartv$
                             union all
                             select obj#, part# from tabcompartv$
                            ) tp
                       where u.user# = op.owner# and op.type# = 19 and op.obj# = tp.obj#
                      ) d
                   where s.c5 = :1 and s.c1 = :2
                     and s.type in ('T','C','E','P','H','B','t','c','M','U','G','L')
                     and s.n13 is not null and s.c2 is not null and s.c3 is null
                     and s.c5 = d.table_owner and s.c1 = d.table_name
                     and s.n13 = d.partition_position
                     and s.c2 != d.partition_name and s.statid is null
Bind Variables :
 1 -  (VARCHAR2(32)):STG
 2 -  (VARCHAR2(32)):MTH_MBR_COV_PLAN_PCP_STG
```

Real World Example | SQL Tuning Advisor in Action

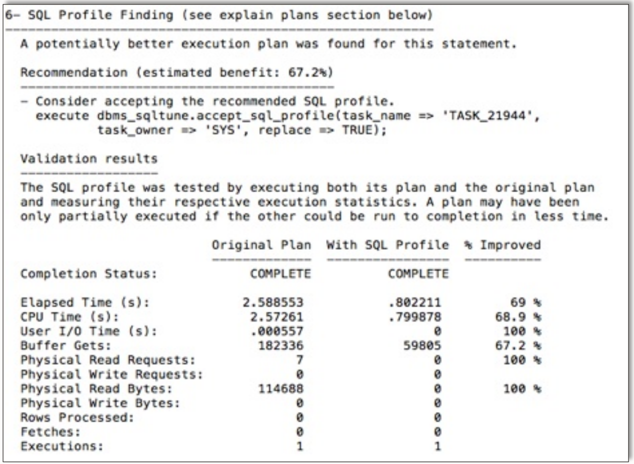
5. View report results



```
-------------------------------------------------------------------------------
FINDINGS SECTION (8 findings)
-------------------------------------------------------------------------------

1- Statistics Finding
---------------------
  Optimizer statistics for table "SYS"."TABPART$" and its indices are stale.

  Recommendation
  --------------
  - Consider collecting optimizer statistics for this table.
    execute dbms_stats.gather_table_stats(ownname => 'SYS', tabname =>
            'TABPART$', estimate_percent => DBMS_STATS.AUTO_SAMPLE_SIZE,
            method_opt => 'FOR ALL COLUMNS SIZE AUTO');

  Rationale
  ---------
    The optimizer requires up-to-date statistics for the table in order to
    select a good execution plan.
```

Real World Example | SQL Tuning Advisor in Action

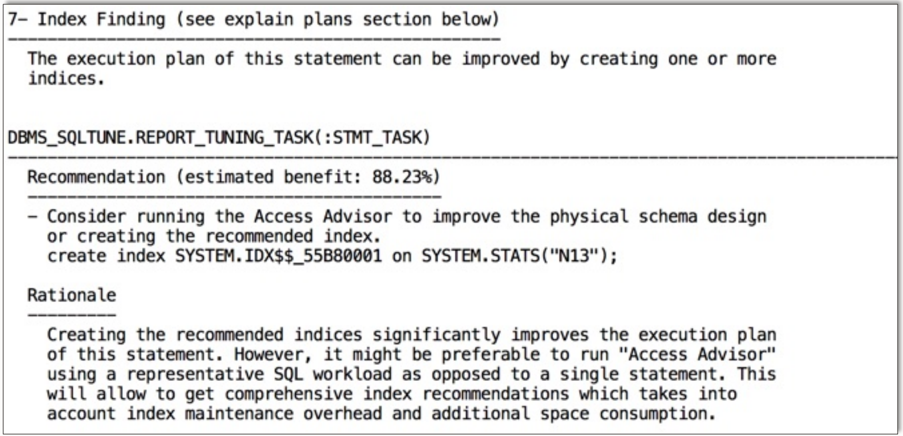
5. View report results



```
6- SQL Profile Finding (see explain plans section below)
--------------------------------------------------------
  A potentially better execution plan was found for this statement.

  Recommendation (estimated benefit: 67.2%)
  -----------------------------------------
  - Consider accepting the recommended SQL profile.
    execute dbms_sqltune.accept_sql_profile(task_name => 'TASK_21944',
            task_owner => 'SYS', replace => TRUE);

  Validation results
  ------------------
  The SQL profile was tested by executing both its plan and the original plan
  and measuring their respective execution statistics. A plan may have been
  only partially executed if the other could be run to completion in less time.

                           Original Plan  With SQL Profile  % Improved
                           -------------  ----------------  ----------
  Completion Status:            COMPLETE          COMPLETE

  Elapsed Time (s):             2.588553           .802211       69 %
  CPU Time (s):                  2.57261           .799878     68.9 %
  User I/O Time (s):             .000557                 0      100 %
  Buffer Gets:                    182336             59805     67.2 %
  Physical Read Requests:              7                 0      100 %
  Physical Write Requests:             0                 0
  Physical Read Bytes:            114688                 0      100 %
  Physical Write Bytes:                0                 0
  Rows Processed:                      0                 0
  Fetches:                             0                 0
  Executions:                          1                 1
```

Real World Example | SQL Tuning Advisor in Action

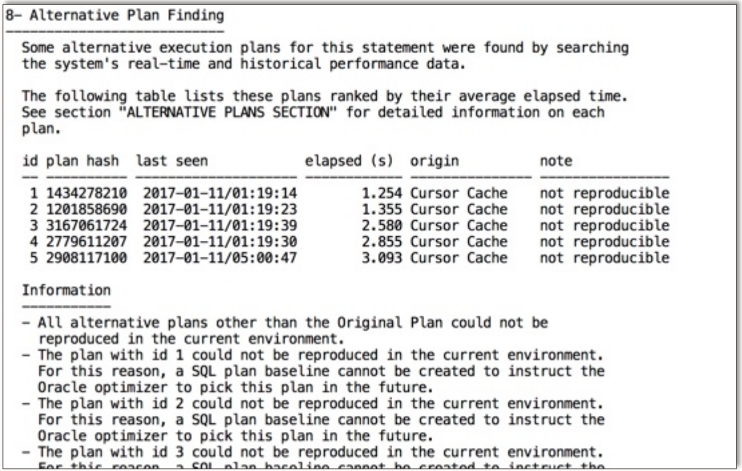
5. View report results



```
7- Index Finding (see explain plans section below)
--------------------------------------------------
  The execution plan of this statement can be improved by creating one or more
  indices.


DBMS_SQLTUNE.REPORT_TUNING_TASK(:STMT_TASK)
-------------------------------------------------------------------------------------------
  Recommendation (estimated benefit: 88.23%)
  ------------------------------------------
  - Consider running the Access Advisor to improve the physical schema design
    or creating the recommended index.
    create index SYSTEM.IDX$$_55B80001 on SYSTEM.STATS("N13");

  Rationale
  ---------
    Creating the recommended indices significantly improves the execution plan
    of this statement. However, it might be preferable to run "Access Advisor"
    using a representative SQL workload as opposed to a single statement. This
    will allow to get comprehensive index recommendations which takes into
    account index maintenance overhead and additional space consumption.
```

Real World Example | SQL Tuning Advisor in Action

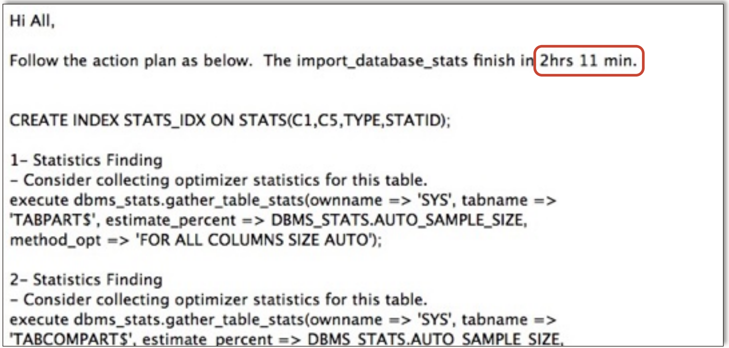
5. View report results



```
8- Alternative Plan Finding
---------------------------
  Some alternative execution plans for this statement were found by searching
  the system's real-time and historical performance data.

  The following table lists these plans ranked by their average elapsed time.
  See section "ALTERNATIVE PLANS SECTION" for detailed information on each
  plan.

  id plan hash  last seen            elapsed (s)  origin          note
  -- ---------- -------------------- ------------ --------------- ----------------
   1 1434278210  2017-01-11/01:19:14        1.254 Cursor Cache    not reproducible
   2 1201858690  2017-01-11/01:19:23        1.355 Cursor Cache    not reproducible
   3 3167061724  2017-01-11/01:19:39        2.580 Cursor Cache    not reproducible
   4 2779611207  2017-01-11/01:19:30        2.855 Cursor Cache    not reproducible
   5 2908117100  2017-01-11/05:00:47        3.093 Cursor Cache    not reproducible

  Information
  -----------
  - All alternative plans other than the Original Plan could not be
    reproduced in the current environment.
  - The plan with id 1 could not be reproduced in the current environment.
    For this reason, a SQL plan baseline cannot be created to instruct the
    Oracle optimizer to pick this plan in the future.
  - The plan with id 2 could not be reproduced in the current environment.
    For this reason, a SQL plan baseline cannot be created to instruct the
    Oracle optimizer to pick this plan in the future.
  - The plan with id 3 could not be reproduced in the current environment.
    For this reason, a SQL plan baseline cannot be created to instruct the
```

Real World Example | SQL Tuning Advisor in Action

6. Act on findings
   - Follow 5 statistics recommendations to gather stats on 5 tables



Hi All,

Follow the action plan as below. The import_database_stats finish in 2hrs 11 min.

```
CREATE INDEX STATS_IDX ON STATS(C1,C5,TYPE,STATID);
```

1- Statistics Finding
- Consider collecting optimizer statistics for this table.

```
execute dbms_stats.gather_table_stats(ownname => 'SYS', tabname =>
'TABPART$', estimate_percent => DBMS_STATS.AUTO_SAMPLE_SIZE,
method_opt => 'FOR ALL COLUMNS SIZE AUTO');
```

2- Statistics Finding
- Consider collecting optimizer statistics for this table.

```
execute dbms_stats.gather_table_stats(ownname => 'SYS', tabname =>
'TABCOMPART$', estimate_percent => DBMS_STATS.AUTO_SAMPLE_SIZE,
```

   - Result: 20x improvement!

Photo by Ussama Azam on Unsplash

SQL Patch

SQL Patch | Overview

Repair SQL statements

- Add hints
- Transparent
- Persistent
- Introduced in Oracle 11g
  - Oracle 11g and 12.1: `DBMS_SQLDIAG_INTERNAL`
  - Oracle 12.2 and newer: `DBMS_SQLDIAG`
- Available in EE and SE2

- Documentation: https://docs.oracle.com/en/database/oracle/oracle-database/19/arpls/DBMS_SQLDIAG.html#GUID-0F29CD05-6BF3-4EEB-90F5-E2465865C255
- Useful scripts, e.g., create_sql_patch.sql: http://kerryosborne.oracle-guy.com/2013/06/06/sql-gone-bad-but-plan-not-changed/

SQL Patch | Version Differences

Oracle 11.2 and 12.1

- `DBMS_SQLDIAG_INTERNAL`
  (undocumented)

```
BEGIN
  SYS.DBMS_SQLDIAG_INTERNAL.i_create_patch(
    sql_text  => 'select * big_table',
    hint_text => 'PARALLEL(big_table,10)',
    name      => 'big_table_sql_patch');
END;
/
```

Oracle 12.2 and newer

- `DBMS_SQLDIAG`

```
DECLARE
  l_patch_name  VARCHAR2(4000);
BEGIN
  l_patch_name :=
 SYS.DBMS_SQLDIAG.create_sql_patch(
    sql_text  => 'select * from big_table',
    hint_text => 'PARALLEL(big_table,10)',
    name      => 'big_table_sql_patch');
END;
/
```

Pro tip: You can use *SQL ID* instead of the full *SQL text*

SQL Patch | Demo

SQL Patch

Copyright © 2021, Oracle and/or its affiliates

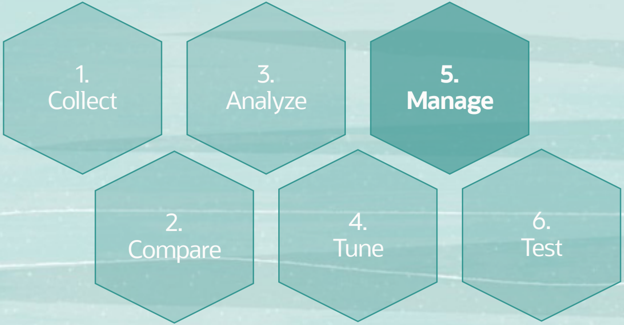
Watch on YouTube

SQL Repair Advisor | Overview

Also check out SQL Repair Advisor

- Available through Cloud Control
- DBMS_SQLDIAG
    - 19c enhancement SQL_DIAGNOSE_AND_REPAIR

Performance Stability Prescription



1. Collect

2. Compare

3. Analyze

4. Tune

5. Manage

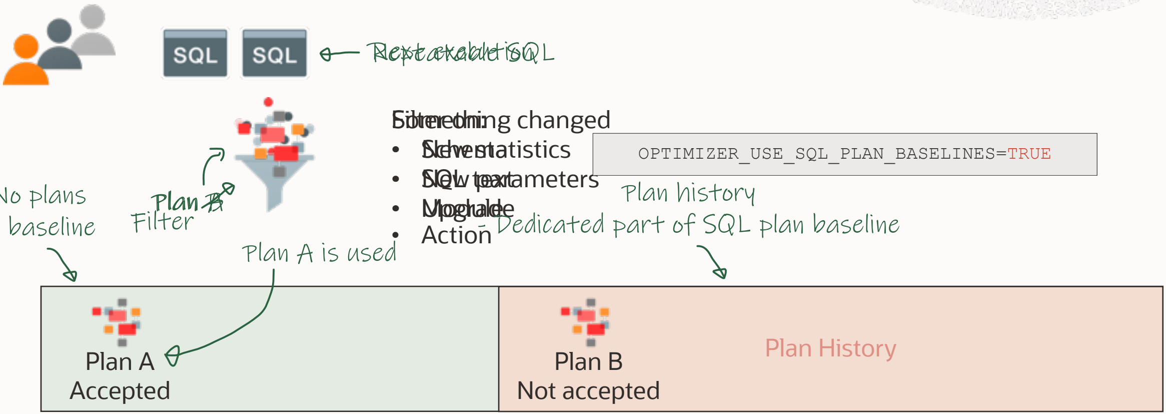
6. Test

SQL Plan Management | SPM

> *SQL plan management uses a mechanism called a* ***SQL plan baseline****, which is a set of accepted plans that the optimizer is allowed to use for a SQL statement.*
>
> *...*
>
> *SQL plan management prevents performance regressions caused by plan changes.*

Database 19c SQL Tuning Guide, chapter 27

SPM | Concept



Something changed

- New statistics
- New parameters
- Upgrade

Filter on:

- Schema
- SQL text
- Module
- Action

```
OPTIMIZER_USE_SQL_PLAN_BASELINES=TRUE
```

Plan A Accepted	Plan B Not accepted Plan History

SPM | Plans

The plans in a SQL plan baseline can be:

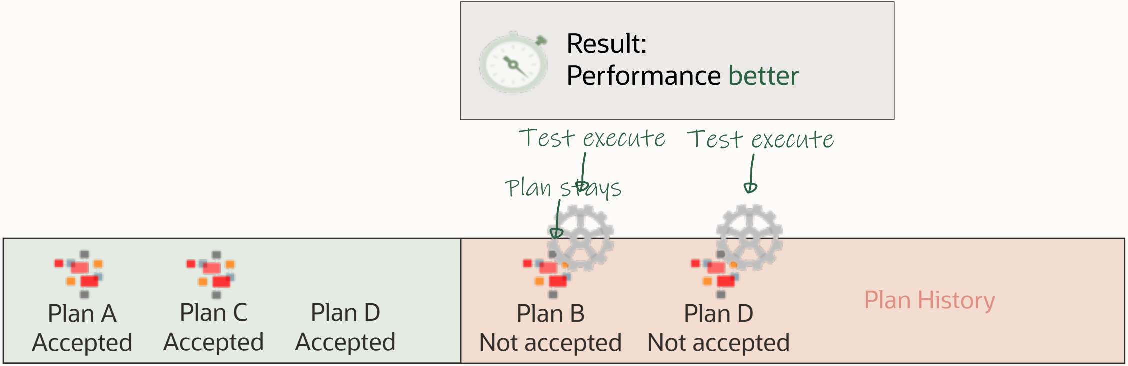
- Enabled
- Accepted
- Fixed

To change status use `DBMS_SPM.ALTER_SQL_PLAN_BASELINE`

You can also prevent plans from getting purged by setting the `autopurge` property.

Pro tip: The *Accepted* attribute can only be set by a test execution

SPM | Evolve



Result:
Performance better

Test execute Test execute

Plan stays

Plan A
Accepted

Plan C
Accepted

Plan D
Accepted

Plan B
Not accepted

Plan D
Not accepted

Plan History

SPM | Evolve

Evolving happens in maintenance task SYS_AUTO_SPM_EVOLVE_TASK

- Part of Automatic SQL Tuning Task

You decide whether recommendations are implemented automatically

```
SQL> BEGIN
   DBMS_SPM.SET_EVOLVE_TASK_PARAMETER(
      parameter => 'accept_plans',
      value     => 'true');
   END;
/
```

You can evolve plans manually

SPM | Management Base

- SQL Management Base is stored in SYSAUX tablespace
- Plans are stored in a LOB
- Unused plans are deleted after 53 weeks
- Space budget is 10 %

SPM | Management Base

Check your settings

```
SQL> select parameter_name, parameter_value from dba_sql_management_config;
```

```
PARAMETER_NAME                     PARAMETER_VALUE
__________________________________ __________________
AUTO_CAPTURE_ACTION
```

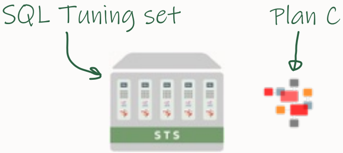
```
SQL> exec DBMS_SPM.CONFIGURE('plan_retention_weeks', 5);
```

```
AUTO_CAPTURE_PARSING_SCHEMA_NAME
AUTO_CAPTURE_SQL_TEXT
AUTO_SPM_EVOLVE_TASK               OFF
AUTO_SPM_EVOLVE_TASK_INTERVAL      3600
```

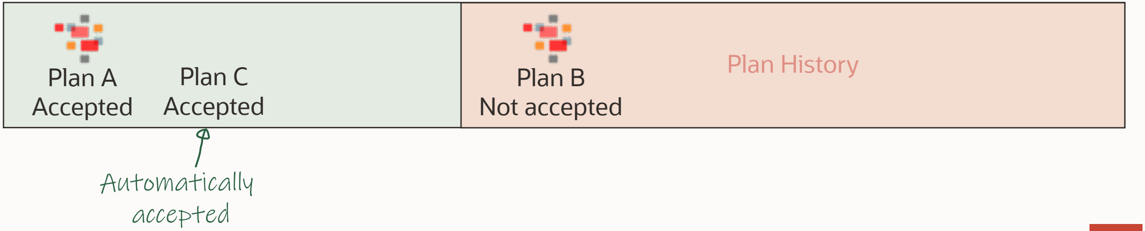
```
SQL> exec DBMS_SPM.CONFIGURE('space_budget_percent', 5);
```

```
PLAN_RETENTION_WEEKS               53
SPACE_BUDGET_PERCENT               10
```

SPM | Load from STS



```
SQL> DECLARE
        cnt number;
   BEGIN
        cnt := DBMS_SPM.LOAD_PLANS_FROM_SQLSET('UPG_STS_1');
   END;
/
```



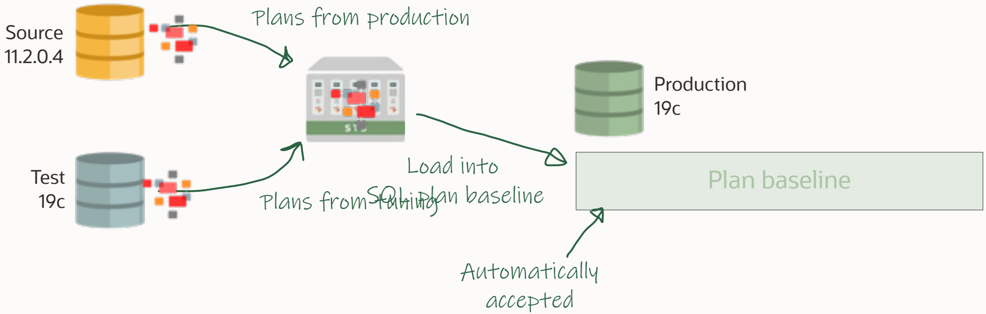
SPM | What if ... literals

SQL Plan Management in a system with literals is not a good fit

- Many distinct statements
- CURSOR_SHARING = FORCE? No!
- SQL profiles can do force matching

Optimal solution: Change your application to use bind variables

SPM | Use Case


Source
11.2.0.4
Plans from production
Test
19c
Plans from tuning
STS
Load into
SQL plan baseline
Production
19c
Plan baseline
Automatically
accepted

SPM | Use Case

STS
Plan baseline

```
SQL> DECLARE
  plans_loaded NUMBER;
  filter VARCHAR2(255);
BEGIN
  filter := 'sql_id=''czzzubf8fjz96'' AND plan_hash_value=''1165613724''';

  plans_loaded := DBMS_SPM.LOAD_PLANS_FROM_SQLSET (
    sqlset_name  => 'UPG_STS_1',
    basic_filter => filter
  );
END;
/
```

Pro tip: The function `LOAD_PLANS_FROM_SQLSET` can also *fix* the plans

Photo by Julia Joppien on Unsplash

Transporting SQL Plan Baselines

Transporting | SQL Plan Baseline

Prepare
Extract
Transfer
Load

SQL Plan Baselines are stored in data dictionary

To transfer - information must be converted to a transportable format and stored in a staging table

```
SQL> BEGIN
   DBMS_SPM.CREATE_STGTAB_BASELINE (
      table_name   => 'SPB_STAGING',
      table_owner  => 'SPM');
   END;
/
```

Pro tip: Create the staging table in a schema and/or tablespace that you are migrating

Transporting | SQL Plan Baseline

Prepare

Extract

Transfer

Load

Select the baselines that you want to transfer

To extract the fixed and accepted plans

```
SQL> DECLARE
   l_count NUMBER;
BEGIN
   l_count := DBMS_SPM.PACK_STGTAB_BASELINE (
      table_name    => 'SPB_STAGING',
      table_owner   => 'SPM',
      enabled       => 'YES',
      fixed         => 'YES');
   END;
/
```

Pro tip: You can also use `dba_sql_plan_baselines` to find plans

Transporting | SQL Plan Baseline

Prepare

Extract

Transfer

Load

If the staging table is migrated together with the user data, you can skip this step

Use Data Pump to transfer that single table

```
SQL> CREATE DATABASE LINK src_link ... ;

$ impdp system network_link=src_link tables=SPM.SPB_STAGING ...
```

Pro tip: You can also import from dump file if there is no network connectivity to source database

Transporting | SQL Plan Baseline

Prepare

Extract

Transfer

Load

Finally, load the baselines from the staging table into the data dictionary

```
SQL> DECLARE
   l_count NUMBER;
BEGIN
   l_count := DBMS_SPM.UNPACK_STGTAB_BASELINE (
      table_name   => 'SPB_STAGING',
      table_owner  => 'SPM');
   END;
/
```

Pro tip: You can apply filters to limit the baselines to import

Transporting | SQL Plan Baseline

Additional resources

Documentation

Technical Brief, SQL Plan Management in Oracle Database 19c

Database 19c, SQL Tuning Guide, Overview of SQL Plan Management

Transporting | SQL Plan Baseline

You can migrate SQL Plan Baselines before downtime.

SPB are not bound to user objects, so you can move them into an empty database

In the empty database, be sure to stop automatic evolution of SPB. You don't want that happening in an empty database.

Disable auto-task or set:

```
BEGIN
   DBMS_SPM.SET_EVOLVE_TASK_PARAMETER(
      task_name => 'SYS_AUTO_SPM_EVOLVE_TASK' ,
      parameter => 'ACCEPT_PLANS',
      value => 'FALSE');
END;
/
```

Transporting | SQL Plan Baseline

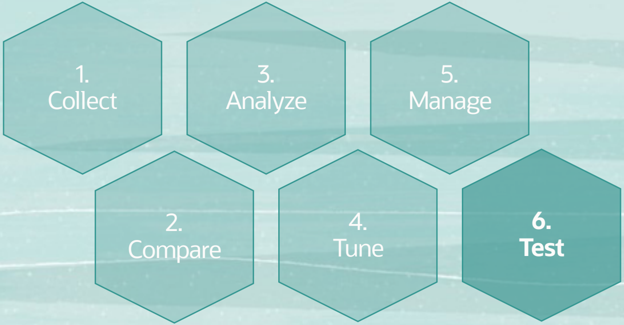
Use Data Pump object path = SMB

```
Note this exclude the following as well:
SQL Profiles
SQL Patches
SQL Plan Directives
```

Different PLAN STABILITY options

	SQL Plan Management	SQL Profile	SQL Patch
Edition	EE (subset in SE2)	EE + Tuning	All
Method	Restrict plan usage	Improves cardinality estimates	Applies hints
Stores	Entire plan	Statistics / hints	Hints
Transportable	Yes	Yes	Yes
Plan guarantee	Yes	-	-
Maintenance	Automatic evolve	Manual inspection	Manual inspection

Performance Stability Prescription



1. Collect

2. Compare

3. Analyze

4. Tune

5. Manage

6. Test

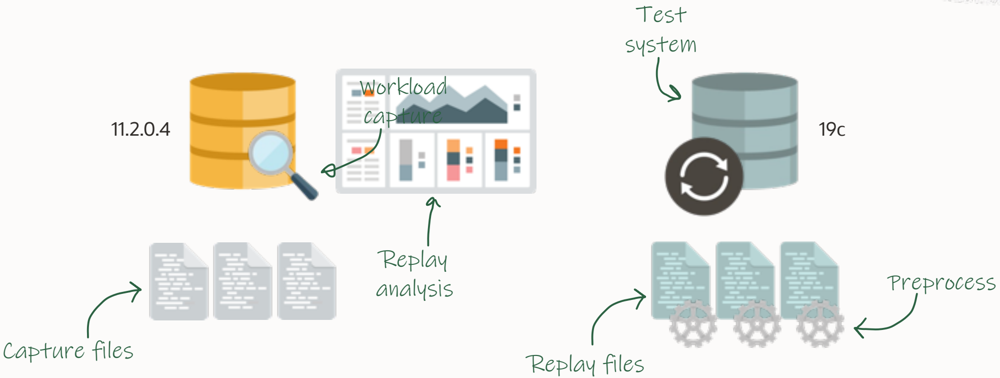
Database Replay | Overview

> *You can use Database Replay to capture a workload on the production system and replay it on a test system with the exact timing, concurrency, and transaction characteristics of the original workload.*
>
> *This enables you to test the effects of a system change without affecting the production system.*

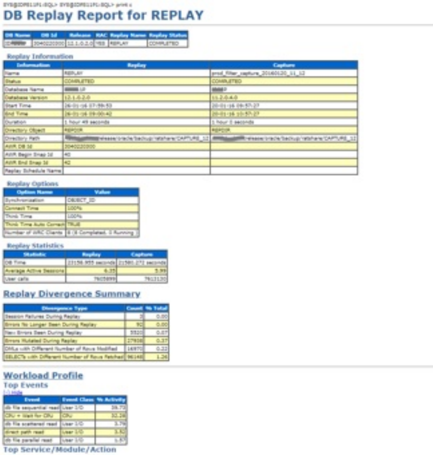
Database 19c Testing Guide, chapter 9

- Requires Enterprise Edition + Real Application Testing
- Available in Autonomous Database

Database Replay | Overview



Database Replay | Overview



SYS@IDRE11P1:SQL> SYS@IDRE11P1:SQL> print c

DB Replay Report for REPLAY

DB Name	DB Id	Release	RAC	Replay Name	Replay Status
ID[redacted]	2040220300	12.1.0.2.0	YES	REPLAY	COMPLETED

Replay Information

Information	Replay	Capture
Name	REPLAY	prod_filter_capture_20160120_11_12
Status	COMPLETED	COMPLETED
Database Name	[redacted] 1P	[redacted]P
Database Version	12.1.0.2.0	11.2.0.4.0
Start Time	26-01-16 07:59:53	20-01-16 09:57:27
End Time	26-01-16 09:00:42	20-01-16 10:57:27
Duration	1 hour 49 seconds	1 hour 0 seconds
Directory Object	REPDIR	REPDIR
Directory Path	[redacted]release/oracle/backup/ratshare/CAPTURE_12	[redacted]release/oracle/backup/ratshare/CAPTURE_12
AWR DB Id	2040220300	
AWR Begin Snap Id	40	
AWR End Snap Id	42	
Replay Schedule Name		

Replay Options

Option Name	Value
Synchronization	OBJECT_ID
Connect Time	100%
Think Time	100%
Think Time Auto Correct	TRUE
Number of WRC Clients	8 (8 Completed, 0 Running)

Replay Statistics

Statistic	Replay	Capture
DB Time	22156.955 seconds	21580.272 seconds
Average Active Sessions	6.25	5.99
User calls	7605899	7613120

Replay Divergence Summary

Divergence Type	Count	% Total
Session Failures During Replay	0	0.00
Errors No Longer Seen During Replay	90	0.00
New Errors Seen During Replay	5520	0.07
Errors Mutated During Replay	27938	0.37
DMLs with Different Number of Rows Modified	16970	0.22
SELECTs with Different Number of Rows Fetched	96148	1.26

Workload Profile

Top Events

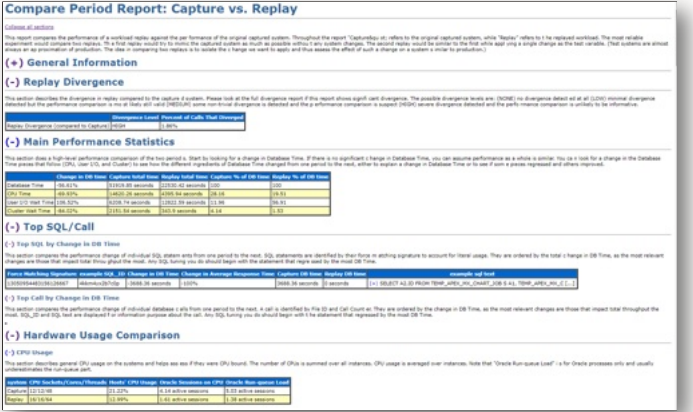
(-) Hide

Event	Event Class	% Activity
db file sequential read	User I/O	38.73
CPU + Wait for CPU	CPU	32.28
db file scattered read	User I/O	3.79
direct path read	User I/O	3.52
db file parallel read	User I/O	1.57

Top Service/Module/Action

Ulrike Schwinn on blogs.oracle.com

Database Replay | Overview



Compare Period Report: Capture vs. Replay

Collapse all sections

This report compares the performance of a workload replay against the performance of the original captured system. Throughout the report "Capture&quot;" refers to the original captured system, while "Replay" refers to the replayed workload. The most reliable experiment would compare two replays. The first replay would try to mimic the captured system as much as possible without any system changes. The second replay would be similar to the first while applying a single change as the test variable. (Test systems are almost always an approximation of production. The idea in comparing two replays is to isolate the change we want to apply and thus assess the effect of such a change on a system similar to production.)

(+) General Information

(-) Replay Divergence

This section describes the divergence in replay compared to the captured system. Please look at the full divergence report if this report shows significant divergence. The possible divergence levels are: (NONE) no divergence detected at all (LOW) minimal divergence detected but the performance comparison is most likely still valid (MEDIUM) some non-trivial divergence is detected and the performance comparison is suspect (HIGH) severe divergence detected and the performance comparison is unlikely to be informative.

	Divergence Level	Percent of Calls That Diverged
Replay Divergence (compared to Capture)	HIGH	1.86%

(-) Main Performance Statistics

This section does a high-level performance comparison of the two periods. Start by looking for a change in Database Time. If there is no significant change in Database Time, you can assume performance as a whole is similar. You can look for a change in the Database Time pieces that follow (CPU, User I/O, and Cluster) to see how the different ingredients of Database Time changed from one period to the next, either to explain a change in Database Time or to see if some pieces regressed and others improved.

	Change in DB time	Capture total time	Replay total time	Capture % of DB time	Replay % of DB time
Database Time	-56.61%	51919.85 seconds	22530.42 seconds	100	100
CPU Time	-69.93%	14620.26 seconds	4395.94 seconds	28.16	19.51
User I/O Wait Time	106.52%	6208.74 seconds	12822.59 seconds	11.96	56.91
Cluster Wait Time	-84.02%	2151.54 seconds	343.9 seconds	4.14	1.53

(-) Top SQL/Call

(-) Top SQL by Change in DB Time

This section compares the performance change of individual SQL statements from one period to the next. SQL statements are identified by their force matching signature to account for literal usage. They are ordered by the total change in DB Time, as the most relevant changes are those that impact total throughput the most. Any SQL tuning you do should begin with the statement that regressed by the most DB Time.

Force Matching Signature	example SQL_ID	Change in DB Time	Change in Average Response Time	Capture DB time	Replay DB time	example sql text
13050954483156126667	4l4km4ux2b7c0p	-3688.36 seconds	-100%	3688.36 seconds	0 seconds	[+] SELECT A2.ID FROM TEMP_APEX_MX_CHART_JOB S A1, TEMP_APEX_MX_C [...]

(-) Top Call by Change in DB Time

This section compares the performance change of individual database calls from one period to the next. A call is identified by File ID and Call Counter. They are ordered by the change in DB Time, as the most relevant changes are those that impact total throughput the most. SQL_ID and SQL text are displayed for information purpose about the call. Any SQL tuning you do should begin with the statement that regressed by the most DB Time.

(-) Hardware Usage Comparison

(-) CPU Usage

This section describes general CPU usage on the systems and helps assess if they were CPU bound. The number of CPUs is summed over all instances. CPU usage is averaged over instances. Note that "Oracle Run-queue Load" is for Oracle processes only and usually underestimates the run-queue part.

system	CPU Sockets/Cores/Threads	Hosts' CPU Usage	Oracle Sessions on CPU	Oracle Run-queue Load
Capture	12/12/48	21.22%	4.14 active sessions	5.03 active sessions
Replay	16/16/64	12.99%	1.41 active sessions	1.38 active sessions

Ulrike Schwinn on blogs.oracle.com

Database Replay | Facts

1. Platform independent
2. RAC compliant - optionally, change number of nodes
3. Per-PDB capture/replay NEW IN 19c
4. Capture and replay across database releases

Database Replay | To Consider

1. Workload capture restrictions

2. Not suitable with external dependencies
   - Database link, external tables, UTL_HTTP

3. Recommended to restart database before capture
   - Startup in restricted mode, capture automatically sets unrestricted mode

4. Work best from dedicated SCN
   - Data Pump `FLASHBACK_SCN` or restore to specific SCN

Database Replay | Info

- Start out with a small capture, then go full-scale
  - This allows you to iron-out issues and prepare for the full-scale capture
- Capture is a light-weight tracing, typically adds 4-5 % overhad
- Capture files are written to disk, disks must be fast
  - Otherwise it will affect the database
- Real-life example: 24h capture produced 4 TB of replay files
- Replay happens on a cold system
  - Use a "Replay Query Only" run to warm-up the system

Performance Stability Prescription

1. Collect
2. Compare
3. Analyze
4. Tune
5. Manage
6. Test

Secrets, Surprises, Underscores

Photo by Jan Szwagrzyk on Unsplash

Surprise | Automatic SQL Plan Management

Enabled by default in Oracle 19.3.0

Disabled by default since Oracle 19.4.0, but only on non-Exadata systems

- Scans AWR
- Verifies and enables SQL Plan Baselines without DBA intervention
- Exadata-only feature

ENABLED

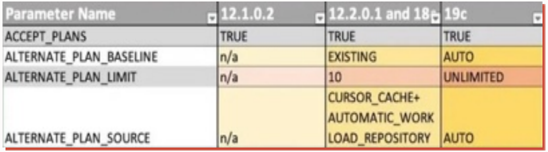
```
BEGIN
 DBMS_SPM.SET_EVOLVE_TASK_PARAMETER(
   task_name => 'SYS_AUTO_SPM_EVOLVE_TASK',
   parameter => 'ALTERNATE_PLAN_BASELINE',
   value     => 'AUTO'
  );
END;
/
```

DISABLED

```
BEGIN
 DBMS_SPM.SET_EVOLVE_TASK_PARAMETER(
   task_name => 'SYS_AUTO_SPM_EVOLVE_TASK',
   parameter => 'ALTERNATE_PLAN_BASELINE',
   value     => 'EXISTING'
  );
END;
/
```

Surprise | Automatic SQL Plan Management

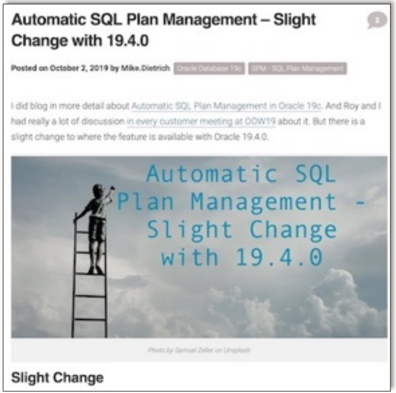
Different settings per release



Parameter Name	12.1.0.2	12.2.0.1 and 18	19c
ACCEPT_PLANS	TRUE	TRUE	TRUE
ALTERNATE_PLAN_BASELINE	n/a	EXISTING	AUTO
ALTERNATE_PLAN_LIMIT	n/a	10	UNLIMITED
ALTERNATE_PLAN_SOURCE	n/a	CURSOR_CACHE+AUTOMATIC_WORKLOAD_REPOSITORY	AUTO

Revert to previous behavior

- Blog Post: Automatic SPM in Oracle 19c
- Blog Post: Slight change with Auto SPM in 19.4.0



Automatic SQL Plan Management – Slight Change with 19.4.0

Posted on October 2, 2019 by Mike.Dietrich Oracle Database 19c SPM - SQL Plan Management

I did blog in more detail about Automatic SQL Plan Management in Oracle 19c. And Roy and I had really a lot of discussion in every customer meeting at OOW19 about it. But there is a slight change to where the feature is available with Oracle 19.4.0.

Photo by Samuel Zeller on Unsplash

Slight Change

Surprise | Automatic SQL Tuning Set

In Oracle 19.7.0, an Automatic SQL Tuning Set gets populated

- Some customers reported high growth and consumption in SYSAUX
- Disabled since 19.8.0

```
select to_char(max(last_schedule_time),'DD-MON-YY hh24:mi') LATEST, task_name, enabled
from dba_autotask_schedule_control group by task_name, status, enabled

LATEST             TASK_NAME                        ENABLED
------------------ -------------------------------- -------
15-JUL-20 09:56    Auto STS Capture Task            TRUE
15-APR-20 00:16    Auto SPM Task                     FALSE
```

- Blog Post: Do you love unexpected surprises?
- Disable it manually:

```
exec DBMS_AUTO_TASK_ADMIN.DISABLE(client_name=>'Auto STS Capture Task',
operation=>NULL, window_name=>NULL);
```

Underscores | `optimizer_adaptive_*`

Parameter `optimizer_adaptive_plans`

- Default: `TRUE`
- Adjust join methods, bitmap pruning and parallel distribution methods during runtime after parsing

Parameter `optimizer_adaptive_statistics`

- Default: `FALSE`
- Create dynamic statistics, SQL Plan Directives and do automatic reoptimization

Recommendation

- Leave the defaults
- For Oracle 12.2.0.1 and newer

Underscores | `_sql_plan_directive_mgmt_control`

Parameter `_sql_plan_directive_mgmt_control`

- SQL Plan Directives get collected in the background
- Even when `optimizer_adaptive_statistics=false` (default)
- But SPDs won't be used
- MOS Note: 2209560.1 - How To Disable SQL Plan Directive (SPD)

Recommendation

- Set `_sql_plan_directive_mgmt_control=0` always everywhere
- For Oracle 12.2.0.1 and newer

Underscores | `_cursor_obsolete_threshold`

Parameter `_cursor_obsolete_threshold`

- Parent cursors not getting obsoleted
- Thus, the child cursors under the parent are getting extended beyond 1024 (default in 12.1)
- Massive concurrency issues with cursor mutexes
- MOS Note: 2431353.1
  High Version Counts For SQL (>1024) Post Upgrade To 12.2 and Above Causing Slow Performance

Recommendation

- Set `_cursor_obsolete_threshold=1024` always everywhere
- For Oracle 12.2.0.1 and newer

Underscores | optimizer_real_time_statistics

Parameter `optimizer_real_time_statistics`

- Real time and high frequency statistics gathering on DML operations
- Exadata-only feature
- ON by default until 19.9.0
- OFF by default since 19.10.0
- Documentation, Optimizer Blog and Upgrade Blog

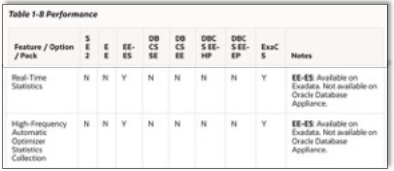


Table 1-8 Performance

Feature / Option / Pack	SE2	EE	EE-ES	DBCS SE	DBCS EE	DBCS EE-HP	DBCS EE-EP	ExaCS	Notes
Real-Time Statistics	N	N	Y	N	N	N	N	Y	**EE-ES**: Available on Exadata. Not available on Oracle Database Appliance.
High-Frequency Automatic Optimizer Statistics Collection	N	N	Y	N	N	N	N	Y	**EE-ES**: Available on Exadata. Not available on Oracle Database Appliance.

Recommendation

- Until 19.9.0
  - `_optimizer_gather_stats_on_conventional_dml=FALSE`
  - `_optimizer_use_stats_on_conventional_dml=FALSE`
- From 19.10.0 on: `optimizer_real_time_statistics=FALSE`

Underscores | deferred_segment_creation

Parameter `deferred_segment_creation`

- Table/index created but no row inserted? No space used
- But performance penalty when first row gets inserted
- Only useful in environments where objects get created and dropped massively
- Several corruption bugs, contention issues and more
  - See: MOS Note 1216282.1 - Parameter "DEFERRED_SEGMENT_CREATION"

Recommendation

- Set `deferred_segment_creation=false` unless you really need this feature
- For Oracle 11.2 and newer

Standard Edition SE2

Photo by Call Me Fred on Unsplash

SE2 | Selection of various Tuning Tools

Many work very well with SE2

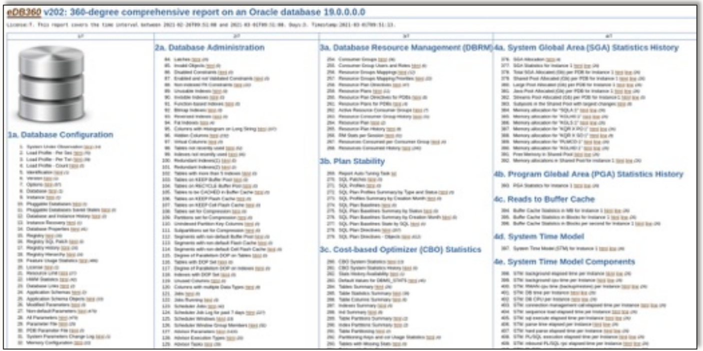
- Presentation about free tuning tools by Björn Rost (2016)

Some very useful examples:

- MOATS (latest version is from: 2020)
- Snapper (2020)
- SQLdb360 (2020)
- TUNAs360 (2016)
- Pathfinder (2016)
- many more

And don't forget:

- Event 10046 – MOS Note: 376442.1



eDB360 v202: 360-degree comprehensive report on an Oracle database 19.0.0.0.0

1a. Database Configuration

2a. Database Administration

3a. Database Resource Management (DBRM)

3b. Plan Stability

3c. Cost-based Optimizer (CBO) Statistics

4a. System Global Area (SGA) Statistics History

4b. Program Global Area (PGA) Statistics History

4c. Reads to Buffer Cache

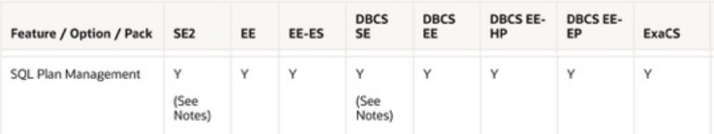
4d. System Time Model

4e. System Time Model Components

SE2 | SQL Plan Management

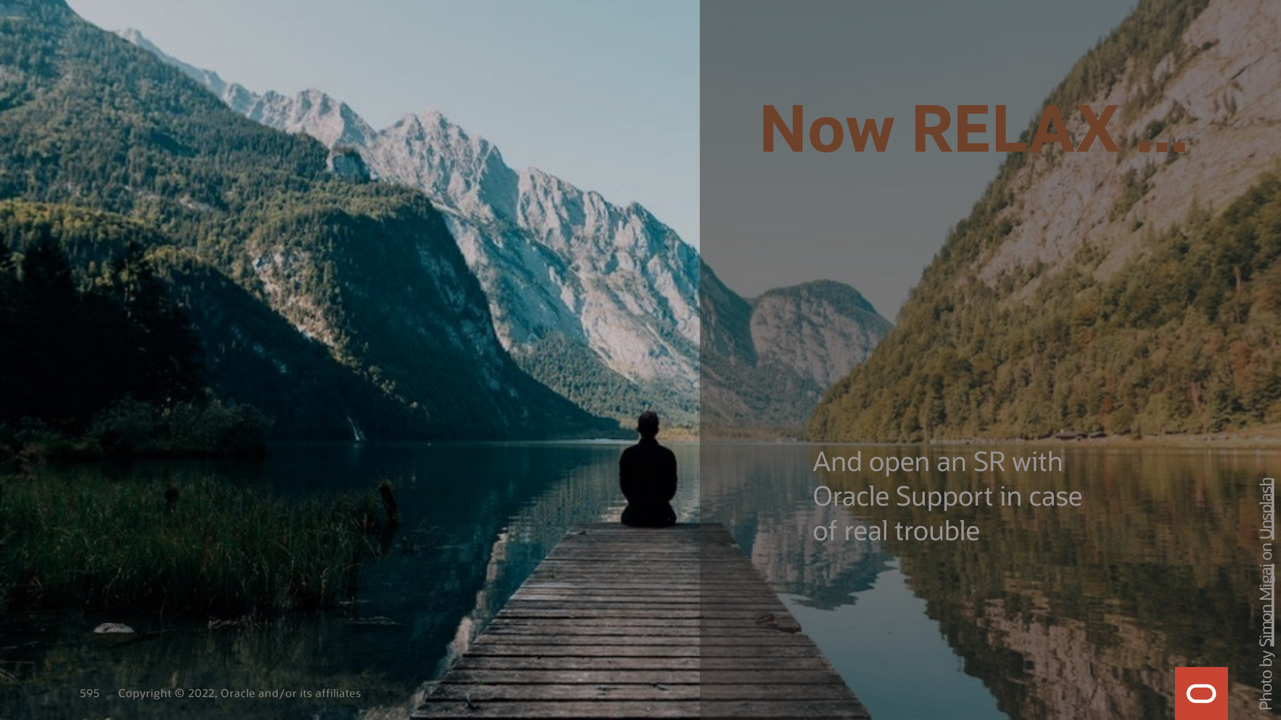
Parts of SQL Plan Management can be used without Diag/Tuning Pack

- Oracle 19c License Guide



Feature / Option / Pack	SE2	EE	EE-ES	DBCS SE	DBCS EE	DBCS EE-HP	DBCS EE-EP	ExaCS
SQL Plan Management	Y (See Notes)	Y	Y	Y (See Notes)	Y	Y	Y	Y

  - Only one SQL plan baseline per SQL statement is allowed and SQL plan evolution is disabled
  - Excerpt from the Oracle 19c License Guide – please check for full and current details



Now RELAX ...

And open an SR with Oracle Support in case of real trouble

Photo by Simon Migaj on Unsplash

Photo by Dušan veverkolog on Unsplash

Want to Know More?

Webinar: Performance Stability

Recording

Slides

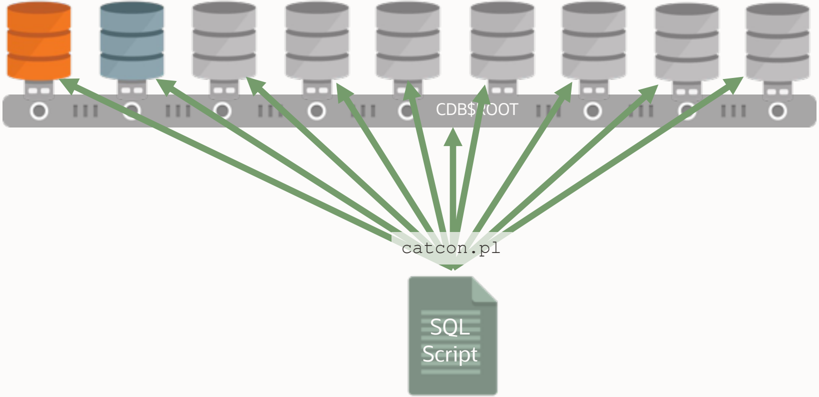
Chapter 5
Multitenant Migration

Photo by Krzysztof Kotkowicz on Unsplash

Multitenant | Script Execution

catcon.pl

- MOS Note: 1932340.1 - How to execute sql scripts in Multitenant environment (catcon.pl)



and finally

Multitenant Upgrades

when you adopted the CDB architecture

Everything at Once

Upgrade the entire CDB with all PDBs

Unplug / Plug / Upgrade

Upgrade one or multiple PDBs in a higher version, new CDB

More Options

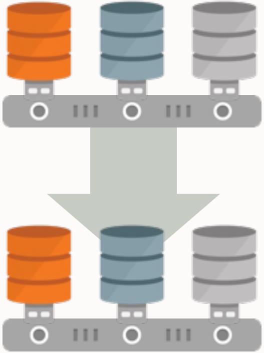
Refreshable Clones as an efficient way to test and perform upgrades

More Power

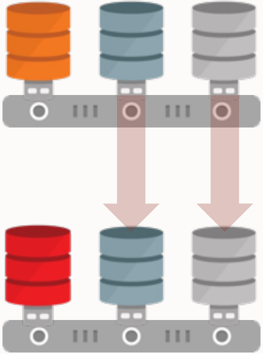
Speed Up "Everything at Once" CDB upgrades

CDB Upgrades | Options

Everything at Once

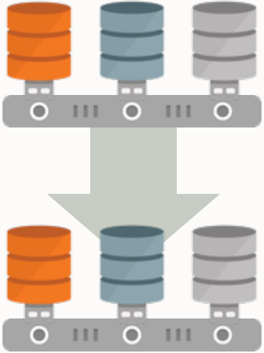


Unplug / Plug / Upgrade

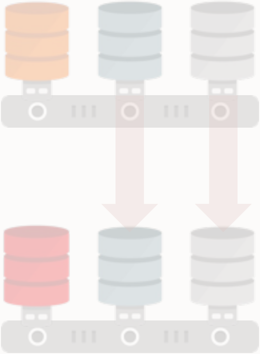


CDB Upgrades | Option 1

Everything at Once



Unplug / Plug / Upgrade



Parallel Upgrade | Container Database

A number of processors are assigned

- Minimum 4
- Maximum unlimited
- Default CPU count

```
$ dbupgrade -n 4
```

Parallel Upgrade | Container Database

Each PDB gets a number of parallel processes

- Minimum 1
- Maximum 8
- Default 2

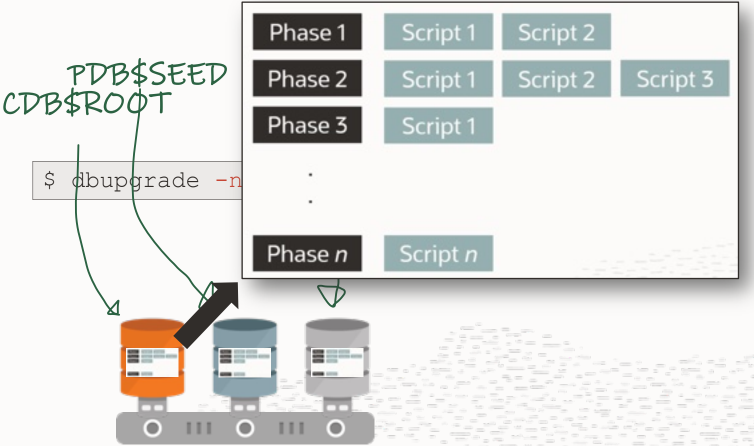
```
$ dbupgrade -N 2
```

Parallel Upgrade | Container Database

But - there is another limit

$$\frac{\text{Total number of processors (n)}}{\text{Processor per PDB (N)}} = \text{PDBs upgraded simultaneously}$$

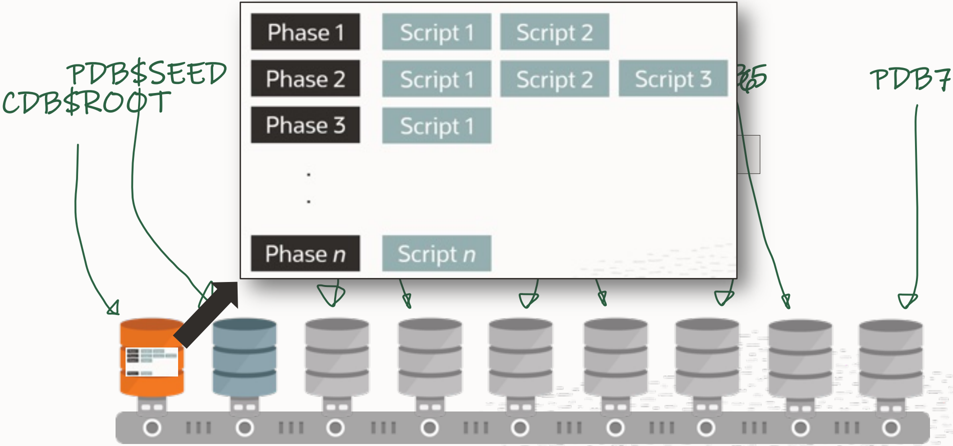
Parallel Upgrade | Single Tenant



```
$ dbupgrade -n
```

Phase 1	Script 1	Script 2	
Phase 2	Script 1	Script 2	Script 3
Phase 3	Script 1		
.			
.			
Phase *n*	Script *n*		

Parallel Upgrade | Multitenant



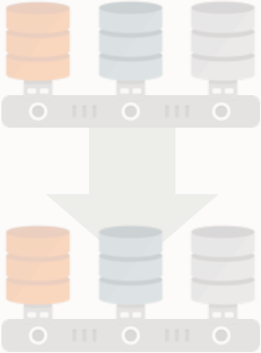
Phase 1	Script 1	Script 2	
Phase 2	Script 1	Script 2	Script 3
Phase 3	Script 1		
.			
.			
Phase *n*	Script *n*		

Parallel Upgrade | Multitenant

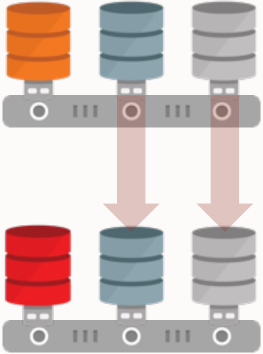
Scale by upgrading more PDBs simultaneously

CDB Upgrades | Option 2

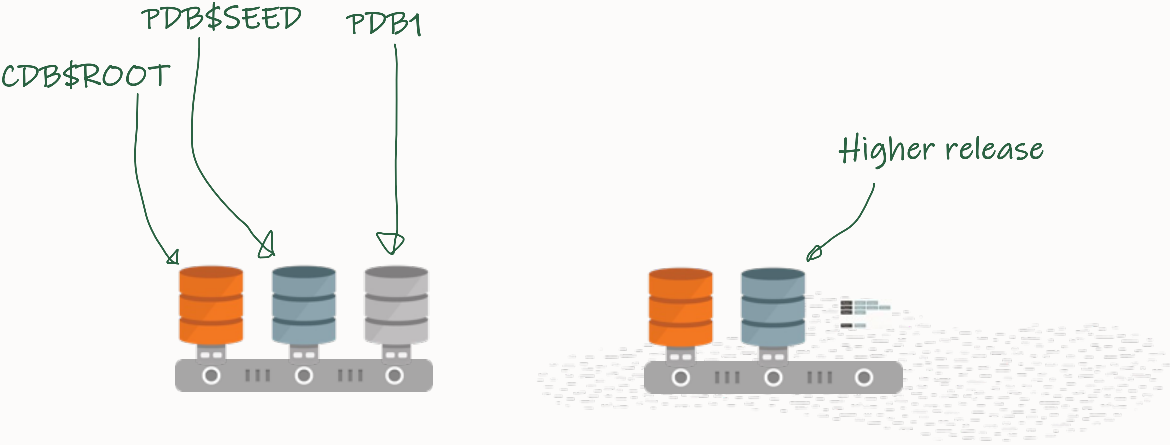
Everything at Once



Unplug / Plug / Upgrade



Parallel Upgrade | Unplug-plug Upgrade



Parallel Upgrade | Unplug-plug

Unplug-plug always faster than

Non-CDB

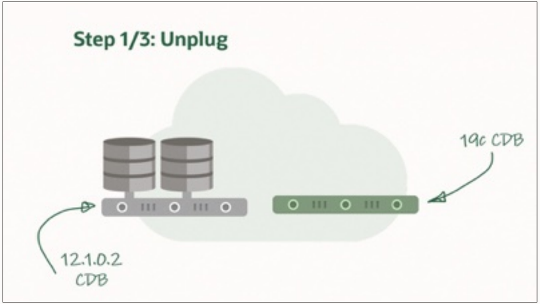
Single Tenant

Multitenant

AutoUpgrade | Unplug-plug Upgrade

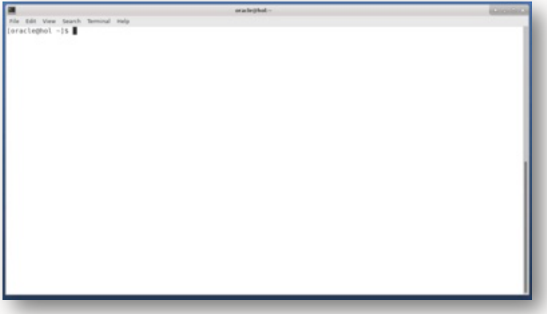
Upgrade a single PDB

- Faster
- More flexible
- Requires compatible target CDB
- Not compatible with Flashback Database
  - Consider using Refreshable PDBs
  - Copy data files (`target_pdb_copy_option`)



AutoUpgrade | Unplug-plug Upgrade

```
upg1.sid=CDB12102
upg1.target_cdb=CDB19
upg1.pdbs=pdb1
upg1.source_home=/u01/app/oracle/product/12102
upg1.target_home=/u01/app/oracle/product/19
```



```
[oracle@hol ~]$
```

Watch on YouTube

AutoUpgrade | Unplug-plug Upgrade

Upgrade several PDBs

```
upg1.pdbs=pdb1,pdb2,pdb3
```

Rename a PDB

```
upg1.pdbs=pdb1
upg1.target_pdb_name.pdb1=sales
```

Copy data files on plug-in

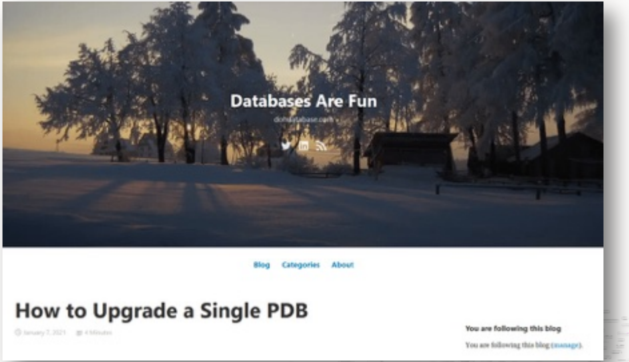
```
upg1.pdbs=pdb1
upg1.target_pdb_copy_option.pdb1=file_name_convert=('pdb1','sales')
```

AutoUpgrade | Unplug-plug Upgrade

Current limitations:

- Does not support Data Guard
- Does not support TDE Tablespace Encryption

AutoUpgrade | Unplug-plug Upgrade



Databases Are Fun

dohdatabase.com

Blog Categories About

How to Upgrade a Single PDB

January 7, 2021 4 Minutes

You are following this blog

You are following this blog (manage).

https://dohdatabase.com/how-to-upgrade-a-single-pdb

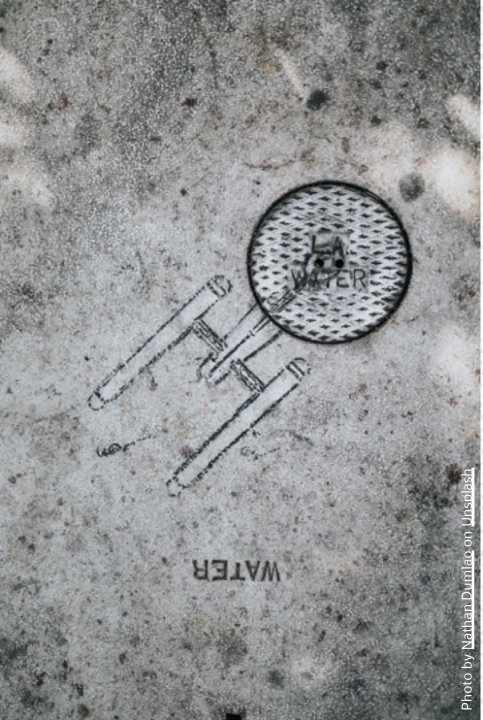


Photo by Nathan Dumlao on Unsplash

I Need More Power

I can't hold her together, Captain!

Faster Upgrades | Statement

During upgrade CPU is a vital resource

Faster Upgrades | Overview

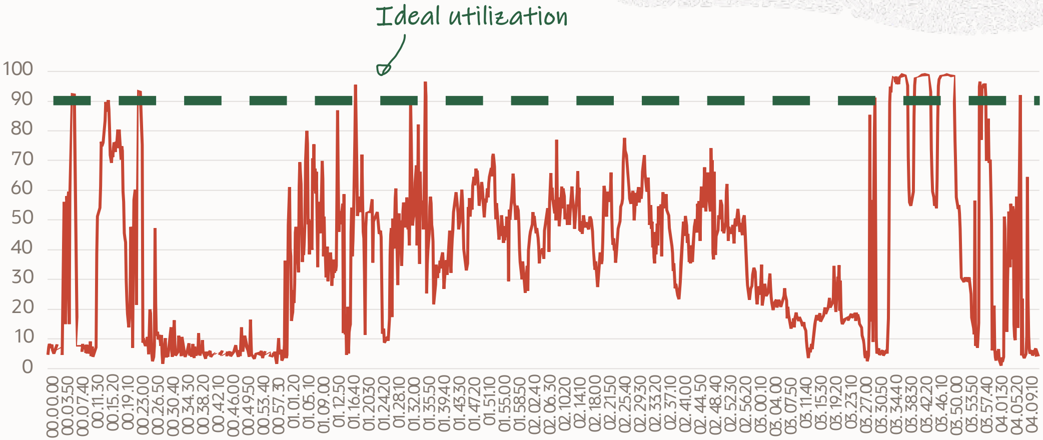
- OCI Bare Metal host
  - 16 OPCUs
  - 768 GB memory
  - NVMe disks

- CDB with 52 PDBs
  - `CPU_COUNT` = 32
  - `SGA_TARGET` = 80G
  - `PGA_AGGREGATE_TARGET` = 20G

- Many database components (17 in total)

- **Upgrade from 12.1.0.2 to 19**

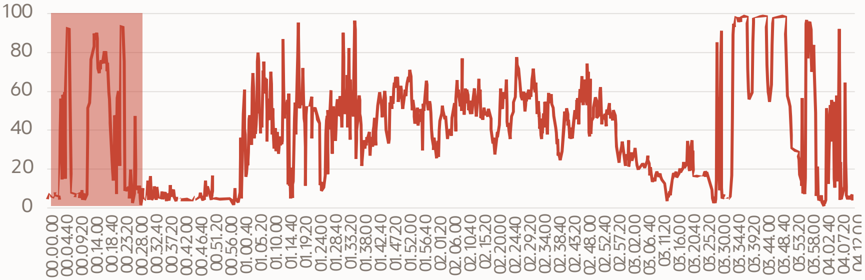
Faster Upgrades | CPU Utilization



Ideal utilization

Total upgrade time: 4 hours 8 minutes

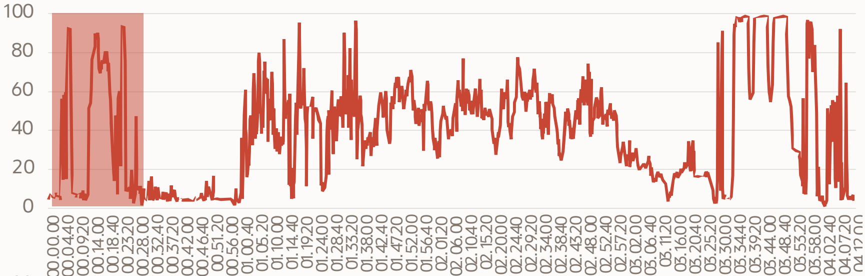
Faster Upgrades | CPU Utilization

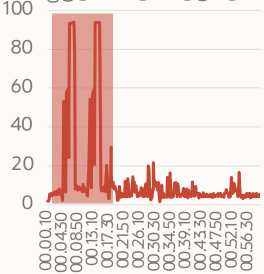


Preupgrade check and fixups

- Gather dictionary and fixed objects stats in advance (7 days)

Faster Upgrades | CPU Utilization

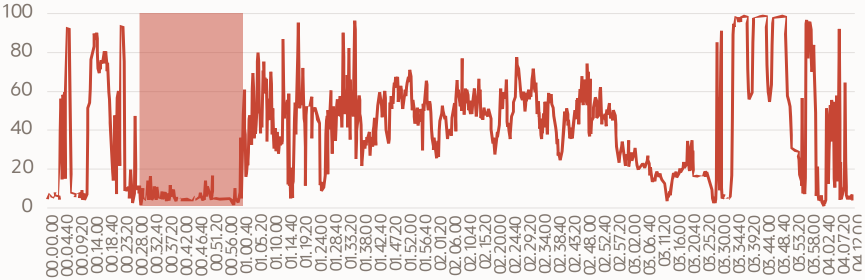




Gathering stats in advance saves 12 minutes

Dictionary and fixed objects

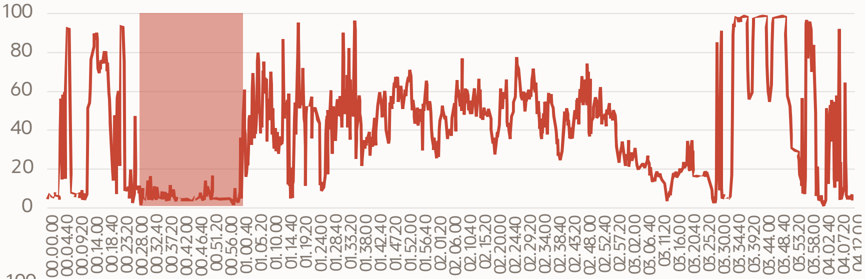
Faster Upgrades | CPU Utilization



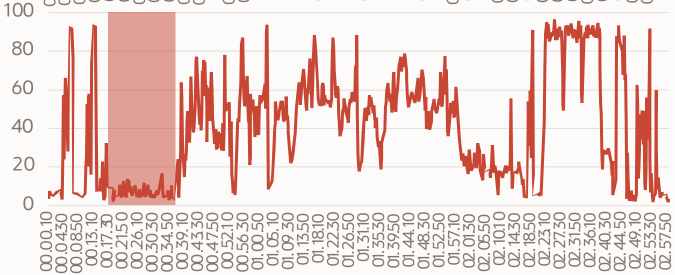
Upgrade CDB$ROOT

- Remove components
- AutoUpgrade automatically assigns 8 parallel processes to CDB$ROOT upgrade

Faster Upgrades | CPU Utilization



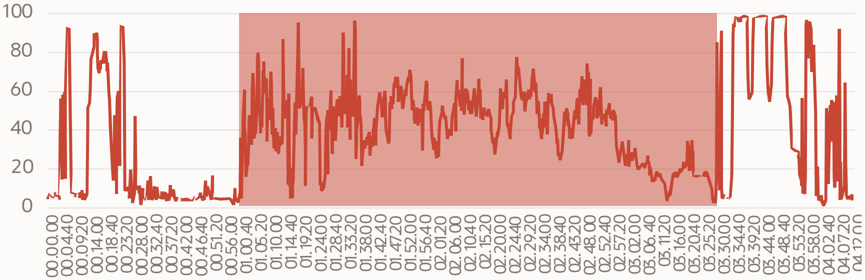
All components installed



Removing all components

13 minutes faster

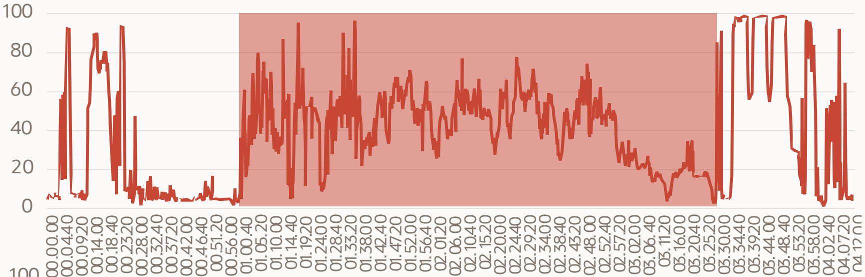
Faster Upgrades | CPU Utilization



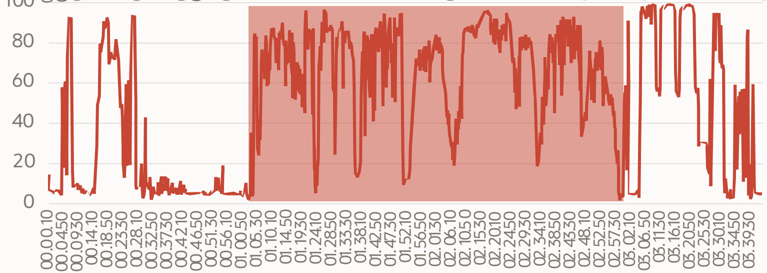
Upgrade PDB$SEED and user PDBs

- Add more PDBs (`catctl -n`)
- Keep parallel processes per PDB at default (2)
- Remove components from PDBs

Faster Upgrades | CPU Utilization



32 parallel processes



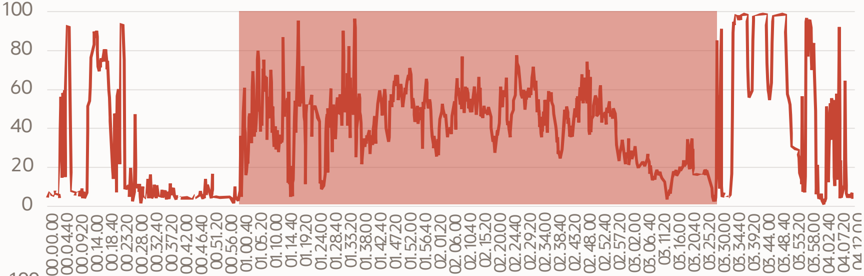
54 parallel processes

`upg1.catctl_options=-n 54`

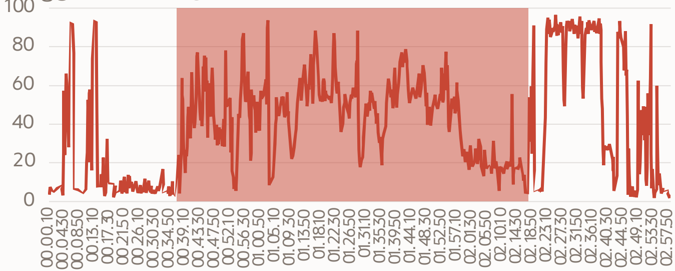
26 minutes faster

Pro tip: Remember to increase `PROCESSES` dramatically

Faster Upgrades | CPU Utilization



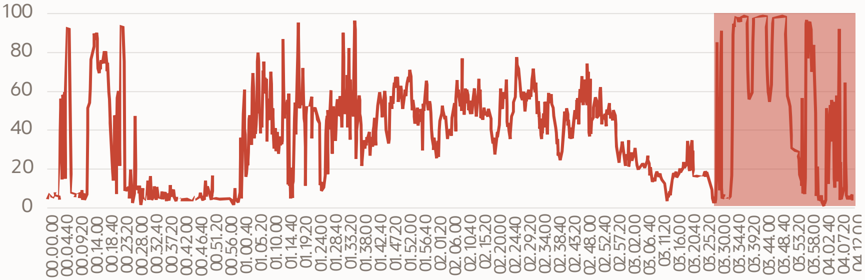
All components installed



Removing all components

48 minutes faster

Faster Upgrades | CPU Utilization

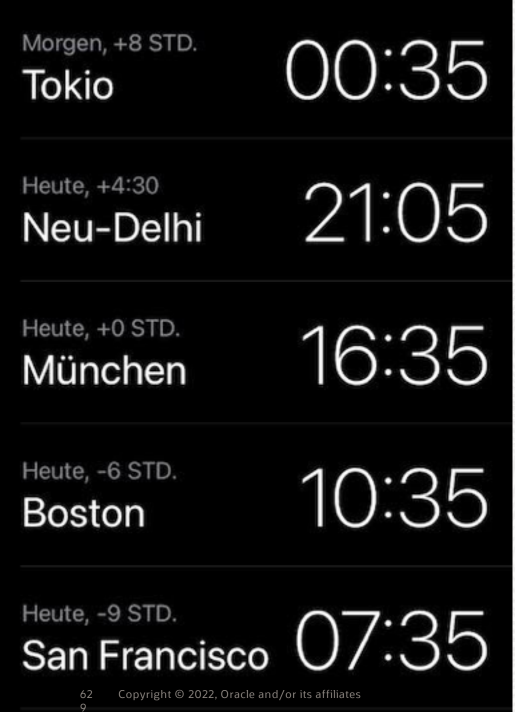


Post upgrade checks and fixups

- Recompilation (`utlrp`) already highly parallelized
- Postpone timezone file upgrade

Faster Upgrades | Conclusion

- Gather stats in advance
- Allow more PDBs per cycle
- Remove components



Morgen, +8 STD.
Tokio
00:35

Heute, +4:30
Neu-Delhi
21:05

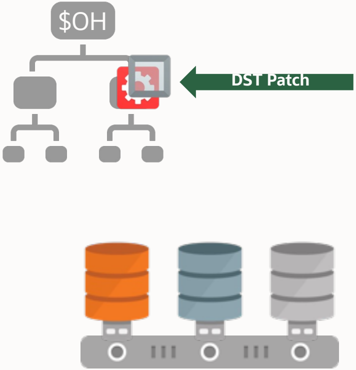
Heute, +0 STD.
München
16:35

Heute, -6 STD.
Boston
10:35

Heute, -9 STD.
San Francisco
07:35

Multitenant and Time Zone Patching

Time Zone | Multitenant DST Version and Patching

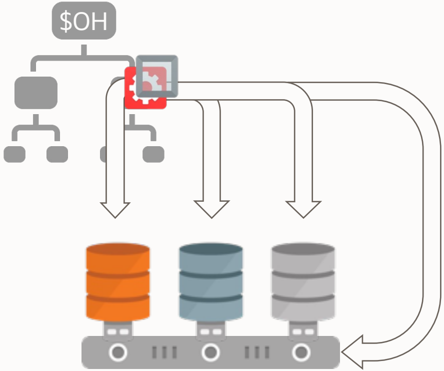


- New 19c CDB gets created with DST V32

Database Release	Default TZ Version
12.1.0.1, 12.1.0.2	DST V18
12.2.0.1	DST V26
18c	DST V31
19c	DST V32
21c	DST V35

- Patching $ORACLE_HOME
  - Containers need to be "TZ upgraded"
  - PDBs and CDB$ROOT can stay on different TZ values

Time Zone | Multitenant Time Zone Upgrade



- Upgrading by default will adjust time zone
  - AutoUpgrade default: `upg1.timezone_upg=yes`
  - 2 restarts will happen
  - Time zone upgrade happens for all containers

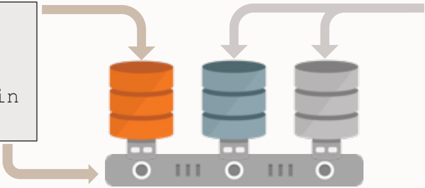
- Manual time zone upgrade is still possible
  - `?/rdbms/admin/utltz_countstar.sql`
  - `?/rdbms/admin/utltz_upg_check.sql`
  - `?/rdbms/admin/utltz_upg_apply.sql`

Time Zone | Updating Time Zone - Check

Download DST patch with: MOS Note:412160.1

Check script:

```
perl catcon.pl -n 1 -s
-l /home/oracle
-b tz_check_ROOT_SEED
-d $ORACLE_HOME/rdbms/admin
utltz_upg_check.sql
```



```
perl catcon.pl -n 1 -S
-l /home/oracle
-b tz_check_PDBs
-d $ORACLE_HOME/rdbms/admin
utltz_upg_check.sql
```

Time Zone | Updating Time Zone - Apply

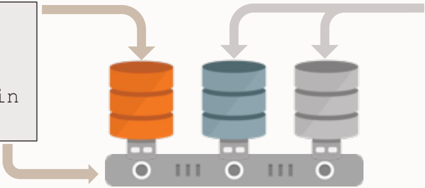
This will restart the database twice, first in UPGRADE mode, then in normal mode

- Exclusive locks may happen

Apply script:

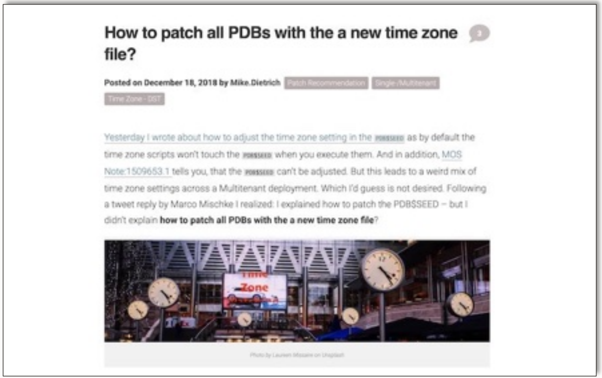
```
SQL> alter pluggable database all open;
```

```
perl catcon.pl -n 1 -s
-l /home/oracle
-b tz_apply_ROOT_SEED
-d $ORACLE_HOME/rdbms/admin
utltz_upg_apply.sql
```



```
perl catcon.pl -n 1 -S
-l /home/oracle
-b tz_apply_PDBs
-d $ORACLE_HOME/rdbms/admin
utltz_upg_apply.sql
```

Time Zone | Summary



How to patch all PDBs with the a new time zone file?

Posted on December 18, 2018 by Mike.Dietrich Patch Recommendation Single-/Multitenant Time Zone - DST

Yesterday I wrote about how to adjust the time zone setting in the PDB$SEED as by default the time zone scripts won't touch the PDB$SEED when you execute them. And in addition, MOS Note:1509653.1 tells you, that the PDB$SEED can't be adjusted. But this leads to a weird mix of time zone settings across a Multitenant deployment. Which I'd guess is not desired. Following a tweet reply by Marco Mischke I realized: I explained how to patch the PDB$SEED – but I didn't explain **how to patch all PDBs with the a new time zone file**?

Photo by Laurens Massaro on Unsplash

How to patch all your PDBs with a new time zone patch?

Getting ready for the future

Upgrade / Migrate and Consolidate to Multitenant at La Mobilière, Switzerland

Customer Case | La Mobilière

Customer

Project

Constraints

Preparation

Upgrade

Success?

Remarks

Swiss Mobiliar

- Switzerland's most personal insurer
- Founded 1826 in Bern, oldest Swiss insurance
- Legal form:
  - Cooperative association (mutual company)

Customer Case | La Mobilière

Customer

Project

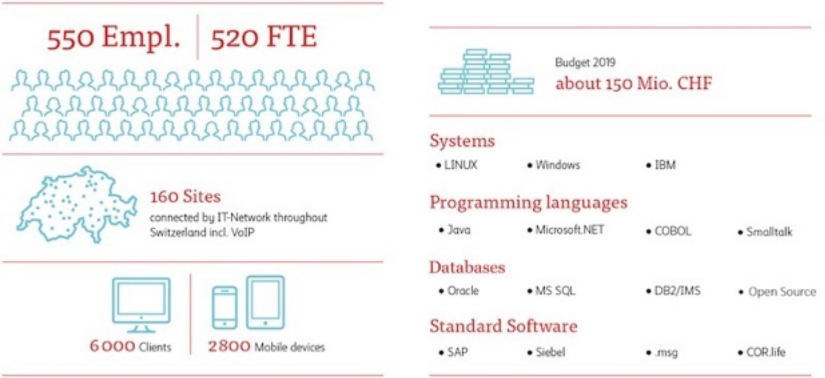
Constraints

Preparation

Upgrade

Success?

Remarks



550 Empl. | 520 FTE

160 Sites
connected by IT-Network throughout Switzerland incl. VoIP

6000 Clients | 2800 Mobile devices

Budget 2019
about 150 Mio. CHF

Systems

- LINUX
- Windows
- IBM

Programming languages

- Java
- Microsoft.NET
- COBOL
- Smalltalk

Databases

- Oracle
- MS SQL
- DB2/IMS
- Open Source

Standard Software

- SAP
- Siebel
- .msg
- COR.life

Customer Case | La Mobilière

Customer

Project 2017

Constraints

Preparation

Upgrade

Success?

Remarks

Upgrade 337 databases

- Oracle Database 12.1.0.2 to Oracle Database 12.2.0.1
- 82 production databases
- 8 container databases
  - 350 PDBs
  - Max of 50 PDB's in one CDB in dev

Move from schema-based consolidation to PDBs

PDB-only architecture with Oracle 12.2

- Except 3rd party app restrictions

Customer Case | La Mobilière

Customer

Project 2017

Constraints

Preparation

Upgrade

Success?

Remarks

Motivation

- Developers want Oracle 12.2 features
- Cost savings with Multitenant
- Reduce admin costs by automation

Customer Case | La Mobilière

Customer

Project 2017

Constraints

Preparation

Upgrade

Success?

Remarks

Regression tests

- Done during the testing phase of the Mobiliar Software Release
- Database RELEASE UPDATE (RU): 12.2.0.1.170718

Dual Oracle Home strategy

Upgrade to Oracle Database 12.2

- With catctl.pl embedded into home-built shell script

Performance tests performed by application owner

Customer Case | La Mobilière

Customer

Project 2017

Constraints

Preparation

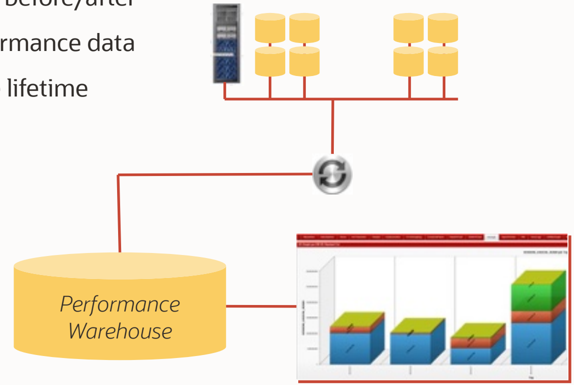
Upgrade

Success?

Remarks

Performance tracking with Mobiliar's own AWR Warehouse

- Compare performance before/after
- 7 TB of historical performance data
- Covers entire database lifetime



Customer Case | La Mobilière

- Customer
- Project 2017
- Constraints
- Preparation
- **Upgrade**
- Success?
- Remarks

50 PDBs - Upgrade in one pass

Upgrade Steps (parallel degree: -n 32)	Duration
PRE TASKS (online)	**13 min**
Execution of preupgrade.jar	5 min
Execution of pre upgrade fixup scripts	8 min
Upgrade TASKS (offline)	**3h 29 min**
Create guarantee restore point and change oracle home to 12.2	5 min
All in one Upgrade of CDB$ROOT and all PDB's	2h 46 min
Recompile of CDB$ROOT and all PDB's after upgrade to 12.2	32 min
Enable local undo mode for container database	6 min
POST TASKS (online)	**11 min**
Immediate create level 0 backup of container database	10 min
Drop guarantee restore point after successful upgrade	1 min
Total Upgrade Time	**3h 53 min**

Customer Case | La Mobilière

Customer

Project 2017

Constraints

Preparation

Upgrade

Success?

Remarks

Parallel upgrade `catctl.pl` unfolds its full power when upgrading many PDBs at the same time

- 50 PDBs upgraded in less than 4 hours

When we encounter issues, we fix them before going live

- Follow their projects on: https://mobiliar.ch/db-blog

100% Multitenant Consolidation reached in Oct 1, 2019

Customer Case | La Mobilière

Customer

Project 2017

Constraints

Preparation

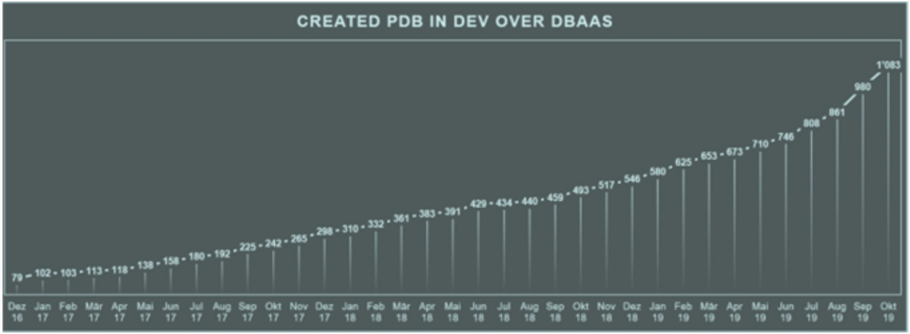
Upgrade

Success?

Remarks

High increase in number of PDBs, often driven by Microservices

- Self-developed DBaaS interface to provision, alter and drop PDBs
- Used to create >1000 PDBs



Migrating to the Future

Exadata Migration
at La Mobilière, Switzerland

Customer Case | La Mobilière

Customer

Project 2019

Constraints

Preparation

Migration

Success?

Remarks

Move many PDBs to a new Exadata

- 2 Exadata X8
- 3 DB Nodes each

Customer Case | La Mobilière

Customer

Project 2019

Constraints

Preparation

Migration

Success?

Remarks

Downtime <24 hours

Customer Case | La Mobilière

Customer

Project 2019

Constraints

Preparation

Migration

Success?

Remarks

Move to Exadata

- Complete script automation
  - Including error handling

Pre-Actions

- Lock the app user on source PDB
- Deactivate the app service on the PDB
- Create DB Link for remote clone
- Remove PDB from Cloud Control

Post actions

- Perform datapatch - newer RU
- Perform backup of each PDB
- Unlock the app user on target PDB
- Create the new app service
- Update Cloud Control with new PDB
- Delete Clone DB Link

Customer Case | La Mobilière

Customer

Project 2019

Constraints

Preparation

Migration

Success?

Remarks

Automated script

- Kick off: Friday, 22:00h
- 8 parallel script loops
- Monitored first clone loop - looked good!
- Went to sleep …

Customer Case | La Mobilière

Customer

Project 2019

Constraints

Preparation

Migration

Success?

Remarks

352 PDBs

- 346 moved to the new CDBs fully automatically without errors
- 8 PDB’s aborted with errors
  - Got identified quickly and moved manually
- Loop scripts needed between 3 - 6 hours to move the 150 PDBs
  - Including pre and post tasks

Customer Case | La Mobilière

Customer

Project 2019

Constraints

Preparation

Migration

Success?

Remarks

Read the full story at:

- https://mobiliardbblog.wordpress.com/2019/12/16/consolidating-350-pdbs-in-less-than-6-hours/

Upgrade NOW!

Upgrade to Oracle 19c at La Mobilière, Switzerland

Photo by Ricardo Gomez Angel on Unsplash

Customer Case | La Mobilière

Customer

Project 2020

Constraints

Preparation

Upgrade

Success?

Remarks

Upgrade 2000 PDBs

- Oracle Database 12.2.0.1 to 19c
- Up to 50 PDBs per CDB in PROD
- Up to 150 PDBs per CDB in Dev/Test

Customer Case | La Mobilière

Customer

Project 2020

Constraints

Preparation

Upgrade

Success?

Remarks

CPU resources limited

- Solution: Sequential upgrades
- One DBA covers 1-2 CDB upgrades
- Once done, next DBA steps in

Customer Case | La Mobilière

Customer

Project 2020

Constraints

Preparation

Upgrade

Success?

Remarks

Adopt AutoUpgrade

- Download newest version from MOS: 2485457.1
- Phase 1: 735 PDBs on a single weekend

CDB1	144 PDBs
CDB2	148 PDBs
CDB3	148 PDBs
CDB4	147 PDBs
CDB5	148 PDBs

Customer Case | La Mobilière

Customer

Project 2020

Constraints

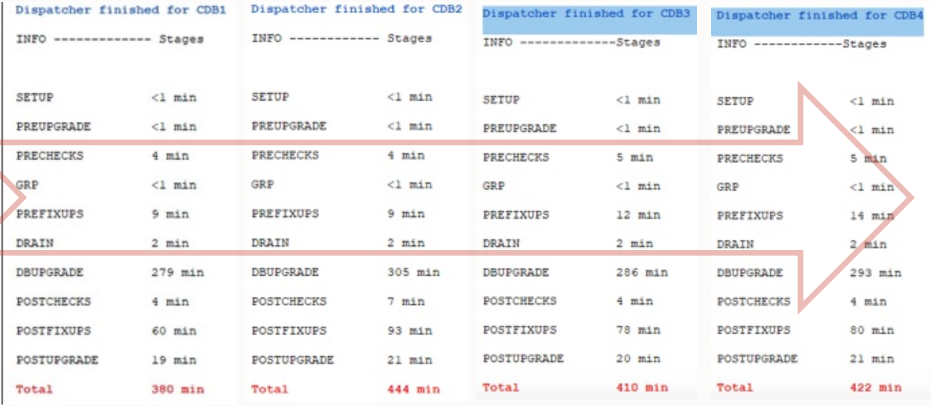
Preparation

Upgrade

Success?

Remarks

Upgrade timings *



```
Dispatcher finished for CDB1
INFO ------------- Stages

SETUP           <1 min
PREUPGRADE      <1 min
PRECHECKS        4 min
GRP             <1 min
PREFIXUPS        9 min
DRAIN            2 min
DBUPGRADE      279 min
POSTCHECKS       4 min
POSTFIXUPS      60 min
POSTUPGRADE     19 min
Total          380 min
```

```
Dispatcher finished for CDB2
INFO ------------ Stages

SETUP           <1 min
PREUPGRADE      <1 min
PRECHECKS        4 min
GRP             <1 min
PREFIXUPS        9 min
DRAIN            2 min
DBUPGRADE      305 min
POSTCHECKS       7 min
POSTFIXUPS      93 min
POSTUPGRADE     21 min
Total          444 min
```

```
Dispatcher finished for CDB3
INFO -------------Stages

SETUP           <1 min
PREUPGRADE      <1 min
PRECHECKS        5 min
GRP             <1 min
PREFIXUPS       12 min
DRAIN            2 min
DBUPGRADE      286 min
POSTCHECKS       4 min
POSTFIXUPS      78 min
POSTUPGRADE     20 min
Total          410 min
```

```
Dispatcher finished for CDB4
INFO ------------Stages

SETUP           <1 min
PREUPGRADE      <1 min
PRECHECKS        5 min
GRP             <1 min
PREFIXUPS       14 min
DRAIN            2 min
DBUPGRADE      293 min
POSTCHECKS       4 min
POSTFIXUPS      80 min
POSTUPGRADE     21 min
Total          422 min
```

* Logs of CDB5 are lost

Customer Case | La Mobilière

Customer

Project 2020

Constraints

Preparation

Upgrade

Success?

Remarks

Upgrade timings

- Average: 6 - 7.5 hours
- Including recompilation
- Only 10 oCPUs used per CDB

Customer Case | La Mobilière

Customer

Project 2020

Constraints

Preparation

Upgrade

Success?

Remarks

Post-fixups issue

- Solution: Use the newest OPatch

Processes parameter

- Upgrade in PDBs spawns a lot of processes
- Solution: Increase it for the upgrade

Recompile PDBs before upgrade

-
  ```
  ORACLE_HOME/perl/bin/perl $ORACLE_HOME/rdbms/admin/catcon.pl -n 32 -e -d $ORACLE_HOME/rdbms/admin -l /tmp -b utlrp_log utlrp.sql
  ```

Read the full story here:

- https://mobiliardbblog.wordpress.com/2020/05/05/what-does-it-take-to-migrate-735-databases-to-19c-in-a-weekend-the-multitenant-architecture-and-great-dbas/

*"We upgraded 735 databases to 19c, and the task was mostly relatively relaxed.
Start the AutoUpgrade tool and monitor the progress from time to time.*

Sitting in front of the screen the whole time is not necessary."

Alain Fuhrer
Head IT Database Services
La Mobilière
Bern, Switzerland

Photo by Todd Trapani on Unsplash

Chapter 6
Migration Strategies

Migration Strategies

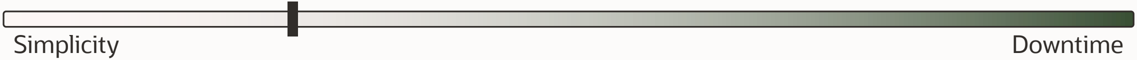
Overview

What do we call a migration?

Migration | Move data from A to B

Which one is the best technique?

Migration | Techniques



Simplicity

Downtime

Techniques include:

- Data Pump
- Transportable Tablespaces
- Full Transportable Export/Import
- Data Guard
- Incremental Backups
- Oracle GoldenGate

We will give you detailed insights!

For important databases, execute **database health checks** before upgrade

Health Checks

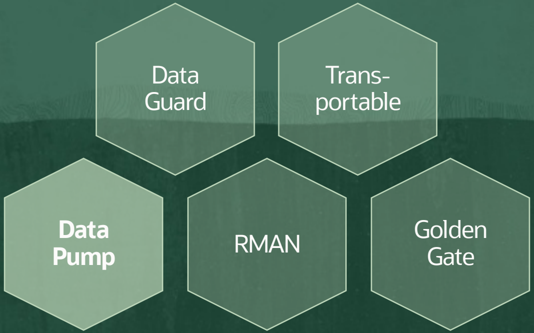
Health check script

- Download from MOS Note: 1366971
- In Multitenant, it must be run in each PDB separately

ORAchk Upgrade Readiness Assessment

- Part of Autonomous Health Framework (AHF)
- Download from MOS Note: 1457357.1
- Upgrade Readiness Check – MOS Note: 2550798.1

Different **MIGRATION** techniques



Data Guard

Trans-portable

Data Pump

RMAN

Golden Gate

Migration Strategies

Data Pump

Data Pump

Advantages

- Ease of use
- Universal
- Change structures, character set, and much more
- Platform independent
- Architecture independent
- Works across versions
- Backwards compatible

Documentation

- Oracle Database 19c Utilities Guide

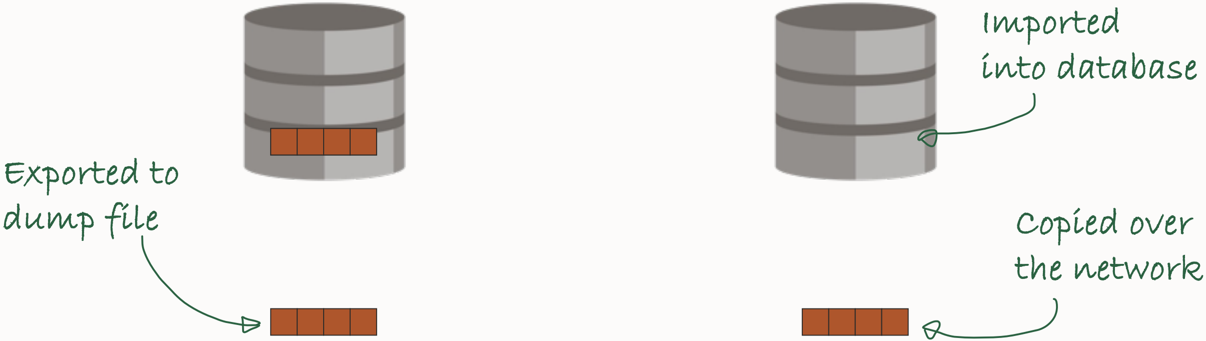
Consideration

- Duration for large amounts of data and complex structures

Data Pump

Basics

Data Pump | Dump File

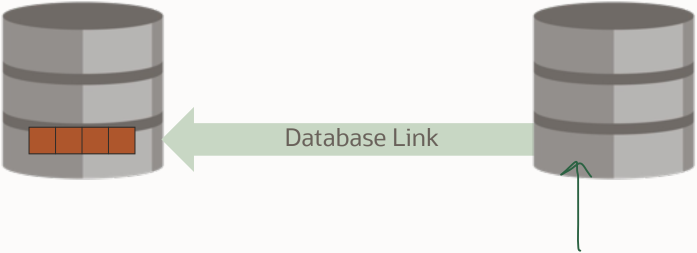


Exported to dump file

Imported into database

Copied over the network

Data Pump | Network Mode



Imported directly over network

Data Pump | Mode Comparison

DUMP FILE	NETWORK
Requires access to file system	SQL*Net connectivity
Requires disk space for dump files	No extra disk space needed
Full functionality	Limited functionality

Pro tip: Read more about how Data Pump moves data

Data Pump | Architecture

DBMS_DATAPUMP
DBMS_METADATA

expdp

DBMS_DATAPUMP
DBMS_METADATA

impdp

Data Pump | Prerequisites

Privilege

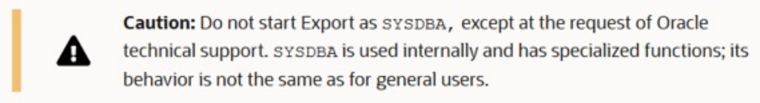
Directory

Streams Pool

Two predefined roles:

- `DATAPUMP_EXP_FULL_DATABASE`
- `DATAPUMP_IMP_FULL_DATABASE`

Don't use `SYS AS SYSDBA`



Caution: Do not start Export as `SYSDBA`, except at the request of Oracle technical support. `SYSDBA` is used internally and has specialized functions; its behavior is not the same as for general users.

Database 19c, Utilities Guide

Pro tip: These roles are powerful - use caution when granting them

Data Pump | Prerequisites

Privilege

Directory

Streams Pool

Needed to store dump files and log files

```
$ mkdir /home/oracle/dp

SQL> create directory DP_DIR as '/home/oracle/dp';

SQL> grant read, write on directory DP_DIR to SYSTEM;
```

Pro tip: Data Pump runs server side, thus, the directory must be accessible to the database server

Data Pump | Prerequisites

Privilege

Directory

Streams Pool

Ensure `STREAMS_POOL_SIZE` is at a reasonable value

```
SQL> alter system set streams_pool_size=128m scope=both;
```

Typically, in the range of 64M to 256M is adequate

Pro tip: Read about how other parameters affect Data Pump

Data Pump

General Best Practices

Data Pump | Best Practices

- **Parameter file**
- Consistency
- Dictionary Statistics
- Diagnostics
- Parallel
- Statistics
- LOBs
- Dump files
- Compression

Do not add parameters to command line

```
$ expdp system directory=dp_dir schemas=scott
logfile=export_scott.log parallel=8 ...
```

Use a parameter file

```
$ more export.par
directory=dp_dir
schemas=scott
logfile=export_scott.log
parallel=8
...

$ expdp system parfile=export.par
```

Data Pump | Best Practices

Parameter file
Consistency
Dictionary Statistics
Diagnostics
Parallel
Statistics
LOBs
Dump files
Compression

Perform consistent exports

```
flashback_time=systimestamp
```

To a specific SCN

```
flashback_scn=<scn>
```

Requires UNDO

- Export at off hours
- Export from Data Guard
  - Convert temporarily to snapshot standby

Pro tip: In Data Pump Legacy mode you can use `CONSISTENT=Y`

Data Pump | Best Practices

Parameter file

Consistency

Dictionary Statistics

Diagnostics

Parallel

Statistics

LOBs

Dump files

Compression

Gather dictionary stats:

- Right before an export
- Immediately after an import

Gathering stats

```
SQL> BEGIN
        DBMS_STATS.GATHER_SCHEMA_STATS('SYS');
        DBMS_STATS.GATHER_SCHEMA_STATS('SYSTEM');
     END;
```

```
$ORACLE_HOME/perl/bin/perl $ORACLE_HOME/rdbms/admin/catcon.pl \
  -l /tmp \
  -b gatherstats -- \
  --x"begin dbms_stats.gather_schema_stats('SYS');
dbms_stats.gather_schema_stats('SYSTEM'); end;"
```

Data Pump | Best Practices

Parameter file
Consistency
Dictionary Statistics
Diagnostics
Parallel
Statistics
LOBs
Dump files
Compression

Always include diagnostic information

```
logtime=all
metrics=yes
```

Adds

- Timestamps
- Internal timings
- Access method

Pro tip: Parameter `LOGTIME` was added in Oracle Database 12.1

Data Pump | Best Practices

No diagnostics

```
Processing object type SCHEMA_EXPORT/TABLE/TABLE
Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA
. . imported "METAL"."ALBUMS"                           988.8 KB   28069 rows
. . imported "METAL"."BANDS"                            3.444 MB   37723 rows
. . imported "METAL"."REVIEWS"                          66.47 MB   21510 rows
```

All diagnostics

```
16-OCT-20 17:26:57.158: Processing object type SCHEMA_EXPORT/TABLE/TABLE
16-OCT-20 17:26:58.262: Startup took 1 seconds
16-OCT-20 17:26:58.264: Startup took 1 seconds
16-OCT-20 17:26:59.082:      Completed 3 TABLE objects in 1 seconds
16-OCT-20 17:26:59.082:      Completed by worker 1 1 TABLE objects in 1 seconds
16-OCT-20 17:26:59.082:      Completed by worker 2 1 TABLE objects in 0 seconds
16-OCT-20 17:26:59.082:      Completed by worker 3 1 TABLE objects in 0 seconds
16-OCT-20 17:26:59.313: Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA
16-OCT-20 17:27:01.943: . . imported "METAL"."ALBUMS"   988.8 KB   28069 rows in 2 seconds using external_table
16-OCT-20 17:27:03.778: . . imported "METAL"."BANDS"    3.444 MB   37723 rows in 2 seconds using external_table
16-OCT-20 17:27:12.644: . . imported "METAL"."REVIEWS"  66.47 MB   21510 rows in 13 seconds using external_table
```

Data Pump | Best Practices

Parameter file
Consistency
Dictionary Statistics
Diagnostics
Parallel
Statistics
LOBs
Dump files
Compression

Speed up

```
parallel=n
```

Typically, set *n* to *number of CPU cores x 2*

Parallel export/import of metadata
- introduced in Oracle Database 12.2

Except

- Transportable Tablespaces (added in 21c)
- Full Transportable Export/Import (also in 21c)
- Network mode

Pro tip: More details in Parallel Capabilities of Oracle Data Pump (Doc ID 365459.1)

Data Pump | Best Practices

Parallelism for import into 11.2.0.4 / 12.1.0.2

- Apply patch for bug 22273229 to enable parallel import of constraints/indexes

Data Pump | Best Practices

Parameter file
Consistency
Dictionary Statistics
Diagnostics
Parallel
Statistics
LOBs
Dump files
Compression

Always exclude optimizer statistics

```
exclude=statistics
```

Transportable Tablespaces

```
exclude=table_statistics,index_statistics
```

Either

- Gather fresh statistics
- Transport statistics with `DBMS_STATS`

Data Pump | Best Practices

Parameter file

Consistency

Dictionary Statistics

Diagnostics

Parallel

Statistics

LOBs

Dump files

Compression

On import, always convert to SecureFile LOBs

```
transform=lob_storage:securefile
```

SecureFile LOBs

- Can import in parallel
- Are generally faster
- Offers advanced functionality
- Are a good fit for partitioning
- Are strongly recommended

Data Pump | Best Practices

Importing as BasicFiles

```
10-OCT-20 21:43:21.848: W-3 . . imported "SCHEMA"."TABLE"      31.83 GB  681025 rows in 804 seconds using direct_path
```

Importing as SecureFiles

```
15-OCT-20 18:16:48.663: W-13 . . imported "SCHEMA"."TABLES"     31.83 GB  681025 rows in 261 seconds using external_table
```

Data Pump | Best Practices

Parameter file
Consistency
Dictionary Statistics
Diagnostics
Parallel
Statistics
LOBs
Dump files
Compression

Export to multiple files

```
dumpfile=dumpfile%U.dmp
filesize=5g
```

For more than 99 files

```
dumpfile=dumpfile%L.dmp
filesize=5g
```

Required if you use parallel export

Data Pump | Best Practices

- Parameter file
- Consistency
- Dictionary Statistics
- Diagnostics
- Parallel
- Statistics
- LOBs
- Dump files
- **Compression**

Use compression to speed up your export

```
compression=all
compression_algorithm=medium
```

Requires Advanced Compression Option

Pro tip: `COMPRESSION=METADATA_ONLY` does not require Advanced Compression Option

Data Pump | Best Practices

Parameter file
Consistency
Dictionary Statistics
Diagnostics
Parallel
Statistics
LOBs
Dump files
Compression

Compression algorithms

BASIC	The same algorithm used in previous versions. Good compression, without severely impacting on performance
LOW :	For use when reduced CPU utilization is a priority over compression ratio
MEDIUM:	Recommended option. Similar characteristics to BASIC, but uses a different algorithm
HIGH:	Maximum available compression, but more CPU intensive

Data Pump | Best Practices

Parameter file
Consistency
Dictionary Statistics
Diagnostics
Parallel
Statistics
LOBs
Dump files
Compression

Real-life examples - 12.2 EBS Database export

	FILE SIZE MB	RATIO	TIME
NONE	5.500	1,0	4m 54s
ALL BASIC	622	8,9	4m 58s
ALL LOW	702	7,8	5m 24s
ALL MEDIUM	567	9,7	4m 55s
ALL HIGH	417	13,2	5m 13s

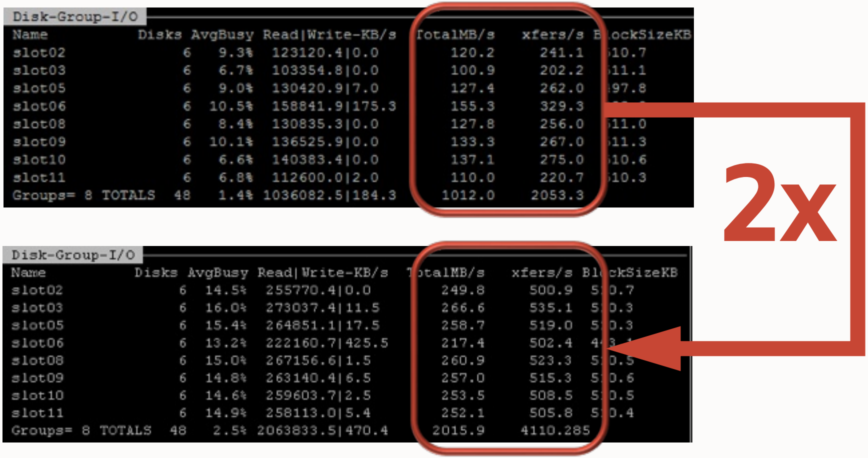
	FILE SIZE MB	RATIO	TIME
NONE	5.800	1,0	2m 33s
ALL BASIC	705	8,2	3m 03s
ALL LOW	870	6,6	8m 11s
ALL MEDIUM	701	8,2	3m 01s
ALL HIGH	509	11,3	12m 16s

Data Pump | Best Practices

Customer evaluation

- `BASIC`
  at 3.5 TB/hour

- `MEDIUM`
  at 7.0 TB/hour



```
Disk-Group-I/O
Name                 Disks  AvgBusy  Read|Write-KB/s    TotalMB/s     xfers/s  BlockSizeKB
slot02                   6     9.3%   123120.4|0.0          120.2       241.1    510.7
slot03                   6     6.7%   103354.8|0.0          100.9       202.2    511.1
slot05                   6     9.0%   130420.9|7.0          127.4       262.0    497.8
slot06                   6    10.5%   158841.9|175.3        155.3       329.3    [illegible]
slot08                   6     8.4%   130835.3|0.0          127.8       256.0    511.0
slot09                   6    10.1%   136525.9|0.0          133.3       267.0    511.3
slot10                   6     6.6%   140383.4|0.0          137.1       275.0    510.6
slot11                   6     6.8%   112600.0|2.0          110.0       220.7    510.3
Groups= 8 TOTALS        48     1.4%  1036082.5|184.3       1012.0      2053.3
```

```
Disk-Group-I/O
Name                 Disks  AvgBusy  Read|Write-KB/s    TotalMB/s     xfers/s  BlockSizeKB
slot02                   6    14.5%   255770.4|0.0          249.8       500.9    510.7
slot03                   6    16.0%   273037.4|11.5         266.6       535.1    510.3
slot05                   6    15.4%   264851.1|17.5         258.7       519.0    510.3
slot06                   6    13.2%   222160.7|425.5        217.4       502.4    [illegible]
slot08                   6    15.0%   267156.6|1.5          260.9       523.3    510.5
slot09                   6    14.8%   263140.4|6.5          257.0       515.3    510.6
slot10                   6    14.6%   259603.7|2.5          253.5       508.5    510.5
slot11                   6    14.9%   258113.0|5.4          252.1       505.8    510.4
Groups= 8 TOTALS        48     2.5%  2063833.5|470.4       2015.9    4110.285
```

Data Pump

Use Cases

Data Pump | With Data Guard

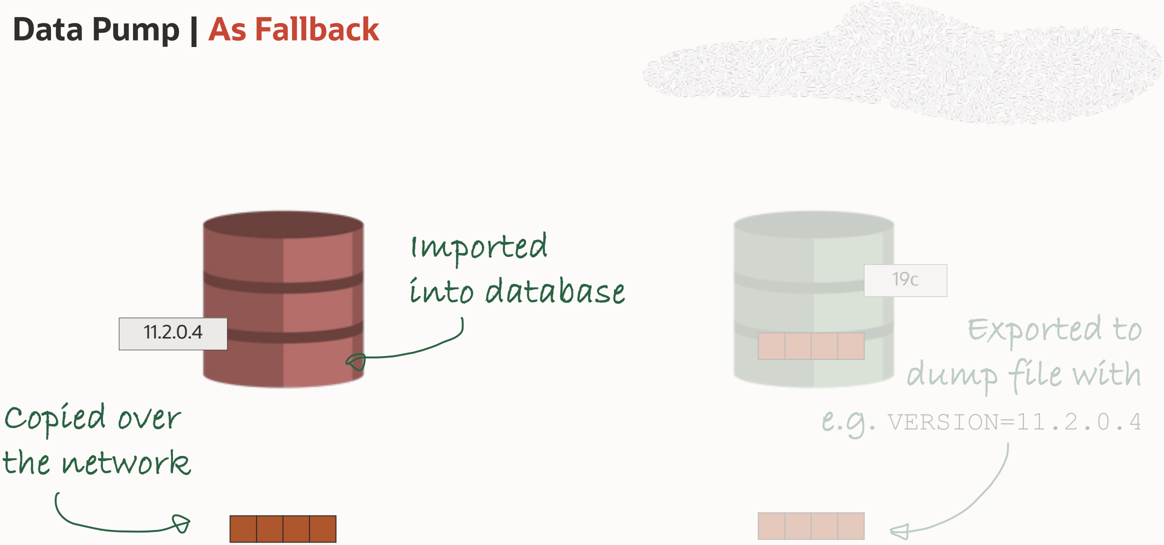
Import into Data Guard environment - works seamlessly

1. Ensure `STANDBY_FILE_MANAGEMENT=AUTO`
   - Optionally, configure `DB_FILE_NAME_CONVERT` as well

2. Create new PDB from PDB$SEED
   - Propagates automatically to standby database

3. Import with Data Pump
   - Import happens implicitly on standby via redo apply
   - Tablespaces are automatically created

Data Pump | As Fallback



Data Pump | As Fallback

To create a dump file compatible with a lower release

```
version=11.2.0.4
```

Other options are

- `COMPATIBLE`
- `LATEST`

Export/Import DataPump Parameter VERSION - Compatibility of Data Pump Between Different Oracle Versions (Doc ID 553337.1)

Pro tip: Read more about VERSION in the documentation

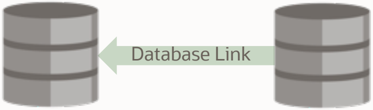
Data Pump | Network Mode

Create database link and start `impdp`

```
SQL> create database link V11204 ... ;

$ impdp ... network_link=V11204
```

`expdp` is invoked implicitly on remote database

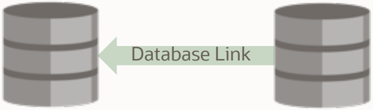
No dump file is created -
data is transmitted directly over database link



Data Pump | Network Mode

Limitations:

- No parallel metadata support (yet)
- No LONG and RAW data
  - Unless source database is 12.2 or newer
  - `ACCESS_METHOD=DIRECT_PATH`
- Data parallelism is restricted to multiple partitions or tables
  - There is no PQ parallelism within a large, unpartitioned table over a dblink
- Does not work for downgrades



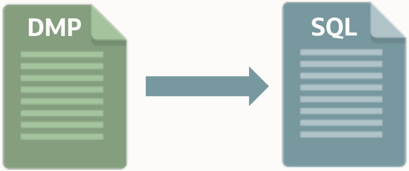
Data Pump

Advanced Features

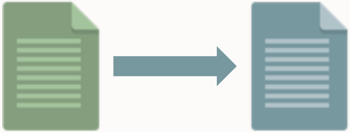
Data Pump | Generate SQL Statements

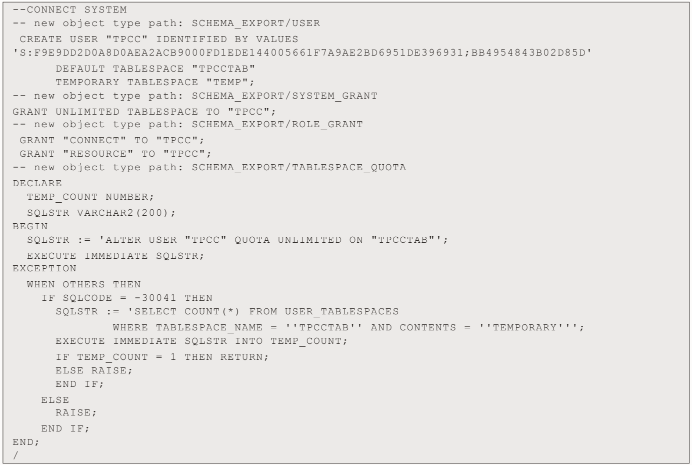
Generate SQLs that `impdp` will execute

```
$ more import.par
...
sqlfile=all_statements.sql
...

$ impdp system parfile=import.par
```



Data Pump | Generate SQL Statements

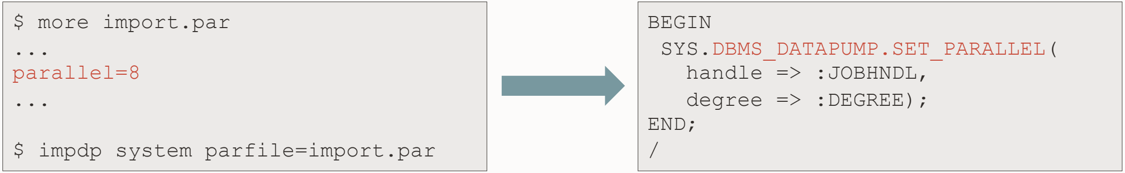




```
--CONNECT SYSTEM
-- new object type path: SCHEMA_EXPORT/USER
 CREATE USER "TPCC" IDENTIFIED BY VALUES
'S:F9E9DD2D0A8D0AEA2ACB9000FD1EDE144005661F7A9AE2BD6951DE396931;BB4954843B02D85D'
      DEFAULT TABLESPACE "TPCCTAB"
      TEMPORARY TABLESPACE "TEMP";
-- new object type path: SCHEMA_EXPORT/SYSTEM_GRANT
GRANT UNLIMITED TABLESPACE TO "TPCC";
-- new object type path: SCHEMA_EXPORT/ROLE_GRANT
 GRANT "CONNECT" TO "TPCC";
 GRANT "RESOURCE" TO "TPCC";
-- new object type path: SCHEMA_EXPORT/TABLESPACE_QUOTA
DECLARE
  TEMP_COUNT NUMBER;
  SQLSTR VARCHAR2(200);
BEGIN
  SQLSTR := 'ALTER USER "TPCC" QUOTA UNLIMITED ON "TPCCTAB"';
  EXECUTE IMMEDIATE SQLSTR;
EXCEPTION
  WHEN OTHERS THEN
    IF SQLCODE = -30041 THEN
      SQLSTR := 'SELECT COUNT(*) FROM USER_TABLESPACES
              WHERE TABLESPACE_NAME = ''TPCCTAB'' AND CONTENTS = ''TEMPORARY''';
      EXECUTE IMMEDIATE SQLSTR INTO TEMP_COUNT;
      IF TEMP_COUNT = 1 THEN RETURN;
      ELSE RAISE;
      END IF;
    ELSE
      RAISE;
    END IF;
END;
/
```

Data Pump | Generate PL/SQL

Generate PL/SQL to use DBMS_DATAPUMP API



```
$ more import.par
...
parallel=8
...

$ impdp system parfile=import.par
```

```
BEGIN
 SYS.DBMS_DATAPUMP.SET_PARALLEL(
   handle => :JOBHNDL,
   degree => :DEGREE);
END;
/
```

Enables you to start Data Pump Export and Import directly from PL/SQL

Data Pump | Generate PL/SQL

1. Enable SQL trace on a test database

```
SQL> alter system set event='10046 trace name context forever, level 4';
```

2. Execute your Data Pump command

```
$ impdp system ... parfile=import.par
```

3. Examine the trace file

```
$ vi ORCL_ora_12345.trc
```

Data Pump | Generate PL/SQL

```
DECLARE
   JOBHNDL NUMBER;
BEGIN
   JOBHNDL := SYS.DBMS_DATAPUMP.OPEN(
      operation => 'EXPORT',
      job_mode => 'SCHEMA',
      remote_link => NULL,
      job_name => NULL,
      version =>  NULL,
      ena_sec_roles => 0);
   SYS.DBMS_DATAPUMP.ADD_FILE(
      handle => JOBHNDL,
      filename => 'demo_exp.log',
      directory => 'DMPDIR',
      filesize => NULL,
      filetype => 3,
      reusefile => NULL);
   SYS.DBMS_DATAPUMP.SET_PARALLEL(
      handle => JOBHNDL,
      degree => 8);
   ...
   ...
   ...
```

Data Pump | Existing Objects

Using `TABLE_EXISTS_ACTION` set to `APPEND` or `TRUNCATE` is slow

Use case: Importing schema with thousands of partitions

Problem: It takes a lot of time to create all those objects

Solution: Create partitions in advance and use `TABLE_EXISTS_ACTION=APPEND`

Problem: That's slow

Solution: Tell Data Pump to trust the objects are fine

```
DATA_OPTION = TRUST_EXISTING_TABLE_PARTITIONS
```

Data Pump | Multitenant

Use case: Avoid *noisy-neighbour* syndrome caused by Data Pump

Problem: Too many Data Pump jobs are started in a PDB, depleting the resources of the CDB

Solution: Restrict number of concurrent Data Pump jobs in a PDB

```
SQL> alter system set max_datapump_jobs_per_pdb=2 container=all;
```

Default:	100
Set to `AUTO`:	50 % of `SESSIONS`

Pro tip: Too many Data Pump jobs result in `ORA-39391`

Data Pump | Multitenant

Use case: Avoid *noisy-neighbour* syndrome caused by Data Pump

Problem: Too many parallel workers are used in a Data Pump job, depleting the resources of the CDB

Solution: Restrict the parallel degree in a Data Pump job

NEW IN 19c

```
SQL> alter system set max_datapump_parallel_per_job=2 container=all;
```

Default:	50
Set to `AUTO`:	50 % of `SESSIONS`

Data Pump | Performance

Blog Posts:

[Why Does It Take so Long to Import Check Constraints?]

Data Pump | Further Information

Technical Briefs:

Data Pump Best Practices for Export and Import

MOS Notes

Master Note for Data Pump (Doc ID 1264715.1)

For Compatibility and version changes (Doc ID 553337.1)

19c Data Pump New Features (Doc ID 2457955.1)

The Second Exadata in Europe

Payback GmbH Germany
HP to Exadata Migration Project 2009

https://www.payback.de/ratgeber/besser-leben/payback-tricks

Customer Case | Payback

Customer

Project 2009

Constraints

Preparation

Migration

Success?

Remarks

Payback GmbH

- Belongs to American Express
- HQ in Munich, Germany
- Develops and operates professional customer loyalty programs based on customized IT solutions

Customer Case | Payback

Customer

Project 2009

Constraints

Preparation

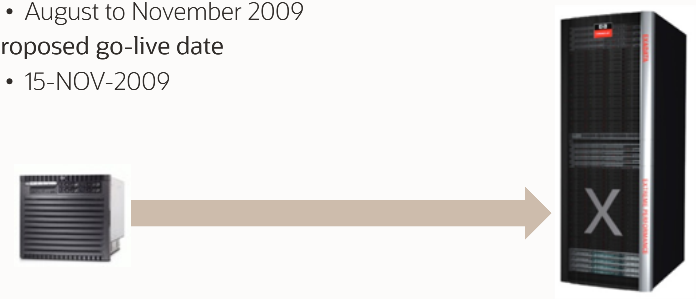
Migration

Success?

Remarks

Migrate 7TB / 1.5TB from HP-UX to Exadata V1

- Cross platform, cross Endianness, cross version
  - Oracle 9.2.0.7 on HP-UX ➩ Oracle 11.1.0.7 on OL
- 4 months planning and migration phase
  - August to November 2009
- Proposed go-live date
  - 15-NOV-2009



Customer Case | Payback

Customer

Project 2009

Constraints

Preparation

Migration

Success?

Remarks

Move everything in less than 24 hrs

Network bottleneck

- Remedy:
  Install extra InfiniBand hardware into HP box
  ➪ ~ 3GB/sec throughput!

Customer Case | Payback

Customer

Project 2009

Constraints

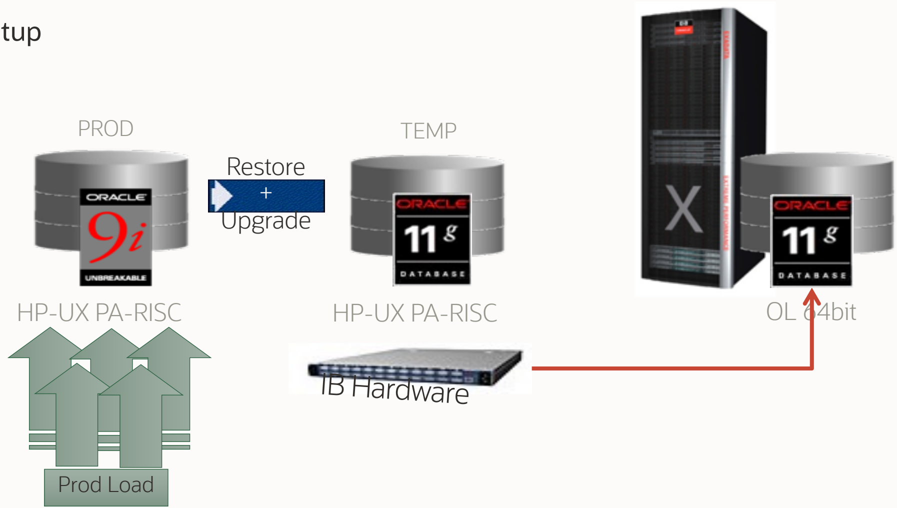
Preparation

Migration

Success?

Remarks

Setup



Customer Case | Payback

Customer

Project 2009

Constraints

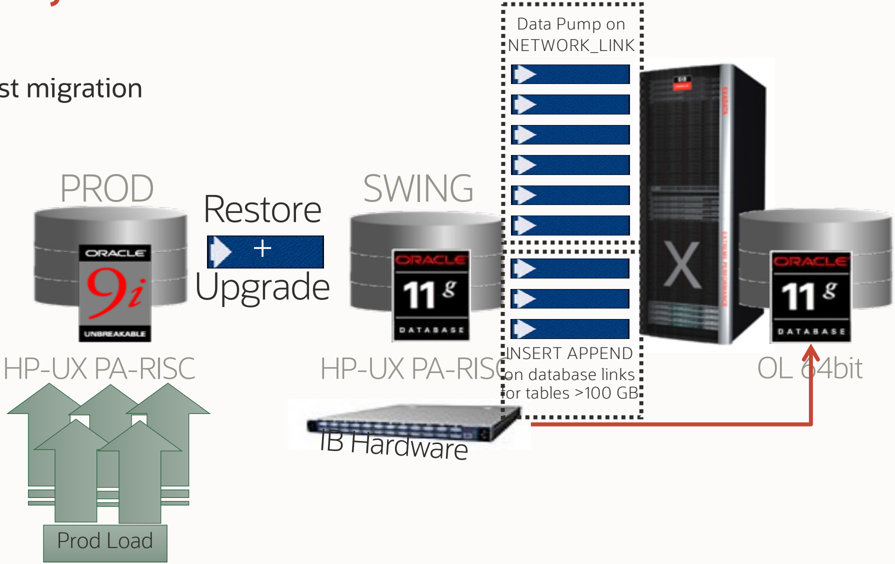
Preparation

Migration

Success?

Remarks

Test migration



Customer Case | Payback

Customer

Project 2009

Constraints

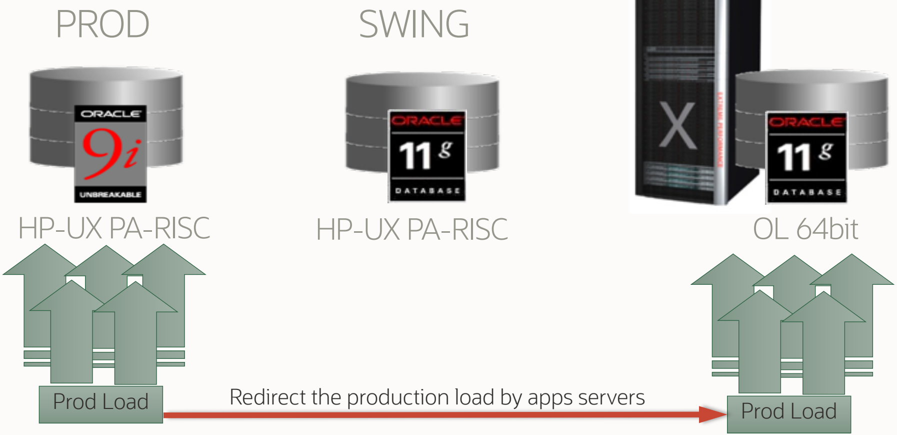
Preparation

Migration

Success?

Remarks

Parallel loads and performance tests



PROD

SWING

HP-UX PA-RISC

HP-UX PA-RISC

OL 64bit

Prod Load

Redirect the production load by apps servers

Prod Load

Customer Case | Payback

Customer

Project 2009

Constraints

Preparation

Migration

Success?

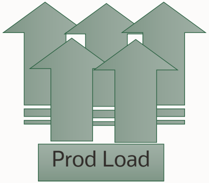
Remarks

Last test came live migration

HP-UX PA-RISC

PROD

OL 64bit



Customer Case | Payback

Customer

Project 2009

Constraints

Preparation

Migration

Success?

Remarks

Live? And alive?

- Yes! Go-live in early November 2009
  - Two weeks earlier than proposed
- Total upgrade and migration time: ~20 hours
  - ~ 8 hours: Restore and recovery
  - ~ 1 hour: Database upgrade to Oracle 11.1.0.7
  - ~10 hours: Data migration to Exadata V1
  - ~ 1 hour: Smoke testing and final verification
- Dramatic performance improvements
  - Job runtimes decreased by 80%
  - User complaints about too fast performance … really!!

Customer Case | Payback

Customer

Project 2009

Constraints

Preparation

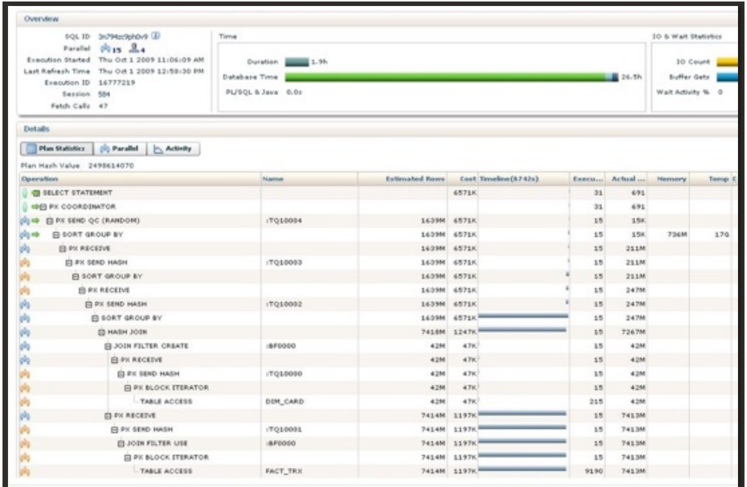
Migration

Success?

Remarks

Not a single piece of SQL got changed!!!

- Most critical job: runtime from 30hrs to < 2hrs



Overview

SQL ID 3n79Mzc9ph0v9

Parallel 15 4

Execution Started Thu Oct 1 2009 11:06:09 AM

Last Refresh Time Thu Oct 1 2009 12:58:30 PM

Execution ID 16777219

Session 584

Fetch Calls 47

Time

Duration 1.9h

Database Time 26.5h

PL/SQL & Java 0.0s

IO & Wait Statistics

IO Count

Buffer Gets

Wait Activity % 0

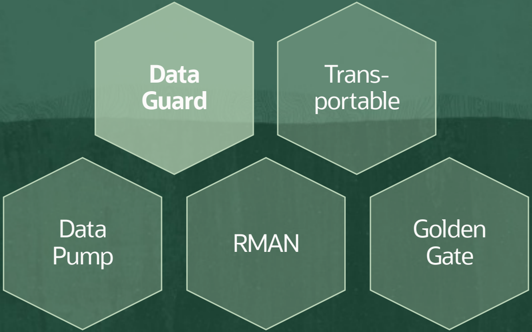
Details

Plan Statistics | Parallel | Activity

Plan Hash Value 2498614070

Operation	Name	Estimated Rows	Cost	Timeline(6742s)	Execu...	Actual ...	Memory	Temp
SELECT STATEMENT			6571K		31	691		
PX COORDINATOR					31	691		
PX SEND QC (RANDOM)	:TQ10004	1639M	6571K		15	15K		
SORT GROUP BY		1639M	6571K		15	15K	736M	17G
PX RECEIVE		1639M	6571K		15	211M		
PX SEND HASH	:TQ10003	1639M	6571K		15	211M		
SORT GROUP BY		1639M	6571K		15	211M		
PX RECEIVE		1639M	6571K		15	247M		
PX SEND HASH	:TQ10002	1639M	6571K		15	247M		
SORT GROUP BY		1639M	6571K		15	247M		
HASH JOIN		7418M	1247K		15	7267M		
JOIN FILTER CREATE	:BF0000	42M	47K		15	42M		
PX RECEIVE		42M	47K		15	42M		
PX SEND HASH	:TQ10000	42M	47K		15	42M		
PX BLOCK ITERATOR		42M	47K		15	42M		
TABLE ACCESS	DIM_CARD	42M	47K		215	42M		
PX RECEIVE		7414M	1197K		15	7413M		
PX SEND HASH	:TQ10001	7414M	1197K		15	7413M		
JOIN FILTER USE	:BF0000	7414M	1197K		15	7413M		
PX BLOCK ITERATOR		7414M	1197K		15	7413M		
TABLE ACCESS	FACT_TRX	7414M	1197K		9190	7413M		

Different **MIGRATION** techniques



Data Guard

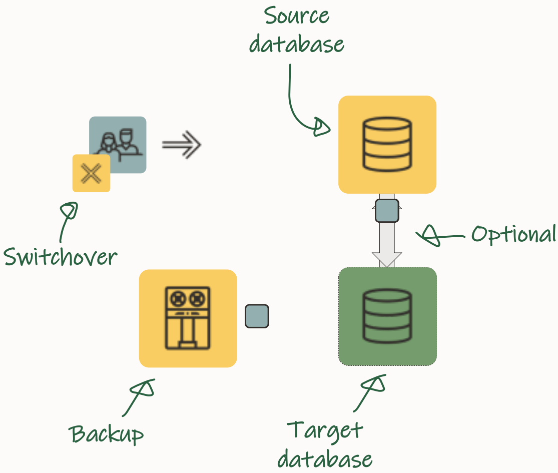
Transportable

Data Pump

RMAN

Golden Gate

Data Guard | Concept



Configure:

- Redo transport
- Redo apply

```
RMAN> RESTORE STANDBY CONTROLFILE ... ;
RMAN> RESTORE DATABASE ... ;
RMAN> RECOVER DATABASE UNTIL ... ;
```

Data Guard | Benefits

- Preferred solution
- Well-known, simple and easy
- Seamless switchover
  with properly configured application
- Some cross-platform capabilities

Source Oracle Home
is needed on target host

Data Guard | Platform Certification

1. Migrate database to new hardware and upgrade from 11.2.0.4 to 19c

2. Target host must run Oracle Linux 9

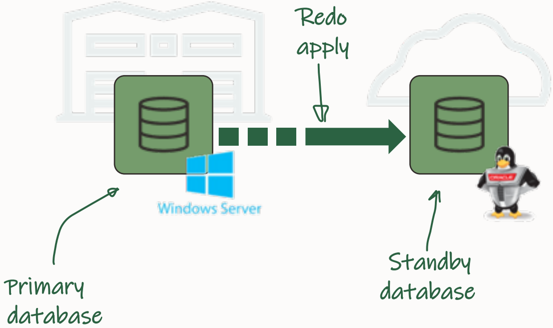
3. To use Data Guard, you must install Oracle Database 11.2.0.4 on target host

4. Oracle Database 11.2.0.4 is not certified on Oracle Linux 9

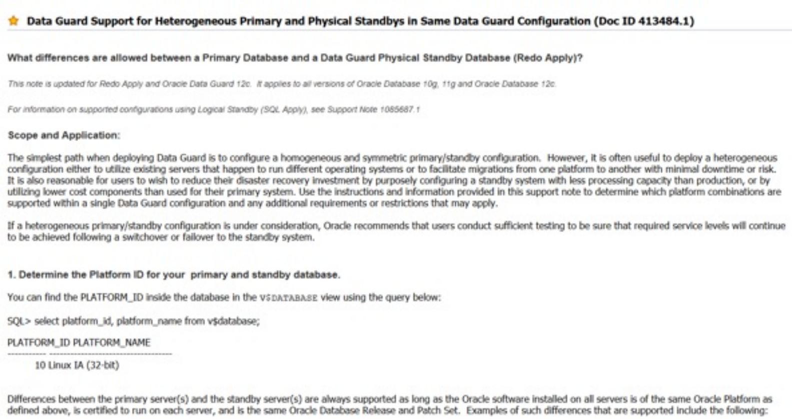
Data Guard not possbile

Do you need the same platform on source and target host?

Data Guard | Heterogeneous



Little Endian | Linux Standby



Data Guard Support for Heterogeneous Primary and Physical Standbys in Same Data Guard Configuration (Doc ID 413484.1)

What differences are allowed between a Primary Database and a Data Guard Physical Standby Database (Redo Apply)?

This note is updated for Redo Apply and Oracle Data Guard 12c. It applies to all versions of Oracle Database 10g, 11g and Oracle Database 12c.

For information on supported configurations using Logical Standby (SQL Apply), see Support Note 1085687.1

Scope and Application:

The simplest path when deploying Data Guard is to configure a homogeneous and symmetric primary/standby configuration. However, it is often useful to deploy a heterogeneous configuration either to utilize existing servers that happen to run different operating systems or to facilitate migrations from one platform to another with minimal downtime or risk. It is also reasonable for users to wish to reduce their disaster recovery investment by purposely configuring a standby system with less processing capacity than production, or by utilizing lower cost components than used for their primary system. Use the instructions and information provided in this support note to determine which platform combinations are supported within a single Data Guard configuration and any additional requirements or restrictions that may apply.

If a heterogeneous primary/standby configuration is under consideration, Oracle recommends that users conduct sufficient testing to be sure that required service levels will continue to be achieved following a switchover or failover to the standby system.

1. Determine the Platform ID for your primary and standby database.

You can find the PLATFORM_ID inside the database in the `V$DATABASE` view using the query below:

```
SQL> select platform_id, platform_name from v$database;

PLATFORM_ID PLATFORM_NAME
----------- ------------------------------
         10 Linux IA (32-bit)
```

Differences between the primary server(s) and the standby server(s) are always supported as long as the Oracle software installed on all servers is of the same Oracle Platform as defined above, is certified to run on each server, and is the same Oracle Database Release and Patch Set. Examples of such differences that are supported include the following:

Data Guard Support for Heterogeneous Primary and Physical Standbys in Same Data Guard Configuration (Doc ID 413484.1)

Little Endian | Linux Standby

```
SQL> SELECT platform_name, endian_format
     FROM v$transportable_platform
     WHERE endian_format='Little';

PLATFORM_NAME                        ENDIAN_FORMAT

____________________________________ ________________
Apple Mac OS (x86-64)                Little
HP IA Open VMS                       Little
HP Open VMS                          Little
HP Tru64 UNIX                        Little
Linux IA (32-bit)                    Little
Linux IA (64-bit)                    Little
Linux x86 64-bit                     Little
Microsoft Windows IA (32-bit)        Little
Microsoft Windows IA (64-bit)        Little
Microsoft Windows x86 64-bit         Little
Solaris Operating System (x86)       Little
Solaris Operating System (x86-64)    Little
```

SQL*Net connectivity is required between source and target database

RMAN Compression can reduce the size and duration of the backup significantly

Pro tip: Most compression algorithms require Advanced Compression Option

👍

Secure your RMAN backup with
TDE Tablespace Encryption or RMAN Encryption

Pro tip: Requires Advanced Security Option

Using multisection backups is important in databases with bigfile tablespaces

Pro tip: The keyword `SECTION SIZE` controls the use of multisection backups

REDO APPLY

benchmark

Redo apply, TB/Day	11.2.0.4	12.1.0.2	12.2	MIRA 2x 12.2	MIRA 4x 12.2
Batch	57	57	57	115	226
OLTP	14	15	15	29	60

Source: Redo Apply Best Practices – Oracle Data Guard and Active Data Guard

How To Calculate The Required Network Bandwidth
Transfer Of Redo In Data Guard Environments (Doc ID 736755.1)

redo TRANSPORT AND APPLY benchmark

Connection, Gbps	11.2.0.4	12.1.0.2	12.2	MIRA 2x 12.2	MIRA 4x 12.2
Batch	57 / 6	57 / 6	57 / 6	115 / 11	226 / 22
OLTP	14 / 2	15 / 2	15 / 2	29 / 3	60 / 6

Source: Redo Apply Best Practices – Oracle Data Guard and Active Data Guard

👍

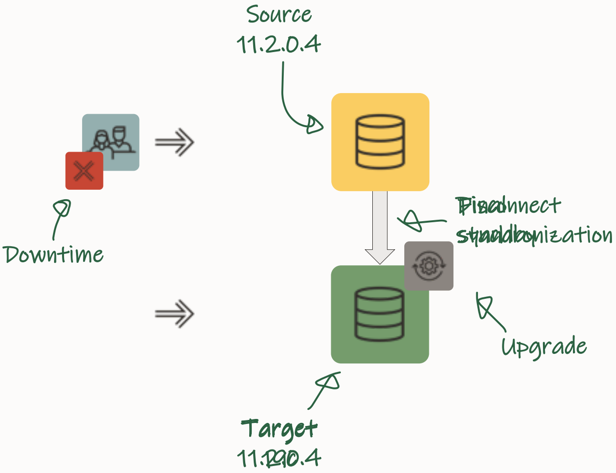
Consider compressing redo when using very slow connections

Pro tip: Requires Advanced Compression Option

What about upgrade?

Downtime is required to upgrade database after switchover

Data Guard | Upgrade



Data Guard | AutoUpgrade

SOURCE DATABASE	TARGET DATABASE
`java -jar autoupgrade.jar -mode analyze`	
DOWNTIME	
`alter system flush redo to ... confirm apply;`	
	`recover managed standby database cancel;`
	`recover standby database;`
	`alter database recover managed standby database finish;`
	`alter database activate physical standby database;`
	`alter database open;`
	`java -jar autoupgrade.jar -mode deploy`

Or use a Transient Logical Standby database for rolling upgrade using DBMS_ROLLING

Pro tip: Watch How Low Can You Go? Zero Downtime Operations for details

What about PDB conversion?

Convert to PDB after migration (and upgrade) using `noncdb_to_pdb.sql`

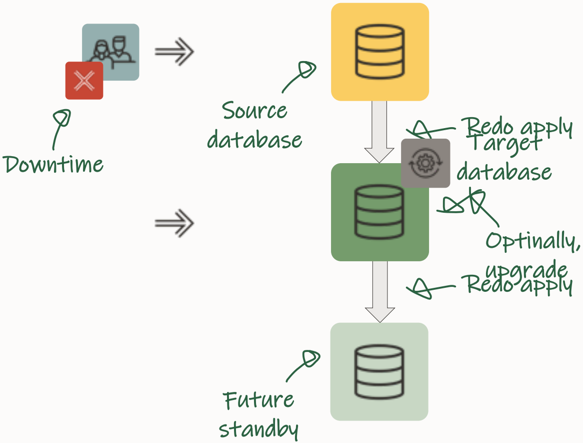
Can you offload the work from the source database?

👍 Yes, you can. Instantiate the standby database from a backup

Your target database must be protected by Data Guard?

👍 You can build the future
standby database in advance
and connect it as a cascading standby

Data Guard | Cascading Standby



Your target database must have a valid backup before go-live?

👍

Before go-live perform level 0 backup of the target database

👍 It works even if you upgrade the database

☒ It does not work if you also convert to PDB

Following a PDB conversion new backups of the data files are required before go-live

?

Your target database must be RAC?

👍 No problem, your standby database can be configured as a RAC database

How about your fallback plan?

Data Guard | Fallback

- To roll back (before go live):
  Source database is untouched
- To fall back (after go live):
  Switchover
- Unless database was upgraded:
  Downgrade
- Unless database was converted:
  Data Pump and GoldenGate

Data Guard | Important Notes

MOS Note: 273015.1
Migrating to RAC using Data Guard

MOS Note: 413484.1
DG Support for Heterogeneous Primary and Physical Standbys in Same DG Configuration

MOS Note: 1079563.1
RMAN DUPLICATE/RESTORE/RECOVER Mixed Platform Support

MOS Note: 2439602.1
Implement Standby in OCI (Bare Metal) and how to gather diagnostic information for any failure

MOS Note: 881421.1
Using Active Database Duplication to Create Cross Platform Data Guard Setup (Windows/Linux)

MOS Note: 1617946.1
Creating a Physical Standby using RMAN Duplicate (RAC or Non-RAC)

MOS Note: 1055938.1
Migrating from HP Oracle Database Machine to Sun Oracle Database Machine 11.2 using Data Guard

Redo Apply Best Practices - Oracle Data Guard and Active Data Guard

Using Data Guard as migration vehicle

Payback GmbH Germany
Exadata Migration Project 2012

Source: obs / Payback GmbH / Bild: Payback

Customer Case | Payback

Customer

Project 2012

Constraints

Preparation

Upgrade

Success?

Remarks

Payback GmbH

- Belongs to American Express
- HQ in Munich, Germany
- Develops and operates professional customer loyalty programs based on customized IT solutions

Customer Case | Payback

Customer

Project 2012

Constraints

Preparation

Upgrade

Success?

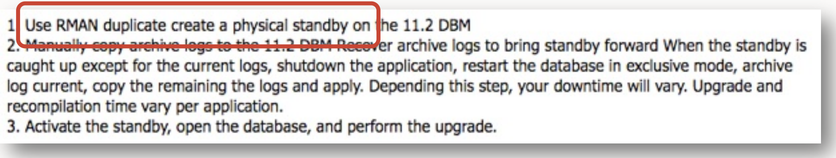
Remarks

Migrate 14TB from Exadata V1 to Exadata X2-2

Project timeline: 2 months including all tests

How to?

- MOS Note: 1055938.1
  Hardware and Oracle Migration using Data Guard (Case 2)



1. Use RMAN duplicate create a physical standby on the 11.2 DBM
2. Manually copy archive logs to the 11.2 DBM Recover archive logs to bring standby forward When the standby is caught up except for the current logs, shutdown the application, restart the database in exclusive mode, archive log current, copy the remaining the logs and apply. Depending this step, your downtime will vary. Upgrade and recompilation time vary per application.
3. Activate the standby, open the database, and perform the upgrade.

Customer Case | Payback

Customer

Project 2012

Constraints

Preparation

Upgrade

Success?

Remarks

Oracle 11.1.0.7 software must not be installed on Exadata X2-2

- Upgrading source Exadata to 11.2.0.3 not an option

Database 14TB

Downtime: less than 8hrs

Network "bottleneck"

- Remedy: Special IB cabled connection from V1 to X2-2

Customer Case | Payback

Customer

Project 2012

Constraints

Preparation

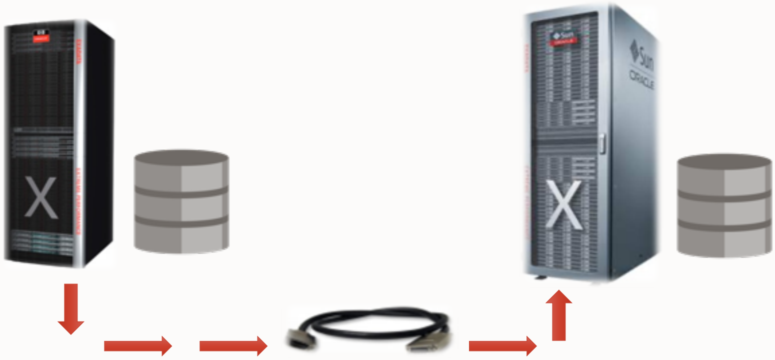
Upgrade

Success?

Remarks

Restoring 14TB with RMAN

- `DUPLICATE FOR STANDBY FROM ACTIVE DATABASE`



InfiniBand cable

Customer Case | Payback

Customer

Project 2012

Constraints

Preparation

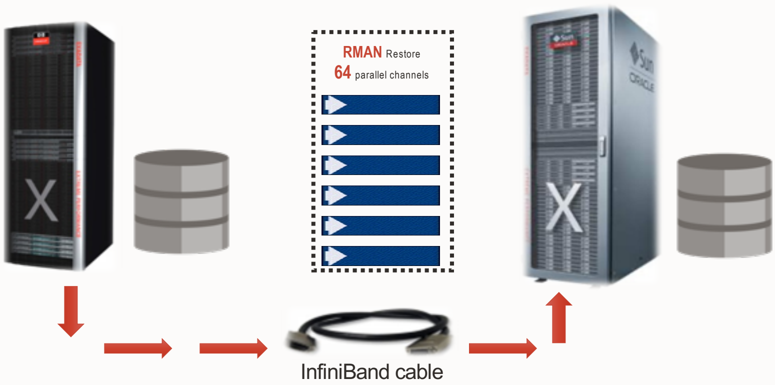
Upgrade

Success?

Remarks

Live upgrade/migration

- RMAN Restore and Recovery: <3 hours
  - 64 parallel RMAN channels allocated: >4TB/hour



Customer Case | Payback

Customer

Project 2012

Constraints

Preparation

Upgrade

Success?

Remarks

Database upgrade 11.1.0.7 ⇨ 11.2.0.3

- Used the **new PARALLEL UPGRADE** tool `catctl.pl` **as Beta customer**
  - Total database upgrade time including recompilation: 20 mins

Customer Case | Payback

Customer

Project 2012

Constraints

Preparation

Upgrade

Success?

Remarks

Live? And alive?

- Yes! Go-live on 3-JUL-2012
  - Almost three weeks earlier than proposed

- Total migration and upgrade time officially: ~4 hours
  - < 3 hours: Restore for Standby and recovery
  - < 20 mins: Database upgrade to Oracle 11.2.0.3
  - ~ 40 mins: Extra tasks (crsctl etc.)

Customer Case | Payback

Customer

Project 2012

Constraints

Preparation

Upgrade

Success?

Remarks

2020?

A few plans did change – but we were prepared ☺

- AWR and SQL Plan Management

Physical standby as migration vehicle was the key technique

- Allows several test runs
- Copy time does not account for downtime

Customer Case | Payback

Customer

Project 2012

Constraints

Preparation

Upgrade

Success?

Remarks

2020?

Today, Payback has many production databases on Oracle 19.8.0

Photo by Alternate Skate on Unsplash

Migration Strategies

Rolling Upgrades with Transient Logical Standby

Rolling Upgrade | Transient Logical Standby

Use a logical standby database to upgrade
with very little downtime.

The only downtime is as little
as it takes to perform a switchover.

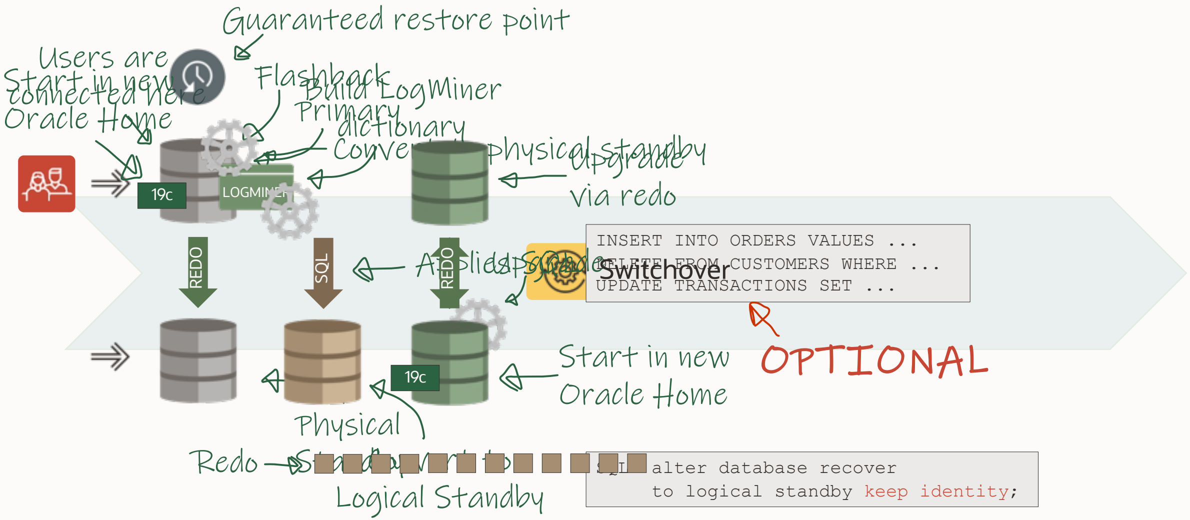
Pro tip: Also useful for other maintenance activities

Rolling Upgrade | Standby Types

PHYSICAL	LOGICAL
Redo apply	SQL apply
Updated by changing data block	Updated by executing SQLs
Exact copy - block-by-block	Copy - data is the same

Pro tip: Read more about standby types in [Data Guard Concepts and Administrations]

Rolling Upgrade | Concept



```
INSERT INTO ORDERS VALUES ...
DELETE FROM CUSTOMERS WHERE ...
UPDATE TRANSACTIONS SET ...
```

```
SQL> alter database recover
     to logical standby keep identity;
```

Rolling Upgrade | Options

MANUAL	DBMS_ROLLING
Part of Enterprise Edition	Requires Active Data Guard
Source must be 11.1.0.7	Source must be 12.1.0.2 or newer
Manual approach	Automated
Data Guard broker must be disabled	Data Guard broker can be enabled
	Recommended

Rolling Upgrade | Manual

MOS Note: 949322.1

Oracle11g Data Guard: Database Rolling Upgrade Shell Script

- Potentially not adjusted for Oracle 12c and newer
- Requires source is 11.2.0.3 or newer
- Does not work with Multitenant
- Not supported in 19c

Rolling Upgrade | Options

MANUAL	DBMS_ROLLING
Part of Enterprise Edition	Requires Active Data Guard
Source must be 11.1.0.7	Source must be 12.1.0.2 or newer
Manual approach	Automated
Data Guard broker must be disabled	Data Guard broker can be enabled
	Recommended

Rolling Upgrade | DBMS_ROLLING

6 SIMPLE STEPS

```
SQL> exec dbms_rolling.init_plan;
SQL> exec dbms_rolling.build_plan;
SQL> exec dbms_rolling.start_plan;
```

Upgrade database

```
SQL> exec dbms_rolling.switchover;
SQL> exec dbms_rolling.finish_plan;
```

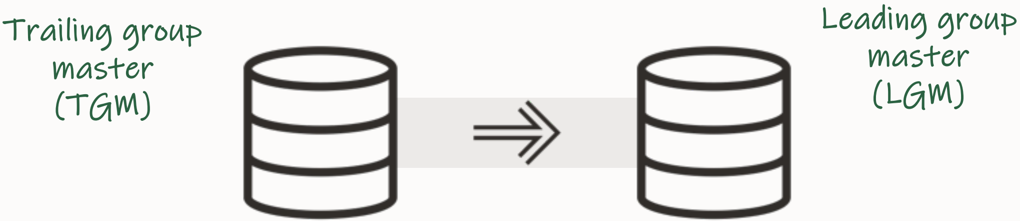
Rolling Upgrade | DBMS_ROLLING

```
...
Get current redo branch of the primary database
Wait until recovery is active on the primary's redo
branch
Reduce to a single instance if database is a RAC
Verify only a single instance is active if future
primary is RAC
Stop media recovery
Execute dbms_logstdby.build
Convert into a transient logical standby
Open database including instance-peers if RAC
Verify logical standby is open read/write
Get redo branch of transient logical standby
Get reset scn of transient logical redo branch
Configure logical standby parameters
Start logical standby apply
```

86 INSTRUCTIONS OR CHECKS

```
Stop logical standby apply
Start logical standby apply
Wait until apply lag has fallen below 600 seconds
Notify Data Guard broker that switchover to logical
standby database is starting
Log post-switchover instructions to events table
Switch database to a logical standby
Notify Data Guard broker that switchover to logical
standby database has completed
Wait until end-of-redo has been applied
...
```

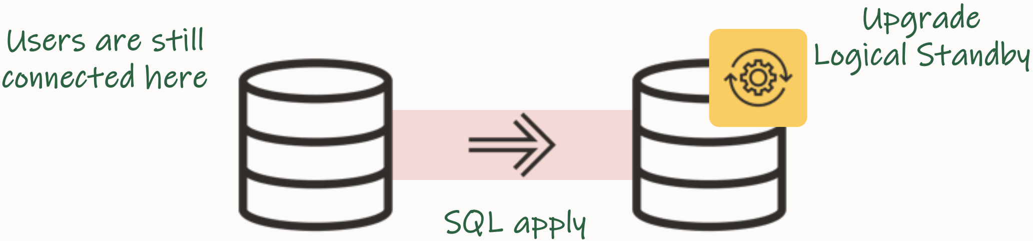
Rolling Upgrade | Demo



SID	PROD
DB_UNIQUE_NAME	PROD1
Host	bm1
Current primary database	

SID	PROD
DB_UNIQUE_NAME	PROD2
Host	bm2
Future primary database	

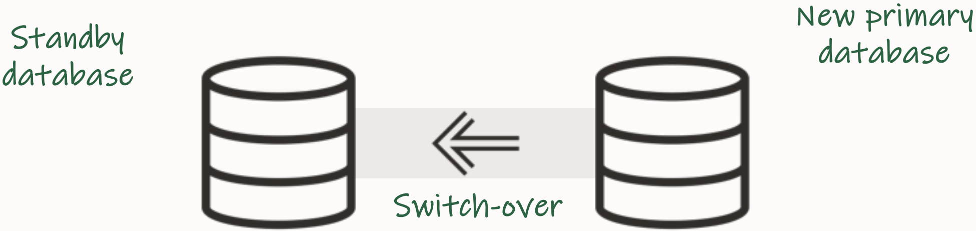
Rolling Upgrade | Demo



SID	PROD
DB_UNIQUE_NAME	PROD1
Host	bm1

SID	PROD
DB_UNIQUE_NAME	PROD2
Host	bm2

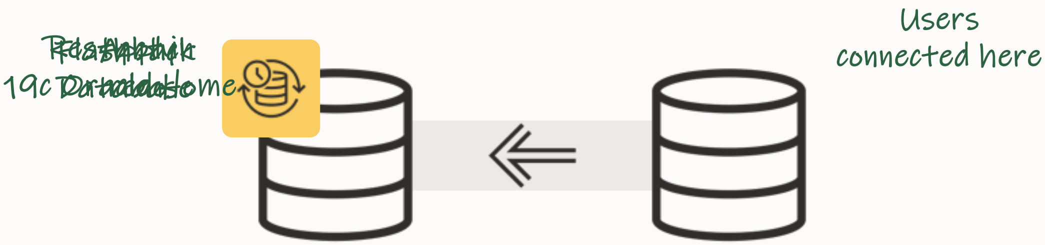
Rolling Upgrade | Demo



SID	PROD
DB_UNIQUE_NAME	PROD1
Host	bm1

SID	PROD
DB_UNIQUE_NAME	PROD2
Host	bm2

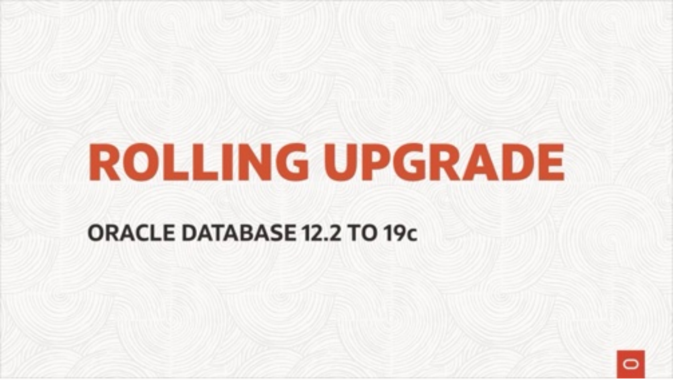
Rolling Upgrade | Demo



SID	PROD
DB_UNIQUE_NAME	PROD1
Host	bm1

SID	PROD
DB_UNIQUE_NAME	PROD2
Host	bm2

Rolling Upgrade | Demo



ROLLING UPGRADE

ORACLE DATABASE 12.2 TO 19c

Watch on YouTube

Rolling Upgrade | Backups

After converting to logical standby database, take a level 0 backup

Rolling Upgrade | Database Readiness

Can I use rolling upgrade on my database?

Rolling Upgrade | Database Readiness

Do not create the logical standby
on the same server as the primary database

Rolling Upgrade | Database Readiness

Supplemental logging is enabled automatically which introduces an overhead and increases the amount of redo generated

Pro tip: The change happens implicitly when the log miner dictionary is built

Rolling Upgrade | Database Readiness

When supplemental logging is enabled all DML cursors are invalidated

Rolling Upgrade | Database Readiness

☒ Not all data types and partitioning types are supported

Pro tip: Check the documentation for details

Rolling Upgrade | Database Readiness

Logical Standby Database supports most Oracle Text features

Some restrictions apply

Rolling Upgrade | Multitenant

- Rolling upgrade on container databases is fully supported
- Upgrade happens on CDB level - when you switchover - the entire CDB switches over
- The Transient Logical Standby can have a subset of the PDBs
- Adding new PDBs in primary after instantiating logical standby is possible, but cumbersome

Rolling Upgrade | Best Practices

Tips and tricks to ease your migration

Rolling Upgrade | Best Practice

For optimal performance all tables should have primary keys or unique keys

Pro tip: For further information, read Prerequisite Conditions for Creating a Logical Standby Database

Rolling Upgrade | Best Practice

Use `ALTER DATABASE GUARD` to prevent accidental changes on logical standby database

Rolling Upgrade | Best Practice

It is recommended to use three standby databases for maximum protection

Rolling Upgrade | Best Practice

Upgrade Grid Infrastructure to new release before you start the process

Rolling Upgrade | Best Practice

👍

Patch the source database
to the latest Release Update

Rolling Upgrade | Best Practice

Before starting rolling maintenance,
test your Data Guard config

Rolling Upgrade | Best Practice

👍 Plan your switchover to an off-peak period

Rolling Upgrade | Exadata Cloud Service

Step-by-step instructions in
Exadata Cloud Database 19c Rolling Upgrade With DBMS_ROLLING (Doc ID 2832235.1)

Rolling Upgrade | Additional Information - 1

Technical Briefs:

- Oracle Database Rolling Upgrades Using a Data Guard Physical Standby Database

Documentation:

- Oracle 19c Data Guard Concepts and Administration

MOS Notes:

- Transient Rolling Upgrade Using DBMS_ROLLING - Beginners Guide
- Rolling upgrade using DBMS_ROLLING - Complete Reference (Doc ID 2086512.1)
- MAA Whitepaper: SQL Apply Best Practices (Doc ID 1672310.1)
- Step by Step How to Do Swithcover/Failover on Logical Standby Environment (Doc ID 2535950.1)
- How To Skip A Complete Schema From Application on Logical Standby Database (Doc ID 741325.1)
- How to monitor the progress of the logical standby (Doc ID 1296954.1)
- How To Reduce The Performance Impact Of LogMiner Usage On A Production Database (Doc ID 1629300.1)

Rolling Upgrade | Additional Information - 2

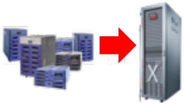
MOS Notes:

- Handling ORA-1403 ora-12801 on logical standby apply (Doc ID 1178284.1)
- Troubleshooting Example - Rolling Upgrade using DBMS_ROLLING (Doc ID 2535940.1)
- DBMS Rolling Upgrade Switchover Fails with ORA-45427: Logical Standby Redo Apply Process Was Not Running (Doc ID 2696017.1)
- SRDC - Collect Logical Standby Database Information (Doc ID 1910065.1)
- MRP fails with ORA-19906 after Flashback of Transient Logical Standby used for Rolling Upgrade (Doc ID 2069325.1)
- What Causes High Redo When Supplemental Logging is Enabled (Doc ID 1349037.1)
- Logical Standby SQL APPLY Tuning Tips (Doc ID 2674154.1)

Bugs:

- BUG 22541208 - REPLICATION FAILS WITH ORA-02149 DROPING PARTITION WITH SYSTEM GENERATED NAME (fixed in 12.2 backport available for 12.1)
- BUG 31412209 - TRANSIENT LOGICAL STANDBY UPGRADE FAILING WITH ORA-600[KRVXSAU_122_12202_LCR_OP] (fixed in 12.2 backport available for lower versions)

Rolling Upgrade | Nippon Steel & Sumitomo Metal

Benefits	Consolidati on of	Minimize planned	High performance
"Consolidating 4 Databases including Steel factory systems onto Exadata providing High performance and reliability, Enabling making use of High Quality of infrastructure."		5 minutes 	

Business Objectives

- High availability
- DB Infra consolidation

Solution

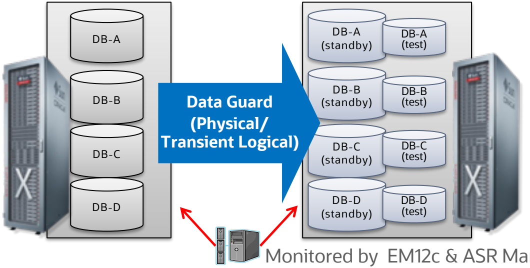
- Adopting Exadata providing high performance and high availability
- Rolling upgrade using Data Guard minimizes planned downtime

- Rolling upgrade using Transient Logical Standby realized minimizing downtime of upgrading DB (11.2.0.3→11.2.0.4)
- 5 minutes downtime x 2times (switchover) per 1 DB

- Mainframe migration
- Consolidating 4 DBs including Steel factory system & DWH onto Exadata

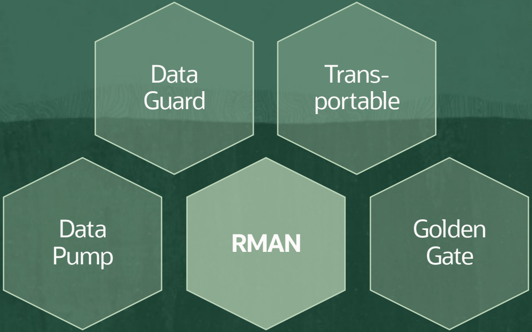
X3-2 Eighth (production)

X3-2 Eighth (Standby /Dev/ Test)



Monitored by EM12c & ASR Manager

Different MIGRATION techniques



Data Guard

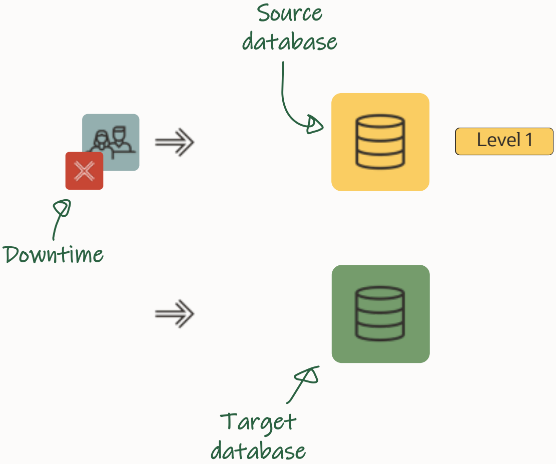
Trans-portable

Data Pump

RMAN

Golden Gate

Incremental | Concept



```
RMAN> BACKUP INCREMENTAL
      LEVEL 1 ...
```

```
RMAN> RECOVER DATABASE ...
```

Incremental | Benefits

- Simple and easy
- Well-known process
- Use existing backups
- Independent of file system,
  raw devices and ASM
- Some cross-platform capabilities

Incremental | Procedure

SOURCE DATABASE	TARGET DATABASE
`backup incremental level 0 database ... ;`	
	`restore database;`
`backup incremental level 1 database ... ;`	
	`recover database;`
DOWNTIME	
`backup incremental level 1 database ... ;`	
	`recover database;`
	`alter database open resetlogs;`

Incremental backups are useful when there is no SQL*Net connectivity between source and target

👍 Incremental backups are useful when source database release can't be installed on target host

Pro tip: Any release of RMAN can restore and recover a previous release backup

Block Change Tracking is recommended to speed up incremental backups

Pro tip: BCT is an Enterprise Edition feature, but requires Active Data Guard if enabled on standby database

RMAN Compression can significantly reduce the size and duration of the backup

Pro tip: Most compression algorithms require Advanced Compression Option

Secure your RMAN backup with
TDE Tablespace Encryption or RMAN Encryption

Pro tip: Requires Advanced Security Option

Using multisection backups is important in databases with bigfile tablespaces

Pro tip: The keyword `SECTION SIZE` controls the use of multisection backups

To recover the latest changes use an incremental backup or archive logs

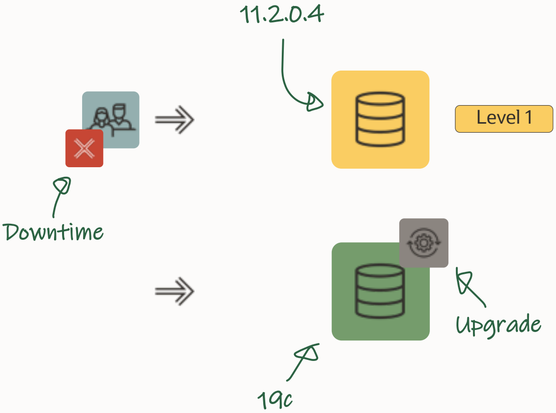
What about upgrade?

👍

RMAN can restore and recover from backups made in a previous release

Pro tip: The database must be opened immediately in `UPGRADE` mode and upgraded

Incremental | Upgrade



```
RMAN> BACKUP INCREMENTAL
      LEVEL 1 ...
```

```
RMAN> RECOVER DATABASE ...

SQL> ALTER DATABASE OPEN RESETLOGS UPGRADE;
```

Incremental | AutoUpgrade

SOURCE DATABASE	TARGET DATABASE
`backup incremental level 0 database ... ;`	
	`restore database;`
`java -jar autoupgrade.jar -mode analyze`	
DOWNTIME	
`java -jar autoupgrade.jar -mode analyze`	
`java -jar autoupgrade.jar -mode fixups`	
`backup incremental level 1 database ... ;`	
	`recover database;`
	`alter database open resetlogs upgrade;`
	`java -jar autoupgrade.jar -mode upgrade`

What about PDB conversion?

👍

Convert to PDB after migration (and upgrade) using `noncdb_to_pdb.sql`

Cloning a non-CDB directly
into a CDB (NON$CDB cloning)
is not recommended for large databases

Can you offload the work from the source database?

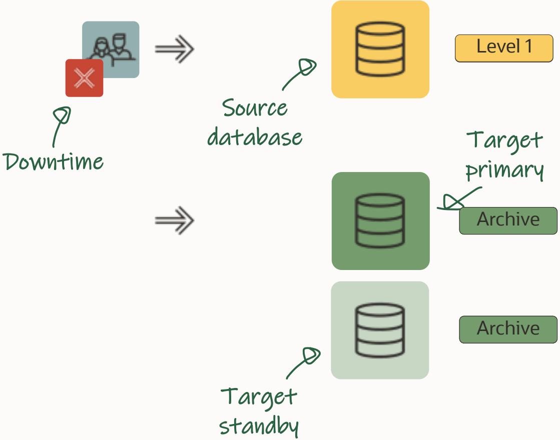
👍 Yes, you can perform the backups on a standby database

👍 Or simly re-use any backups that are being taken already

Your target database must be protected by Data Guard?

You can restore source data files on to future standby database in advance

Incremental | Data Guard



```
SQL> ALTER DATABASE OPEN RESETLOGS;
```

Configure:

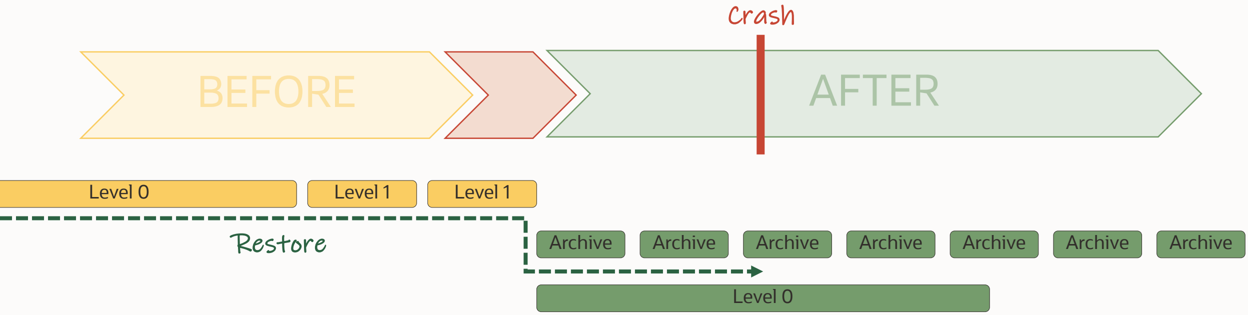
- redo transport
- Redo apply

```
RMAN> RECOVER DATABASE ...
```

Your target database must have a valid backup before go-live?

👍 The backup pieces used by the migration, can be used for disaster recovery as well

Incremental | Backup



It works even if you upgraded the database

☒ It does not work if you also converted to a PDB ℹ Following a PDB conversion, new backups of the data files are required before go-live

Your target database must be RAC?

👍

No problem, you can even restore a single instance to a RAC

To make recovery as easy as possible, use shared storage as much as possible

Incremental | RAC

- Backups on shared storage enables multi-instance recovery
- Recovery is easier with SPFile and password file on shared storage
- For encrypted databases also place keystore on shared storage

How about your fallback plan?

Incremental | Fallback

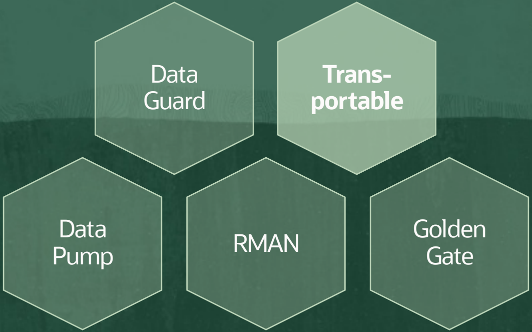
- To roll back (before go live):
  Source environment is preserved

- To fall back (after go live):
  Redo process in reverse order

- Unless database was upgraded:
  Downgrade

- Unless database was converted:
  Data Pump and GoldenGate

Different **MIGRATION** techniques



Data Guard

Transportable

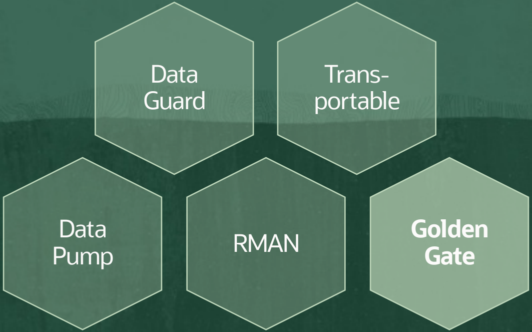
Data Pump

RMAN

Golden Gate

See separate decks for M5 and XTTS v4 Perl scripts

Different **MIGRATION** techniques



Data Guard

Trans-portable

Data Pump

RMAN

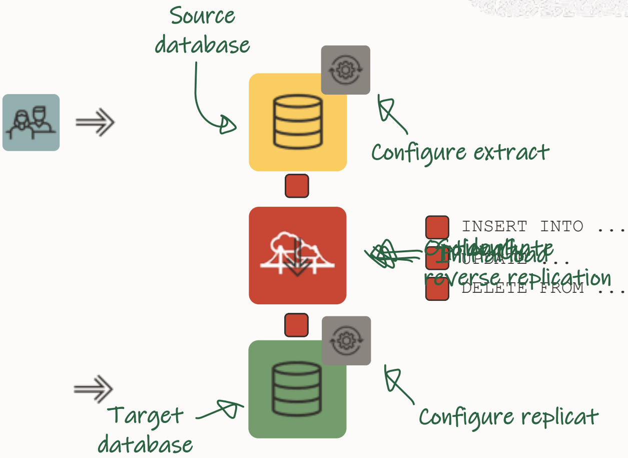
Golden Gate

GoldenGate | Overview

GoldenGate offers true zero downtime upgrades combined with excellent fallback capabilities and extreme flexibility

Pro tip: Active Data Guard included in GoldenGate license

GoldenGate | Concept



```
INSERT INTO ...
UPDATE ...
DELETE FROM ...
```

GoldenGate | Benefits

- True zero downtime
- Extremely flexible
- Cross-version and cross-architecture
- Cross-platform and cross-endian
- Test before go-live using Flashback Database (Doc ID 966212.1)

Pro tip: Active Data Guard included in GoldenGate license

GoldenGate | Architecture

- SQL*Net connection between databases
  - Alternatively, a distribution path between two GoldenGate instances

- Compress trail files to reduce network load

- Example: Database generating 10 TB redo
  - Trail files typically 30-40 % of redo
  - Compress at least 1:4, most likely up to 1:8
  - Result: 400 GB to 1000 GB trail files

GoldenGate | Initial load

If your initial load is a Data Pump import:

- Export or re-create public and other not exported objects
  - Synonyms
  - Database links
  - ...

- Diagnostic and tuning related information
  - AWR
  - SQL Plan Baselines
  - SQL Profiles
  - SQL Patches
  - ...

To strengthen security, you can encrypt the GoldenGate trail files

GoldenGate requires
database minimal supplemental logging
which does not impose a significant overhead

GoldenGate | Considerations

- Target database time zone file version must be equal to or higher than source

```
SQL> select * from v$timezone_file;
```

- Possibly patches are recommended on source database to support GoldenGate
  - 11g
  - 12c and newer
- DDL replication
  - Truncate
  - Sequences

GoldenGate | Data Pump Integration

No longer needed to specify from which SCN replication should start

New Replicat parameter
`DBOPTIONS ENABLE_INSTANTIATION_FILTERING`

Requires Oracle GoldenGate 12.2

MOS Note: 1276058.1

```
SQL> select source_object_name,
instantiation_scn from
dba_apply_instantiated_objects where
source_object_owner = 'APPS' ;

SOURCE_OBJECT_NAME INSTANTIATION_SCN
------------------------------------
TCUSTMER                   829723224
TCUSTORD                   829723223




2017-07-17 15:02:51 INFO OGG-10155
Instantiation CSN filtering is enabled
on table APPS.TCUSTMER at CSN
829,723,224.

2017-07-17 15:02:51 INFO OGG-10155
Instantiation CSN filtering is enabled
on table APPS.TCUSTORD at CSN
829,723,223.
```

GoldenGate | Network recommendations

Running Oracle GoldenGate remotely

Network round trip ping time:

Extract	less than 80 ms
Replicat	less than 5 ms

Bandwidth:

Integrated Extract - only the changes to tables that are being captured will be sent to the Extract process itself

GoldenGate | Database Readiness

Can you use GoldenGate on your database?

GoldenGate | Database Readiness

```
SQL> select * from dba_goldengate_support_mode;

OWNER OBJECT_NAME          SUPPORT_MODE
CO    CUSTOMERS            ID KEY
CO    ORDERS               ID KEY
CO    ORDER_ITEMS          FULL
CO    PRODUCTS             ID KEY
CO    STORES               ID KEY
```

GoldenGate | Database Readiness

	✓	ℹ	✗
SUPPORT_MODE	FULL	ID KEY PLSQL	INTERNAL NONE

Pro tip: Visit the documentation for more details

GoldenGate | Database Readiness

	✓	ℹ	✗
SUPPORT_MODE	FULL	ID KEY PLSQL	INTERNAL NONE

Pro tip: Visit the documentation for more details

GoldenGate | Database Readiness

	✓	ℹ	✗
SUPPORT_MODE	FULL	ID KEY PLSQL	INTERNAL NONE

Pro tip: Visit the documentation for more details

GoldenGate | Database Readiness

What's wrong in this Oracle Database running 12.2?

Identify columns supported as of Oracle Database 18c ...

```
SQL> select * from dba_goldengate_support_mode;

OWNER OBJECT_NAME          SUPPORT_MODE
CO    CUSTOMERS            ID KEY
CO    ORDERS               ID KEY
CO    ORDER_ITEMS          FULL
CO    PRODUCTS             ID KEY
CO    STORES               ID KEY
```

GoldenGate | Database Readiness

Oracle Database 21c New Feature

```
SQL> select * from dba_goldengate_support_mode;

OWNER OBJECT_NAME        SUPPORT_MODE       DESCRIPTION
CO    CUSTOMERS          ID KEY             A very good explanation
CO    ORDERS             ID KEY             Another good explanation
CO    ORDER_ITEMS        FULL
CO    PRODUCTS           ID KEY             A third explanation
CO    STORES             ID KEY             Good explanation comes in
                                            abundance these days
```

GoldenGate | Database Readiness

For optimal performance all tables should have primary keys or unique keys

GoldenGate | Database Readiness

```
SQL> select * from dba_goldengate_not_unique;

OWNER  TABLE_NAME                         BAD_COLUMN
IX     AQ$_ORDERS_QUEUETABLE_L            N
IX     AQ$_STREAMS_QUEUE_TABLE_L          N
SH     SALES                              N
SH     COSTS                              N
SH     SUPPLEMENTARY_DEMOGRAPHICS         N
SH     CAL_MONTH_SALES_MV                 N
SH     FWEEK_PSCAT_SALES_MV               N
```

GoldenGate | Database Readiness

If the application maintains uniqueness,
but it is not enforced on the database,
use a `KEYCOLS` clause to let GoldenGate use it

Pro tip: For further information, read [Ensuring Row Uniqueness in Source and Target Tables]()

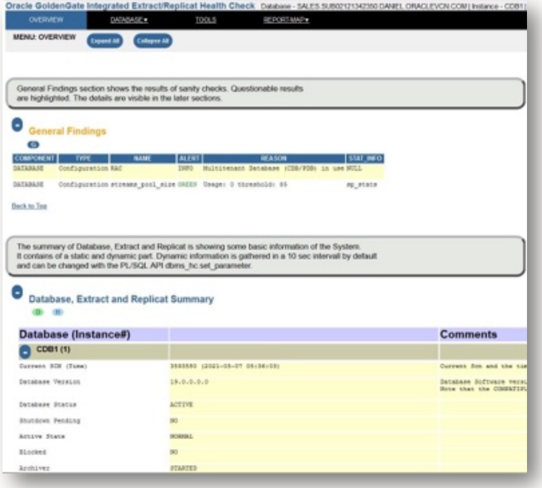
GoldenGate | Database Readiness

GoldenGate requires
database minimal supplemental logging
which does not impose a significant overhead

GoldenGate | Health Check

Generate report:

- Check prerequisites
- Database characteristics
- Find database objects of interest
- Extract/replicat statistics
- Check database readiness



Oracle GoldenGate Integrated Extract/Replicat Health Check Database - SALES.SUB02121342350.DANIEL.ORACLEVCN.COM | Instance - CDB1 |

General Findings section shows the results of sanity checks. Questionable results are highlighted. The details are visible in the later sections.

General Findings

COMPONENT	TYPE	NAME	ALERT	REASON	STAT_INFO
DATABASE	Configuration	RAC	INFO	Multitenant Database (CDB/PDB) in use	NULL
DATABASE	Configuration	streams_pool_size	GREEN	Usage: 0 threshold: 85	sp_state

Back to Top

The summary of Database, Extract and Replicat is showing some basic information of the System.
It contains of a static and dynamic part. Dynamic information is gathered in a 10 sec intervall by default
and can be changed with the PL/SQL API dbms_hc.set_parameter.

Database, Extract and Replicat Summary

Database (Instance#)		Comments
CDB1 (1)		
Current SCN (Time)	3593580 (2021-05-07 05:36:03)	Current Scn and the tim
Database Version	19.0.0.0.0	Database Software versi More than the COMPATIBL
Database Status	ACTIVE	
Shutdown Pending	NO	
Active State	NORMAL	
Blocked	NO	
Archiver	STARTED	

GoldenGate | Health Check

Generate report by:

- Installing objects in database: `ogghc_install.sql`
- Execute health check: `ogghc_run.sql`
- Optionally, clean-up objects: `ogghc_uninstall.sql`

For GoldenGate MicroServices Architecture find the scripts:

```
/u01/app/ogg/oraclenn/lib/sql/healthcheck
```

How about upgrading?

GoldenGate can extract from one release, and replicate into another

Pro tip: You can even migrate from very old releases using multiple instances of GoldenGate

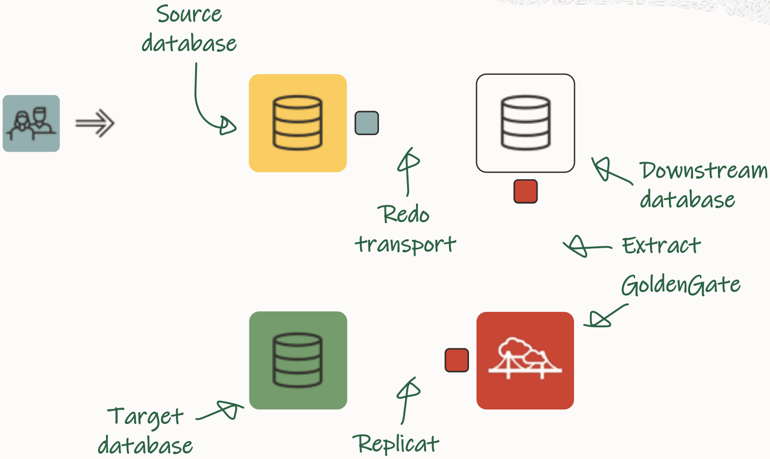
How about PDB conversion?

GoldenGate can replicate
from non-CDB directly into a PDB

Can I offload the work
from the source database?

👍 Yes, you can extract from a *downstream* database

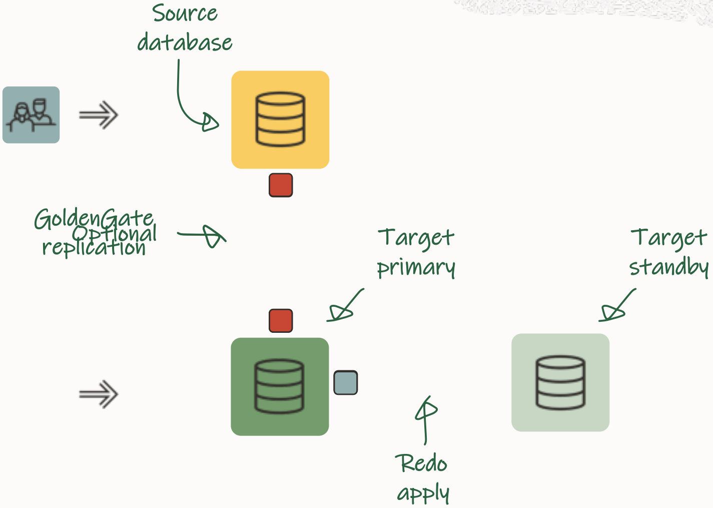
GoldenGate | Downstream



? Your target database must be protected by Data Guard?

👍 After the initial load on the target database, start building your Data Guard

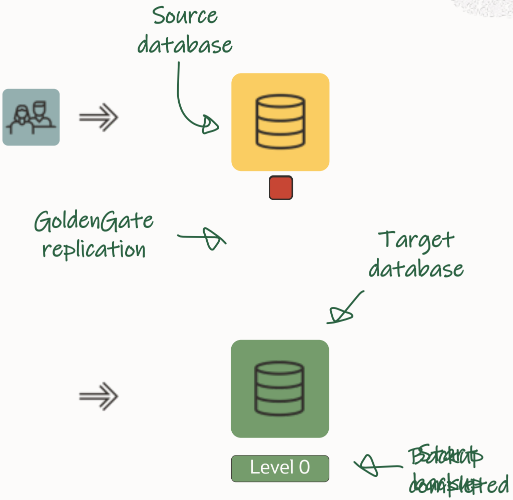
GoldenGate | Data Guard



Your target database must
have a valid backup before go-live?

👍 Perform and verify your backups
after initial load, but before switchover

GoldenGate | **Backup**

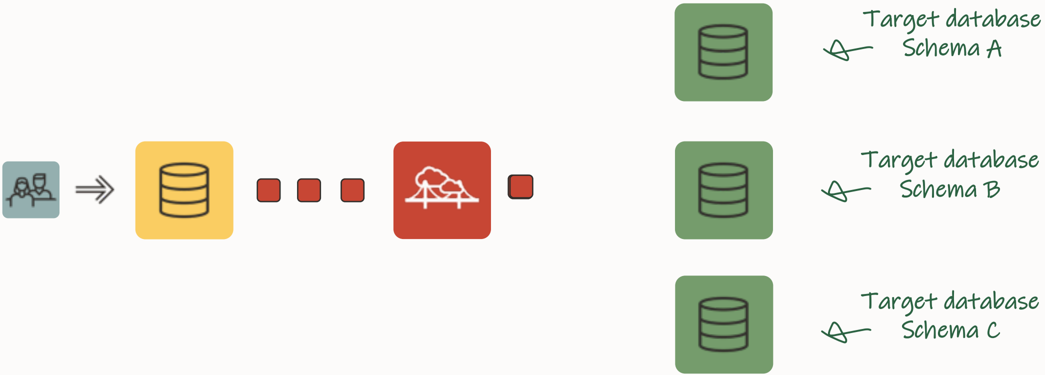


Your target database must be RAC?

👍 No problem, GoldenGate can extract from and replicate to a RAC database

How about schema isolation?

GoldenGate | Schema Isolation



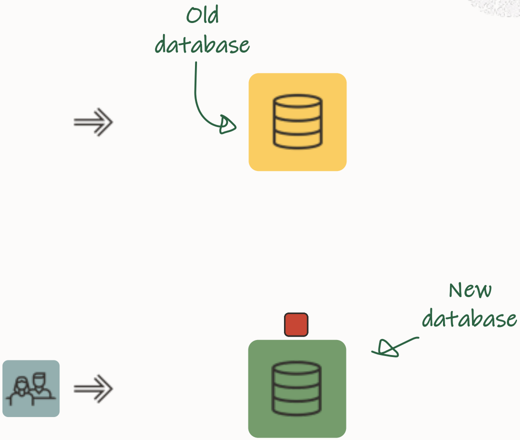
GoldenGate | Customization

You can customize the target schema even more:

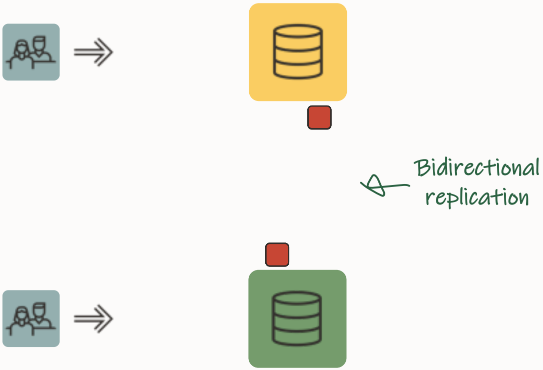
- Replicate from schema A in source to schema B in target (schema rename)
- Replicate data from non-partitioned to partitioned table
- Replicate from one character set to another
- Replicate from fragmented table to compacted table

How about your fallback plan?

GoldenGate | Fallback



GoldenGate | Fallback



GoldenGate | Technical Briefs

Oracle Database Migration with an Oracle GoldenGate Hub Configuration

Zero Downtime Database Upgrade Using Oracle GoldenGate

Oracle GoldenGate with Oracle RAC Configuration Best Practices

Transparent Role Transitions With Oracle Data Guard and Oracle GoldenGate

OCI GoldenGate | Cloud Native

New Cloud Native service: OCI GoldenGate

Runs GoldenGate 21c, managed by Oracle

Auto-scale: true cloud elasticity, low operations cost

Very attractive pricing

Supports:

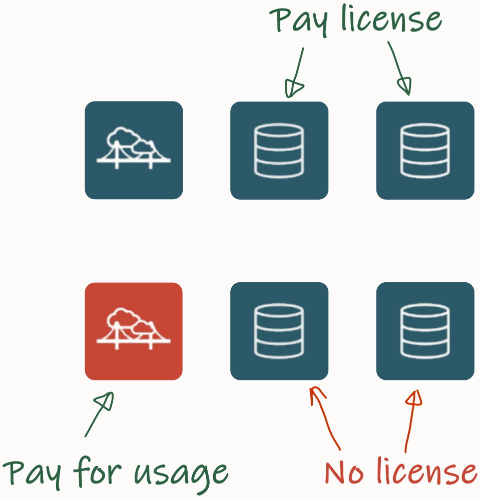
- Oracle Database 11.2.0.4 and higher

Pro Tip: Watch a short intro on YouTube

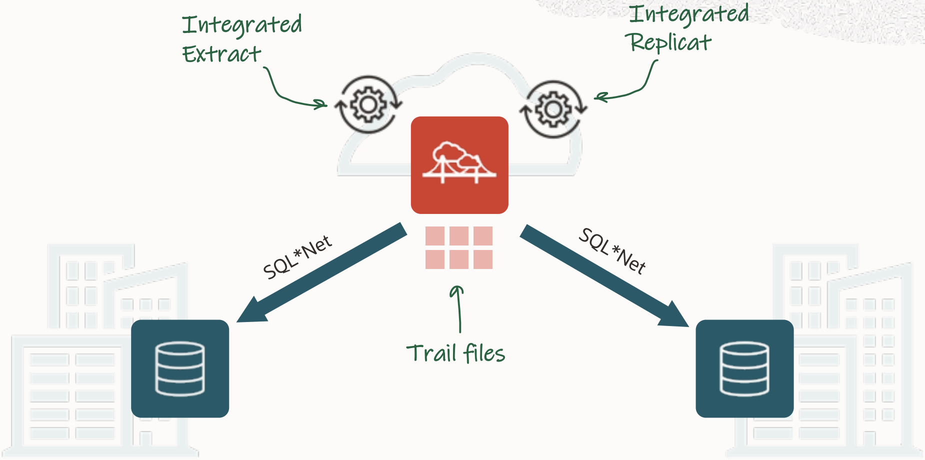
OCI GoldenGate | Pricing

Traditional

OCI GoldenGate



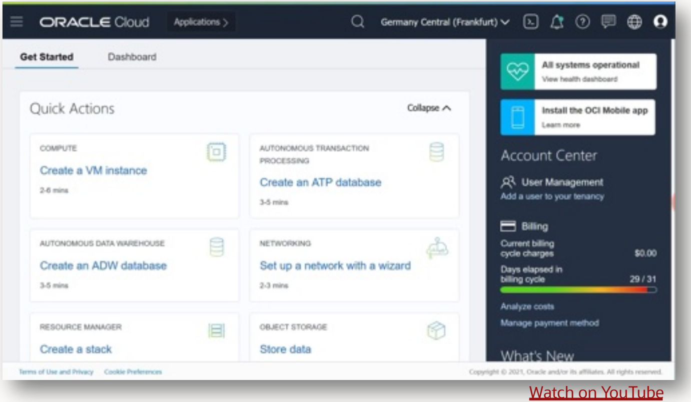
OCI GoldenGate | Architecture



OCI GoldenGate | Overview

	GoldenGate	GG OCI Marketplace	OCI GoldenGate
Solution Management			
Create and Manage GoldenGate Deployments	<-------- customer responsibility -------->		
Platform Services			
Oracle Cloud Automations	*Not Available*	*Not Available*	Oracle Managed
Automatic Scaling (up to 3x)			
OCI Monitoring / Service Telemetry			
Metering and Billing per second			
Full REST API for Control Plane and Data Plane	Customer Managed	Customer Managed	
Disaster Recovery, Backup and Restore			
Upgrades and Patching			
Private Endpoints and Secure Vault			
Wallet Integration w/Autonomous DB			
Operating System Administration			
Infrastructure Management			
Virtualization & Terraform Stack Automation	Customer Provided	Oracle Provided	Oracle Provided
Install / Rapid Provisioning			
Server Administration			
Storage and Durability Guarantees			
Core Networking			

OCI GoldenGate | Cloud Native



Quick Actions

COMPUTE
Create a VM instance
2-6 mins

AUTONOMOUS TRANSACTION PROCESSING
Create an ATP database
3-5 mins

AUTONOMOUS DATA WAREHOUSE
Create an ADW database
3-5 mins

NETWORKING
Set up a network with a wizard
2-3 mins

RESOURCE MANAGER
Create a stack

OBJECT STORAGE
Store data

All systems operational
View health dashboard

Install the OCI Mobile app
Learn more

Account Center

User Management
Add a user to your tenancy

Billing

Current billing cycle charges	$0.00
Days elapsed in billing cycle	29 / 31

Analyze costs
Manage payment method

What's New

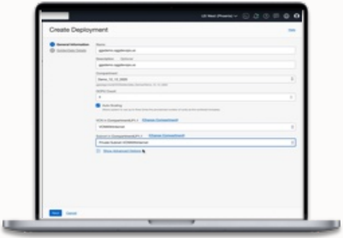
Terms of Use and Privacy Cookie Preferences

Copyright © 2021, Oracle and/or its affiliates. All rights reserved.

Watch on YouTube

OCI GoldenGate | Start small, grow to massive scale

Get started for
$1.34 per OCPU per hour



Create Deployment

Try it for free:
https://www.oracle.com/cloud/free/

GoldenGate moves petabytes of real-time data per day at Web scale

84%

of Fortune 100 use GoldenGate

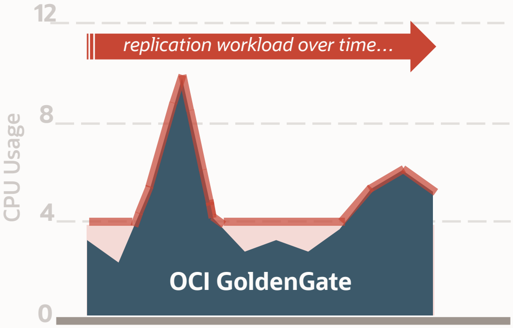
OCI GoldenGate | Auto scaling

Same experience as Autonomous Database

- Choose a base size
- Turn on auto scale feature
- Automatic 3x scaling factor

Pay only for what you use

- Scaling happens online / no downtime
- Per-second billing



Dynamic Auto-Scale

Automatically scale up to 3X
Scaling with zero downtime

OCI GoldenGate | Recommended sizing

Development / Trials

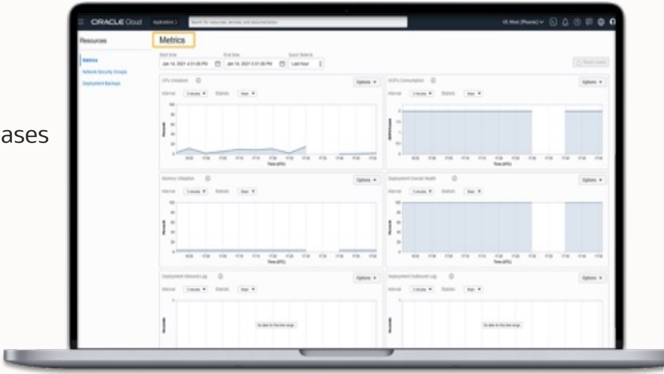
- Start with 1 OCPU and Auto-scale on

Typical production use cases

- 4 OCPU with Auto-scale, covers 80% of use cases
- YMMV – can cover >60GB/hr of DB Redo

Extreme scale and performance

- 8 OCPU with Auto-scale to 24 OCPU
- Up to:
    - 24 GB of memory
    - 24 Gbps network
    - 6TB of storage



Easily manage OCI-GG deployments from your console

OCI GoldenGate | On-Prem

OCI GoldenGate and on-prem databases?

OCI GoldenGate | On-Prem

Probably not ...

Unless you have a lightning-fast connection and your database is physically close to OCI

OCI GoldenGate | Network recommendations

Running Oracle GoldenGate remotely

Network round trip ping time:

Extract	less than 80 ms
Replicat	less than 5 ms

Bandwidth:

Integrated Extract - only the changes to tables that are being captured will be sent to the Extract process itself

GoldenGate | Additional Resources

Certifications

GoldenGate 19.1: Using Oracle GoldenGate on Oracle Cloud Marketplace

OCI Marketplace: Oracle GoldenGate for Oracle

Oracle GoldenGate Best Practices: Instantiation from an Oracle Source Database (Doc ID 1276058.1)

Effects of ADD TRANDATA and ADD SCHEMATRANDATA on an Oracle databases' Supplemental Logging (Doc ID 2070331.1)

GoldenGate | Technical Briefs

Oracle Database Migration with an Oracle GoldenGate Hub Configuration

Zero Downtime Database Upgrade Using Oracle GoldenGate

Oracle GoldenGate with Oracle RAC Configuration Best Practices

Transparent Role Transitions With Oracle Data Guard and Oracle GoldenGate

ORACLE 12c

Zero Downtime Database Upgrade Using Oracle GoldenGate

ORACLE

Migration with Oracle GoldenGate

Amadeus
OOW Presentation 2012

Source: https://amadeus.com/images/en/amadeus/offices/amadeus-madrid-sign.jpg

Customer Case | Amadeus

Customer

Project 2012

Constraints

Preparation

Migration

Success?

Remarks

Amadeus is a leading transaction processor for the global travel and tourism industry

DISTRIBUTION BUSINESS

711 airlines
110,000+ hotel properties
30 car rental companies
50+ cruise and ferry lines
207 tour operators
24 insurance companies
95 railways

IT SOLUTIONS

Inventory
Departure Control
e-Commerce

Airlines
Airports
Hotels
Rail

20,000+ tx/sec (peak)
< 0.3 sec response time
10 Petabytes of storage
3+ million net bookings/day
> 1 billion tx/day

** All numbers are from 2012*

Customer Case | Amadeus

Customer

Project 2012

Constraints

Preparation

Migration

Success?

Remarks

Migrate Oracle 10g production databases to Oracle 11g

Migrate to new HW and/or OS platform

Source		Target
Oracle 10.2.0.3 RAC HPUX v2	→	Oracle 11.2.0.2/3 RAC HPUX v3
	→	Oracle 11.2.0.2/3 RAC RHE Linux
Oracle 10.2.0.3 Single Instance HPUX v2	→	Oracle 11.2.0.2/3 RAC One RHE Linux

Customer Case | Amadeus

Customer

Project 2012

Constraints

Preparation

Migration

Success?

Remarks

Fixed quarterly outage windows

Maximum of 5 minutes database downtime

No service impact outside the outage window

Endian change: HP-UX ⇨ to Linux (big ⇨ little endian)

Possibility of fallback during and after the outage

High volume of DB changes (redo of up to 20MB/sec)

Large database sizes (up to 14TB)

Possibility for physical re-organization

- Fresh data dictionary
- Tablespace and partitioning redesign

Customer Case | Amadeus

Customer

Project 2012

Constraints

Preparation

Migration

Success?

Remarks

In-depth proof of concept (supported by Oracle)

- Focusing on functional aspects
- Focusing on data volume

Standardized migration process model with timeline

Home-made scripts and procedures to support setup, monitoring, tuning and switch over

Training of in-house specialist supporting the DBAs

Customer Case | Amadeus

Customer

Project 2012

Constraints

Preparation

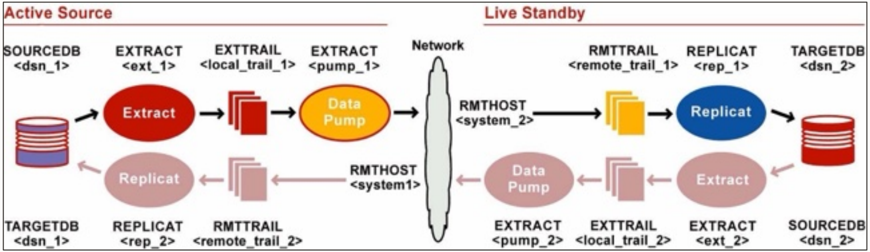
Migration

Success?

Remarks

Instantiation of new 11g database: expdp from Physical Standby

Installation, configuration, tuning of GG replication



Comparison of source/target DB content (Veridata)

Rehearsals of switch over and fallback

Switch over: Stop replication / Start reverse-replication

Customer Case | Amadeus

Customer

Project 2012

Constraints

Preparation

Migration

Success?

Remarks

15 databases successfully migrated in phase 1 (*Oct 2012*)

Source	Target	Migrated
Oracle 10.2.0.3 RAC HPUX v2	Oracle 11.2.0.2/3 RAC HPUX v3	6 ✓
	Oracle 11.2.0.2/3 RAC RHE Linux	3 ✓
Oracle 10.2.0.3 Single Instance HPUX v2	Oracle 11.2.0.2/3 RAC One RHE Linux	6 ✓

- Switchover duration: 2-6 minutes
- No fallback performed

Customer Case | Amadeus

Customer

Project 2012

Constraints

Preparation

Migration

Success?

Remarks

The concept proved to handle a smooth and secure migration across different DB versions and HW/OS platforms

To be considered …

- Instantiation of target database (incl. Plan Stability)
- Customized GG setup per database
- Handling of unsupported data types (e.g., ANYDATA)
- Impact of supplemental logging on source DB
- Effort of tuning GG for DBs with high DML rate (e.g., parallel replicate processes)

Oracle GoldenGate | Further Information

WP: Zero Downtime Database Upgrade Using Oracle GoldenGate
https://www.oracle.com/technetwork/middleware/goldengate/overview/ggzerodowntimedatabaseupgrades-174928.pdf

MOS Note: 1448324.1
GoldenGate Integrated Capture and Integrated Replicat Healthcheck Script

MOS Note: 2193391.1
Latest GoldenGate/Database (OGG/RDBMS) Patch recommendations

Complete Database Profile OGG readiness check

- MOS Note:1298562.1:
  Oracle GoldenGate database Complete Database Profile check script for Oracle DB (All Schemas) Classic Extract

Check OGG readiness for Schema Only

- MOS Note: 1296168.1
  Oracle GoldenGate database Schema Profile check script for Oracle DB

Oracle Streams | Desupport

> *Starting in Oracle Database 19c (19.1), Oracle Streams is desupported. Oracle GoldenGate is the replication solution for Oracle Database.*

Database 19c Upgrade Guide

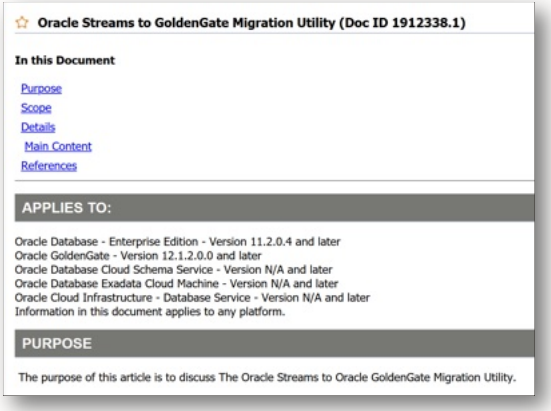
- Oracle Database Advanced Queuing is not deprecated
- Fully supported in Oracle Database 19c

Oracle Streams | GoldenGate Migration

How to migrate from Streams to GoldenGate

[Oracle Streams to Oracle GoldenGate Conversion (Doc ID 1383303.1)]

[Oracle Streams to GoldenGate Migration Utility (Doc ID 1912338.1)]



☆ **Oracle Streams to GoldenGate Migration Utility (Doc ID 1912338.1)**

In this Document

- Purpose
- Scope
- Details
  - Main Content
- References

APPLIES TO:

Oracle Database - Enterprise Edition - Version 11.2.0.4 and later
Oracle GoldenGate - Version 12.1.2.0.0 and later
Oracle Database Cloud Schema Service - Version N/A and later
Oracle Database Exadata Cloud Machine - Version N/A and later
Oracle Cloud Infrastructure - Database Service - Version N/A and later
Information in this document applies to any platform.

PURPOSE

The purpose of this article is to discuss The Oracle Streams to Oracle GoldenGate Migration Utility.

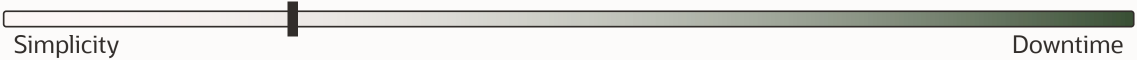
Summary

Further Information

Different **MIGRATION** techniques

	Data Pump	Data Guard	Transient Logical Standby	RMAN duplicate	RMAN Incr.	TTS	FTEX	FTEX Incr.	Golden Gate
Simplicity	Simple	Simple	Complex	Simple	Simple	Complex	Moderate	Moderate	Complex
Downtime	Significant	Near Zero	Near Zero	Significant	Low	Significant	Significant	Low	Zero
Version Change	+		+			+	+	+	+
Same-Endianness OS Change	+	(+)	(+)	(+)	(+)	+	+	+	+
Big/Little Endianness OS Change	+					+	+	+	+
Same Hardware	+	+		+	+	+	+	+	+
Hardware Exchange	+	+	+	+	+	+	+	+	+
non-CDB to CDB/PDB	+					+	+	+	+
Encrypt	+	+	+	+	+				+
Fallback After Go-Live / Upgrade	+								+
Character Set Change	+								+

Migration | Keep It Simple



Simplicity

Downtime

Performance Stability | After Migration

Performance Stability

Tips, Tricks & Underscores - Thursday 4 March 2021

ORACLE

Performance Stability

Tips, Tricks & Underscores

Roy Swonger, Mike Dietrich & Daniel Overby Hansen

Database Upgrade, Utilities and Patching

Watch the recording

Get the slides

Photo by Dušan veverkolog on Unsplash

Want to Know More?

Webinar: Migration Strategies – Insights, Tips and Secrets

Recording

Slides

Chapter 7
Cool Features

Photo by Jason Blackeye on Unsplash

Spatial and Graph + Machine Learning

Oracle Spatial and Graph (OSG) included with Oracle Database

Oracle Machine Learning (Advanced Analytics) include as well

- Since Dec 5, 2019
- Valid for all editions, includes Enterprise Edition, SE2 and DBCS
- Applies to all database versions back to Oracle 11.2.0.4

- https://mikedietrichde.com/2019/12/06/great-license-news-spatial-and-graph-machine-learning/

Extended VARCHAR2

Extended VARCHAR2 | Overview

NEW IN
12.1

Available since Oracle 12.1

Requires `COMPATIBLE=12.1.0` or higher

Needs to be turned on explicitly

VARCHAR2 and NVARCHAR2 columns can fit up to 32k bytes

- Allows length >4000 byte

Irreversible

Parameter:

- `MAX_STRING_SIZE=EXTENDED`
- Can be used on PDB level

Extended VARCHAR2 | How to?

NEW IN
12.1

Step by step:

```
ALTER SYSTEM set MAX_STRING_SIZE=EXTENDED scope=SPFILE;
```

```
SHUTDOWN IMMEDIATE
STARTUP UPGRADE
```

```
@?/rdbms/admin/utl32k.sql
```

```
SHUTDOWN IMMEDIATE
STARTUP
```

```
CREATE TABLE applicants
  (id            NUMBER GENERATED AS IDENTITY,
   first_name    VARCHAR2(30),
   last_name     VARCHAR2(30),
   application   DATE,
   cv            VARCHAR2(32767)
 );
```

Extended VARCHAR2 | The Fine Print

NEW IN 12.1

Caution!

- For new tables:
  - Up to 3964 bytes will be stored in a regular VARCHAR2
  - Above this limit, data will be stored in an inline SecureFile LOB
    - See: http://www.ludovicocaldara.net/dba/extended-data-types-storage/
    - `_scalar_type_lob_storage_threshold=4000` by default
    - Inline SecureFile LOBs does not support `NOLOGGING`
- For existing tables:
  - Row chaining
  - Workaround: `DBMS_REDEFINITION` or Online Table Move

Extended VARCHAR2 | The Fine Print

NEW IN 12.1

Performance

- Extended VARCHAR2 can save roundtrips
- See: https://blog.dbi-services.com/12c-extended-datatypes-better-than-clob/

```
SELECT * FROM TEST_CLOB_COLUMN;

Statistics
----------------------------------------------
    5  recursive calls
    0  db block gets
  136  consistent gets
   80  physical reads
    0  redo size
16310  bytes sent via SQL*Net to client
11890  bytes received via SQL*Net from client
   52  SQL*Net roundtrips to/from client
    0  sorts (memory)
    0  sorts (disk)
   10  rows processed
```

```
SELECT * FROM TEST_VARCHAR2_9000;

Statistics
----------------------------------------------
    4  recursive calls
    0  db block gets
   28  consistent gets
    0  physical reads
    0  redo size
90721  bytes sent via SQL*Net to client
  380  bytes received via SQL*Net from client
    2  SQL*Net roundtrips to/from client
    0  sorts (memory)
    0  sorts (disk)
   10  rows processed
```

Extended VARCHAR2 | The Fine Print

NEW IN 12.1

Potential pitfall

- `COMPATIBLE < 19.0.0`
  - SecureFile LOB requires at least 16 blocks in an extent
- `COMPATIBLE ≥ 19.0.0`
  - SecureFile LOB requires at least 32 blocks in an extent

Example:

- DB_BLOCK_SIZE: 16K
- Tablespace uniform extent size: 128k

  ➔ One extents contains only 8 blocks

    ➔ No SecureFile LOB can be created

Online Data File Move

Online Data File Move | Overview

NEW IN 12.1

Rename:

```
SQL> ALTER DATABASE
     MOVE DATAFILE '/u01/oracle/rbdb1/user1.dbf'
     TO            '/u01/oracle/rbdb1/user01.dbf';
```

Relocate to ASM:

```
SQL> ALTER DATABASE
     MOVE DATAFILE '/u01/oracle/rbdb1/user1.dbf'
     TO            '+DATA';
```

Pro tip: Works for `SYSTEM`, `UNDO` and `SYSAUX` as well.

Online Data File Move | Overview

NEW IN
12.1

Generate OMF name:

```
SQL> ALTER DATABASE
     MOVE DATAFILE 12;
```

Pro tip: On 12.1 and 12.2 be aware of bug 24836489:
DATAFILES ARE CREATED WRONG LOCATION IN OMF DEFINED PDB DATABASE

Online Data File Move | Overview

NEW IN 12.1

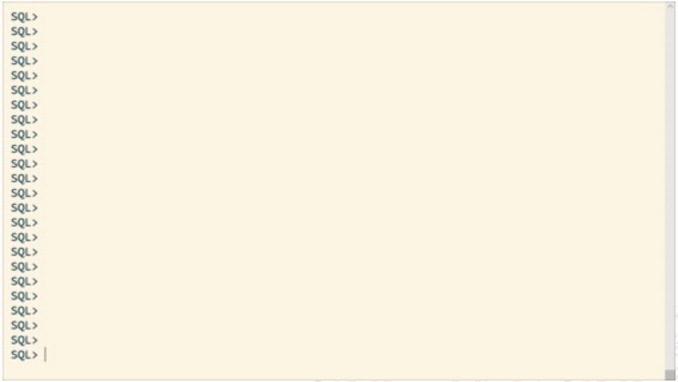
Only works for data files that belong to the current container

Data file is copied block-by-block

- Physical file size remains the same
- High Water Mark is not affected

Documentation: Concept and syntax

Online Data File Move | Demo



```
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
```

Watch on YouTube

Online Table Move

Online Table Move | Overview

NEW IN 12.2

Move table:

```
SQL> alter table lots_of_data move online tablespace users;
```

In parallel:

```
SQL> alter table lots_of_data move online tablespace users parallel 4;
```

Pro tip: Requires a short lock at the end of the operation

Online Table Move | Overview

NEW IN 12.2

Indexes remain VALID during and after online move

Optionally, change index as well:

```
SQL> alter table lots_of_data
     move online tablespace users
     update indexes(i1 tablespace users);
```

Pro tip: You can also move IOTs online, but not partitioned IOTs

Online Table Move | Overview

NEW IN 12.2

To also move LOB segments:

```
SQL> alter table lots_of_data
     move online tablespace users
     lob(clob1) store as (tablespace users);
```

Pro tip: Unused columns are preserved during a move operation

Online Table Move | Overview

NEW IN 12.2

Compress:

```
SQL> alter table lots_of_data
     move online tablespace users
     row store compress advanced;
```

Uncompress:

```
SQL> alter table lots_of_data
     move online tablespace users
     nocompress;
```

Documentation: Syntax

Pro tip: You can also move individual partitions of a partitioned table

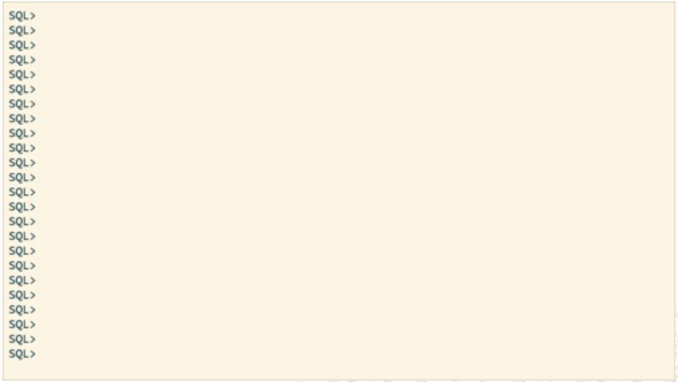
Online Table Move | Overview

NEW IN
12.2

Caution:

- Move invalidates statistics
- ROWIDs change
- Free space needed

Online Table Move | Demo



```
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
```

Watch on YouTube

Online Convert to Partitioned Table | Overview

NEW IN
12.2

Convert:

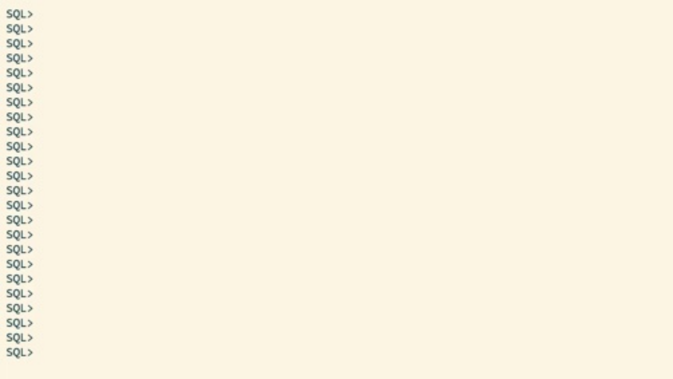
```
SQL> alter table lots_of_data
     modify partition by hash (object_id) partitions 8
     online
     update indexes (i_lots_of_data global);
```

Does not work for an already partitioned table

Pro tip: Number of hash partitions should always be power of 2

Documentation: Syntax and partition options

Online Convert to Partitioned Table | Demo



```
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
```

Watch on YouTube

DBMS_REDEFINITION

DBMS_REDEFINITION | Concept

> *You can redefine tables online with the DBMS_REDEFINITION package.*
>
> *… it is accessible to both queries and DML during much of the redefinition process. Typically, the table is locked in the exclusive mode only during a very small window …*

Database 19c Administrator'sGuide

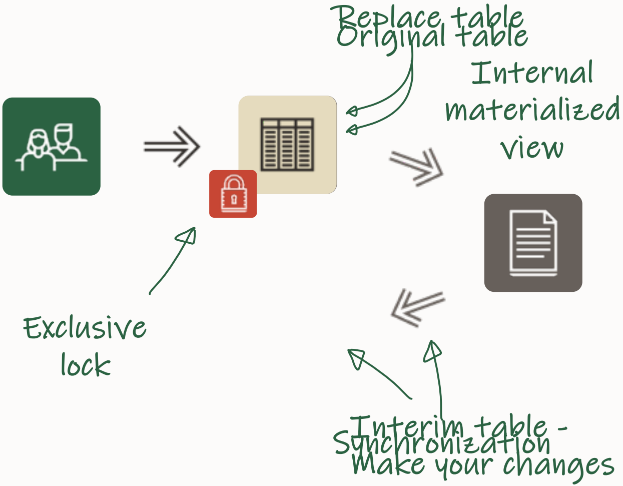
- Lock duration independent of table size
- Requires Enterprise Edition
- Use for bulk updates as well

DBMS_REDEFINITION | Use Cases

Some of the use cases:

- Convert BasicFile LOBs to SecureFile LOBs
- Modify the storage parameters of a table
- Add, modify, or drop one or more columns
- Add or drop partitioning support
- Change partition structure
- Convert to IOT (or reverse)
- Add attribute clustering to a table
- Optimized bulk updates
- … plus, more in the documentation

DBMS_REDEFINITION | Concept



DBMS_REDEFINITION | Benefits

In case of failures the redefinition process is often resumable

Pro tip: For more information about restarting, check the documentation

DBMS_REDEFINITION | Benefits

You can roll back the redefinition process, even after it has successfully completed

Pro tip: Rollbacks are described in detail in the documentation

DBMS_REDEFINITION | Considerations

Requires space to hold a copy of the table

DBMS_REDEFINITION | Considerations

Increased redo generation affects
Fast Recovery Area, backups and standby redo apply

DBMS_REDEFINITION | Example

```
SQL> exec dbms_redefinition.can_redef_table( ...
```

DBMS_REDEFINITION | Example

```
SQL> exec dbms_redefinition.can_redef_table( ...
SQL> create table interim_table ( ...
```

Pro tip: Use `DBMS_METADATA.GET_DDL` to create the interim table

DBMS_REDEFINITION | Example

```
SQL> exec dbms_redefinition.can_redef_table( ...
SQL> create table interim_table ( ...
SQL> exec dbms_redefinition.start_redef_table( ...
```

Pro tip: Speed up this step by enabling parallel query and DML in the session

DBMS_REDEFINITION | Example

```
SQL> exec dbms_redefinition.can_redef_table( ...
SQL> create table interim_table ( ...
SQL> exec dbms_redefinition.start_redef_table( ...
SQL> exec dbms_redefinition.sync_interim_table( ...
```

Pro tip: The more you run this procedure, the less time the final lock will need

DBMS_REDEFINITION | Example

```
SQL> exec dbms_redefinition.can_redef_table( ...
SQL> create table interim_table ( ...
SQL> exec dbms_redefinition.start_redef_table( ...
SQL> exec dbms_redefinition.sync_interim_table( ...
SQL> exec dbms_redefinition.copy_table_dependents( ...
```

Pro tip: You can also do this manually

DBMS_REDEFINITION | Example

```
SQL> exec dbms_redefinition.can_redef_table( ...
SQL> create table interim_table ( ...
SQL> exec dbms_redefinition.start_redef_table( ...
SQL> exec dbms_redefinition.sync_interim_table( ...
SQL> exec dbms_redefinition.copy_table_dependents( ...
SQL> select * from dba_redefinition_errors;
```

DBMS_REDEFINITION | Example

```
SQL> exec dbms_redefinition.can_redef_table( ...
SQL> create table interim_table ( ...
SQL> exec dbms_redefinition.start_redef_table( ...
SQL> exec dbms_redefinition.sync_interim_table( ...
SQL> exec dbms_redefinition.copy_table_dependents( ...
SQL> select * from dba_redefinition_errors;
SQL> exec dbms_redefinition.finish_redef_table( ...
```

Pro tip: The original table is shortly locked during this phase

DBMS_REDEFINITION | Considerations

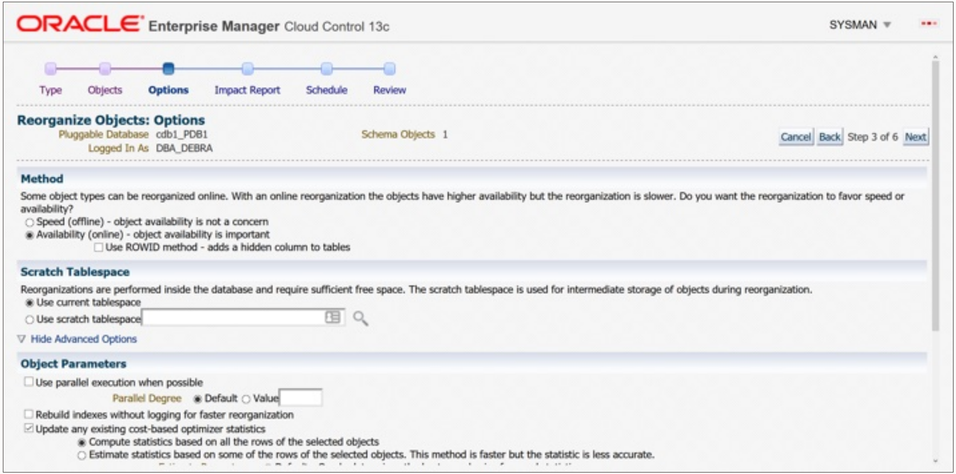
ℹ Statistics can be copied from source table.
Optionally, gather statistics using `DBMS_STATS`

DBMS_REDEFINITION | Easy

One button approach:

`DBMS_REDEFINITION.REDEF_TABLE`

DBMS_REDEFINITION | Enterprise Manager



Reorganize Objects: Options

Pluggable Database	cdb1_PDB1
Logged In As	DBA_DEBRA
Schema Objects	1

Method

Some object types can be reorganized online. With an online reorganization the objects have higher availability but the reorganization is slower. Do you want the reorganization to favor speed or availability?

○ Speed (offline) - object availability is not a concern
● Availability (online) - object availability is important
□ Use ROWID method - adds a hidden column to tables

Scratch Tablespace

Reorganizations are performed inside the database and require sufficient free space. The scratch tablespace is used for intermediate storage of objects during reorganization.

● Use current tablespace
○ Use scratch tablespace

Object Parameters

□ Use parallel execution when possible
Parallel Degree ● Default ○ Value
□ Rebuild indexes without logging for faster reorganization
☑ Update any existing cost-based optimizer statistics
● Compute statistics based on all the rows of the selected objects
○ Estimate statistics based on some of the rows of the selected objects. This method is faster but the statistic is less accurate.

DBMS_REDEFINITION | Nice To Know

Documentation:

- Redefining Tables Online, Database Administrator's Guide 19c

Views:

- `V$ONLINE_REDEF`
- `DBA_REDEFINITION_STATUS`

Installation & Patching

OUI | Install and Patch at the same time

NEW IN
18c

Since Oracle 18c, you can install and patch at the same time

- For GI homes

```
$ mkdir /u01/app/grid/1990
$ cd /u01/app/grid/1990

$ unzip LINUX.X64_193000_grid_home.zip
$ unzip p31750108_19000_Linux_x86-64.zip

$ ./gridSetup -applyRU 31750108
```

OUI | Install and Patch at the same time

NEW IN 18c

Since Oracle 18c, you can install and patch at the same time

- For DB homes

```
$ mkdir /u01/app/oracle/product/1990
$ cd /u01/app/oracle/product/1990

$ unzip LINUX.X64_193000_db_home.zip
$ unzip -d p31771877_190000_Linux-x86-64.zip /u01/app/oracle/product/1990/patch

$ ./runInstaller -applyRU patch/31771877
```

OUI | Install and Patch at the same time

NEW IN
18c

Apply multiple one-off patches in addition

- For DB homes

```
$ ./runInstaller -applyRU patch/30899722 -applyOneOffs x/30805684,y/30524762
```

- With multiple patches, you need to separate the subdirectories as otherwise the patch xml file gets overwritten and patches won't be found

- Unfortunately, this feature hasn't been implemented on MS Windows yet

Read Only Oracle Homes | Overview

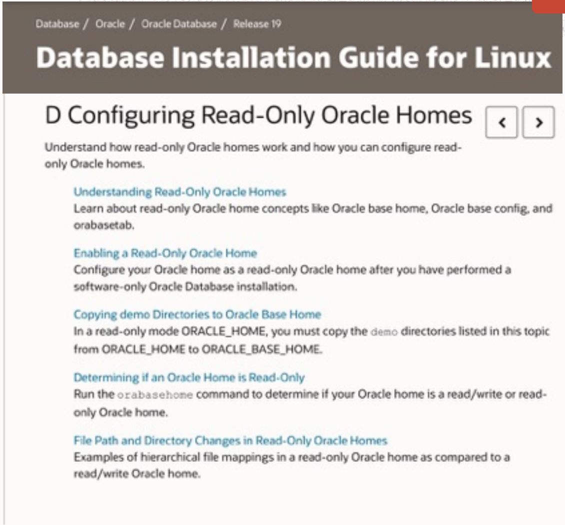
NEW IN 18c

Simple and easy cloning and provisioning

Configuration and log files stay outside $OH

Documentation:

- https://docs.oracle.com/en/database/oracle/oracle-database/19/ladbi/configuring-read-only-oracle-homes.html#GUID-906DA159-AC83-4ACC-A8A6-5B4A39EB72E1



Database / Oracle / Oracle Database / Release 19

Database Installation Guide for Linux

D Configuring Read-Only Oracle Homes

Understand how read-only Oracle homes work and how you can configure read-only Oracle homes.

Understanding Read-Only Oracle Homes
Learn about read-only Oracle home concepts like Oracle base home, Oracle base config, and orabasetab.

Enabling a Read-Only Oracle Home
Configure your Oracle home as a read-only Oracle home after you have performed a software-only Oracle Database installation.

Copying demo Directories to Oracle Base Home
In a read-only mode ORACLE_HOME, you must copy the `demo` directories listed in this topic from ORACLE_HOME to ORACLE_BASE_HOME.

Determining if an Oracle Home is Read-Only
Run the `orabasehome` command to determine if your Oracle home is a read/write or read-only Oracle home.

File Path and Directory Changes in Read-Only Oracle Homes
Examples of hierarchical file mappings in a read-only Oracle home as compared to a read/write Oracle home.

Read Only Oracle Homes | Configuration

Setup

1. Install as usual

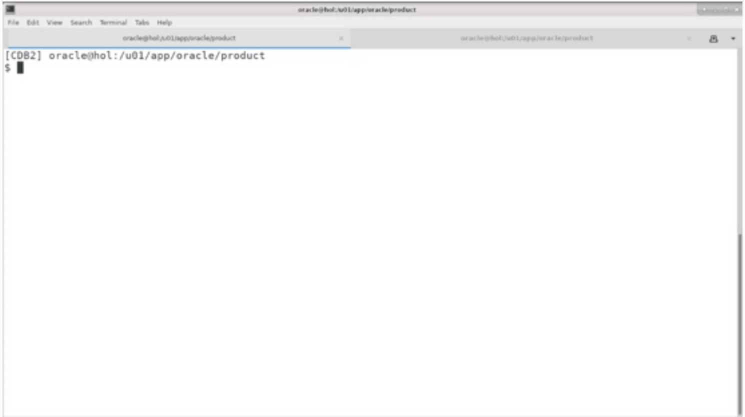
2. `$ORACLE_HOME/bin/roohctl -enable`

Documentation:

- https://docs.oracle.com/en/database/oracle/oracle-database/19/ladbi/configuring-read-only-oracle-homes.html#GUID-906DA159-AC83-4ACC-A8A6-5B4A39EB72E1

```
[oracle@hol ~]$ cd /u01/app/oracle/product/ROOH19/
[oracle@hol ROOH19]$ cd bin

[oracle@hol bin]$ ./roohctl -enable

Enabling Read-Only Oracle home.
Update orabasetab file to enable Read-Only Oracle home.
Orabasetab file has been updated successfully.
Create bootstrap directories for Read-Only Oracle home.
Bootstrap directories have been created successfully.
Bootstrap files have been processed successfully.
Read-Only Oracle home has been enabled successfully.

Check the log file
/u01/app/oracle/cfgtoollogs/roohctl/roohctl-
201124PM045139.log for more details.
```

Read Only Oracle Homes | Demo


oracle@hol:/u01/app/oracle/product
File Edit View Search Terminal Tabs Help
oracle@hol:/u01/app/oracle/product
oracle@hol:/u01/app/oracle/product

```
[CDB2] oracle@hol:/u01/app/oracle/product
$ 
```

Watch on YouTube

Read Only Oracle Homes | Directories

Important directories

```
cd $(orabaseconfig)
/u01/app/oracle

cd $(orabasehome)
/u01/app/oracle/homes/OraDB19Home2
```

Read Only Oracle Homes | Directory Structure

```
$ tree -a $(orabaseconfig)/dbs

/u01/app/oracle/dbs
├── hc_ROOH19.dat
├── initROOH19.ora
├── lkROOH19
├── orapwROOH19
└── spfileROOH19.ora
```

```
$ tree -a -d $(orabasehome)

/u01/app/oracle/homes/OraDB19Home2
├── assistants
│   └── dbca
│       └── templates
├── dbs
├── install
├── network
│   ├── admin
│   ├── log
│   └── trace
└── rdbms
    ├── audit
    └── log
        └── opatch
            └── lsinv
```

Long Identifiers

Be careful when you tell developers about it … 🤩

Long Identifiers | Overview

NEW IN 12.2

Standard since Oracle 12.2.0.1

You can't turn it off when `COMPATIBLE=12.2.0` or higher

Everything can be named now up to 128 bytes length

Exception:

- 8 bytes or less:
  - Database name
- 30 bytes or less
  - Tablespace
  - Disk group
  - PDB
  - Rollback segment

```
create table
 MY_NEW_TABLE_GETS_NOW_AN_EXTRA_LONG_NAME_BECAUSE_I_CAN
 (
  BUT_THIS_WORKS_WITH_COLUMNS_FOR_SURE_TOO varchar2(16000),
  AND_BELIEVE_ME_THIS_FEATURE_WILL_BE_USED_ASAP  number(3)
 )
;
```

Long Identifiers | Pitfalls

NEW IN 12.2

You can't turn it off!

Exchanging data to lower version database can be a problem

- Data Pump
    - With or without `VERSION`

```
Connected to: Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Warning: Oracle Data Pump is exporting from a database that supports long
 identifiers to a version that does not support long identifiers.

ORA-39373: cannot export TABLE_DATA:"SYSTEM"."T123456789012345678901234567890" to
 version 12.1 due to long identifiers
```

- https://mikedietrichde.com/2018/07/09/export-with-data-pump-and-long-identifiers/

- Database links

Expression Based Parameters

Expression Based Parameters | Overview

NEW IN
21

Numeric operation:

```
SQL> alter system set cpu_count='8/2' scope=both;
```

Other parameters:

```
SQL> alter system set sga_target=sga_max_size scope=both;
```

Combination:

```
SQL> alter system set shared_pool_size='sga_target*0.2' scope=both;
```

Expression Based Parameters | Overview

NEW IN 21

Min/max - and override operator precedence:

```
SQL> alter system set shared_pool_size='max(8000M, (sga_target-5000M)*0.2)';
```

Environment variable:

```
SQL> alter system set cpu_count='$NUMBER_OF_PROCESSORS/2';
```

Documentation: Syntax

Pro tip: Applies to `ALTER SESSION` commands as well

Expression Based Parameters | Overview

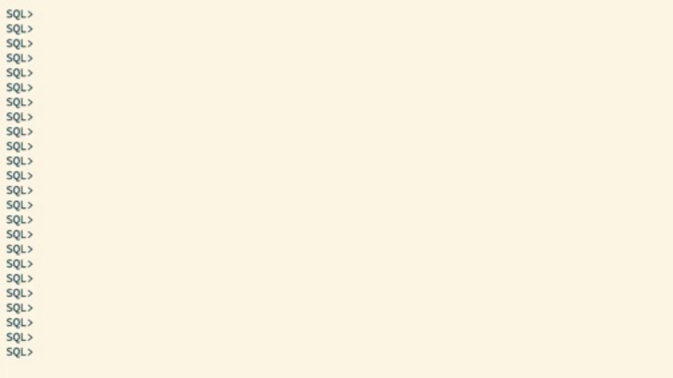
NEW IN 21

PFile:

```
*.cpu_count=(${NUMBER_OF_PROCESSORS} / 2)
*.aq_tm_processes=MIN(40, PROCESSES*0.1)
*.job_queue_processes=processes
```

Documentation: Syntax

Expression Based Parameters | Demo



```
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
```

Watch on YouTube

Data Guard

Data Guard | Overview

NEW IN
19

- Restore points are automatically replicated from primary to standby database
- When primary database is flashed back - standby database follows automatically
- DML on standby gets redirected to primary database (Active Data Guard)

Documentation: Restore Point Replication and Automatic Flashback and DML redirect

Data Guard | Demo

Restore Point Replication

Create Restore Point

Watch on YouTube

Keystore Password External Store

Keystore Password External Store | Overview

NEW IN
12.2

Instead of typing password in cleartext:

```
SQL> administer key management ... keystore identified by "S3cr3tP@@sw0rd";
SQL> --or
SQL> create pluggable database ... keystore identified by "S3cr3tP@@sw0rd";
```

Get it from the external store:

```
SQL> administer key management ... keystore identified by external store;
SQL> --or
SQL> create pluggable database ... keystore identified by external store;
```

Keystore Password External Store | Overview

NEW IN 12.2

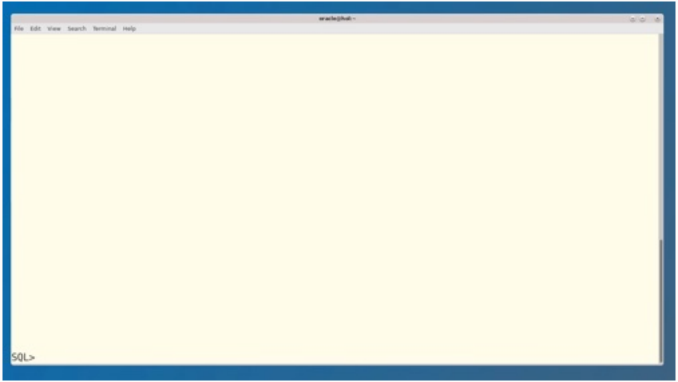
A much more secure solution:

- Supports separation of duties
- Avoid hardcoded keystore password in scripts
- Avoid typing the keystore password in cleartext in terminal

Documentation: Concept

Pro tip: Get all the details in the blog post *How to Stop Hardcoding Your TDE Keystore Password*

Keystore Password External Store | Demo



```
SQL>
```

Watch on YouTube

Online Encryption

Online Encryption | Overview

NEW IN 12.2

Online encryption of existing database tablespace files

- `alter tablespace <tbs> encryption online encrypt;`
- Storage overhead: 2x largest file of tbs

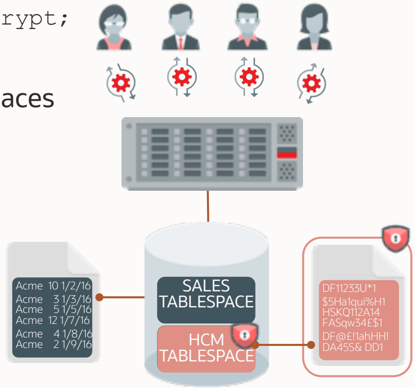
Full encryption of SYSTEM, SYSAUX, and UNDO tablespaces

- Not recommended

Offline tablespace encryption

- No storage overhead

RMAN decrypted restore with 18c



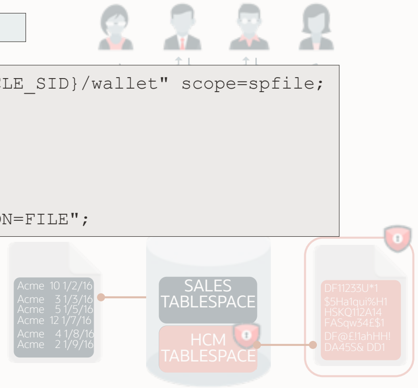
Online Encryption | Step By Step

NEW IN 12.2

1. Wallet Root Preparation

```
$  mkdir -p ${ORACLE_BASE}/admin/${ORACLE_SID}/wallet/tde
```

```
alter system set WALLET_ROOT="${ORACLE_BASE}/admin/${ORACLE_SID}/wallet" scope=spfile;

shutdown

startup

alter system set TDE_CONFIGURATION="KEYSTORE_CONFIGURATION=FILE";
```



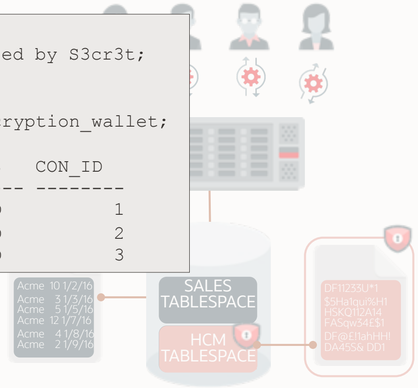
Online Encryption | Step By Step

NEW IN 12.2

2. Create Password Protected Keystore

```
administer key management create keystore
 '${ORACLE_BASE}/admin/${ORACLE_SID}/wallet/tde' identified by S3cr3t;


select WRL_TYPE, WRL_PARAMETER, STATUS, CON_ID from v$encryption_wallet;

WRL_TYPE  WRL_PARAMETER                           STATUS    CON_ID
--------- ---------------------------------------- -------- --------
FILE      /u01/app/oracle/admin/CDB2/wallet/tde/   CLOSED          1
FILE                                               CLOSED          2
FILE                                               CLOSED          3
```

```
$  ls -lrt /u01/app/oracle/admin/CDB2/wallet/tde/

-rw-------. 1 oracle dba 2555 Jan 11 23:26 ewallet.p12
```

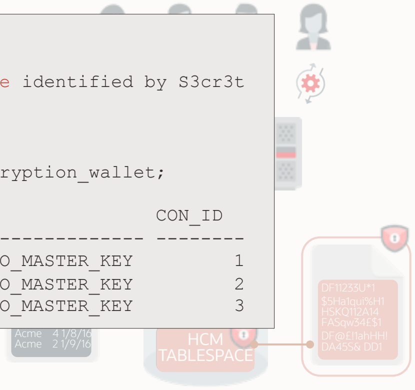


Online Encryption | Step By Step

NEW IN 12.2

3. Open the Keystore

```
alter pluggable database all open;

administer key management set keystore open force keystore identified by S3cr3t
 container=all;


select WRL_TYPE, WRL_PARAMETER, STATUS, CON_ID from v$encryption_wallet;

WRL_TYPE  WRL_PARAMETER                            STATUS                CON_ID
--------- ---------------------------------------- -------------------- --------
FILE      /u01/app/oracle/admin/CDB2/wallet/tde/   OPEN_NO_MASTER_KEY          1
FILE                                               OPEN_NO_MASTER_KEY          2
FILE                                               OPEN_NO_MASTER_KEY          3
```

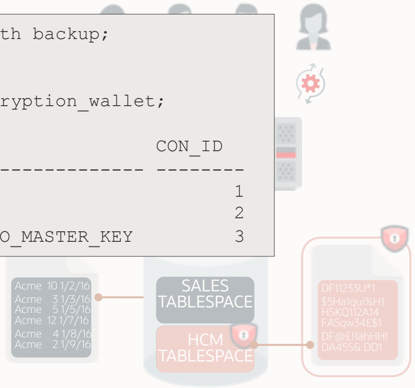


Online Encryption | Step By Step

NEW IN 12.2

4a. Create Master Key CDB$ROOT

```
administer key management set key identified by S3cr3t with backup;


select WRL_TYPE, WRL_PARAMETER, STATUS, CON_ID from v$encryption_wallet;

WRL_TYPE  WRL_PARAMETER                            STATUS              CON_ID
--------- ---------------------------------------- ------------------- --------
FILE      /u01/app/oracle/admin/CDB2/wallet/tde/   OPEN                       1
FILE                                               OPEN                       2
FILE                                               OPEN_NO_MASTER_KEY         3
```

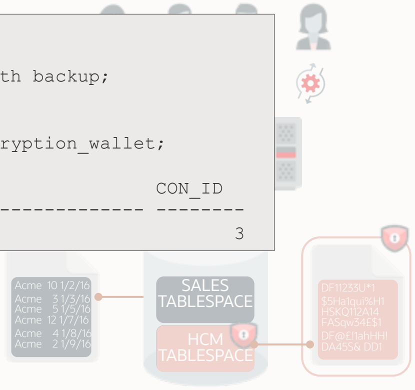


Online Encryption | Step By Step

NEW IN 12.2

4b. Create Master Key PDB

```
alter session set container=PDB1;

administer key management set key identified by S3cr3t with backup;


select WRL_TYPE, WRL_PARAMETER, STATUS, CON_ID from v$encryption_wallet;

WRL_TYPE  WRL_PARAMETER                           STATUS              CON_ID
--------- ---------------------------------------- ------------------- --------
FILE                                              OPEN                       3
```



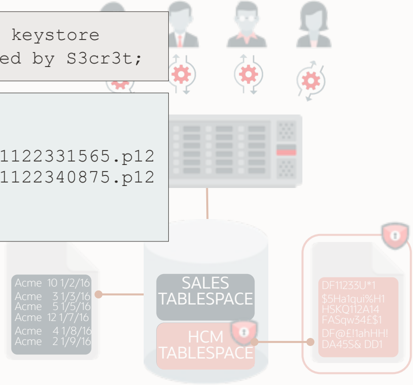
Online Encryption | Step By Step

NEW IN 12.2

5. Autologin Keystore

```
administer key management create auto_login keystore from keystore
 '${ORACLE_BASE}/admin/${ORACLE_SID}/wallet/tde' identified by S3cr3t;
```

```
$  ls -lrt /u01/app/oracle/admin/CDB2/wallet/tde/

-rw-------. 1 oracle dba 2555 Jan 11 23:33 ewallet_2021011122331565.p12
-rw-------. 1 oracle dba 3995 Jan 11 23:34 ewallet_2021011122340875.p12
-rw-------. 1 oracle dba 5467 Jan 11 23:34 ewallet.p12
-rw-------. 1 oracle dba 5512 Jan 11 23:35 cwallet.sso
```

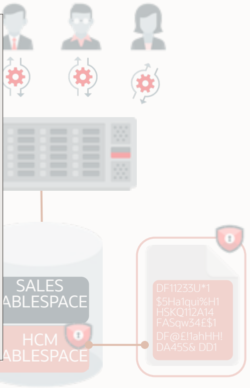


Online Encryption | Step By Step

NEW IN 12.2

6. Encrypt Tablespaces Online

```
alter session set container=PDB1;

alter tablespace TEST encryption online encrypt;


select TABLESPACE_NAME, STATUS, ENCRYPTED from DBA_TABLESPACES;

TABLESPACE_NAME      STATUS    ENC
-------------------- --------- ---
SYSTEM               ONLINE    NO
SYSAUX               ONLINE    NO
UNDOTBS1             ONLINE    NO
TEMP                 ONLINE    NO
TEST                 ONLINE    YES
```

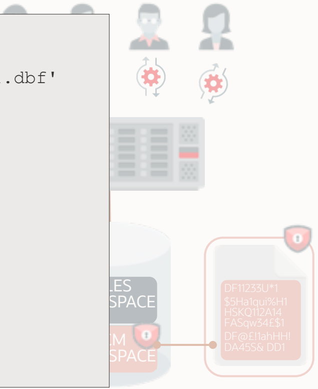


Online Encryption | Step By Step

NEW IN 12.2

7. Encrypt New Tablespaces

```
alter system set ENCRYPT_NEW_TABLESPACES=ALWAYS scope=both;

create tablespace FUTURE datafile '/u02/oradata/CDB2/pdb1/future01.dbf'
 size 100M autoextend on online;


select TABLESPACE_NAME, STATUS, ENCRYPTED from DBA_TABLESPACES;

TABLESPACE_NAME       STATUS    ENC
--------------------- --------- ---
SYSTEM                ONLINE    NO
SYSAUX                ONLINE    NO
UNDOTBS1              ONLINE    NO
TEMP                  ONLINE    NO
TEST                  ONLINE    YES
FUTURE                ONLINE    YES
```

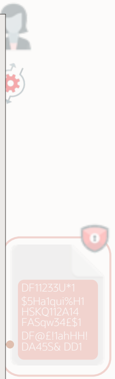


Online Encryption | Recovery Manager

NEW IN 12.2

RMAN Option: Restore as encrypted

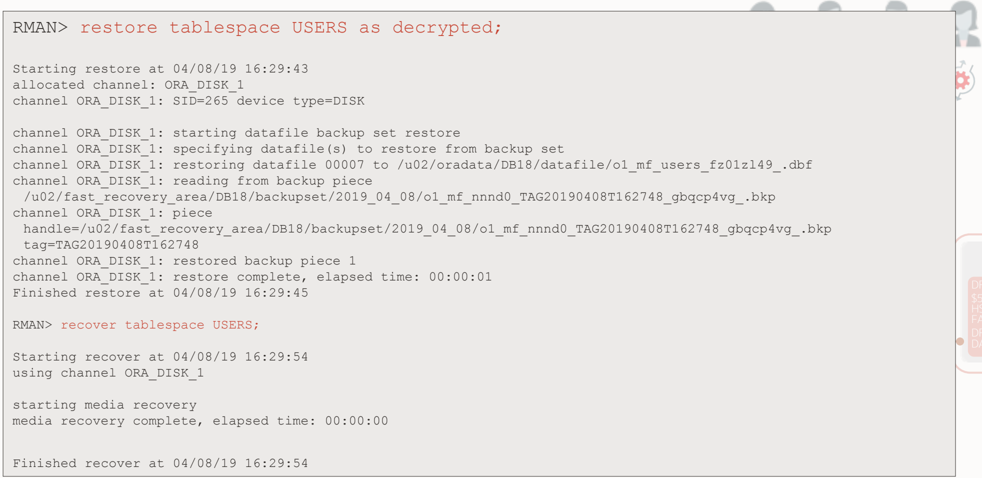
```
RMAN> restore tablespace TEST as encrypted;

Starting restore at 01/01/21 15:00:00
using channel ORA_DISK_1

channel ORA_DISK_1: starting datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
channel ORA_DISK_1: restoring datafile 00009 to /u02/oradata/DB12/o1_mf_test_fz01zl49_.dbf
channel ORA_DISK_1: reading from backup piece
 /u02/fast_recovery_area/DB12/backupset/2021_01_01/o1_mf_nnnd0_TAG20210101T150000_gbq5n71p_.bkp
channel ORA_DISK_1: piece
 handle=/u02/fast_recovery_area/DB12/backupset/2021_01_01/o1_mf_nnnd0_TAG20210101T150000_gbq5n71p_.bkp
 tag=TAG20210101T150000
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:01
Finished restore at 01/01/21 15:02:11
RMAN> recover tablespace TEST;

Starting recover at 01/01/21 15:01:12
using channel ORA_DISK_1

starting media recovery
media recovery complete, elapsed time: 00:00:00

Finished recover at 01/01/21 15:01:12
```



Online Encryption | Recovery Manager

NEW IN 18c

RMAN Option: Restore as decrypted



```
RMAN> restore tablespace USERS as decrypted;

Starting restore at 04/08/19 16:29:43
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=265 device type=DISK

channel ORA_DISK_1: starting datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
channel ORA_DISK_1: restoring datafile 00007 to /u02/oradata/DB18/datafile/o1_mf_users_fz01z149_.dbf
channel ORA_DISK_1: reading from backup piece
 /u02/fast_recovery_area/DB18/backupset/2019_04_08/o1_mf_nnnd0_TAG20190408T162748_gbqcp4vg_.bkp
channel ORA_DISK_1: piece
 handle=/u02/fast_recovery_area/DB18/backupset/2019_04_08/o1_mf_nnnd0_TAG20190408T162748_gbqcp4vg_.bkp
 tag=TAG20190408T162748
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:01
Finished restore at 04/08/19 16:29:45

RMAN> recover tablespace USERS;

Starting recover at 04/08/19 16:29:54
using channel ORA_DISK_1

starting media recovery
media recovery complete, elapsed time: 00:00:00


Finished recover at 04/08/19 16:29:54
```

Online Encryption | Demo

NEW IN
18c

Online
Encryption

Copyright © 2021, Oracle and/or its affiliates

Watch on YouTube

Gradual Password Rollover

Gradual Password Rollover | Overview

NEW IN 19

- Allow a user to have two passwords for a limited amount of time

```
SQL> CREATE PROFILE app_profile LIMIT
         PASSWORD_ROLLOVER_TIME 1;

SQL> CREATE USER app_user
        ...
        PROFILE app_profile;
```

- Minimum one hour (1/24), maximum 60 days

Originally a 21c feature, but backported with 19.12.0

Documentation

Gradual Password Rollover | Overview

NEW IN 19

- Which users are using the old password?

```
SQL> select authentication_type
     from   unified_audit_trail
     where  action_name='LOGON' and dbusername='APP_USER';
```

- The `authentication_type` tells which password is used:

```
(TYPE=(DATABASE));(CLIENT ADDRESS=((PROTOCOL=tcp)(HOST=10.0.1.225)(PORT=24974)));
 (LOGON_INFO=((VERIFIER=12C-NEW)(CLIENT_CAPABILITIES=O5L_NP,O7L_MR,O8L_LI)));

(TYPE=(DATABASE));(CLIENT ADDRESS=((PROTOCOL=tcp)(HOST=10.0.1.225)(PORT=24983)));
 (LOGON_INFO=((VERIFIER=12C-OLD)(CLIENT_CAPABILITIES=O5L_NP,O7L_MR,O8L_LI)));
```

Gradual Password Rollover | Demo

NEW IN 19

Watch on YouTube

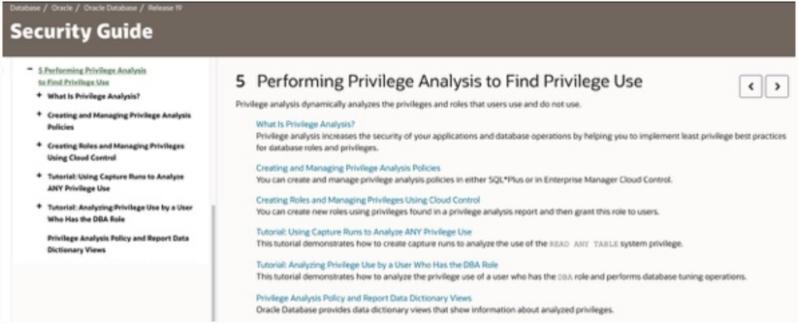
Privilege Capture

```
grant DBA to APP_USER with admin option;
```

Privilege Capture | Overview

Implement the concept of "least privileges"

- Report used privileges and grant path
- Report unused privileges
  - https://docs.oracle.com/en/database/oracle/oracle-database/19/dbseg/performing-privilege-analysis-find-privilege-use.html#GUID-44CB644B-7B59-4B3B-B375-9F9B96F60186



Database / Oracle / Oracle Database / Release 19
Security Guide
− 5 Performing Privilege Analysis to Find Privilege Use
+ What Is Privilege Analysis?
+ Creating and Managing Privilege Analysis Policies
+ Creating Roles and Managing Privileges Using Cloud Control
+ Tutorial: Using Capture Runs to Analyze ANY Privilege Use
+ Tutorial: Analyzing Privilege Use by a User Who Has the DBA Role
Privilege Analysis Policy and Report Data Dictionary Views
‹
›

5 Performing Privilege Analysis to Find Privilege Use

Privilege analysis dynamically analyzes the privileges and roles that users use and do not use.

What Is Privilege Analysis?
Privilege analysis increases the security of your applications and database operations by helping you to implement least privilege best practices for database roles and privileges.

Creating and Managing Privilege Analysis Policies
You can create and manage privilege analysis policies in either SQL*Plus or in Enterprise Manager Cloud Control.

Creating Roles and Managing Privileges Using Cloud Control
You can create new roles using privileges found in a privilege analysis report and then grant this role to users.

Tutorial: Using Capture Runs to Analyze ANY Privilege Use
This tutorial demonstrates how to create capture runs to analyze the use of the `READ ANY TABLE` system privilege.

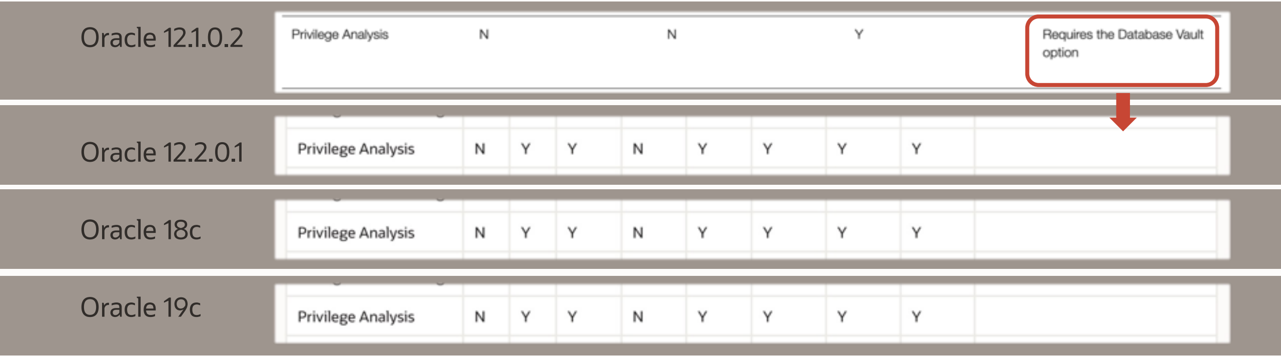
Tutorial: Analyzing Privilege Use by a User Who Has the DBA Role
This tutorial demonstrates how to analyze the privilege use of a user who has the `DBA` role and performs database tuning operations.

Privilege Analysis Policy and Report Data Dictionary Views
Oracle Database provides data dictionary views that show information about analyzed privileges.

Privilege Capture | License

Part of Enterprise Edition

- Changed from "requires Database Vault option" to "included in EE" with Oracle 18c
- Applies to Oracle 12.2.0.1 and newer



Oracle 12.1.0.2

Privilege Analysis	N	N	Y	Requires the Database Vault option

Oracle 12.2.0.1

Privilege Analysis	N	Y	Y	N	Y	Y	Y	Y	

Oracle 18c

Privilege Analysis	N	Y	Y	N	Y	Y	Y	Y	

Oracle 19c

Privilege Analysis	N	Y	Y	N	Y	Y	Y	Y	

Privilege Capture | Create a Capture Policy

DBMS_PRIVILEGE_CAPTURE

CREATE_CAPTURE

- Create a privilege capture analysis policy

ENABLE_CAPTURE

- Enable the analysis policy

Run it for a given period of time

DISABLE_CAPTURE

- Stop the privilege analysis run

GENERATE_RESULT

- Populate dictionary views with analysis results

DROP_CAPTURE / DELETE_RUN

- Drop it if it is not longer needed
- Or delete the results of this run only

```
BEGIN
 DBMS_PRIVILEGE_CAPTURE.CREATE_CAPTURE(
  name =>        'tuning_privs',
  description => 'analyze tuning privs',
  type =>         DBMS_PRIVILEGE_CAPTURE.
                  G_CONTEXT,
  condition =>  'SYS_CONTEXT(''USERENV'',
''SESSION_USER'')=''SMITH''');
END;
/
```

Privilege Capture | Start Capture

DBMS_PRIVILEGE_CAPTURE

CREATE_CAPTURE

- Create a privilege capture analysis policy

ENABLE_CAPTURE

- Enable the analysis policy

Run it for a given period of time

DISABLE_CAPTURE

- Stop the privilege analysis run

GENERATE_RESULT

- Populate dictionary views with analysis results

DROP_CAPTURE / DELETE_RUN

- Drop it if it is not longer needed
- Or delete the results of this run only

```
BEGIN
  DBMS_PRIVILEGE_CAPTURE.ENABLE_CAPTURE
  ('tuning_privs');
END;
/
```

Privilege Capture | Stop Capture

DBMS_PRIVILEGE_CAPTURE

CREATE_CAPTURE

- Create a privilege capture analysis policy

ENABLE_CAPTURE

- Enable the analysis policy

Run it for a given period of time

DISABLE_CAPTURE

- Stop the privilege analysis run

GENERATE_RESULT

- Populate dictionary views with analysis results

DROP_CAPTURE / DELETE_RUN

- Drop it if it is not longer needed
- Or delete the results of this run only

```
BEGIN
  DBMS_PRIVILEGE_CAPTURE.DISABLE_CAPTURE
  ('tuning_privs');
END;
/
```

Privilege Capture | Populate Views

DBMS_PRIVILEGE_CAPTURE

CREATE_CAPTURE

- Create a privilege capture analysis policy

ENABLE_CAPTURE

- Enable the analysis policy

Run it for a given period of time

DISABLE_CAPTURE

- Stop the privilege analysis run

GENERATE_RESULT

- Populate dictionary views with analysis results

DROP_CAPTURE / DELETE_RUN

- Drop it if it is not longer needed
- Or delete the results of this run only

```
BEGIN
  DBMS_PRIVILEGE_CAPTURE.GENERATE_RESULT
  ('tuning_privs');
END;
/
```

Privilege Capture | Used System Privileges

```
DBMS_PRIVILEGE_CAPTURE.GENERATE_RESULT
```

System privileges and roles used by the user "SMITH"

```
SELECT USERNAME, SYS_PRIV, USED_ROLE, PATH
  FROM DBA_USED_SYSPRIVS_PATH
 WHERE USERNAME = 'SMITH'
 ORDER BY 1, 2, 3;


USERNAME SYS_PRIV        USED_ROLE            PATH
-------- --------------- -------------------- ---------------------------------------------------------
SMITH    ANALYZE ANY     IMP_FULL_DATABASE    GRANT_PATH('SMITH', 'DBA')
SMITH    ANALYZE ANY     IMP_FULL_DATABASE    GRANT_PATH('SMITH', 'DBA', 'IMP_FULL_DATABASE')
SMITH    ANALYZE ANY     IMP_FULL_DATABASE    GRANT_PATH('SMITH', 'DBA', 'DATAPUMP_IMP_FULL_DATABASE',
                                                                     'IMP_FULL_DATABASE')
...
```

Privilege Capture | Unused System Privileges

```
DBMS_PRIVILEGE_CAPTURE.GENERATE_RESULT
```

System privileges unused by the user "SMITH"

```
SELECT USERNAME, SYS_PRIV
  FROM DBA_UNUSED_SYSPRIVS
 WHERE USERNAME = 'SMITH'
 ORDER BY 1, 2;

USERNAME SYS_PRIV
-------- ------------------------------
SMITH    ADMINISTER ANY SQL TUNING SET
SMITH    ADMINISTER DATABASE TRIGGER
SMITH    ADMINISTER RESOURCE MANAGER
SMITH    ADMINISTER SQL TUNING SET
SMITH    ALTER ANY ASSEMBLY
SMITH    ON COMMIT REFRESH
...
```

Privilege Capture | Purge Views

DBMS_PRIVILEGE_CAPTURE

CREATE_CAPTURE

- Create a privilege capture analysis policy

ENABLE_CAPTURE

- Enable the analysis policy

Run it for a given period of time

DISABLE_CAPTURE

- Stop the privilege analysis run

GENERATE_RESULT

- Populate dictionary views with analysis results

DROP_CAPTURE / DELETE_RUN

- Drop it if it is not longer needed
- Or delete the results of this run only

```
BEGIN
  DBMS_PRIVILEGE_CAPTURE.DELETE_RUN
  ('tuning_privs');
END;
/

BEGIN
  DBMS_PRIVILEGE_CAPTURE.DROP_CAPTURE
  ('tuning_privs');
END;
/
```

Privilege Capture | Things to Know

`CAPTURE_ADMIN` role required

Analysis with Multitenant only per container

Only one capture policy can be active at a given time

Complete application test necessary

- Risk to revoke a privilege which gets used rarely

Results are kept until capture run or policy gets deleted

Secure External Password Store

Secure External Password Store | Overview

Instead of typing user name and password in cleartext:

```
sqlplus batch/S3cr3tP##sw00rd@DB19
```

Get it from the external store:

```
sqlplus /@DB19
```

Secure External Password Store | Overview

Characteristics:

- TNS alias controls which credential is being used
- SQLNet.ora points to location of keystore

```
WALLET_LOCATION =
  (SOURCE =
    (METHOD = FILE)
    (METHOD_DATA =
  (DIRECTORY = /home/oracle/tns)
  )
 )
```

- Keystore is an encrypted file located in your file system

```
mkstore -wrl $ORACLE_WALLET_LOC -createCredential DB19 batch S3cr3tP##sw00rd
```

Pro tip: Use `SQLNET.WALLET_OVERRIDE = TRUE` to ensure no other external authentication methods are used

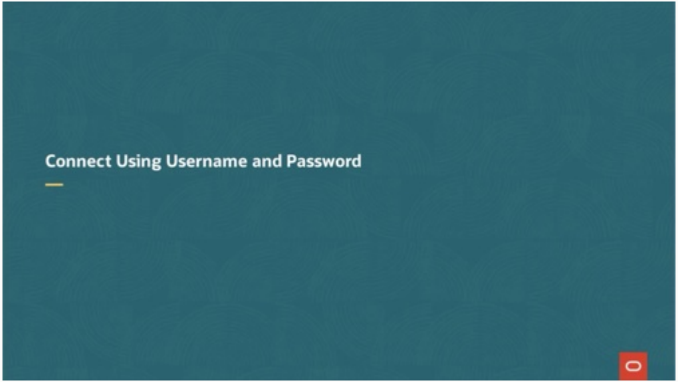
Secure External Password Store | Overview

A much more secure solution:

- Supports separation of duties
- Avoid hardcoded passwords in scripts
- Avoid typing passwords in cleartext in terminal

Documentation: Concept

Secure External Password Store | Demo



Connect Using Username and Password

Watch on YouTube

1.
Intro

Fallback vs Rollback

`COMPATIBLE`

Time Zone File

2.
Methods

3.
Scenarios

TWO CONCEPTS

ROLLBACK
Returns the database to the previous, consistent state

FALLBACK
Returns the database to a previous release without losing changes

The most limiting factor on your fallback options is `COMPATIBLE`

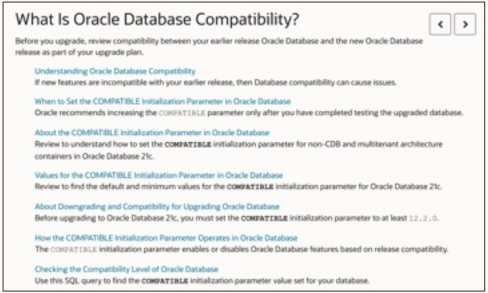
Compatible | Overview

What does `COMPATIBLE` do?

- Enables use of features, e.g.
  - Long identifiers
  - Blockchain tables
- Redo log file structure
- Data file format
- Tablespace header

Where is it documented?

- [Database Upgrade Guide]



What Is Oracle Database Compatibility?

Before you upgrade, review compatibility between your earlier release Oracle Database and the new Oracle Database release as part of your upgrade plan.

Understanding Oracle Database Compatibility
If new features are incompatible with your earlier release, then Database compatibility can cause issues.

When to Set the COMPATIBLE Initialization Parameter in Oracle Database
Oracle recommends increasing the `COMPATIBLE` parameter only after you have completed testing the upgraded database.

About the COMPATIBLE Initialization Parameter in Oracle Database
Review to understand how to set the **`COMPATIBLE`** initialization parameter for non-CDB and multitenant architecture containers in Oracle Database 21c.

Values for the COMPATIBLE Initialization Parameter in Oracle Database
Review to find the default and minimum values for the **`COMPATIBLE`** initialization parameter for Oracle Database 21c.

About Downgrading and Compatibility for Upgrading Oracle Database
Before upgrading to Oracle Database 21c, you must set the **`COMPATIBLE`** initialization parameter to at least `12.2.0`.

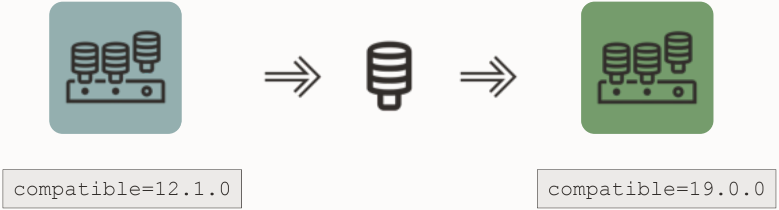
How the COMPATIBLE Initialization Parameter Operates in Oracle Database
The `COMPATIBLE` initialization parameter enables or disables Oracle Database features based on release compatibility.

Checking the Compatibility Level of Oracle Database
Use this SQL query to find the **`COMPATIBLE`** initialization parameter value set for your database.

`COMPATIBLE` can only be changed to a higher value

Changing `COMPATIBLE` prevents the use of Flashback Database and downgrade

Compatible | Multitenant



`compatible=12.1.0`

`compatible=19.0.0`

On plug-in:

- PDB silently changes it's `COMPATIBLE` setting
- The change is irreversible

We recommend to change `COMPATIBLE` one or two weeks after the upgrade

Pro tip: Changing `COMPATIBLE` requires a database restart

Compatible | Recommendation

Which value should you use for `COMPATIBLE`?

- The default of the database release
  - 11.2.0
  - 12.1.0
  - 12.2.0
  - 18.0.0
  - 19.0.0

Should you change `COMPATIBLE` when patching?

- NEVER!
  - Except for ...

Compatible | Comparison

Fully independent from each other

`COMPATIBLE`

- Enables features
- Changes on-disk structures

`OPTIMIZER_FEATURES_ENABLE`

- Just reverts to the parameters used in a previous release
- Avoid using it if possible
- This is not a Swiss Army knife!
- You will turn off a lot of great features

> *Modifying the OPTIMIZER_FEATURES_ENABLE parameter generally is strongly discouraged and should only be used as a short term measure at the suggestion of Oracle Global Support.*

Use Caution if Changing the OPTIMIZER_FEATURES_ENABLE Parameter After an Upgrade (Doc ID 1362332.1)

Compatible | AutoUpgrade

AutoUpgrade does not change `COMPATIBLE`

Unless you want it

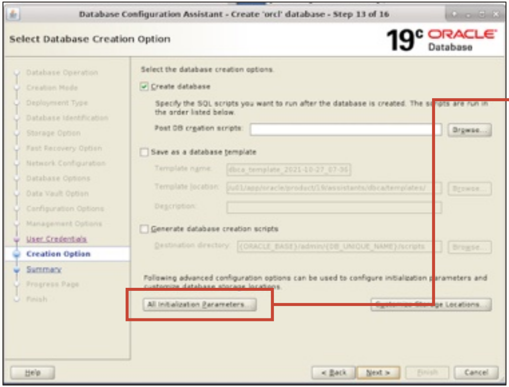
```
upg1.drop_grp_after_upgrade=yes
upg1.raise_compatible=yes
```

How do I create a database with a non-default `COMPATIBLE` setting?

Compatible | DBCA

COMPATIBLE choice is only available via CUSTOM database creation

- General Purpose, OLTP and DWH templates create databases with default COMPATIBLE



Database Configuration Assistant - Create 'orcl' database - Step 13 of 16

Select Database Creation Option

Select the database creation options.

☑ Create database

Specify the SQL scripts you want to run after the database is created. The scripts are run in the order listed below.

Post DB creation scripts:

☐ Save as a database template

Template name: dbca_template_2021-10-27_07-35

Template location: /u01/app/oracle/product/19/assistants/dbca/templates/

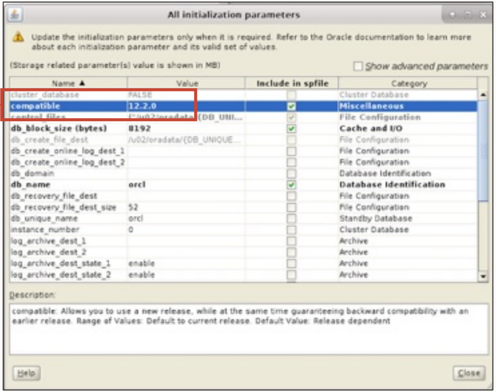
Description:

☐ Generate database creation scripts

Destination directory: {ORACLE_BASE}/admin/{DB_UNIQUE_NAME}/scripts

Following advanced configuration options can be used to configure initialization parameters and customize database storage locations.

Browse...
Browse...
Browse...
All Initialization Parameters...
Customize Storage Locations...
Help
< Back
Next >
Finish
Cancel



All initialization parameters

Update the initialization parameters only when it is required. Refer to the Oracle documentation to learn more about each initialization parameter and its valid set of values.

(Storage related parameter(s) value is shown in MB)

☐ Show advanced parameters

Name ▲	Value	Include in spfile	Category
cluster_database	FALSE	☐	Cluster Database
compatible	12.2.0	☑	Miscellaneous
control_files	("/u02/oradata/{DB_UNI...	☑	File Configuration
db_block_size (bytes)	8192	☑	Cache and I/O
db_create_file_dest	/u02/oradata/{DB_UNIQUE...	☐	File Configuration
db_create_online_log_dest_1		☐	File Configuration
db_create_online_log_dest_2		☐	File Configuration
db_domain		☐	Database Identification
db_name	orcl	☑	Database Identification
db_recovery_file_dest		☐	File Configuration
db_recovery_file_dest_size	52	☐	File Configuration
db_unique_name	orcl	☐	Standby Database
instance_number	0	☐	Cluster Database
log_archive_dest_1		☐	Archive
log_archive_dest_2		☐	Archive
log_archive_dest_state_1	enable	☐	Archive
log_archive_dest_state_2	enable	☐	Archive

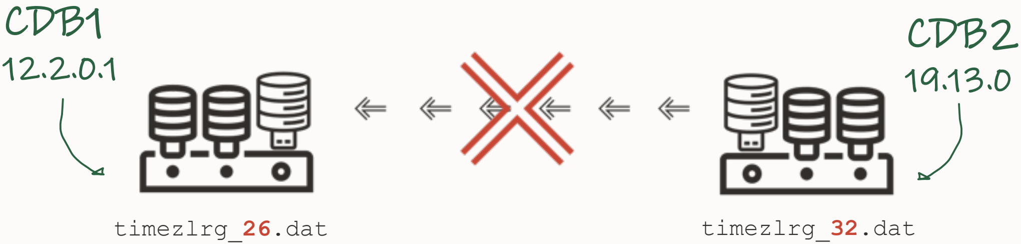
Description:

compatible: Allows you to use a new release, while at the same time guaranteeing backward compatibility with an earlier release. Range of Values: Default to current release. Default Value: Release dependent

Time Zone | Downgrade requirements

Identical zone files must exist in both homes and databases

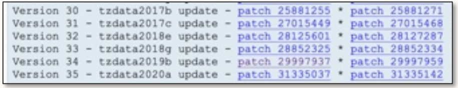
- A lower time zone file's version will prevent downgrade



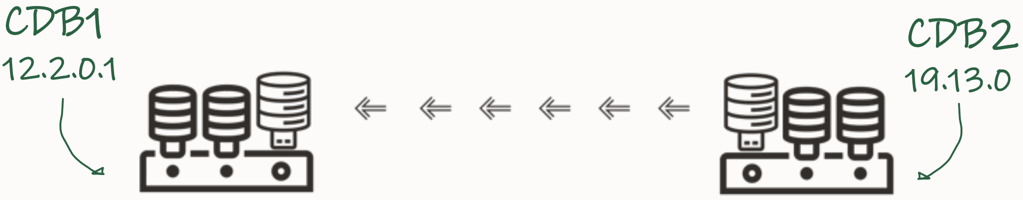
Time Zone | Downgrade Solution

Apply matching time zone patch to source home and database

- MOS Note:412160.1



```
Version 30 - tzdata2017b update - patch 25881255 * patch 25881271
Version 31 - tzdata2017c update - patch 27015449 * patch 27015468
Version 32 - tzdata2018e update - patch 28125601 * patch 28127287
Version 33 - tzdata2018g update - patch 28852325 * patch 28852334
Version 34 - tzdata2019b update - patch 29997937 * patch 29997959
Version 35 - tzdata2020a update - patch 31335037 * patch 31335142
```



~~timezlrg_26.dat~~

Patch: timezlrg_32.dat

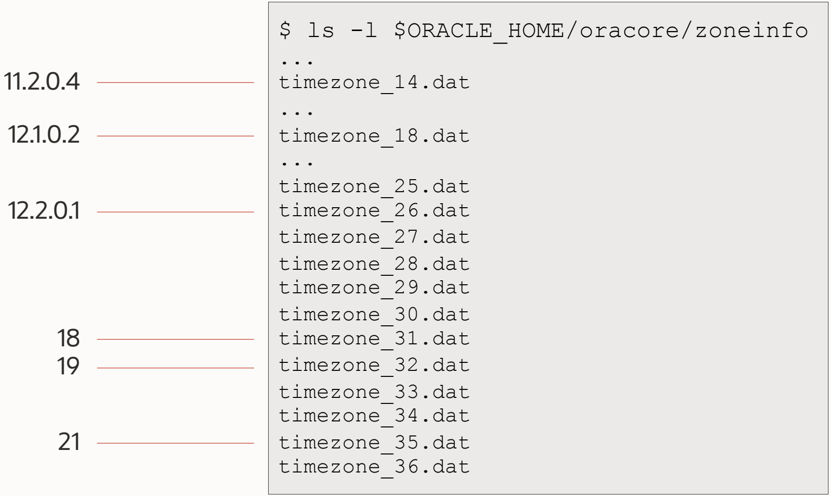
timezlrg_32.dat

Time Zone | Check

Check time zone file version upfront

```
SQL> select * from V$TIMEZONE_FILE;


FILENAME             VERSION    CON_ID
-------------------- ---------- ----------
timezlrg_26.dat              26          0
```

Time Zone | Default Version



```
$ ls -l $ORACLE_HOME/oracore/zoneinfo
...
timezone_14.dat
...
timezone_18.dat
...
timezone_25.dat
timezone_26.dat
timezone_27.dat
timezone_28.dat
timezone_29.dat
timezone_30.dat
timezone_31.dat
timezone_32.dat
timezone_33.dat
timezone_34.dat
timezone_35.dat
timezone_36.dat
```

Time Zone | Patch and Apply

Apply time zone patch to Oracle Home

- Files will be written to `$ORACLE_HOME/oracore/zoneinfo`
- Time zone patches are not part of any RU or patch bundle
  - Not RAC-rolling
- Time zone patches are not patch bundle agnostic but generic per release

Use scripts to adjust time zone

- `?/rdbms/admin/utltz_upg_check.sql`
- `?/rdbms/admin/utltz_upg_apply.sql`

Time Zone | AutoUpgrade

AutoUpgrade upgrades time zone by default

To avoid time zone upgrade

```
upg1.source_home=/u01/app/oracle/product/12.2.0.1
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=CDB1
upg1.pdbs=PDB1
upg1.target_cdb=CDB2
upg1.timezone_upg=no
```

Ease your life by having identical time zone files in all environments

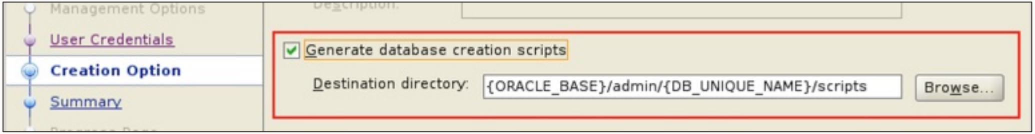
Updating the database time zone file is recommended, not but mandatory

How to create a database with a non-default Time Zone setting?

Time Zone | Custom Setting

DBCA does not offer an implicit solution

Workaround:
Create database creation scripts with DBCA



☑ Generate database creation scripts

Destination directory: {ORACLE_BASE}/admin/{DB_UNIQUE_NAME}/scripts

Set environment variable `ORA_TZFILE`

```
$ export ORA_TZFILE=/u01/app/oracle/product/19/oracore/zoneinfo/timezone_14.dat
```

Create database with script

```
$ ./ORCL.sh
```

Within the same run, you can define time zone and `COMPATIBLE` settings

1.
Intro

**2.
Methods**

- Backup
- Flashback
- Downgrade
- Data Pump
- GoldenGate

3.
Scenarios

4.
Practice

5.
Summary

various methods to

fallback and rollback

BACKUP

Backup

- You should always have backups as one of your fallback methods
- But it should never be the primary fallback method
- Because it takes too long to restore a backup

👍

You should perform a backup before and after an event

If time allows, you should use level 0 backups,
If not, level 1 offers the same level of protection

Pro tip: Cumulative incremental might offer a good balance between time to backup and time to recover

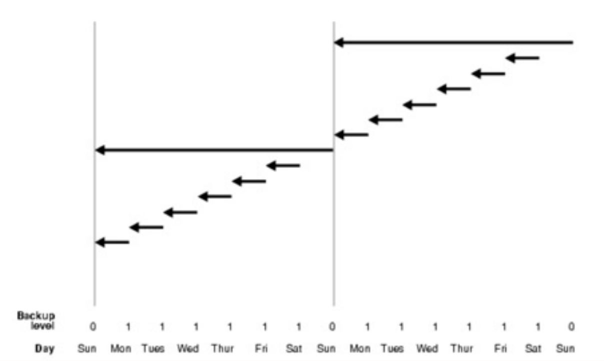
Block Change Tracking can significantly speed up incremental backups

Pro tip: BCT is an Enterprise Edition feature, but requires Active Data Guard if enabled on standby database

Backup | Level 0 / 1 Incremental

Differential:

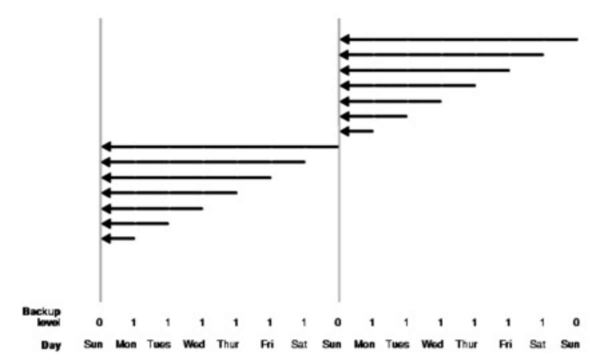
```
RMAN> BACKUP INCREMENTAL LEVEL 1 DATABASE;
```



Backup | Level 0 / 1 Incremental

Cumulative:

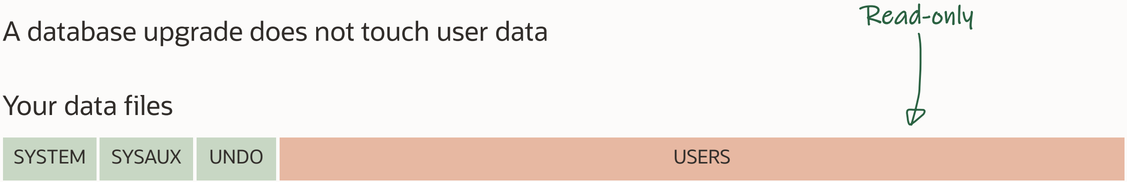
```
RMAN> BACKUP INCREMENTAL LEVEL 1 CUMULATIVE DATABASE;
```



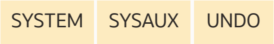
Backup | Partial Offline

A database upgrade does not touch user data

Your data files



Partial offline backup (plus redo log and control files)



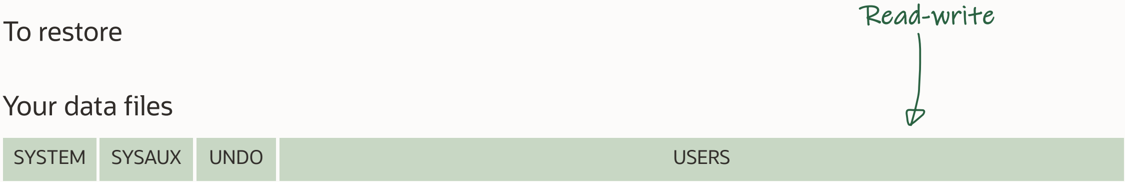
Start upgrade

Pro tip: Works for SE2 and databases in NOARCHIVELOG mode

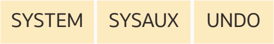
Backup | Partial Offline

To restore

Your data files



Your backup (plus redo log and control files)



Pro tip: Partial offline backup is not applicable for unplug-plug upgrade

various methods to

fallback and rollback

FLASHBACK

Flashback | Overview

- Reverts the database back in time,
  all changes to the database is undone

- Often preferred because it is easy,
  and very fast

- Requires:
  - Enterprise Edition
  - ARCHIVELOG mode
  - 10-20 GB for Flashback Logs
  - `COMPATIBLE` must not be changed

Flashback | Concept

PRE-UPGRADE ENVIRONMENT

```
SQL> create restore point BEFORE_UPG
     guarantee flashback database;
```

POST-UPGRADE ENVIRONMENT

FLASHBACK

```
SQL> shutdown immediate
SQL> startup mount
SQL> flashback database
     to restore point BEFORE_UPG;
SQL> shutdown immediate
```

```
SQL> startup mount
SQL> alter database open resetlogs;
```

👍 Always use guaranteed restore points, and remember to drop them again

Pro tip: Forgetting to drop a GRP will eventually bring the database to a complete halt

various methods to

fallback and rollback

DOWNGRADE

Downgrade | Overview

- Brings the database back to a previous release
- Works days, weeks or months after upgrade
- No data loss
- Requires:
  - `COMPATIBLE` must not be changed
  - Time zone file version must match

Downgrade | Data Dictionary

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

The data dictionary will be different - but compatible

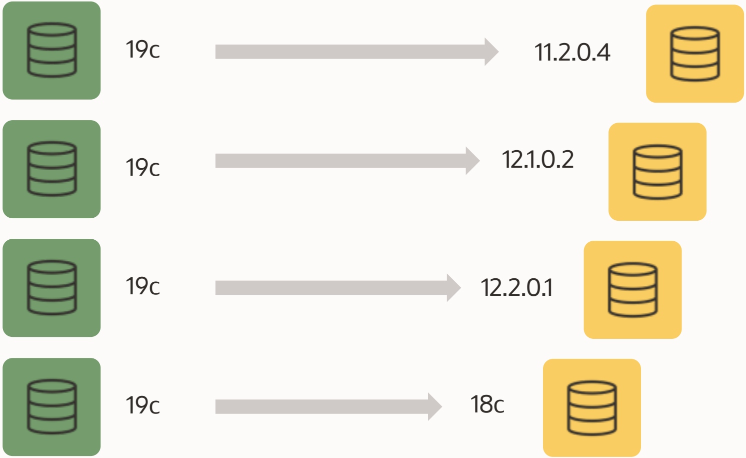
Examples:

- New dictionary tables are not dropped, but truncated
- New indexes are not dropped
- Generally, dropping is avoided

Downgrade reverts only the data dictionary to a state compatible with a previous release

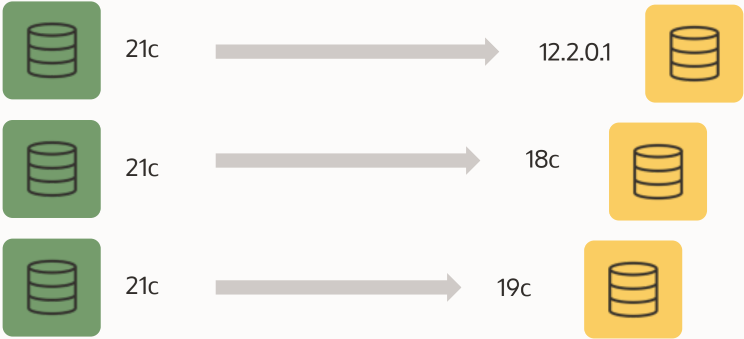
Pro tip: Keep your old Oracle Home available after the upgrade, *just in case*

Downgrade | Oracle Database 19c



For CDB architecture, you can only downgrade to these release
 - not 11.2.0.4

Downgrade | Oracle Database 21c



CDB-only architecture

Downgrade | Fallback Concept

PRE-UPGRADE ENVIRONMENT

POST-UPGRADE ENVIRONMENT

```
SQL> startup downgrade
```

DOWNGRADE

```
SQL> @?/rdbms/admin/catdwgrd.sql
SQL> shutdown immediate
```

```
SQL> startup upgrade
SQL> @?/rdbms/admin/catrelod.sql
SQL> @?/rdbms/admin/utlrp.sql
```

Before downgrading, there is no need to rollback any patches, neither RUs or RURs nor one-off patches

Donwgrade | Pro Tips

Always set these options before using `catdwgrd.sql`

```
SQL> set serverout on
SQL> set termout on
SQL> set timing on
SQL> set echo on
```

And these before using `catrelod.sql`

```
SQL> set termout on
SQL> set timing on
SQL> set echo on
```

And check the log files for errors

```
$ grep '^ORAC-' $ORACLE_HOME/cfgtoollogs/downgrade/catdwgrd*.log
```

various methods to

fallback and rollback

DATA PUMP

Data Pump | Overview

- Universal fallback solution
- When all other fails,
  Data Pump works
- Often not used because considerable downtime is needed
- Requires:
  - Time zone file version must match

ℹ When downgrading, tables using new features are not imported

Data Pump | Overview

Create a dump file compatible with a lower release

```
version=11.2.0.4
```

Other options are

- `COMPATIBLE` (default)
- `LATEST`

Pro tip: See MOS Doc ID 553337.1 for further information

☒

Data Pump over database link does not work for downgrades

Pro tip: Data Pump over database link is using the `NETWORK_LINK` parameter

Data Pump | Fallback Concept

PRE-UPGRADE ENVIRONMENT

POST-UPGRADE ENVIRONMENT

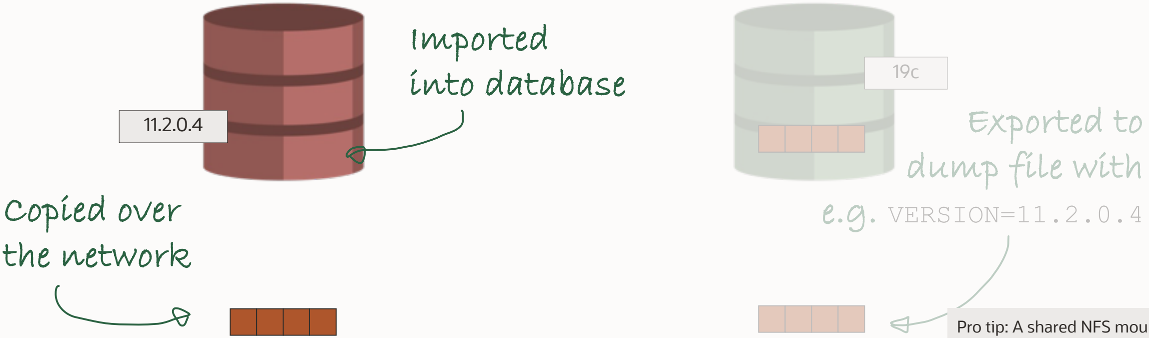
DATA PUMP

```
expdp ... VERSION=<source release>
```

```
impdp ...
```

Pro tip: Keep your old Oracle Home available and create an empty database

Data Pump | Fallback Explained



Pro tip: A shared NFS mount point eliminates the network copy

various methods to

fallback and rollback

GOLDENGATE

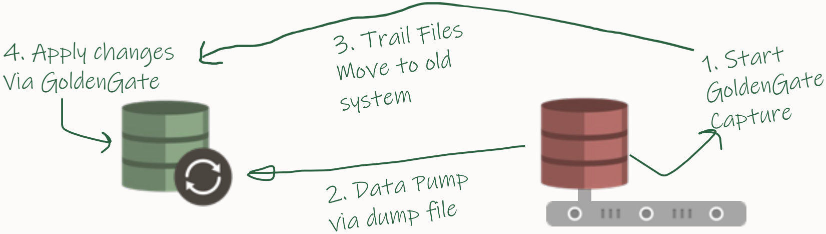
GoldenGate | Overview

- Zero downtime fallback option
- Universal and flexible,
  but complex
- Use in combination with Data Pump

GoldenGate | Fallback

Original database: 11.2.0.4 non-CDB
- Upgrade
- Plug-in
- Convert

Upgraded database: 19c PDB
- Capture
- Export
- Apply

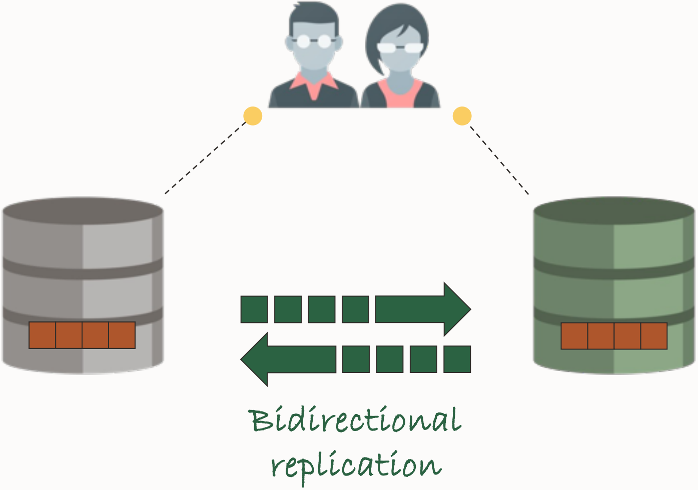


5. Switch clients back
when synchronization
has caught up

```
expdp ... version=11.2.0.4
```

Pro tip: If you keep your old database in place, you can skip the data pump step

GoldenGate | Phased Migration



various methods to

fallback and rollback

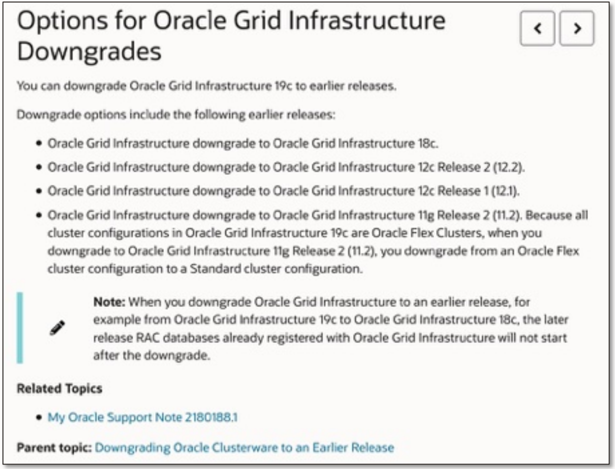
SUMMARY

comparing **FALLBACK** methods

	Backup	Flashback	Downgrade	Data Pump	GoldenGate
Data Loss	x	x			
Use after go-live			x	x	x
Downtime	Considerable	Almost none	Some	Considerable	None
Phased migration					x
Revert PDB conversion				x	x

Whichever method you choose,
be sure to practice, practice and practice

Fallback | Grid Infrastructure Downgrade



Options for Oracle Grid Infrastructure Downgrades

You can downgrade Oracle Grid Infrastructure 19c to earlier releases.

Downgrade options include the following earlier releases:

- Oracle Grid Infrastructure downgrade to Oracle Grid Infrastructure 18c.
- Oracle Grid Infrastructure downgrade to Oracle Grid Infrastructure 12c Release 2 (12.2).
- Oracle Grid Infrastructure downgrade to Oracle Grid Infrastructure 12c Release 1 (12.1).
- Oracle Grid Infrastructure downgrade to Oracle Grid Infrastructure 11g Release 2 (11.2). Because all cluster configurations in Oracle Grid Infrastructure 19c are Oracle Flex Clusters, when you downgrade to Oracle Grid Infrastructure 11g Release 2 (11.2), you downgrade from an Oracle Flex cluster configuration to a Standard cluster configuration.

> **Note:** When you downgrade Oracle Grid Infrastructure to an earlier release, for example from Oracle Grid Infrastructure 19c to Oracle Grid Infrastructure 18c, the later release RAC databases already registered with Oracle Grid Infrastructure will not start after the downgrade.

Related Topics

- My Oracle Support Note 2180188.1

Parent topic: Downgrading Oracle Clusterware to an Earlier Release

Documentation

1. Intro

2. Methods

3. Scenarios

Upgrade
Upgrade with PDB Conversion
Unplug / Plug / Upgrade

4. Practice

5. Summary

typical scenarios for

fallback and rollback

Upgrade

non-CDB to non-CDB
CDB to CDB

Conversion

non-CDB to PDB

Unplug-Plug

PDB to PDB

FLASHBACK

- Preferred
- Data Loss

Flashback | AutoUpgrade

Guaranteed Restore Points

```
upg1.source_home=/u01/app/oracle/product/12.2.0.1
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=CDB1
upg1.restoration=yes
upg1.drop_grp_after_upgrade=no
```

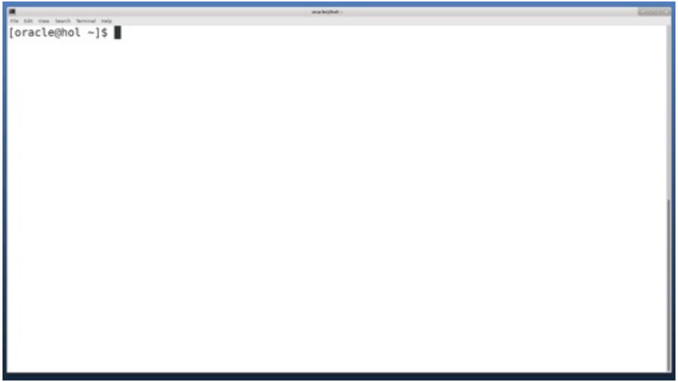
- Default behavior:
  - AutoUpgrade creates GRP except for
    - Standard Edition 2
    - `restoration=no`
  - GRP will be kept
  - GRP needs to be removed manually except for
    - `drop_grp_after_upgrade=yes` will only remove it when upgrade completed successfully

Flashback | AutoUpgrade

AutoUpgrades handles everything, including

- `/etc/oratab`
- Clusterware registration
- Moving files
  - PFile
  - SPFile
  - Password file
  - Etc.

Flashback | AutoUpgrade



```
[oracle@hol ~]$ 
```

Watch on YouTube

What about Data Guard?

👍

You can use Flashback Database without compromising your standby databases

Flashback | Data Guard

- Restore Points are needed on primary and standby databases
- Important: First standby database, then primary
- Always use guaranteed restore points
  - also, on standby databases
- Don't rely on restore point propagation

Flashback | Data Guard

PRIMARY

```
SQL> create restore point ...
     guarantee flashback database;
```

STANDBY

```
SQL> create restore point ...
     Guarantee flashback database;
```

FLASHBACK

PRIMARY

```
SQL> shutdown immediate
SQL> startup mount
SQL> flashback database ...;
SQL> alter database open resetlogs;
```

STANDBY

```
SQL> shutdown immediate
```

```
SQL> startup mount
SQL> flashback database ...
SQL> alter database recover managed
     standby database ...;
```

Data Guard broker does not support flashing back to a previous release

Pro tip: Check the Data Guard Broker documentation for details ℹ Data Guard broker must be shut down during a flashback to a previous release

Flashback | Data Guard

You should either:

- Backup the broker configuration files before the upgrade

```
SQL> select value from v$parameter where name like 'dg\_broker\_config\_%' escape '\';
```

- Recreate the Data Guard broker configuration

```
DGMGRL> create configuration ...
DGMGRL> add database ...
DGMGRL> enable configuration
```

In Oracle Database 19c you can `EXPORT CONFIGURATION` and `IMPORT CONFIGURATION` in Data Guard CLI (DGMGRL)

What about RAC?

👍

You can use Flashback Database on a RAC database

Only one instance should be running during a flashback operation

Flashback | RAC

Stop database (all instances) and mount one instance

```
$ srvctl stop database -d $ORACLE_UNQNAME
$ srvctl start instance -d $ORACLE_UNQNAME -i $ORACLE_SID -o MOUNT
```

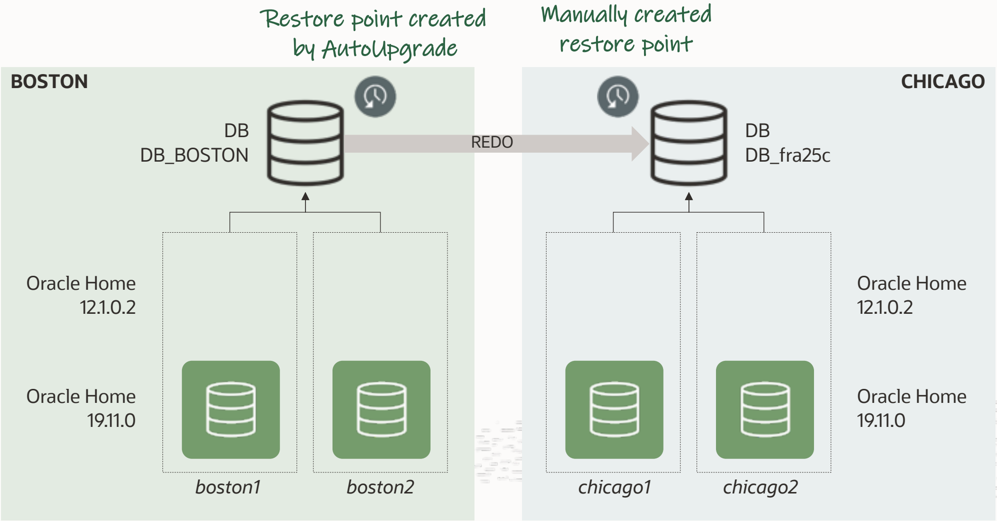
Flash back the database

```
SQL> flashback database ...            --Higher release Oracle Home
SQL> alter database open resetlogs;    --Lower release Oracle Home
```

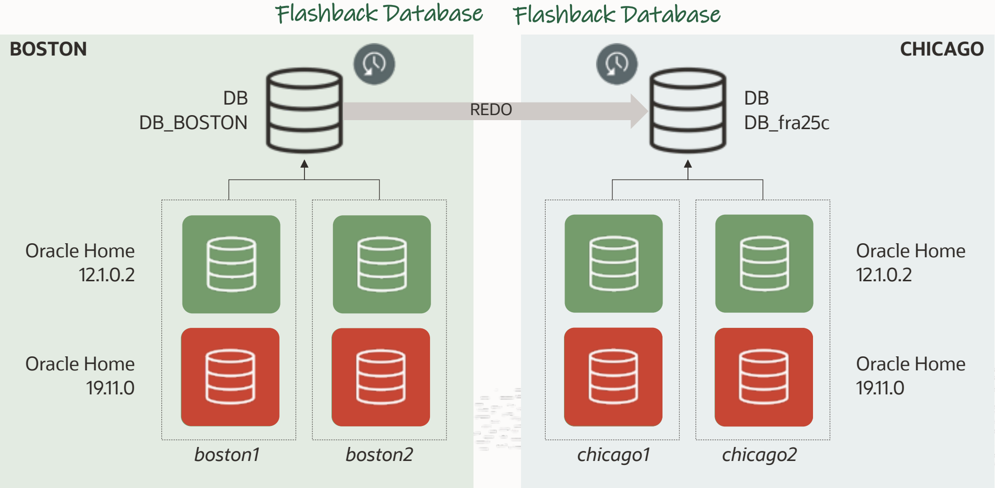
Start database (all instances)

```
$ srvctl stop instance -d $ORACLE_UNQNAME -i $ORACLE_SID
$ srvctl downgrade database -d $ORACLE_UNQNAME -o <lower_release_home> -t 12.2.0.1
$ srvctl start database -d $ORACLE_UNQNAME
```

Flashback | Demo



Flashback | Demo



Flashback | Demo

STOP STANDBY

Flashback Primary

Flashback Standby

Re-enable Data Guard

Watch on YouTube

DOWNGRADE

- No Data Loss

Downgrade | Releases

You can downgrade from Oracle Database 19 to:

- 18
- 12.2
- 12.1.0.2
- 11.2.0.4 (non-CDB only)

Pro tip: Check the Upgrade Guide for details

Perform level 0 backup before and after downgrading, or if time does not allow, at least a level 1 backup

It is recommended to install the latest Release Update in higher and lower release Oracle Homes before you start the downgrade

Check MOS notes 2539751.1 and 2548962.1
for important patches to apply before downgrading

Downgrade | Statistics

What about statistics?

- Dictionary statistics
  - Gather immediately after downgrade

- Fixed objects statistics
  - Gather when database is warmed-up

- Optimizer statistics
  - Regather stale statistics

What about Data Guard?

You can downgrade a database
without compromising your standby database

Downgrade | Data Guard

PRIMARY

```
SQL> startup downgrade

$ ./dbdowngrade
```

Restart database in lower release Oracle Home

```
SQL> @catrelod
SQL> @utlrp

$ datapatch -verbose
```

STANDBY

Wait for all redo to be applied

Restart database in lower release Oracle Home

Wait for all redo to be applied

Data Guard broker does not support downgrading

Pro tip: Check the Data Guard Broker documentation for details

Data Guard broker must be shut down during a downgrade

Downgrade | Data Guard

You should either:

- Backup the broker configuration files before the upgrade

```
SQL> select value from v$parameter where name like 'dg\_broker\_config\_%';
```

- Recreate the Data Guard broker configuration

```
DGMGRL> create configuration ...
DGMGRL> add database ...
DGMGRL> enable configuration
```

In Oracle Database 19c you can `EXPORT CONFIGURATION` and `IMPORT CONFIGURATION` in Data Guard CLI (DGMGRL)

What about RAC?

👍 It is possible to downgrade a RAC database

During a downgrade, the parameter `CLUSTER_DATABASE` must be set to `FALSE`

Downgrade | RAC

Stop database (all instances) and start one instance in higher release Oracle Home

```
SQL> alter system set cluster_database=false scope=spfile sid='*';

$ srvctl stop database -d $ORACLE_UNQNAME

SQL> startup downgrade
```

Downgrade

```
$ ./dbdowngrade
```

Downgrade | RAC

Restart one instance in lower release Oracle Home

```
SQL> startup upgrade
```

Reload, recompile and datapatch

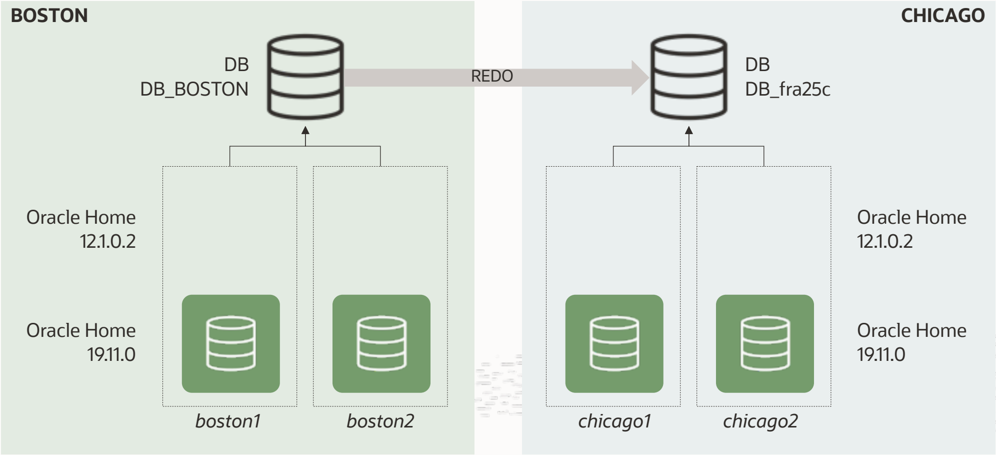
```
SQL> @catrelod
SQL> @utlrp

$ datapatch -verbose
```

Stop instance, clusterware downgrade and restart database (all instances)

```
SQL> alter system set cluster_database=true scope=spfile sid='*';
SQL> shutdown immediate

$ srvctl downgrade database -d $ORACLE_UNQNAME -o <lower_release_home> -t 12.2.0.1
$ srvctl start database -d $ORACLE_UNQNAME
```

MOS note 25489621 contains detailed step-by-step instructions on downgrade

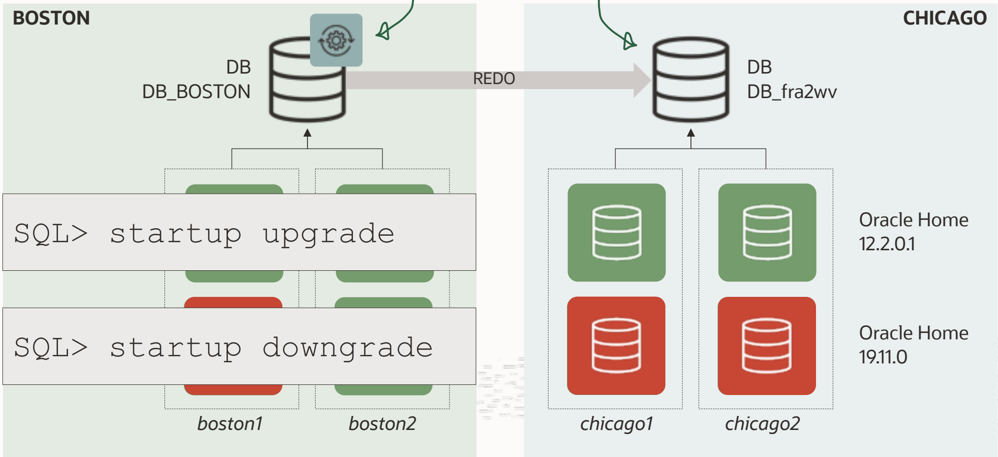
Downgrade | Demo



Downgrade | Demo

Reload
PL/SQL
Downgrade

All redo
must be applied



```
SQL> startup upgrade
```

```
SQL> startup downgrade
```

Downgrade | Demo

Watch on YouTube

Bug | Flashback and downgrade

Restarting database in previous Oracle Home

```
$ $ORACLE_HOME/bin/srvctl start database -d $ORACLE_UNQNAME
PRCR-1079 : Failed to start resource ora.db_fra25c.db
CRS-5017: The resource action "ora.db_fra25c.db start" encountered the following error:
ORA-01078: failure in processing system parameters
LRM-00101: unknown parameter name '__unified_pga_pool_size'
. For details refer to "(:CLSN00107:)" in
"/u01/app/grid/diag/crs/chicago2/crs/trace/crsd_oraagent_oracle.trc".
```

Create PFile in idle instance, then start up on corrected PFile

```
SQL> create pfile='/tmp/init.ora' from spfile

...

SQL> startup pfile='/tmp/init.ora';
```

Bug | Flashback and downgrade

- Parameter unknown to previous release
- Parameter gets written to SPFile
  when database is running new release
- New parameters should not be written
  to SPFile unless `COMPATIBLE` is raised
- Bug 30072483

typical scenarios for

fallback and rollback

Upgrade

non-CDB to non-CDB
CDB to CDB

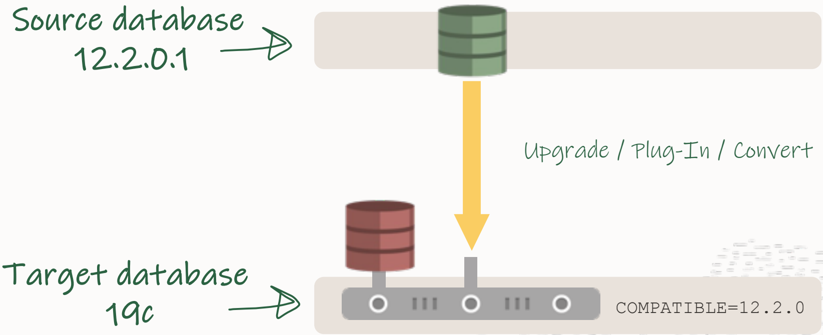
Conversion

non-CDB to PDB

Unplug-Plug

PDB to PDB

12.2 to 19c | Concept



12.2 to 19c | AutoUpgrade

Upgrade - and plug in

```
upg1.source_home=/u01/app/oracle/product/12.2.0.1
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=db12
upg1.target_cdb=cdb19c
```

Command

```
java -jar autoupgrade.jar -config DB19.cfg -mode deploy
```

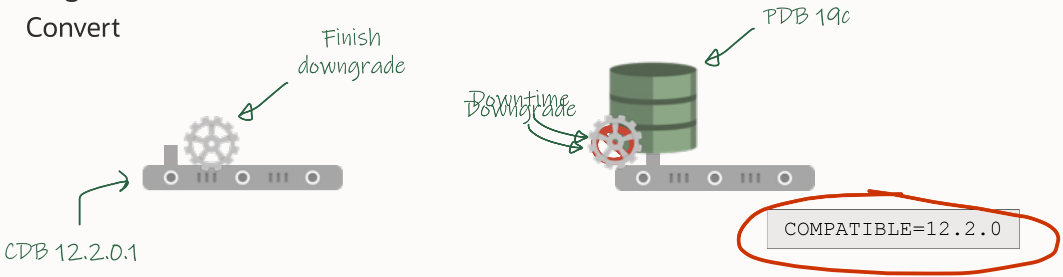
Blog post: Oracle AutoUpgrade between two servers – and Plugin?

Pro tip: You can also plug in manually and upgrade PDB with `dbupgrade -c DB19`

Fallback | Lower Release CDB

Source database: 12.2 CDB

- Upgrade
- Plug-in
- Convert



COMPATIBLE=12.2.0

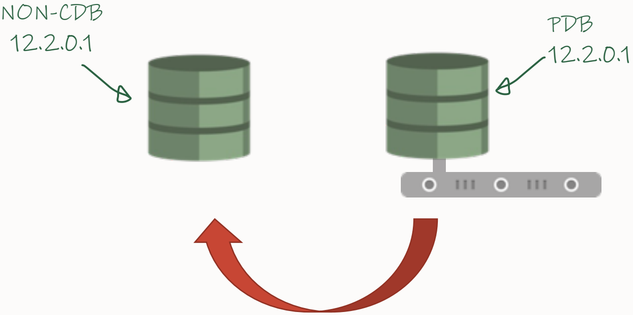
Fallback | PDB Downgrade

Downgrade works for CDB/PDB entirely as well as for single/multiple PDBs

- Manual tasks
  - `catdwgrd.sql` in current (*after* upgrade) environment
  - `catrelod.sql` in previous (*before* upgrade) environment
  - Don't change `COMPATIBLE`

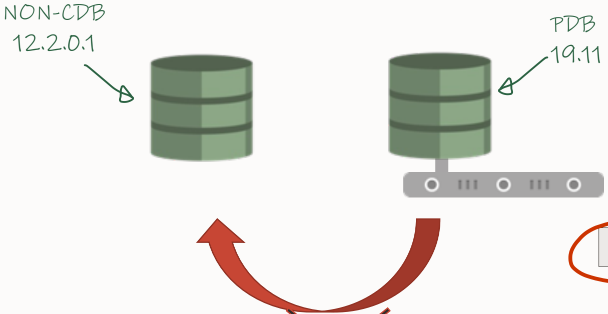
- MOS Note: 2172185.1
  How to Downgrade a Single Pluggable Oracle Database (PDB) to previous release

Fallback | Lower Release Non-CDB



~~@$ORACLE_HOME/rdbms/admin/pdb_to_noncdb.sql~~

Fallback | Lower Release Non-CDB



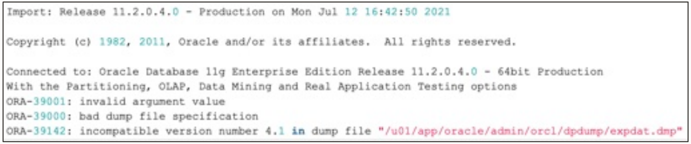
~~Transportable TTS~~

```
COMPATIBLE=12.2.0
```

Fallback | Transportable TTS

Not possible to lower releases

- Blog post



```
Import: Release 11.2.0.4.0 - Production on Mon Jul 12 16:42:50 2021

Copyright (c) 1982, 2011, Oracle and/or its affiliates.  All rights reserved.

Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
ORA-39001: invalid argument value
ORA-39000: bad dump file specification
ORA-39142: incompatible version number 4.1 in dump file "/u01/app/oracle/admin/orcl/dpdump/expdat.dmp"
```

"While a transport tablespace to a lower release may work in some cases, it is not an action that Oracle supports."

Compatibility and New Features when Transporting Tablespaces with Export and Import (Doc ID 291024.1)

How to ...

ℹ Non-CDB to PDB conversion is irreversible

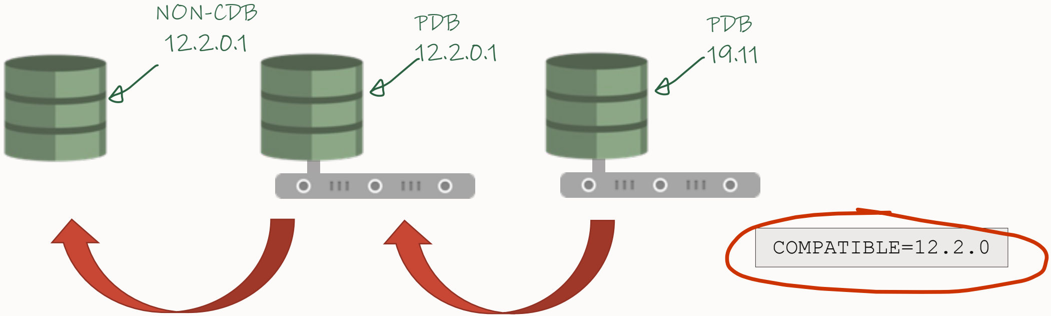
Fallback | Concept

Move to Multitenant architecture = Migration

1. Usual fallback techniques don't work
   - No downgrade
   - No flashback to Restore Point

2. Only possible fallback options
   - Data Pump
   - GoldenGate
   - Transportable Tablespaces (only same version)
   - Plug into a source-version CDB

Fallback | Lower Release Non-CDB



COMPATIBLE=12.2.0

Transportable TTS

Downgrade/Unplug/Plug/Reload

Fallback | Full Process

```
[oracle@odb-12-2 ~]$ 
```

Watch on YouTube

typical scenarios for

fallback and rollback

Upgrade

non-CDB to non-CDB
CDB to CDB

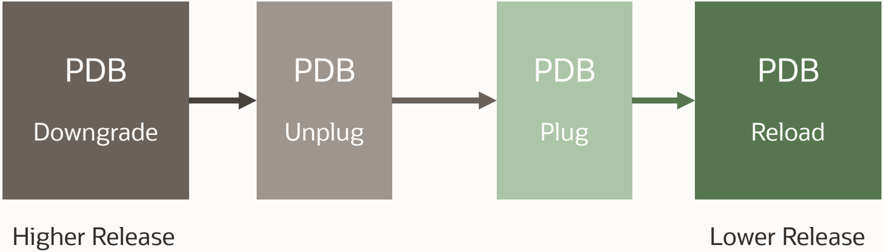
Conversion

non-CDB to PDB

Unplug-Plug

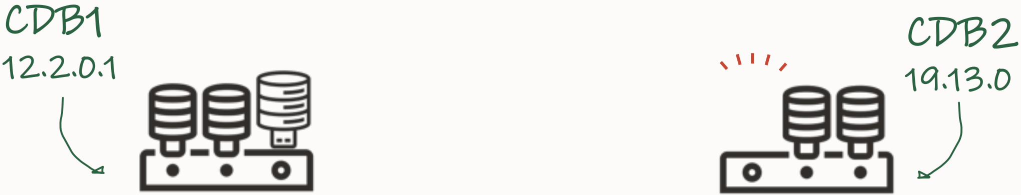
PDB to PDB

Downgrade-Unplug-Plug | Concept



Downgrade-Unplug-Plug | Silent Compatible Change

Beware of the silent `COMPATIBLE` change in Multitenant



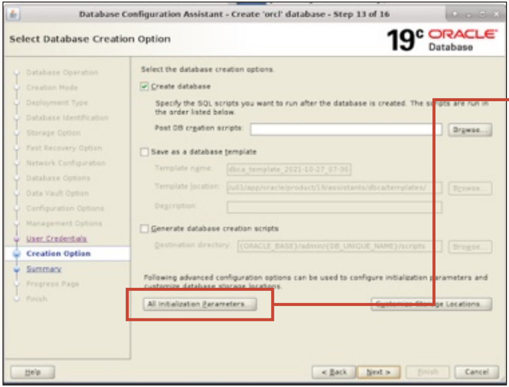
`COMPATIBLE=12.2.0`

`COMPATIBLE=19.1.0`

Downgrade-Unplug-Plug | Preparation

Target CDB must have the same `COMPATIBLE` setting

- CUSTOM database creation



Database Configuration Assistant - Create 'orcl' database - Step 13 of 16

Select Database Creation Option

Select the database creation options.

☑ Create database

Specify the SQL scripts you want to run after the database is created. The scripts are run in the order listed below.

Post DB creation scripts:

□ Save as a database template

Template name: dbca_template_2021-10-27_07-36

Template location: /u01/app/oracle/product/19/assistants/dbca/templates/

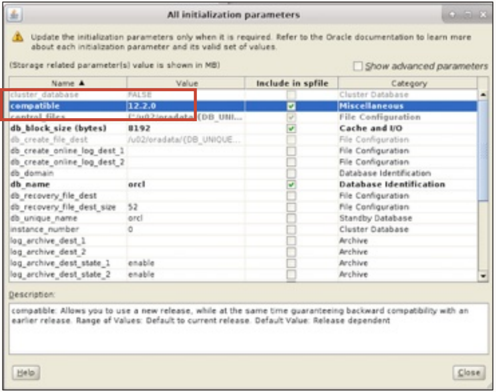
Description:

□ Generate database creation scripts

Destination directory: {ORACLE_BASE}/admin/{DB_UNIQUE_NAME}/scripts

Following advanced configuration options can be used to configure initialization parameters and customize database storage locations.

All Initialization Parameters...
Customize Storage Locations...
Help
< Back
Next >
Finish
Cancel



All initialization parameters

Update the initialization parameters only when it is required. Refer to the Oracle documentation to learn more about each initialization parameter and its valid set of values.

(Storage related parameter(s) value is shown in MB)

□ Show advanced parameters

Name ▲	Value	Include in spfile	Category
cluster_database	FALSE	□	Cluster Database
compatible	12.2.0	☑	Miscellaneous
control_files	("/u02/oradata/{DB_UNI...	☑	File Configuration
db_block_size (bytes)	8192	☑	Cache and I/O
db_create_file_dest	/u02/oradata/{DB_UNIQUE...	□	File Configuration
db_create_online_log_dest_1		□	File Configuration
db_create_online_log_dest_2		□	File Configuration
db_domain		□	Database Identification
db_name	orcl	☑	Database Identification
db_recovery_file_dest		□	File Configuration
db_recovery_file_dest_size	52	□	File Configuration
db_unique_name	orcl	□	Standby Database
instance_number	0	□	Cluster Database
log_archive_dest_1		□	Archive
log_archive_dest_2		□	Archive
log_archive_dest_state_1	enable	□	Archive
log_archive_dest_state_2	enable	□	Archive

Description:

compatible: Allows you to use a new release, while at the same time guaranteeing backward compatibility with an earlier release. Range of Values: Default to current release. Default Value: Release dependent

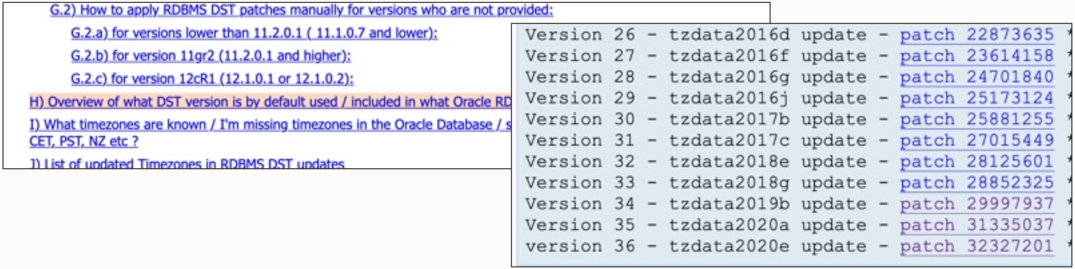
Downgrade-Unplug-Plug | Preparation

Source and target CDB must have the identical time zone files present

- Check time zone file in source and target CDBs

```
select * from V$TIMEZONE_FILE;
```

- Install matching time zone patch to source or target home
  - MOS Note:412160.1



G.2) How to apply RDBMS DST patches manually for versions who are not provided:

G.2.a) for versions lower than 11.2.0.1 (11.1.0.7 and lower):

G.2.b) for version 11gr2 (11.2.0.1 and higher):

G.2.c) for version 12cR1 (12.1.0.1 or 12.1.0.2):

H) Overview of what DST version is by default used / included in what Oracle RD

I) What timezones are known / I'm missing timezones in the Oracle Database / s CET, PST, NZ etc ?

J) List of updated Timezones in RDBMS DST updates

```
Version 26 - tzdata2016d update - patch 22873635
Version 27 - tzdata2016f update - patch 23614158
Version 28 - tzdata2016g update - patch 24701840
Version 29 - tzdata2016j update - patch 25173124
Version 30 - tzdata2017b update - patch 25881255
Version 31 - tzdata2017c update - patch 27015449
Version 32 - tzdata2018e update - patch 28125601
Version 33 - tzdata2018g update - patch 28852325
Version 34 - tzdata2019b update - patch 29997937
Version 35 - tzdata2020a update - patch 31335037
version 36 - tzdata2020e update - patch 32327201
```

Downgrade-Unplug-Plug | Preparation

Source and target CDB must have the identical time zone files present

- Apply time zone patch to CDB$ROOT

```
SQL> start $ORACLE_HOME/rdbms/admin/utltz_upg_check.sql
SQL> start $ORACLE_HOME/rdbms/admin/utltz_upg_apply.sql
```

  - Attention: Restart will happen

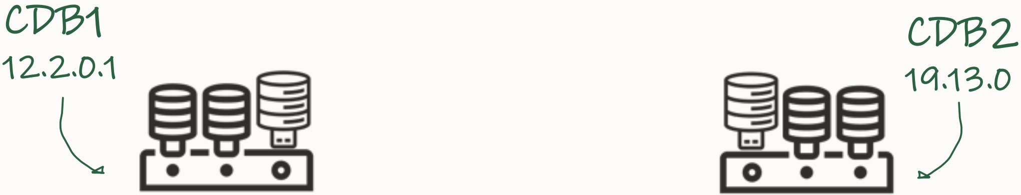
- Check time zone file version again

```
select * from V$TIMEZONE_FILE;
```

Downgrade-Unplug-Plug | No Compatible Change

Both CDBs must have identical `COMPATIBLE` settings

- Downgrade possibility



`COMPATIBLE=12.2.0`

`COMPATIBLE=12.2.0`

Downgrade-Unplug-Plug | PDB Downgrade

Downgrade PDB in higher release CDB

- Cleanup unified audit trail

```
SELECT COUNT(*) FROM UNIFIED_AUDIT_TRAIL;
```

```
exec DBMS_AUDIT_MGMT.CLEAN_AUDIT_TRAIL(DBMS_AUDIT_MGMT.AUDIT_TRAIL_UNIFIED, FALSE);
```

Downgrade-Unplug-Plug | PDB Downgrade

Downgrade PDB in higher release CDB

- Shutdown PDB
- Start PDB in `DOWNGRADE` mode

```
alter pluggable database PDB1 open downgrade;
```

Downgrade-Unplug-Plug | PDB Downgrade

Downgrade the PDB

- Logs default: `$ORACLE_HOME/cfgtoollogs/downgrade`

```
$> chmod +x $ORACLE_HOME/bin/dbdowngrade
```

```
$> dbdowngrade -c 'PDB1'
```

Downgrade-Unplug-Plug | PDB Unplug

Unplug the PDB

- Shutdown
- Unplug

```
alter pluggable database PDB1 close;
```

```
alter pluggable database PDB1 unplug into '/tmp/pdb1.xml';
```

Downgrade-Unplug-Plug | PDB Plugin

Plugin the PDB into the source CDB

- Cleanup potentially existing files in destination
- `FILE_NAME_CONVERT` will trigger the copy operation

```
create pluggable database PDB1 using '/tmp/pdb1.xml'
  file_name_convert=('CDB2','CDB1');
```

Downgrade-Unplug-Plug | PDB Reload

Reload all packages and code

- Open the PDB in `UPGRADE` mode
- Start reload script `catrelod.sql`

```
alter pluggable database PDB1 open upgrade;
```

```
alter session set container=PDB1;
```

```
start ?/rdbms/admin/catrelod.sql
```

Pro tip: Spool the output of `catrelod.sql` into a logfile

Downgrade-Unplug-Plug | PDB Recompilation

Recompilation

- Start `utlrp.sql`

```
@$ORACLE_HOME/rdbms/admin/utlrp.sql
```

Downgrade-Unplug-Plug | Finalize

Stop and restart the PDB

```
alter pluggable database PDB1 close;
```

```
alter pluggable database PDB1 open;
```

Plan carefully and ensure `COMPATIBLE` and time zone are equal between source and target CDBs – and test it!

Can you use a refreshable clone PDB for downgrades?

Downgrade | Refreshable Clone PDB

PDB has been plugged in and upgrade to 19c

- Refreshable clone back into previous release

```
SQL> create pluggable database PDB1 from PDB1@clonemypdb
     REFRESH MODE MANUAL file_name_convert=('CDB2','CDB1');
create pluggable database PDB1 from PDB1@clonemypdb REFRESH MODE MANUAL ...
*
ERROR at line 1:
ORA-65156: pluggable database version 19.0.0.0.0 not allowed
```

  - You can clone only to the same or a higher release CDB

Unplug-Plug-Downgrade | More Information

- MOS Note: 2421060.1
  How to Downgrade a Single Pluggable Oracle Database (PDB) from to previous release

- MOS Note: 2172185.1
  How to Downgrade a Single Pluggable Oracle Database (PDB) to previous release

Our Hands-On Lab has 15 guided exercise taking you from A-Z using five databases and three Oracle Homes

Hands-On Lab

	Database 11.2.0.4	Database 12.2	Database 19c
Upgrade to 19c AutoUpgrade	UPGR →	→	
Convert to PDB AutoUpgrade			UPGR → → CDB2
Migrate to 19c Full transportable export/import	FTEX →	→	CDB2 / PDB2
Unplug-plug upgrade AutoUpgrade		PDB3 →	CDB2
Upgrade to 19c AutoUpgrade		DB12 →	

TWO

OPTIONS

VIRTUAL BOX

Self-contained image

Runs on your laptop

70 GB

LIVELABS

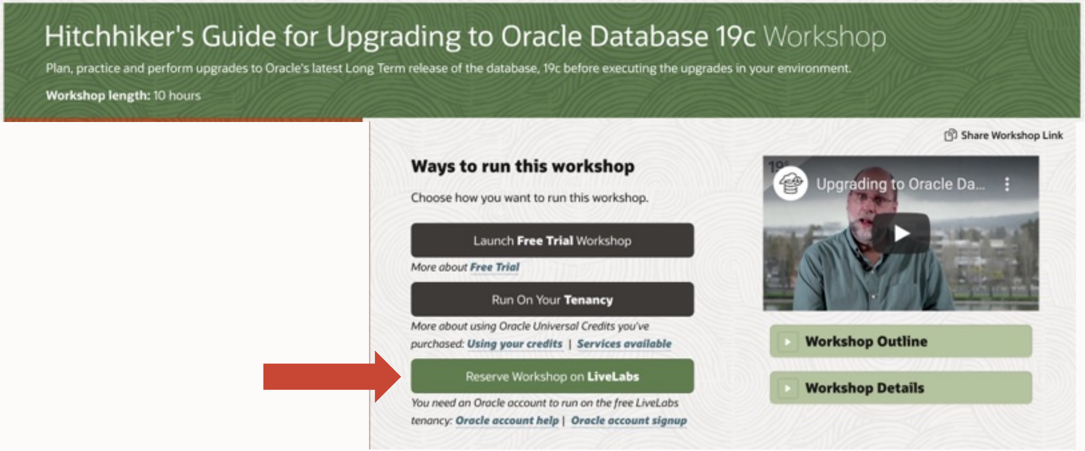
Runs from browser

Runs in OCI

Using Free Tier account

LiveLabs

Hitchhiker's Guide to Database Upgrades



Hitchhiker's Guide for Upgrading to Oracle Database 19c Workshop

Plan, practice and perform upgrades to Oracle's latest Long Term release of the database, 19c before executing the upgrades in your environment.

Workshop length: 10 hours

Share Workshop Link

Ways to run this workshop

Choose how you want to run this workshop.

Launch **Free Trial** Workshop

*More about **Free Trial***

Run On Your **Tenancy**

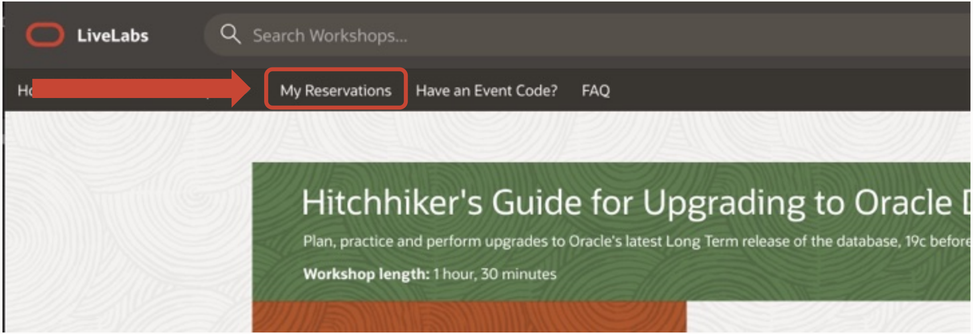
*More about using Oracle Universal Credits you've purchased: **Using your credits** | **Services available***

Reserve Workshop on **LiveLabs**

*You need an Oracle account to run on the free LiveLabs tenancy: **Oracle account help** | **Oracle account signup***

LiveLabs

6-10 minutes later:



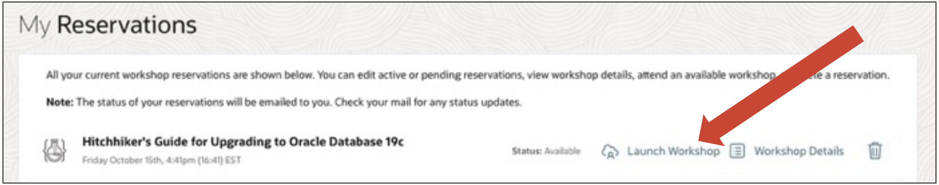
Hitchhiker's Guide for Upgrading to Oracle D

Plan, practice and perform upgrades to Oracle's latest Long Term release of the database, 19c before

Workshop length: 1 hour, 30 minutes

LiveLabs

Then launch the workshop and access it within your browser (noVNC link)

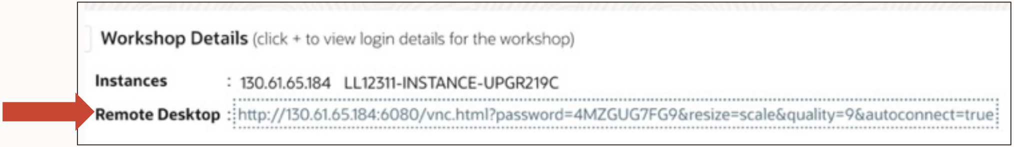


My Reservations

All your current workshop reservations are shown below. You can edit active or pending reservations, view workshop details, attend an available workshop or delete a reservation.

Note: The status of your reservations will be emailed to you. Check your mail for any status updates.

Hitchhiker's Guide for Upgrading to Oracle Database 19c
Friday October 15th, 4:41pm (16:41) EST



Workshop Details (click + to view login details for the workshop)

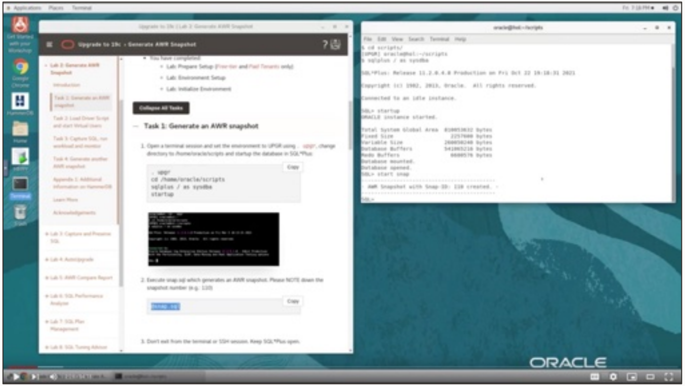
Instances : 130.61.65.184 LL12311-INSTANCE-UPGR219C

Remote Desktop : http://130.61.65.184:6080/vnc.html?password=4MZGUG7FG9&resize=scale&quality=9&autoconnect=true

LiveLabs

No installation of any tool is required if you use the Green Button lab

- Instructions are in the browser inside the lab


Applications
Places
Terminal
Get Started with your Workshop
Google Chrome
HammerDB
Home
Terminal
Trash
Upgrade to 19c | Lab 2: Generate AWR Snapshot
Upgrade to 19c › Generate AWR Snapshot
Lab 2: Generate AWR Snapshot
Introduction
Task 1: Generate an AWR snapshot
Task 2: Load Driver Script and start Virtual Users
Task 3: Capture SQL, run workload and monitor
Task 4: Generate another AWR snapshot
Appendix 1: Additional Information on HammerDB
Learn More
Acknowledgements
Lab 3: Capture and Preserve SQL
Lab 4: AutoUpgrade
Lab 5: AWR Compare Report
Lab 6: SQL Performance Analyzer
Lab 7: SQL Plan Management
Lab 8: SQL Tuning Advisor
Collapse All Tasks
Copy
oracle@hol:~/scripts
File
Edit
View
Search
Terminal
Help
ORACLE

- You have completed:
  - Lab: Prepare Setup (Free-tier and Paid Tenants only)
  - Lab: Environment Setup
  - Lab: Initialize Environment

Task 1: Generate an AWR snapshot

1. Open a terminal session and set the environment to UPGR using . upgr, change directory to /home/oracle/scripts and startup the database in SQL*Plus:

```
. upgr
cd /home/oracle/scripts
sqlplus / as sysdba
startup
```

2. Execute snap.sql which generates an AWR snapshot. Please NOTE down the snapshot number (e.g.: 110)

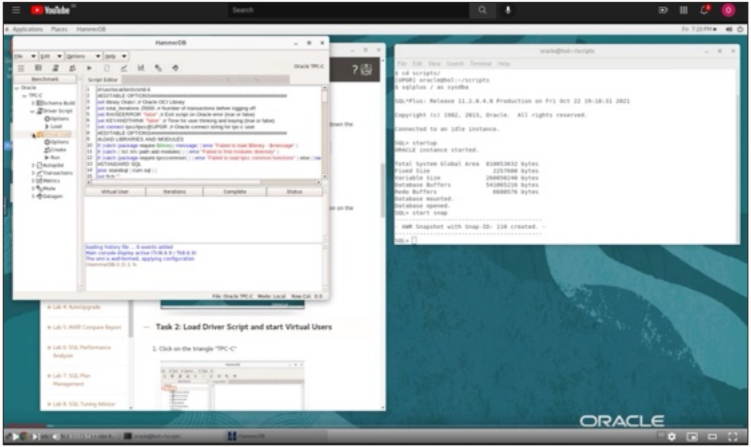
```
@snap.sql
```

3. Don't exit from the terminal or SSH session. Keep SQL*Plus open.

```
$ cd scripts/
[UPGR] oracle@hol:~/scripts
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.4.0 Production on Fri Oct 22 19:18:31 2021

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

Connected to an idle instance.

SQL> startup
ORACLE instance started.

Total System Global Area  [illegible] bytes
Fixed Size                  2257000 bytes
Variable Size             [illegible] bytes
Database Buffers          [illegible] bytes
Redo Buffers                [illegible] bytes
Database mounted.
Database opened.
SQL> start snap
------------------------------------------
- AWR Snapshot with Snap-ID: 110 created. -
------------------------------------------
SQL>
```

LiveLabs

Find a narrated 54 min video covering the entire lab on YouTube

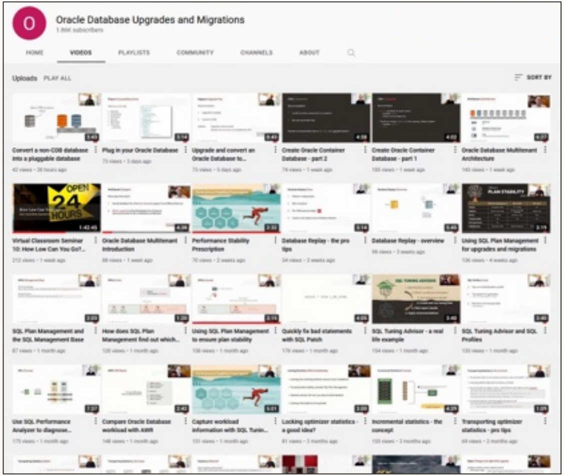


```
$ cd scripts/
[UPGR] oracle@hol:~/scripts
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.4.0 Production on Fri Oct 22 19:18:31 2021

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

Connected to an idle instance.

SQL> startup
ORACLE instance started.

Total System Global Area  810053632 bytes
Fixed Size                  2257600 bytes
Variable Size             260050240 bytes
Database Buffers          541065216 bytes
Redo Buffers                6680576 bytes
Database mounted.
Database opened.
SQL> start snap
------------------------------------------
- AWR Snapshot with Snap-ID: 110 created. -
------------------------------------------
SQL> 
```

Task 2: Load Driver Script and start Virtual Users

1. Click on the triangle "TPC-C"

YouTube | Oracle Database Upgrades and Migrations



Oracle Database Upgrades and Migrations
1.86K subscribers

- Convert a non-CDB database into a pluggable database — 42 views · 20 hours ago
- Plug in your Oracle Database — 73 views · 3 days ago
- Upgrade and convert an Oracle Database to... — 73 views · 5 days ago
- Create Oracle Container Database - part 2 — 74 views · 1 week ago
- Create Oracle Container Database - part 1 — 105 views · 1 week ago
- Oracle Database Multitenant Architecture — 143 views · 1 week ago
- Virtual Classroom Seminar 10: How Low Can You Go?... — 212 views · 1 week ago
- Oracle Database Multitenant Introduction — 88 views · 1 week ago
- Performance Stability Prescription — 70 views · 2 weeks ago
- Database Replay - the pro tips — 54 views · 3 weeks ago
- Database Replay - overview — 98 views · 3 weeks ago
- Using SQL Plan Management for upgrades and migrations — 136 views · 4 weeks ago
- SQL Plan Management and the SQL Management Base — 87 views · 1 month ago
- How does SQL Plan Management find out which... — 120 views · 1 month ago
- Using SQL Plan Management to ensure plan stability — 158 views · 1 month ago
- Quickly fix bad statements with SQL Patch — 176 views · 1 month ago
- SQL Tuning Advisor - a real life example — 154 views · 1 month ago
- SQL Tuning Advisor and SQL Profiles — 135 views · 1 month ago
- Use SQL Performance Analyzer to diagnose... — 175 views · 1 month ago
- Compare Oracle Database workload with AWR — 148 views · 1 month ago
- Capture workload information with SQL Tunin... — 131 views · 1 month ago
- Locking optimizer statistics - a good idea? — 81 views · 2 months ago
- Incremental statistics - the concept — 155 views · 2 months ago
- Transporting optimizer statistics - pro tips — 69 views · 2 months ago

Link

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

SCAN ME

THANK YOU

Visit our blogs:

https://MikeDietrichDE.com

https://DOHdatabase.com

https://www.dbarj.com.br/en

THANK YOU

Webinars:

https://MikeDietrichDE.com/videos

YouTube channel:

OracleDatabaseUpgradesandMigrations

THANK YOU

HANDS-ON LAB

Instructions
Live Labs
Guided tour

THANK YOU