

Oracle Multitenant Database Upgrade Internals



Roy F. Swonger

Senior Director & Product Manager

Database Upgrade & Utilities

Oracle Corporation

ORACLE

Updated: 09-DEC-2014


Oracle Multitenant Database Upgrade Internals

- 1 Introduction
- 2 Database Upgrade News
- 3 Oracle Multitenant Overview
- 4 Multitenant Upgrade
- 5 Inside catctl.pl – and Options
- 6 Performance Figures
- 7 Wrap Up



Reference Involvement

Customer Success
Your success is our aim!



Non-binding Reference Proposal
for

Oracle Database Upgrade Development: Customer Reference Program



Company Name and Logo You agree that your company name (where necessary including your logo) will be called "customer" or "reference customer" for relevant products and projects. Typically we will then use your company name during internal and external presentations or events.

Agree: Yes No

Comments:

Customer Quote The quote is a brief statement about how your company has achieved a technological head start or has benefited economically from using Oracle products. Example: "Since we have been using Oracle iProcurement our acquisition costs have fallen by around 20%" – John Doe, CIO, IT Company Inc. With your agreement we will use your quote, for instance, in product brochures.

Agree: Yes No

Comments:

Reference Involvement: Results



Real World Checkpoint

Customer: **Payback GmbH**

- Belongs to Loyalty Partner GmbH which belongs to American Express
- HQ in Munich, Germany
- Develops and operates professional customer loyalty programs based on customized IT solutions
- Provider for Payback
- Active in Germany, Poland, India and Mexico

Buttons: Project, Constraints, Preparation, Upgrade, Success?, Remarks

199

Real World Checkpoint

Project: **Migrate 7TB / 1.5TB from HP-UX to Exadata V1**

- Cross platform, cross Endianess, 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

Buttons: Project, Constraints, Preparation, Upgrade, Success?, Remarks

200

Real World Checkpoint

Project: **Move everything in less than 24 hrs**

- Network bottlenecks
- Customer initial
- 3GB/sec

Buttons: Project, Constraints, Preparation, Upgrade, Success?, Remarks

201

"The new parallel upgrade script promises to drastically reduce downtime due to planned maintenance. We saw a 37% improvement over the previous upgrade process in our environment."

Harald Stefan
Leiter Datenbanken
Payback GmbH



Real World Checkpoint

Project: **Setup**

Diagram: PROCD Restore Upgrade SWING HP-UX to RISC HP-UX to RISC OL Exdt

Buttons: Project, Constraints, Preparation, Upgrade, Success?, Remarks

202

Real World Checkpoint

Project: **Test migrations**

Diagram: PROCD SWING HP-UX to RISC HP-UX to RISC OL Exdt

Buttons: Project, Constraints, Preparation, Upgrade, Success?, Remarks

203

Real World Checkpoint

Project: **Parallel live load**

Diagram: PROCD HP-UX to RISC HP-UX to RISC OL Exdt

Buttons: Project, Constraints, Preparation, Upgrade, Success?, Remarks

204

Real World Checkpoint

Project: **Final test became LIVE migration**

Diagram: PROCD SWING HP-UX to RISC HP-UX to RISC OL Exdt

Buttons: Project, Constraints, Preparation, Upgrade, Success?, Remarks

205

Real World Checkpoint

Project: **Live? And alive?**

- Yes! Go-live in early November 2009
- Two weeks earlier than proposed
- Total upgrade and migration time: ~20 hours
- 8 hours: Reason and strategy
- 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 60%
- User complaints about too fast performance... really!

Buttons: Project, Constraints, Preparation, Upgrade, Success?, Remarks

206

Real World Checkpoint

Project: **Final piece of S...**

Diagram: PROCD SWING HP-UX to RISC HP-UX to RISC OL Exdt

Buttons: Project, Constraints, Preparation, Upgrade, Success?, Remarks

207

>48000 Downloads since June 2013



Database Upgrade Blog

- <http://blogs.oracle.com/UPGRADE>

Upgrade your Database - NOW!
Ease your Oracle Database upgrades - Best Practices, Workshops, Projects ...

Recent Posts


- Oracle Database 12.1.0.2 EE for HP, AIX and zLinux available
- Incremental Statistics Collection improved in Oracle 12c
- Sleeping Beauties - Upgrade to 11.2.0.4 can be slow
- Beijing, Seoul - and OTN Tour Tokyo - just in one week!
- Premier Support for Oracle 11.2 will end soon ...
- Upgrade to Oracle Database 12c now!
- Utrecht holds the new EMEA record now!
- Plus: Updated Slide Deck
- Maintenance Windows is too small? Autotask Jobs fail
- ORA-20000 Unable to gather statistics concurrently: Resource Manager is not enabled
- ORA-06512: at "SYS.DBMS_STATS"
- Thanks for coming to the Upgrade Workshops in Dublin & Belfast - special thanks to my friends at LUFTHANSA
- PSU October 2014

« Sleeping Beauties ... | Main | Oracle Database ... »

Incremental Statistics Collection improved in Oracle 12c

By Mike Dietrich on Nov 13, 2014

Traveling right now through Asia. It was Beijing for 32 hours, Tokyo for 24 hours - and now we are running an internal 2-day workshop with colleagues from Korea, New Zealand, India and some other countries in Seoul. And yesterday I had the pleasure to listen to Tom Kyte to his optimizer talk at the [OTN Conference in Tokyo](#). And I learned a lot - as always when having the chance to listen to Tom, Graham Wood and the other great experts.



Oracle Database 11.1 offered a great new feature: Incremental Statistics Collection which helped a lot to make stats collection on partitioned tables way more efficient. But it had a few

Slides Download Center

- Comprehensive**
 - Upgrade, Migrate & Consolidate to Oracle Database 12c**
Refreshed 31-OCT-2014
 - Upgrade Best Practices - 12c**
(latest update on 8-NOV-2014)
 - Upgrade Methods**
(Refresh: 8-NOV-2014)
 - What's New with Upgrades to 12c?**
Upload: 8-NOV-2014
 - Why Upgrade to Oracle 12c?**
Upload: 8-NOV-2014
- Deep Dive**
 - Behaviour Changes 8i to 11.2**
(changed: 16-JAN-2013)
 - Database Components Cleanup**
(uploaded: 20-JUN-2011)
- OTN Tour / Usergroups / Conf**
 - Hitchhiker's Guide to Upgrades**
Refresh: 8-NOV-2014

Contact me either via XING



Oracle Multitenant Database Upgrade Internals

- 1 Introduction
- 2 Database Upgrade News**
- 3 Oracle Multitenant Overview
- 4 Multitenant Upgrade
- 5 Inside catctl.pl – and Options
- 6 Performance Figures
- 7 Wrap Up

Upgrade SQL Automation

New Pre-Upgrade Script

- `preupgrd.sql`
- Executes pre-upgrade checks
- Runs in source environment
- Generates fixup scripts
 - `preupgrade_fixups.sql`
 - `postupgrade_fixups.sql`
- **MOS Note:884522.1**

```
====>> PRE-UPGRADE RESULTS for UPGR <<====
```

ACTIONS REQUIRED:

1. Review results of the pre-upgrade checks:
`/u01/app/oracle/cfgtoollogs/UPGR/preupgrade/preupgrade.log`
2. Execute in the SOURCE environment BEFORE upgrade:
`/u01/app/oracle/cfgtoollogs/UPGR/preupgrade/preupgrade_fixups.sql`
3. Execute in the NEW environment AFTER upgrade:
`/u01/app/oracle/cfgtoollogs/UPGR/preupgrade/postupgrade_fixups.sql`

```

                                     O O
*****
*****
Pre-Upgrade Checks in UPGR Completed.
*****
```

preupgrd.sql

preupgrd.sql => utluppkg.sql

TEXT

upgrade.xml

preupgrade.log

preupgrade_fixups.sql

postupgrade_fixups.sql

\$ORACLE_BASE/cfgtoollogs/sid/preupgrade



40%

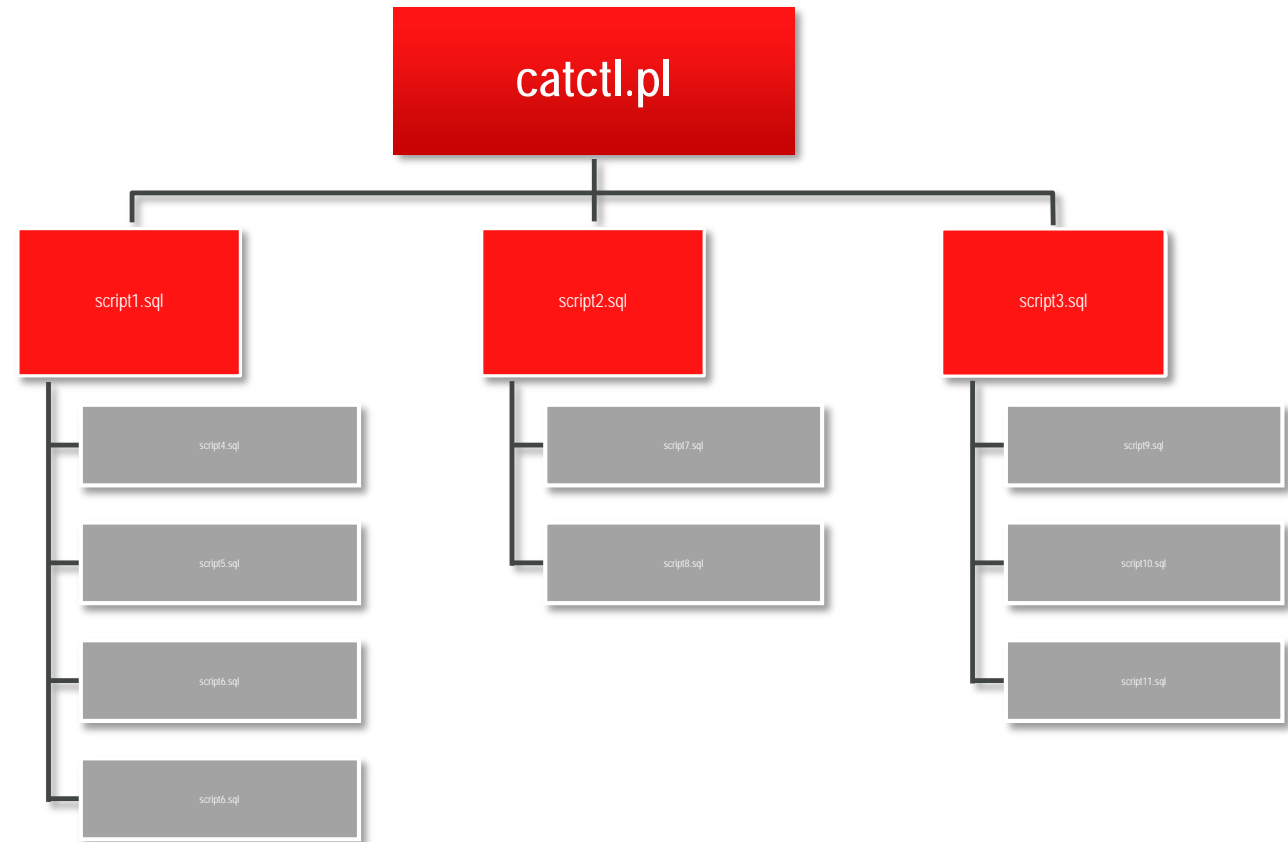
Faster Upgrades in
Oracle Database 12c



Faster Upgrade – Less Downtime

New Parallel Upgrade

- `catctl.pl`
- Runs database upgrade in parallel
 - Only component ORACLE SERVER
- Used and proven by **selected Oracle Database 11g** global customers
 - Telco billing
 - >100 SAP systems
 - Large DWH



“The new parallel upgrade script promises to drastically reduce downtime due to planned maintenance. We saw a 37% improvement over the previous upgrade process in our environment.”


Harald Stefan

Leiter Datenbanken
Payback GmbH

Faster Upgrade – Less Downtime

New Parallel Upgrade

```
$> $ORACLE_HOME/perl/bin/perl catctl.pl catupgrd.sql
```



```
Serial  Phase #:70 Files: 1    Time: 90s  
Serial  Phase #:71 Files: 1    Time: 0s  
Serial  Phase #:72 Files: 1    Time: 0s  
Serial  Phase #:73 Files: 1    Time: 34s
```

```
Grand Total Time: 1588s
```

```
LOG FILES: (catupgrd*.log)
```

```
Upgrade Summary Report Located in:
```

```
/u01/app/oracle/product/12.1.0.2/cfgtoollogs/UPGR/upgrade/upg_summary.log
```

```
Grand Total Upgrade Time:    [0d:0h:26m:28s]
```

Oracle Multitenant Database Upgrade Internals

- 1 Introduction
- 2 Database Upgrade News
- 3 Oracle Multitenant Overview**
- 4 Multitenant Upgrade
- 5 Inside catctl.pl – and Options
- 6 Performance Figures
- 7 Wrap Up



Look Before You Leap! 😊

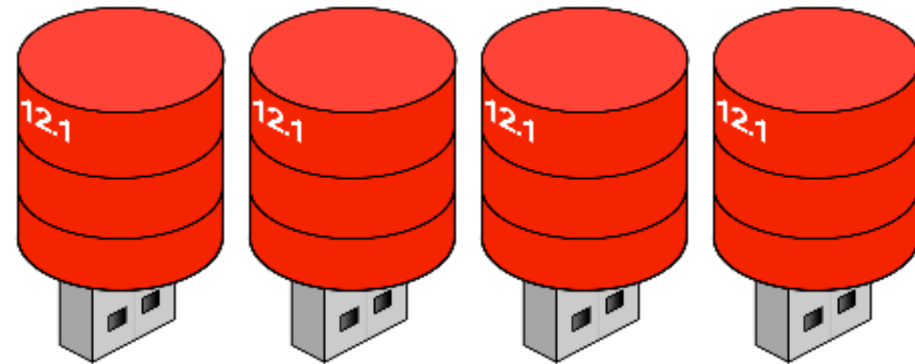
- Some well-known concepts will change
 - >200 pages **new documentation** in the Administrator's Guide

Part VI

Managing a Multitenant Environment

Part VI discusses the Oracle Multitenant option and managing a multitenant environment. It contains the following chapters:

- Chapter 36, "Overview of Managing a Multitenant Environment"
- Chapter 37, "Creating and Configuring a CDB"
- Chapter 38, "Creating and Removing PDBs with SQL*Plus"
- Chapter 39, "Creating and Removing PDBs with Cloud Control"
- Chapter 40, "Administering a CDB with SQL*Plus"
- Chapter 41, "Administering CDBs and PDBs with Cloud Control"
- Chapter 42, "Administering PDBs with SQL*Plus"
- Chapter 43, "Viewing Information About CDBs and PDBs with SQL*Plus"
- Chapter 44, "Using Oracle Resource Manager for PDBs with SQL*Plus"
- Chapter 45, "Using Oracle Resource Manager for PDBs with Cloud Control"
- Chapter 46, "Using Oracle Scheduler with a CDB"



Oracle Multitenant WP:

[http://www.oracle.com/technetwork/
database/multitenant-wp-12c-1949736.pdf](http://www.oracle.com/technetwork/database/multitenant-wp-12c-1949736.pdf)

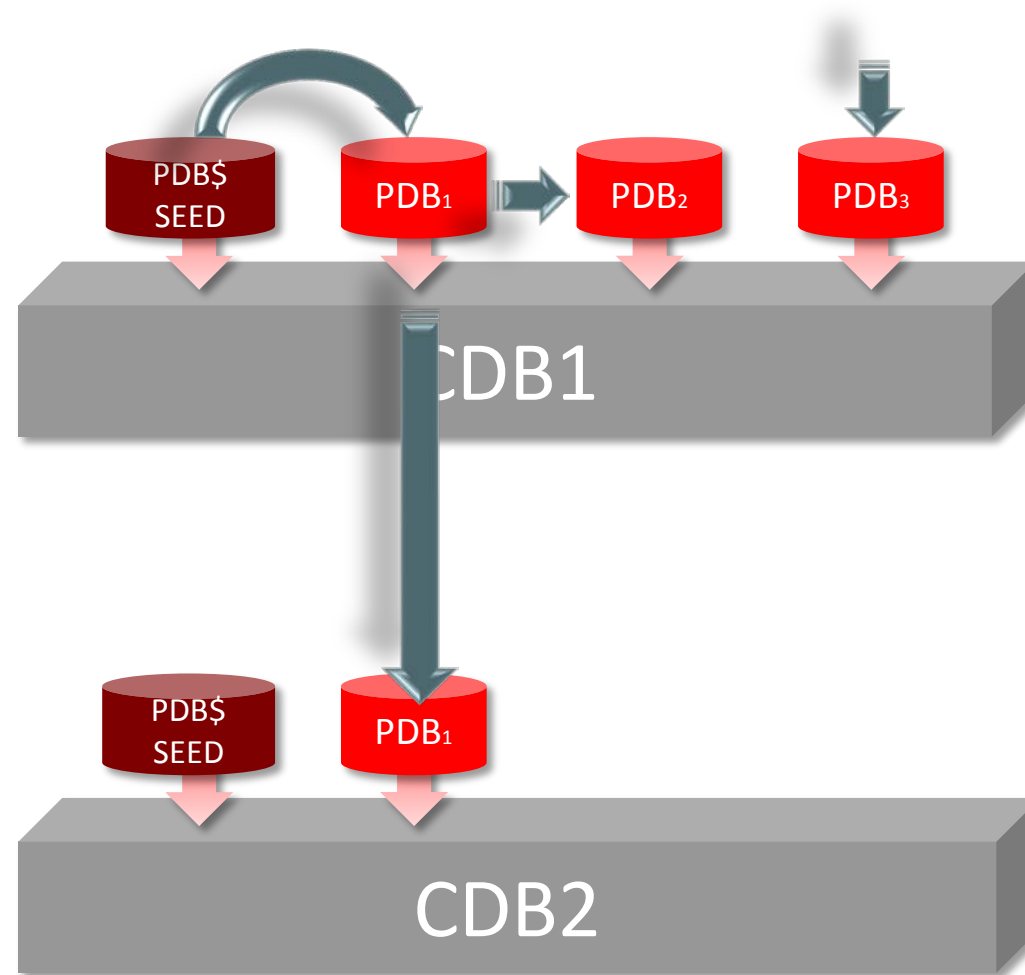
The earth is still a sphere 😊

- You don't have to use Oracle Multitenant ... *not yet*
- Oracle Database 12c non-Multitenant works as expected
 - But you can also use the new features ...
and this will require **changing old habits**

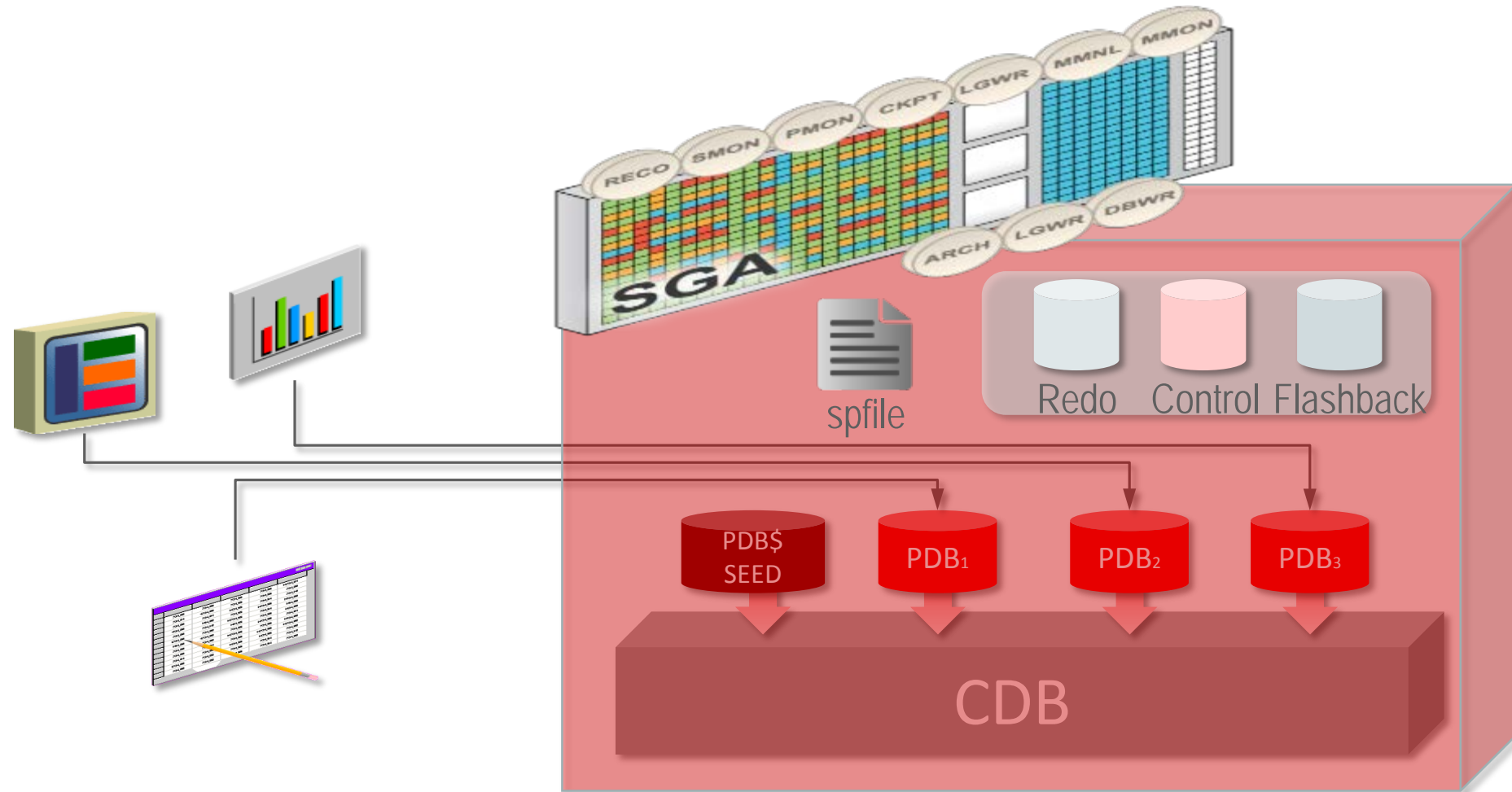
- **START** with a
Single Tenant
database NOW!!!



Oracle Multitenant – Implementation

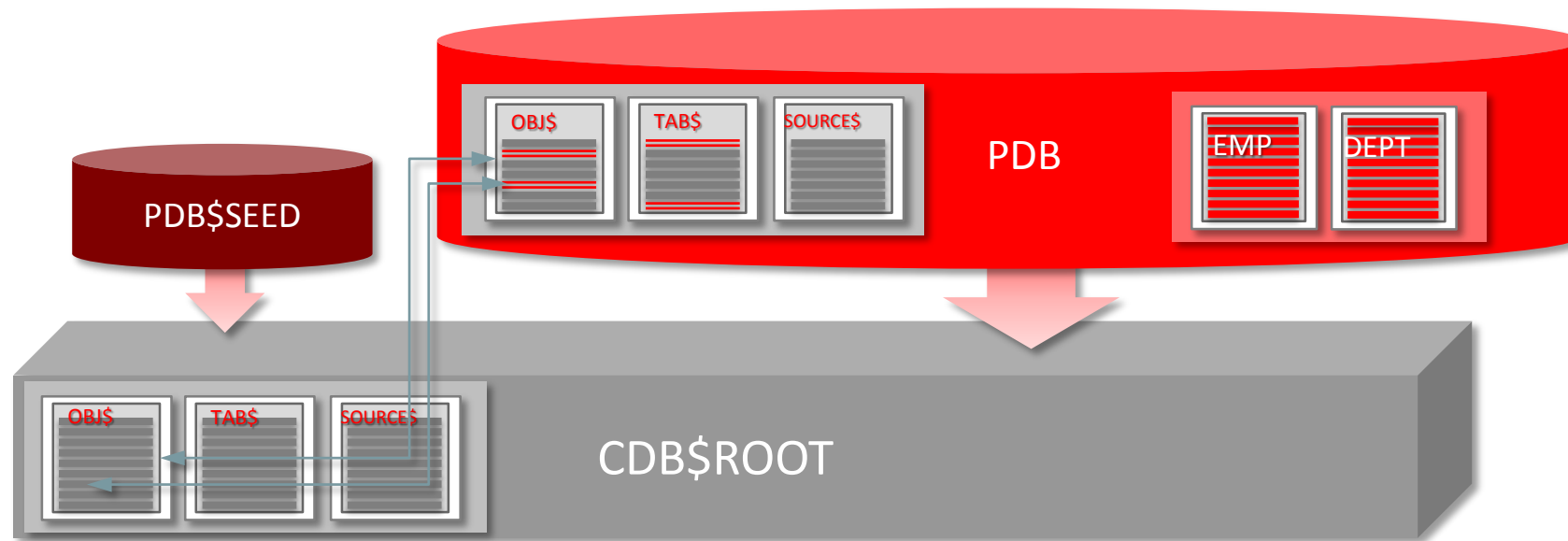


Oracle Multitenant – Behind the scenes



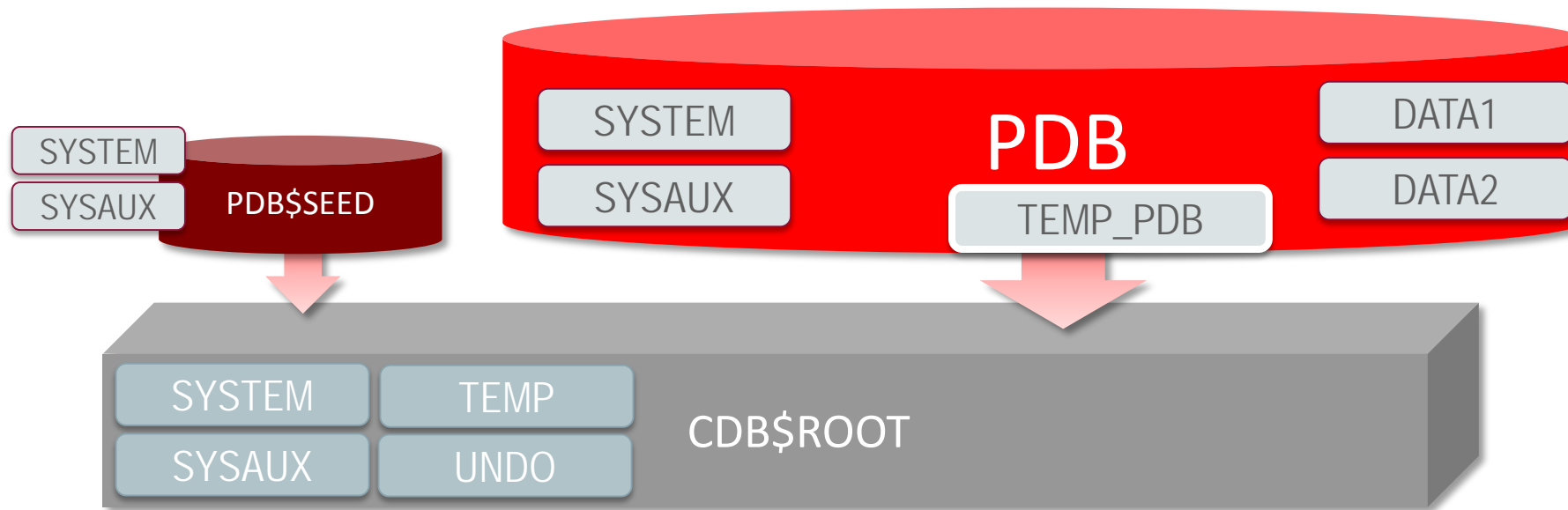
CDB-PDB: Who's who? After Plugin ...

- Data and objects?



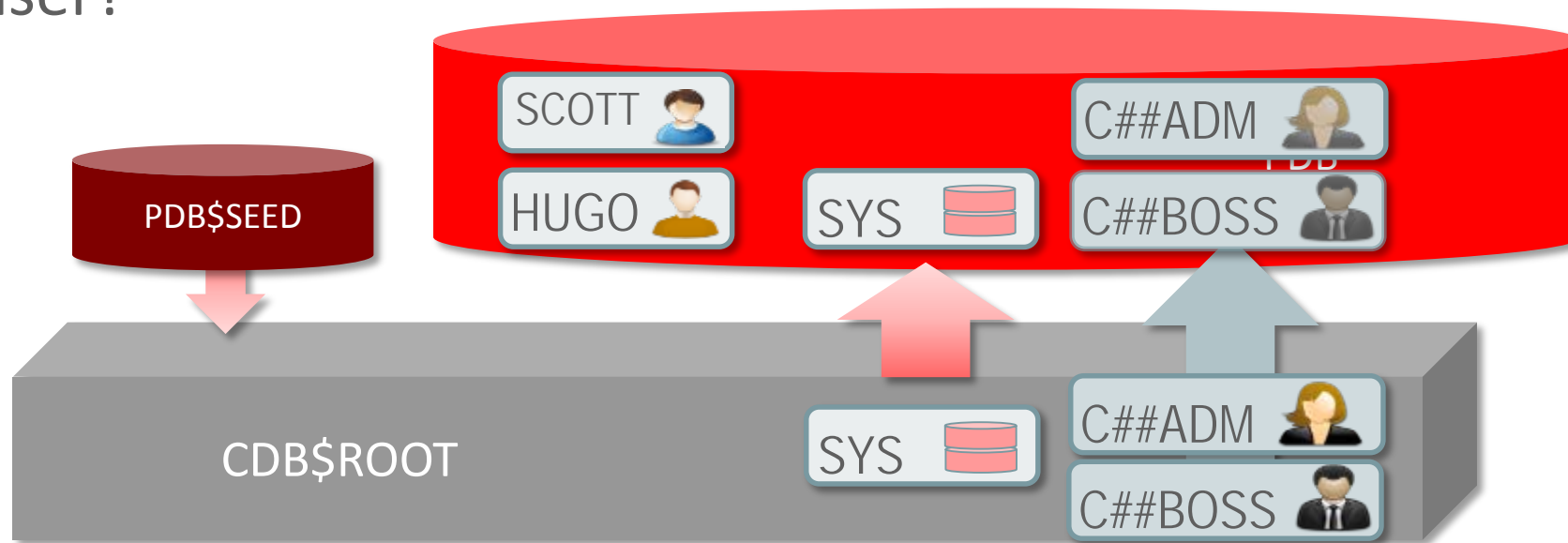
CDB-PDB: Who's who?

- Tablespaces?



CDB-PDB: Who's who?

- Common user?
- Local user?



New Tool For Script Execution

- **Administrative scripts** have to be started via `catcon.pl`:

```
$> perl catcon.pl -u SYS -d $ORACLE_HOME/rdbms/admin -e  
-s -b create_dictionary catcdb.sql
```

– Most useful `catcon.pl` options:

- `-u` Username and optionally password
- `-d` Directory containing the script to execute (default: current directory)
- `-e` Echo on
- `-s` Spools the output of every script
- `-l` Directory to write logfiles into (default: current directory)
- `-b` Base name for logfiles (mandatory option)

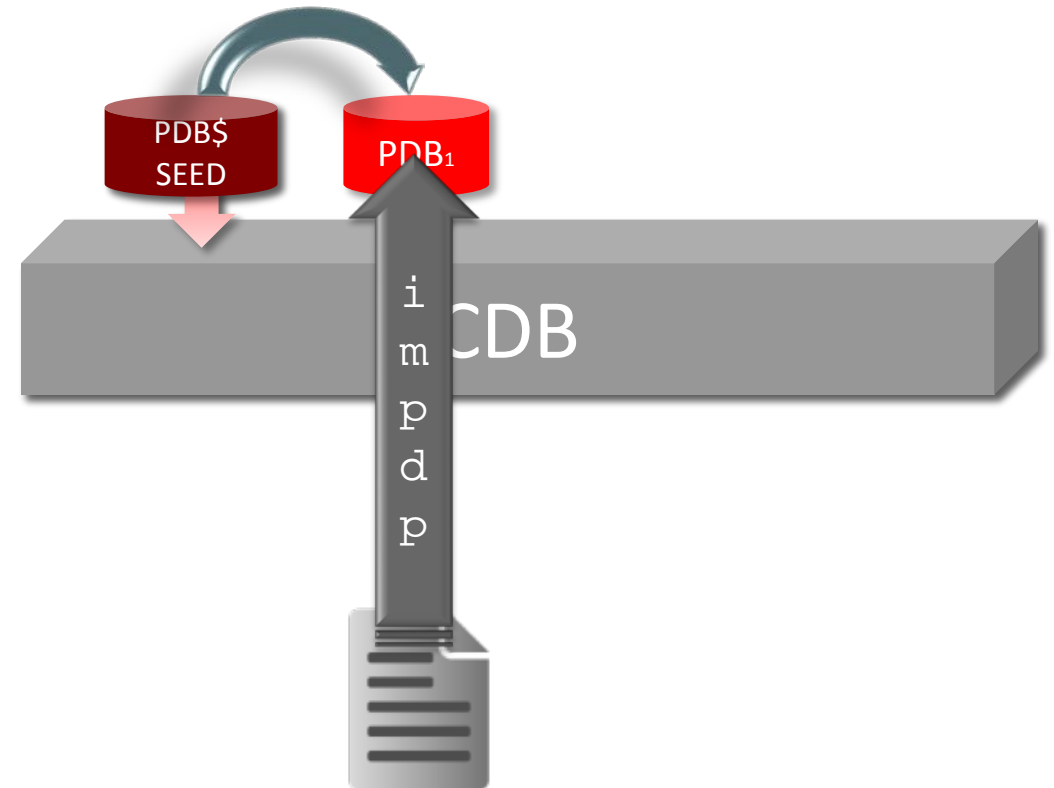
Creation of a New Pluggable Database

- Fast provisioning from PDB\$SEED

```
create pluggable database PDB1  
admin user adm1 identified by pwd;
```

- PDB_FILE_NAME_CONVERT

- Transport with TTS
- Full Transportable Export/Import
- Import data with `impdp`
 - Dump file or NETWORK_LINK
 - `imp` for \leq Oracle 9i



Upgrade and Plugin as PDB

- Database upgrade
- Start database read-only
- Create XML description file

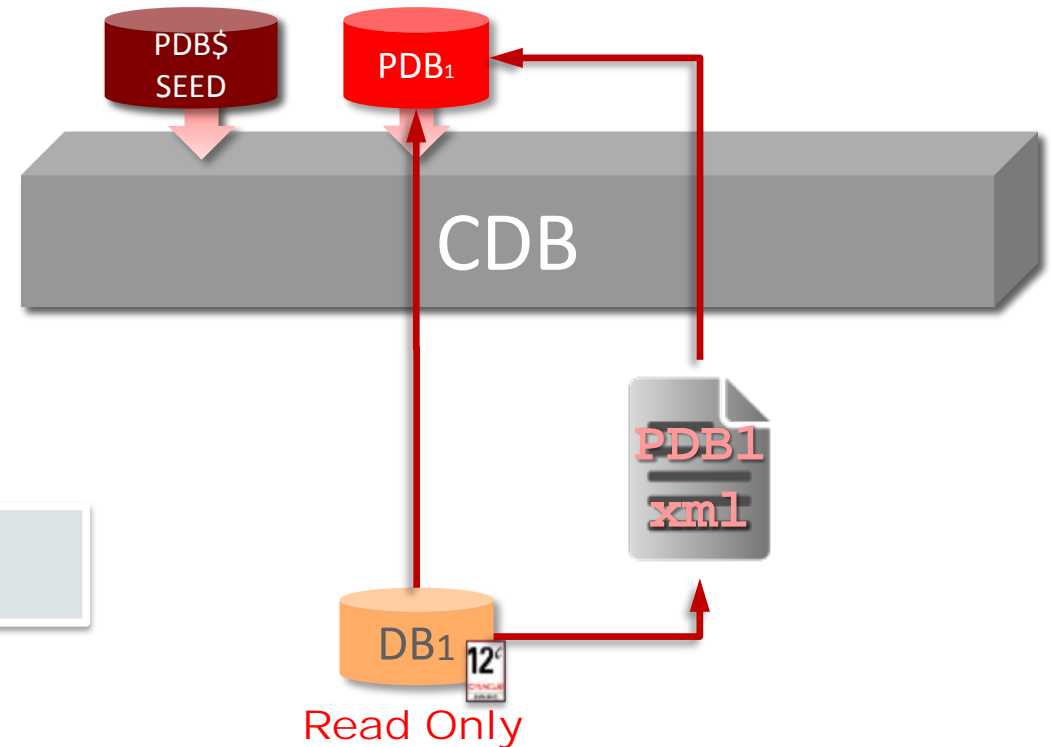
```
exec DBMS_PDB.DESCRIBE('PDB1.xml');
```

- Shutdown database
- Plugin database

```
create pluggable database PDB1  
using ('PDB1.xml') nocopy tempfile reuse;
```

- Sanity operations

```
start ?/rdbms/admin/noncdb_to_pdb.sql
```

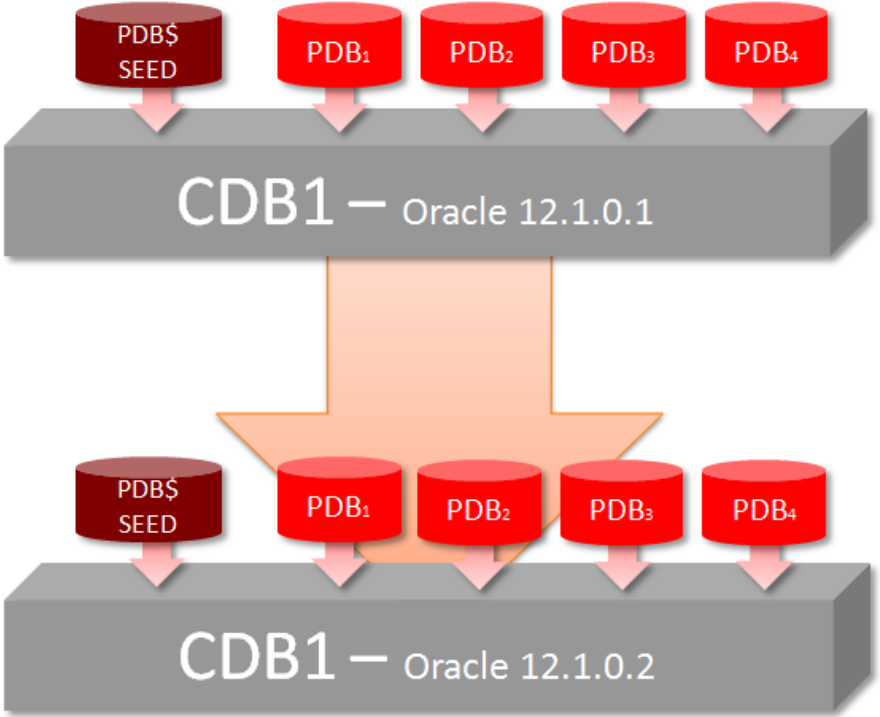


Oracle Multitenant Database Upgrade Internals

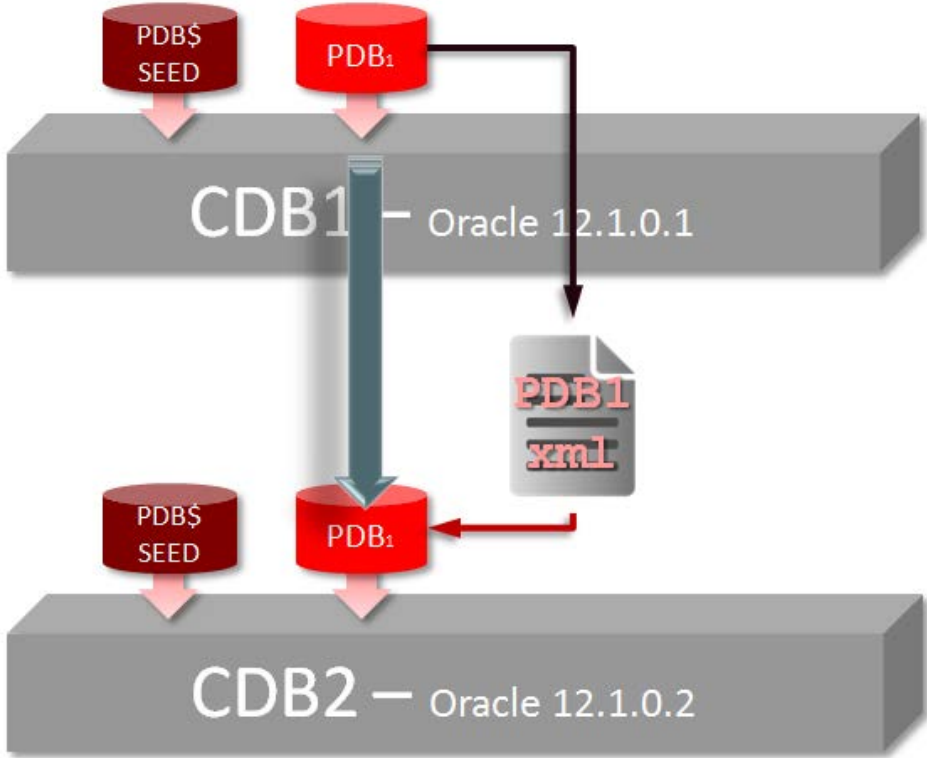
- 1 Introduction
- 2 Database Upgrade News
- 3 Oracle Multitenant Overview
- 4 Multitenant Upgrade**
- 5 Inside catctl.pl – and Options
- 6 Performance Figures
- 7 Wrap Up

Upgrade: Everything at once vs Unplug/Plug

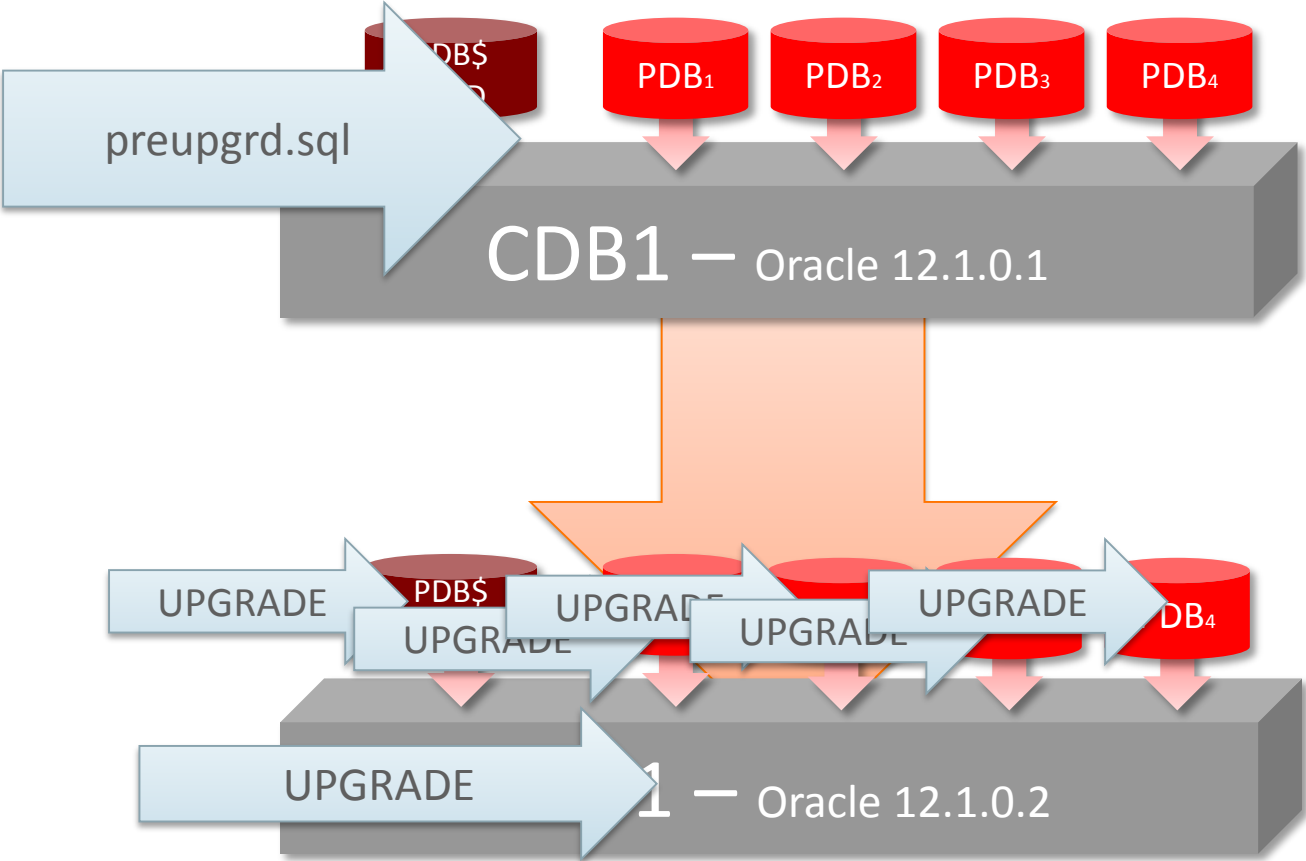
- Everything at Once



- Unplug/Plug



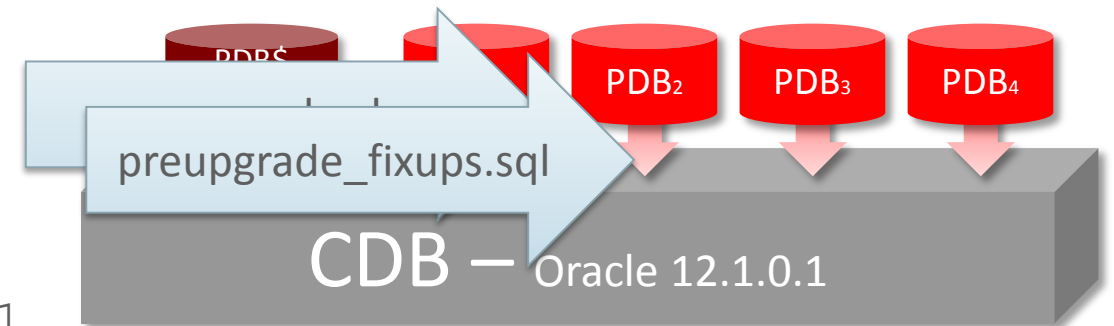
Upgrade: Everything at once



Upgrade: **Everything at once** – Step by Step – 1/2

▪ Preupgrade.sql:

- Copy new `preupgrd.sql` and `utluppkg.sql` into CBD1's `$ORACLE_HOME/rdbms/admin`
- `$ORACLE_HOME/perl/bin/perl`
`$ORACLE_HOME/rdbms/admin/catcon.pl -n 1`
`-d $ORACLE_HOME/rdbms/admin`
`-l /home/oracle/mike -b preupgrd preupgrd.sql`



▪ In CDB – Oracle **12.1.0.1**:

- `SQL> ALTER PLUGGABLE DATABASE ALL OPEN;`
- `$ORACLE_HOME/perl/bin/perl`
`$ORACLE_HOME/rdbms/admin/catcon.pl -n 1`
`-d $ORACLE_HOME/cfgtoollogs/cdbupgr/preupgrade`
`-l /home/oracle/mike -b preupgrade_fixups preupgrade_fixups.sql`

preupgrd.sql in **Everything at Once**

- `preupgrd.sql` runs in every container
 - CDB\$ROOT
 - PDB\$SEED
 - All PDBs
- Executed with `catcon.pl`
 - `catcon.pl -n 1 -e -b preupgrade -d '''.''' preupgrd.sql`
- Results get logged to `preupgrade0.log`
- Specific changes for each container

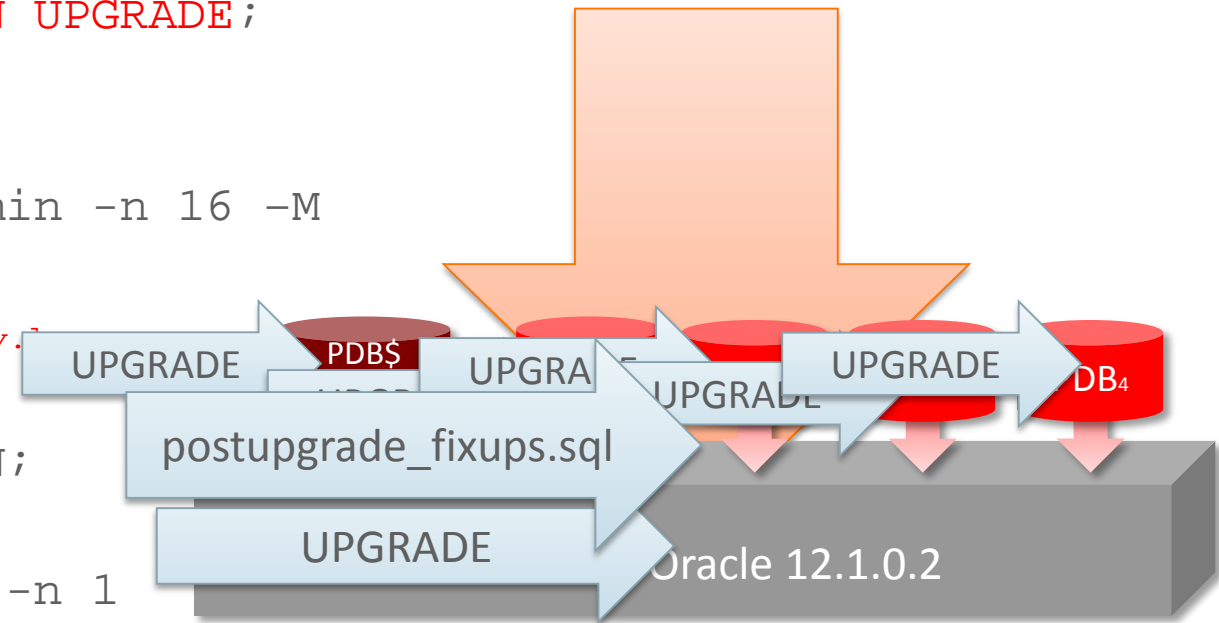
Upgrade: **Everything at once** – Step by Step – 2/2

▪ In CDB – Oracle **12.1.0.2**:

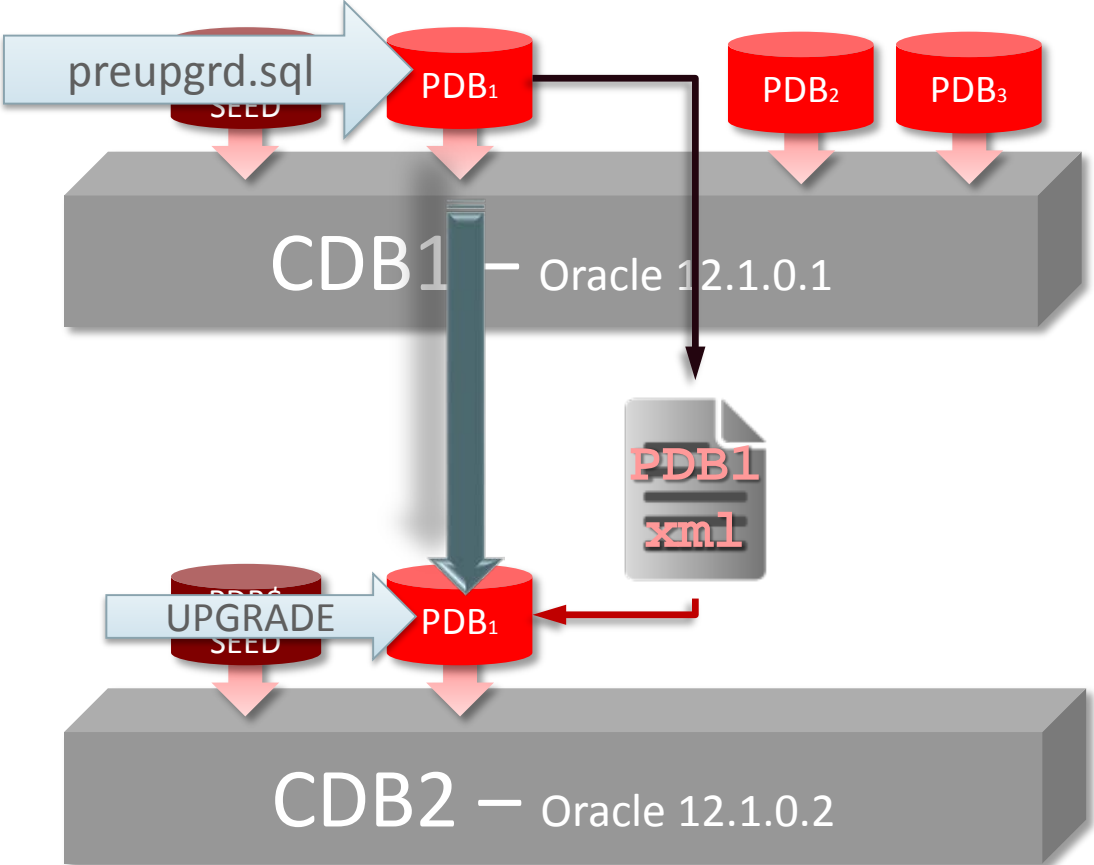
- SQL> STARTUP UPGRADE
- SQL> ALTER PLUGGABLE DATABASE **ALL OPEN UPGRADE**;
- cd \$ORACLE_HOME/rdbms/admin
- \$ORACLE_HOME/perl/bin/perl
catctl.pl -d \$ORACLE_HOME/rdbms/admin -n 16 -M
-l /home/oracle/mike **catupgrd.sql**

▪ See \$ORACLE_HOME/cfgtoollogs/<SID>/upgrade/**upg_summary.**

- SQL> STARTUP
- SQL> ALTER PLUGGABLE DATABASE ALL OPEN;
- \$ORACLE_HOME/perl/bin/perl
\$ORACLE_HOME/rdbms/admin/catcon.pl -n 1
-d \$ORACLE_HOME/cfgtoollogs/cdbupgr/preupgrade
-l /home/oracle/mike
-b postupgrade_fixups **postupgrade_fixups.sql**
- \$ORACLE_HOME/perl/bin/perl catcon.pl -n 1 -e -b utlrlp -d '...' **utlrlp.sql**



Upgrade: One (or many) at a time

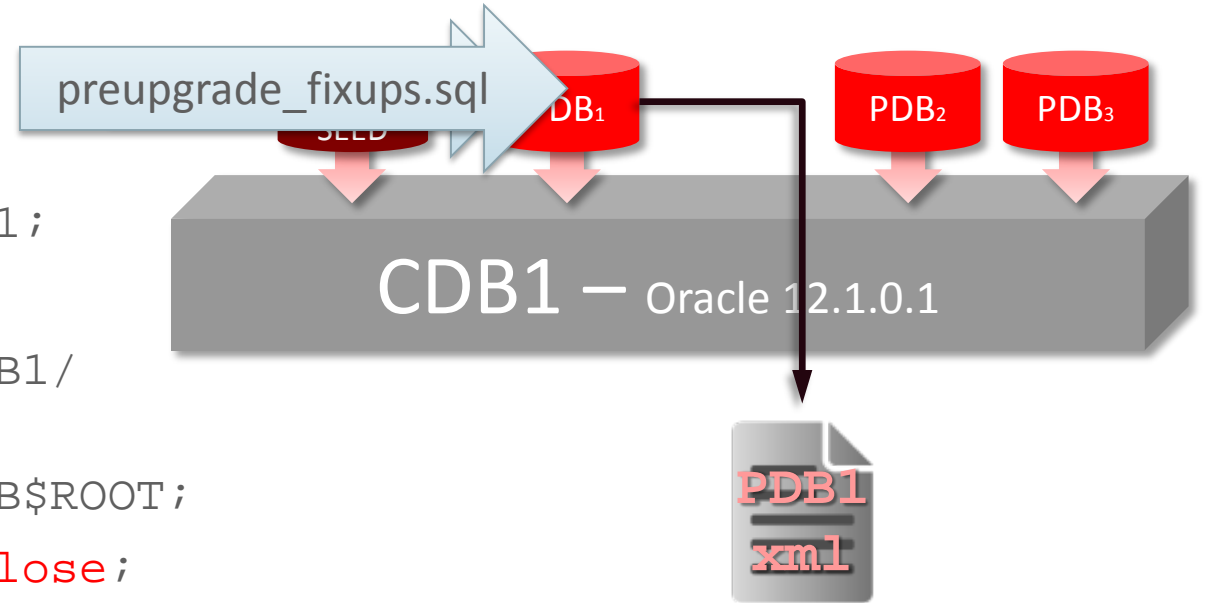


Upgrade: One (or many) at a time – Step by Step – 1/2

▪ In CDB1:

Copy new *preupgrd.sql* and *utluppkg.sql* into CDB1's `$ORACLE_HOME/rdbms/admin`

- `SQL> alter session set container=PDB1;`
- `SQL> @?/rdbms/admin/preupgrd.sql`
- `SQL> @/u01/app/oracle/cfgtoollogs/CDB1/preupgrade/preupgrade_fixups.sql`
- `SQL> alter session set container=CDB$ROOT;`
- `SQL> alter pluggable database PDB1 close;`
- `SQL> alter pluggable database PDB1 unplug into '/stage/pdb1.xml';`
- `SQL> exit`



Upgrade: One (or many) at a time – Step by Step – 2/2

■ In CDB2:

In SQL*Plus:

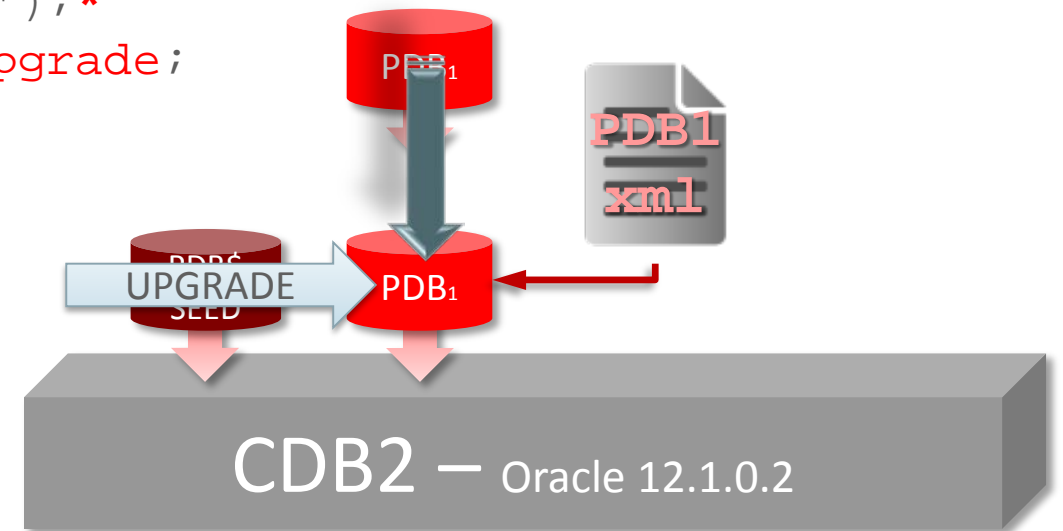
- SQL> alter session set container=CDB\$ROOT;
- SQL> create pluggable database PDB1 using
'/stage/pdb1.xml' file_name_convert=(
'/oradata/CDB1/pdb1', '/oradata/CDB2/pdb1');*
- SQL> alter pluggable database PDB1 open upgrade;
- SQL> exit

On the command prompt:

- \$> cd \$ORACLE_HOME/rdbms/admin
- \$> \$ORACLE_HOME/perl/bin/perl catctl.pl
-c "PDB1" catupgrd.sql

Back in SQL*Plus:

- SQL> alter session set container=PDB1;
- SQL> startup
- SQL> @?/rdbms/admin/utlpr.sql



** A Plug-In-Check can be done before this step – but it will always result in "NO" as COMPATIBLE=12.1.0.2 per default in every Oracle 12.1.0.2 database when created with the DBCA*

Oracle Multitenant Database Upgrade Internals

- 1 Introduction
- 2 Database Upgrade News
- 3 Oracle Multitenant Overview
- 4 Multitenant Upgrade
- 5 Inside catctl.pl – and Options**
- 6 Performance Figures
- 7 Wrap Up



Faster Upgrade – Less Downtime

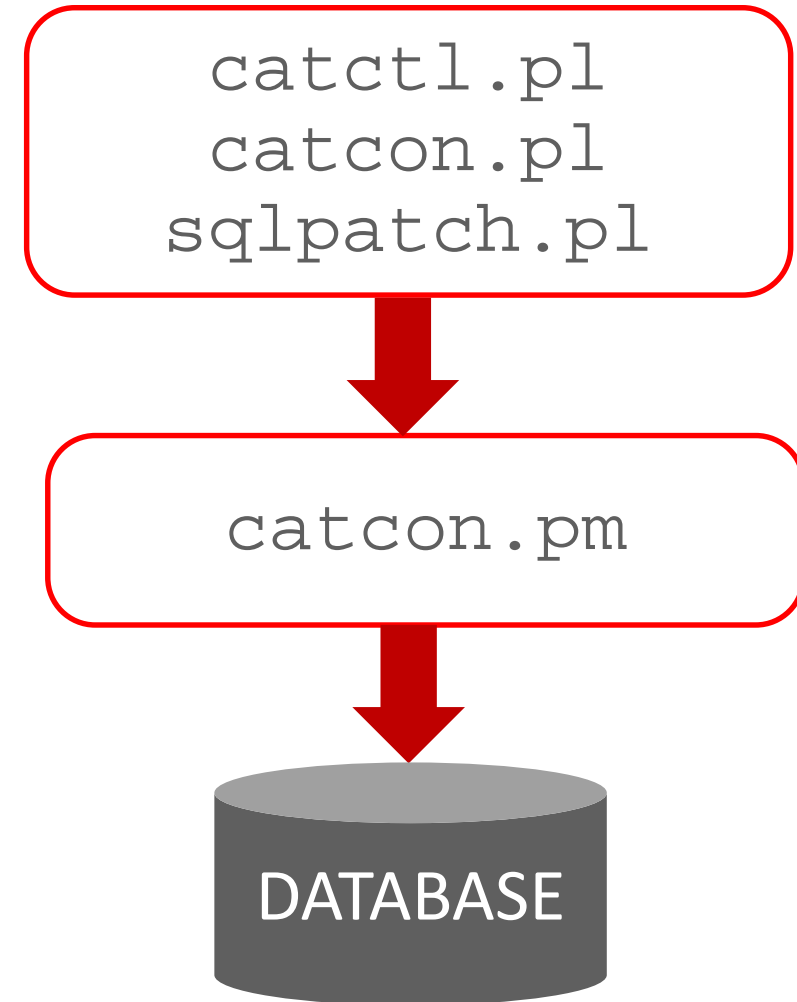
catctl.pl

- Parallel upgrade is performed in phases
- Phases are determined at run time by parsing **CATCTL tags** within the SQL files
- Each phase **builds dependencies** for the next phase
- Each phase **must complete** before moving onto the next phase
- Phases can be run in serial or in parallel

```
Parallel Phase #:34 Files: 14 Time: 113s
Restart Phase #:35 Files: 1 Time: 0s
Parallel Phase #:36 Files: 11 Time: 19s
Restart Phase #:37 Files: 1 Time: 0s
Serial Phase #:38 Files: 1 Time: 8s
Restart Phase #:39 Files: 1 Time: 0s
Serial Phase #:40 Files: 1 Time: 10s
Serial Phase #:41 Files: 1 Time: 3s
Restart Phase #:42 Files: 1 Time: 0s
Parallel Phase #:43 Files: 2 Time: 411s
Restart Phase #:44 Files: 1 Time: 1s
Serial Phase #:45 Files: 2 Time: 510s
Restart Phase #:46 Files: 1 Time: 1s
Parallel Phase #:47 Files: 2 Time: 35s
```

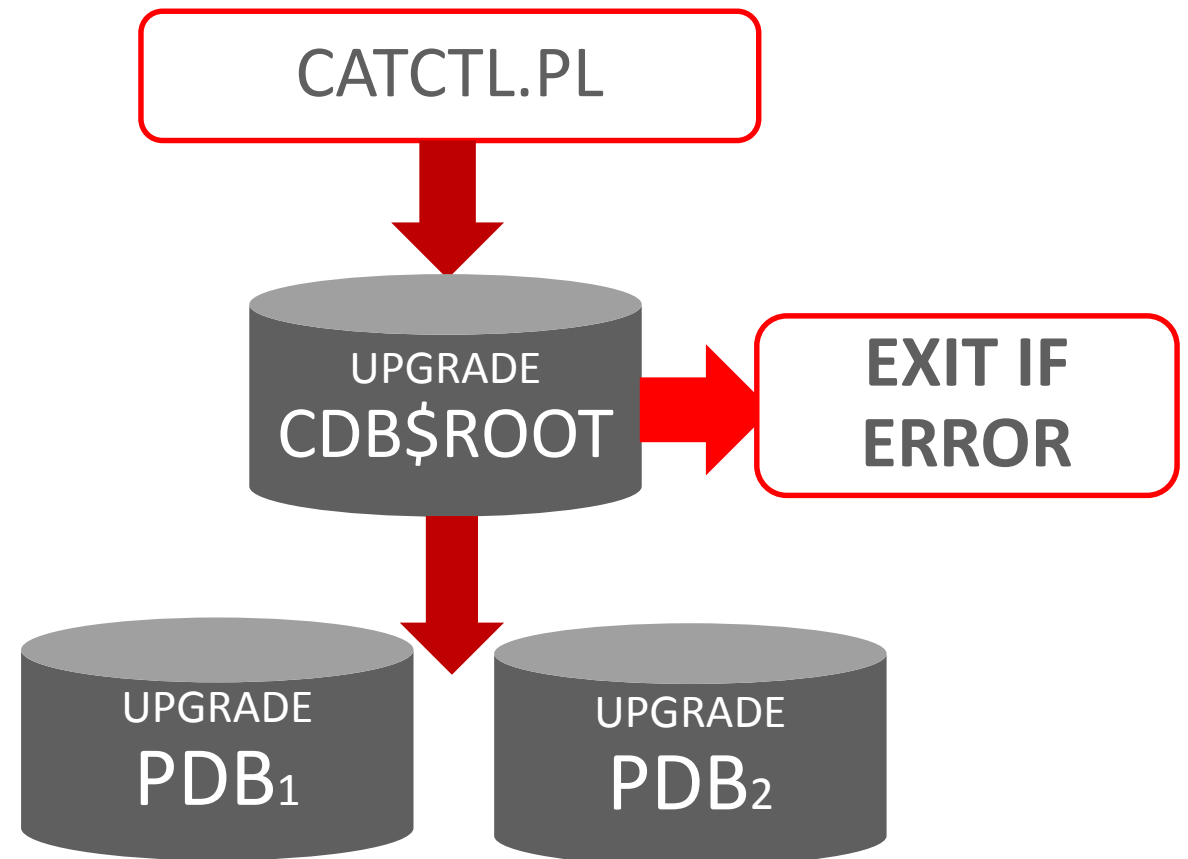
One Interface Used By All

- Since Oracle 12.1.0.2 we broke out `catctl.pl` into a common shared library called `catcon.pm`
- `catcon.pm` used in
 - `catctl.pl`
 - `datapatch` (`sqlpatch.pl`)
 - `catcon.pl`
- `catctl.pl` calls packages within `catcon.pm` to perform the upgrade



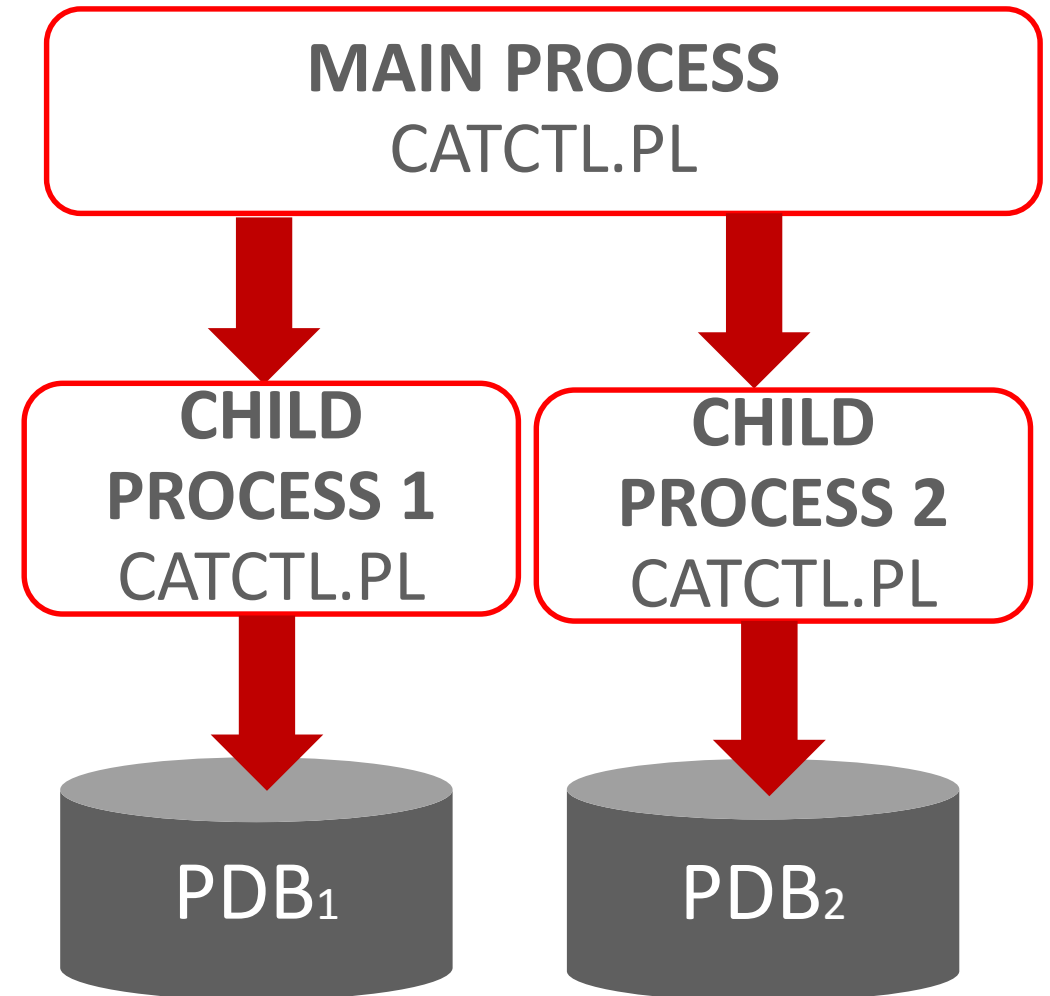
How Upgrade Works in a Multitenant Database

- Upgrade CDB\$ROOT first
 - Exit if there is an error
- Upgrade PDBs next



How Upgrade Works in a Multitenant Database

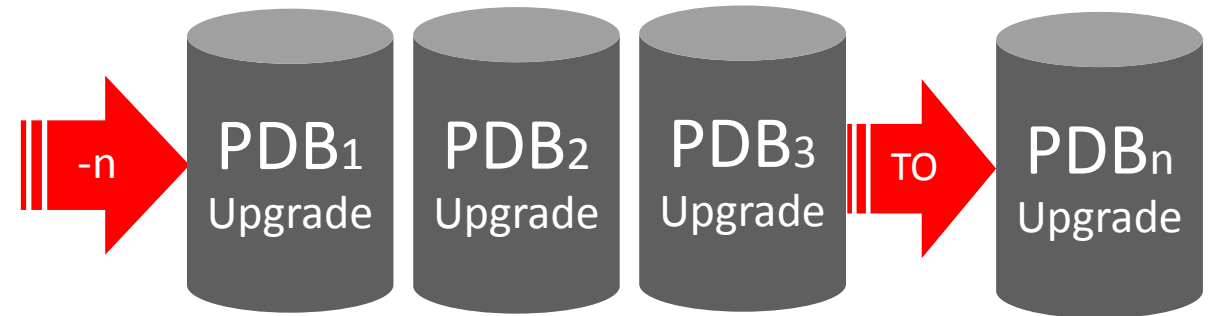
- Main `catctl.pl` process
 - Starts child processes
 - Waits for child processes to complete
- Child processes upgrade PDBs
 - Independent from main
- Process order by `CON_IDs`



Multiple PDB Upgrade

- `catctl.pl -n`
 - Multiple PDB upgrades controlled by `-n`
 - Number of PDBS upgraded in parallel
 - Divided by 2 – rounded down
 - Default: `cpu_count/2`
 - Examples:
 - `-n 24` ==> Upgrade 12 PDBs in parallel
 - `-n 31` ==> Upgrade 15 PDBs in parallel
 - Maximum: 64 => 32 PDBs
 - Minimum: 2 => 1 PDB

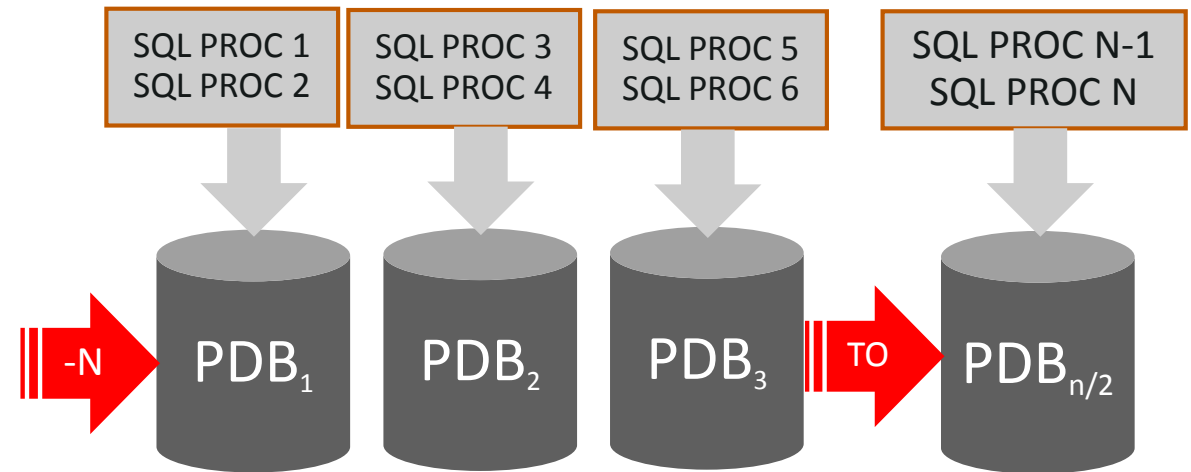
```
$ORACLE_HOME/perl/bin/perl catctl.pl -n 24 catupgrd.sql
```



Parallel Processing Within PDB Upgrades

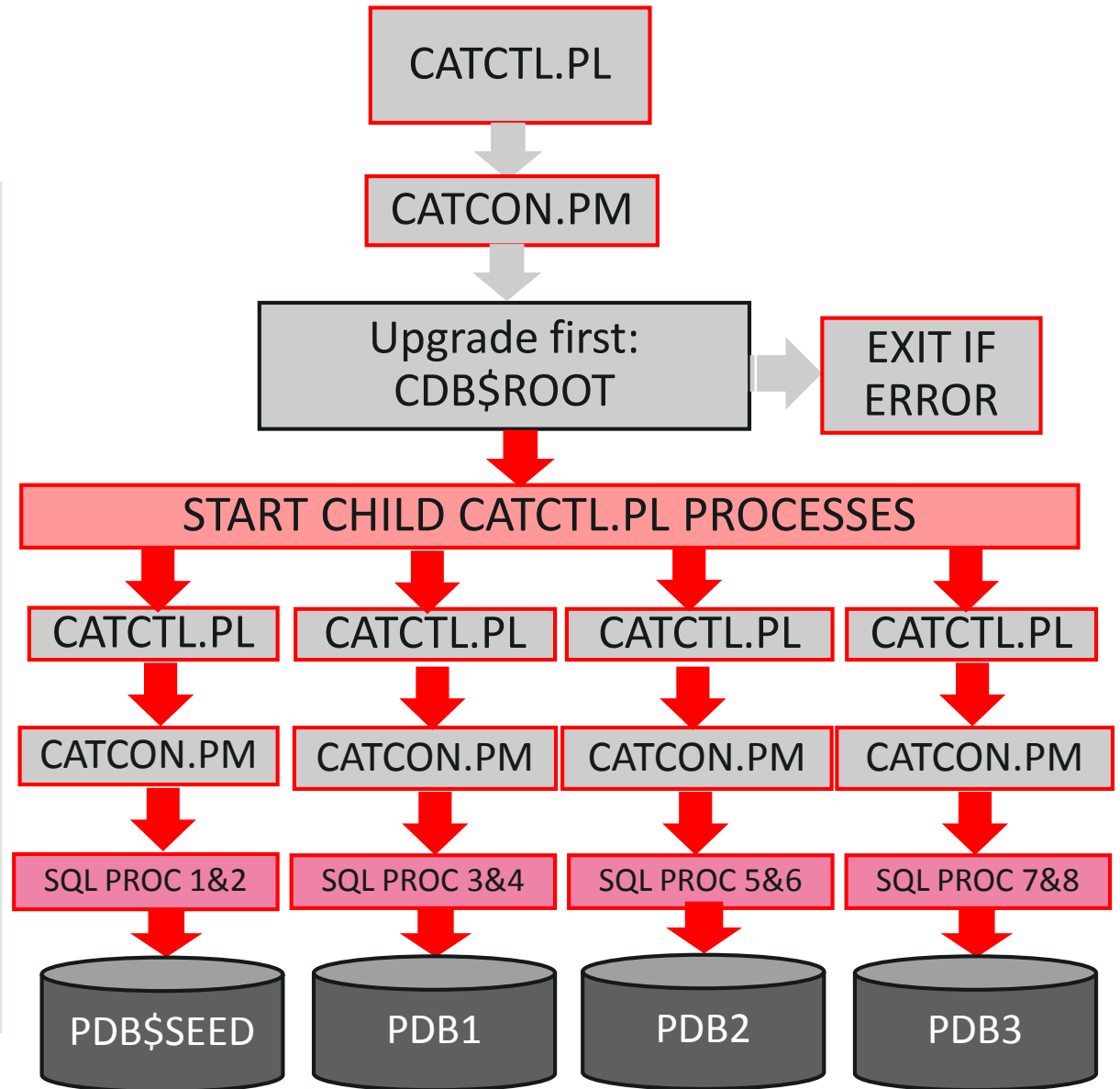
- `catctl.pl -N`
 - Parallel processing within each PDB upgrade controlled by `-N`
 - No. of SQL processes to spawn per PDB
 - Default: 2
 - Example:
 - `cpu_count=32`
 - 16 PDBs per cycle
 - Each PDB 2 parallel workers
 - Maximum: 8
 - Minimum: 1

```
$ORACLE_HOME/perl/bin/perl catctl.pl -N 2 catupgrd.sql
```



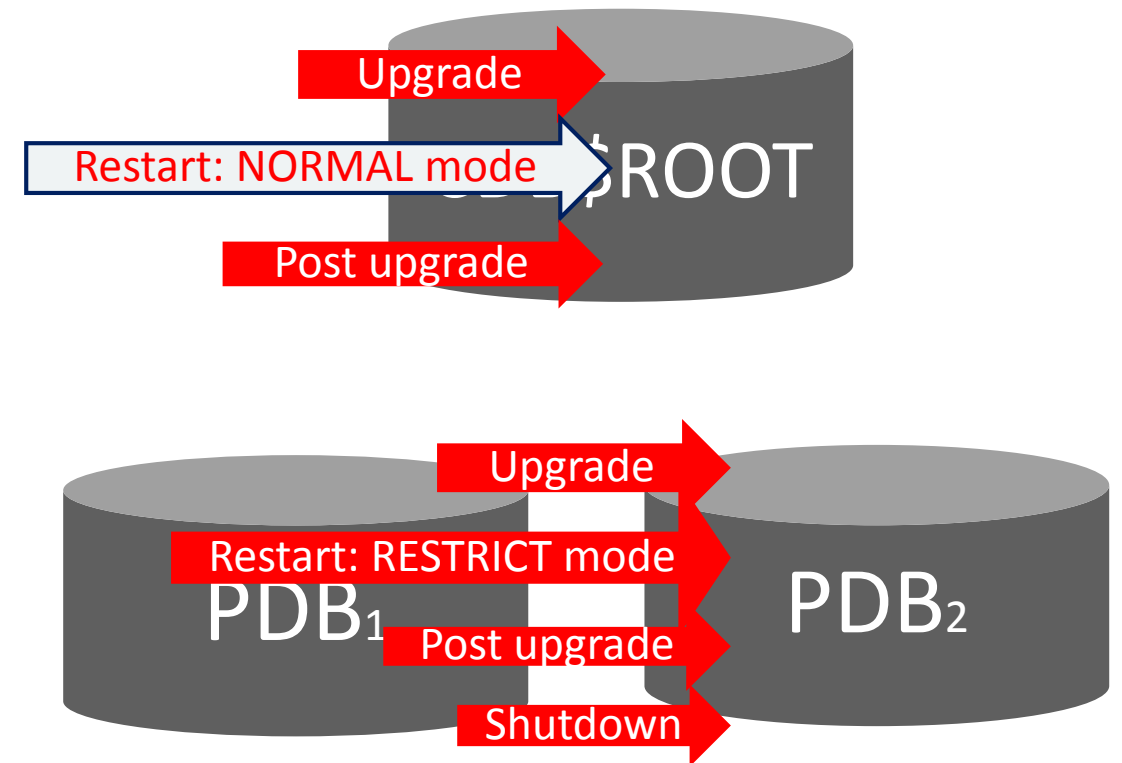
Putting It All Together

- `catctl.pl -n 8`
 - Upgrade CDB\$ROOT first
 - Upgrade PDBs
 - Startup `catctl` child processes
 - 4 PDBs upgraded at a times
 - 2 SQL processes per PDB
 - As soon as one PDB is upgraded then the next PDB can be processed until all the PDBs have been upgraded



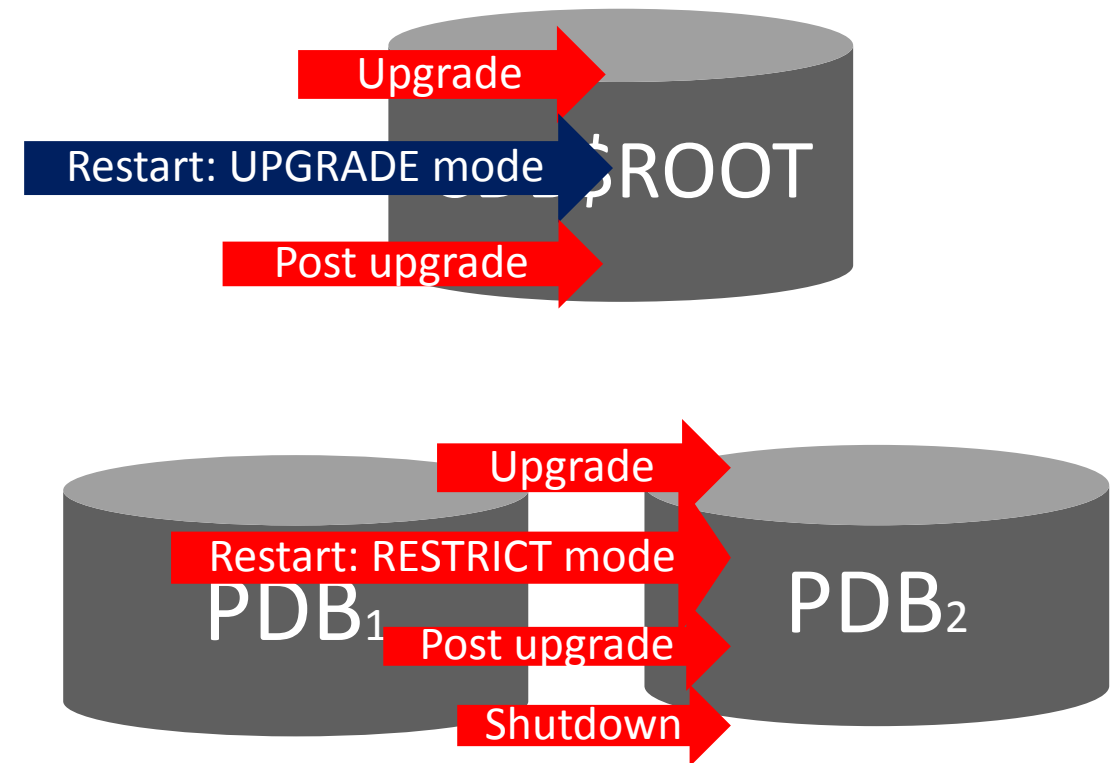
Default: Switch CDB\$ROOT to **Normal** Mode

- `catctl.pl` **default** behavior
 - Upgrade CDB\$ROOT first
 - Switch CDB\$ROOT to NORMAL mode before PDBs get upgraded
 - Advantage:
 - Each PDB becomes available after upgrade
 - Disadvantage:
 - Slower overall due to active processes



Option: Keep CDB\$ROOT in Upgrade Mode

- `catctl.pl` with `-M` option
 - Upgrade CDB\$ROOT first
 - Keep CDB\$ROOT in `UPGRADE` mode while PDBs are upgraded
 - Advantage:
 - Faster overall completion
 - Disadvantage:
 - All PDBs need to be upgraded to become available again



Logging, Prefixing and Directories

- `catctl.pl -l`
 - Directory to spool log files into
 - Default: current working directory
- `catctl.pl -i`
 - Prefix for logfiles
 - Default: none
- `catctl.pl -d`
 - Directory containing files to execute
 - Default: current working directory

```
- catctl.pl -l /home/oracle/upg
```

```
- catctl.pl -i _ORCL_
```

```
- catctl.pl -d  
$ORACLE_HOME/rdbms/admin
```

Logs and Verification

- `upg_summary.log`
 - Summary report
 - Most important log file
 - Different location than other logs:
`$ORACLE_HOME/cfgtoollogs/<SID>/upgrade/upg_summary.log`
 - Information included as well in:
`catupgrd0.log`
 - PDB logs naming:
 - `catupgrd<PDB NAME><proc#>.log`
 - CDB\$ROOT logs naming:
 - `catupgrd<proc#>.log`

Find Errors in `upg_summary.log`
and identify the PDB Name



Check
`catupgrd<pdb_name><proc#>.log`



Search for error noted in summary
report

Example upg_summary.log

Oracle Database 12.1 Post-Upgrade Status Tool		11-17-2014 22:55:29		
[CDB\$ROOT]				
Oracle Database 12.1 Post-Upgrade Status Tool		11-17-2014 23:28:00		
[PDB13]				
Oracle Database 12.1 Post-Upgrade Status Tool		11-18-2014 07:06:23		
[PDB234]				
Component Name	Current Status	Version Number	Elapsed Time	
Oracle Server	UPGRADED	12.1.0.2.0	00:13:55	
Oracle JServer JAVA Virtual Machine	VALID	12.1.0.2.0	00:02:35	
Oracle OLAP	OPTION OFF	12.1.0.2.0	00:00:02	
Oracle Oracle Workspace Manager	VALID	12.1.0.2.0	00:01:47	
Oracle OLAP Analytic Workspace	VALID	12.1.0.2.0	00:01:04	
Oracle Oracle OLAP API	VALID	12.1.0.2.0	00:00:20	
Oracle Oracle Label Security	VALID	12.1.0.2.0	00:00:43	
Oracle Spatial	VALID	12.1.0.2.0	00:02:32	
Oracle Oracle Text	VALID	12.1.0.2.0	00:00:54	
Oracle Oracle XML Database	VALID	12.1.0.2.0	00:02:28	
Oracle Spatial	VALID	12.1.0.2.0	00:01:02	
Oracle Database Java Packages	VALID	12.1.0.2.0	00:01:02	
Oracle Oracle Multimedia	VALID	12.1.0.2.0	00:02:23	
Oracle Spatial	UPGRADED	12.1.0.2.0	00:02:39	
Oracle Oracle Application Express	VALID	4.2.5.00.08	00:02:25	
Oracle Oracle Database Vault	VALID	12.1.0.2.0	00:00:31	
Oracle Final Actions			00:00:49	
Oracle Post Upgrade			00:02:33	
Total Upgrade Time: 00:39:44				[PDB234]
Upgrade Times Sorted In Descending Order				
Total Upgrade Time: 00:42:25				[PDB163]
Total Upgrade Time: 00:42:25				[PDB163]
Total Upgrade Time: 00:21:22				[PDB44]
Total Upgrade Time: 00:21:18				[PDB57]
Total Upgrade Time: 00:21:17				[PDB33]
Total Upgrade Time: 00:21:11				[PDB23]
Total Upgrade Time: 00:21:06				[PDB22]
Total Upgrade Time: 00:21:06				[PDB56]
Total Upgrade Time: 00:20:58				[PDB55]
Total Upgrade Time: 00:20:42				[PDB21]
Total Upgrade Time: 00:20:37				[PDB41]
Total Upgrade Time: 00:20:32				[PDB20]
Total Upgrade Time: 00:20:31				[PDB40]
Total Upgrade Time: 00:20:21				[PDB54]
Total Upgrade Time: 00:20:16				[PDB53]
Total Upgrade Time: 00:20:11				[PDB18]
Total Upgrade Time: 00:20:06				[PDB17]
Total Upgrade Time: 00:20:05				[PDB19]
Total Upgrade Time: 00:20:03				[PDB52]
Total Upgrade Time: 00:19:55				[PDB49]
Total Upgrade Time: 00:19:55				[PDB50]
Total Upgrade Time: 00:19:42				[PDB48]
Total Upgrade Time: 00:19:41				[PDB51]
Total Upgrade Time: 00:19:40				[PDB16]
Grand Total Upgrade Time:				[0d:9h:6m:18s]

Upgrade Success/Failure?

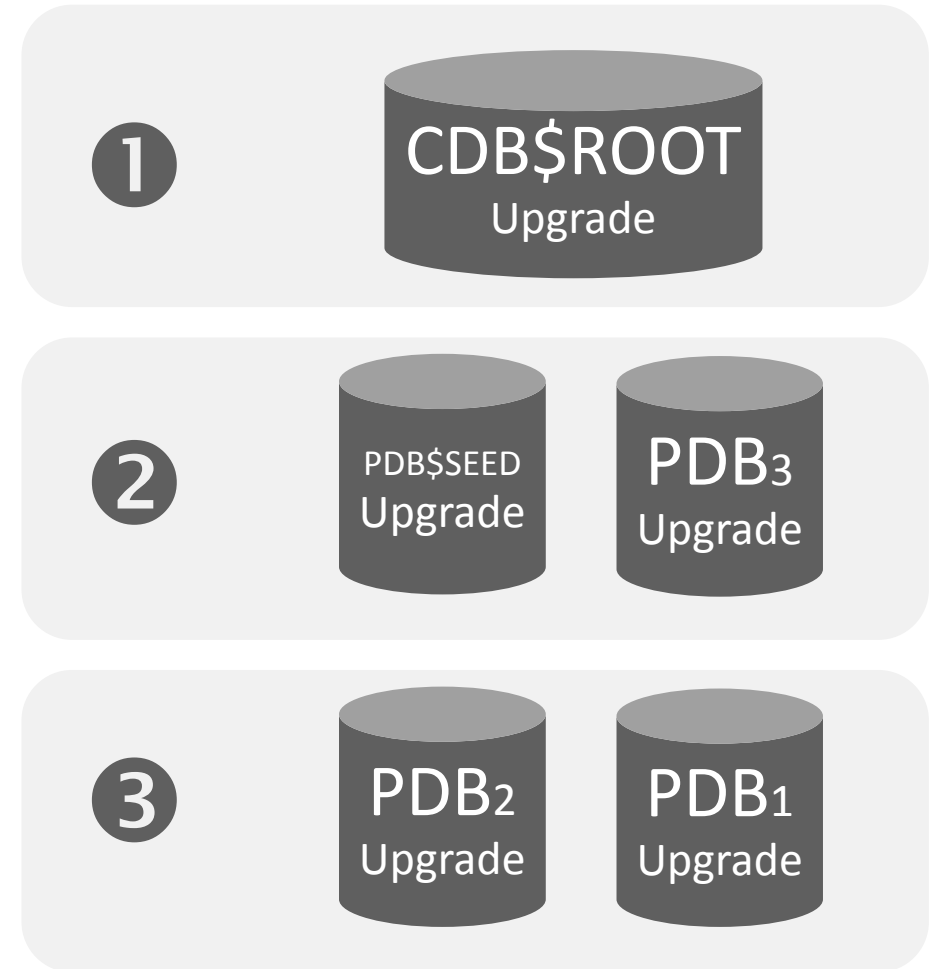
- `REGISTRY$ERROR`
 - Contains all errors
 - `ORACLE SERVER` component determines success/failure of upgrade as a whole
 - Other components have their own validation routines

```
SQL> describe REGISTRY$ERROR;
```

Name	Null?	Type
USERNAME		VARCHAR2(256)
TIMESTAMP		TIMESTAMP(6)
SCRIPT		VARCHAR2(1024)
IDENTIFIER		VARCHAR2(256)
MESSAGE		CLOB
STATEMENT		CLOB

Container Inclusion/Exclusion

- `catctl.pl -c`
 - Include container
 - Define order of upgrades
 - Example:
 - `-c 'CDB$ROOT PDB$SEED PDB3 PDB2 PDB1'`
- `catctl.pl -C`
 - Exclude container
 - Example:
 - `-C 'PDB7 PDB21'`



Postpone and Restart Phases

- `catctl.pl -x`
 - Postpone datapatch/catuppst.sql
 - Possible but **not recommended**
- `catctl.pl -p`
 - Begin phase
- `catctl.pl -P`
 - End phase
- **Example:** `-c 'PDB7' -p 68 -P 73`
 - Upgrade log file identified each phase:
PHASE_TIME____START 68
PHASE_TIME____END 73

```
Parallel Phase #:34 Files: 14 Time: 113s
Restart Phase #:35 Files: 1 Time: 0s
Parallel Phase #:36 Files: 11 Time: 19s
Restart Phase #:37 Files: 1 Time: 0s
Serial Phase #:38 Files: 1 Time: 8s
Restart Phase #:39 Files: 1 Time: 0s
Serial Phase #:40 Files: 1 Time: 10s
Serial Phase #:41 Files: 1 Time: 3s
Restart Phase #:42 Files: 1 Time: 0s
Parallel Phase #:43 Files: 2 Time: 411s
Restart Phase #:44 Files: 1 Time: 1s
Serial Phase #:45 Files: 2 Time: 510s
Restart Phase #:46 Files: 1 Time: 1s
Parallel Phase #:47 Files: 2 Time: 35s
```


Debugging and Tracing

- `catctl.pl -Z catupgrd.sql`
 - Debugs `catcon.pm`
- `catctl.pl -Z catupgrd.sql`
 - Debugs `catctl.pl`
 - Done using PERL to generate trace files in format `catctl_YYYYMMDDHHMNSC_pid_trace.log`

CATCON.PM

```
catconInit: base for log and spool file names
= catupgrd
  running catconInit(User           = 0,
                      InternalUser = ,
                      SrcDir        = 0,
                      LogDir        = 0,
                      LogBase       = catupgrd,
```

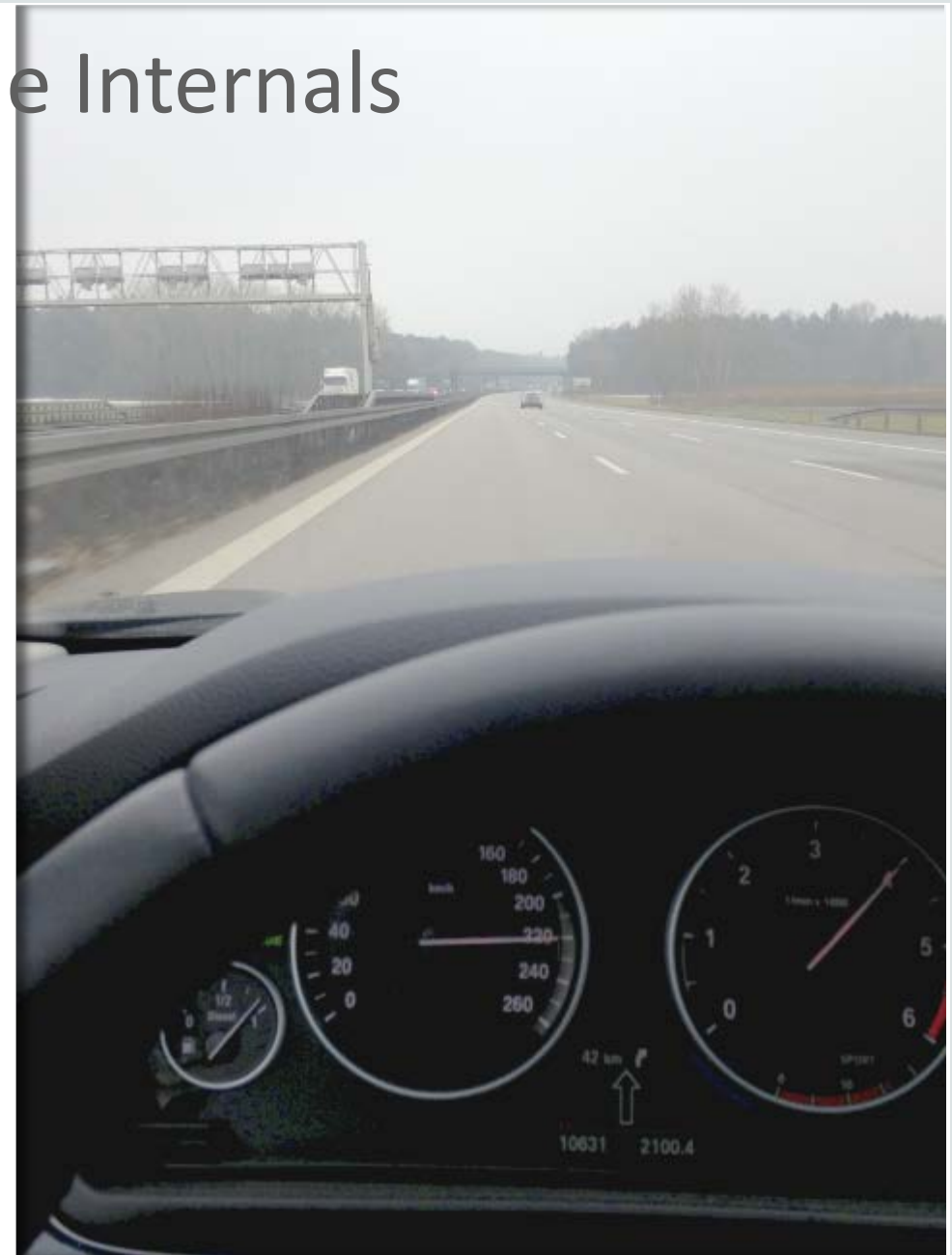
CATCTL.PL

Package `catctl.pl`.

```
269:   my @gArgs;
270:   foreach $argnum (0 .. $#ARGV)
271:   {
272:       push (@gArgs, $ARGV[$argnum]);
272:       push (@gArgs, $ARGV[$argnum]);
272:       push (@gArgs, $ARGV[$argnum]);
```

Oracle Multitenant Database Upgrade Internals

- 1 Introduction
- 2 Database Upgrade News
- 3 Oracle Multitenant Overview
- 4 Multitenant Upgrade
- 5 Inside catctl.pl – and Options
- 6 Performance Figures**
- 7 Wrap Up



Performance Example 1: Exadata V1 25 PDBs –n26

- Exadata V1
 - 2 CPUs – 4 cores each – 64 GB RAM
 - 8GB SGA
 - 25 PDBs
 - 25 GB in size each
 - Swingbench schemas



- Upgrade using just one node:
 - `catctl.pl -M -n 26 -N 2`
 - 3 cycles
 - Cycle 1: CDB\$ROOT
 - Cycle 2: PDB\$SEED, PDB1-PDB12
 - Cycle 3: PDB13-PDB25
 - TOTAL: **2:18:57 hrs**
 - CDB\$ROOT: **20:54 mins**
 - Average (mean) PDB: **59:00 mins**
 - Mapped to single databases: **5:07 mins/db**

Performance Example 2: Linux Server 252 PDBs –n 24

- 24 core Server
 - 2 CPUs – 12 cores each
 - 252 PDBs
 - 70 GB RAM
- Upgrade:
 - `sga_target=40GB`
 - Flush Shared Pool operations removed
 - `catctl.pl -n 24 -N 2`
 - 22 cycles
 - Cycle 1: CDB\$ROOT
 - Cycle 2: PDB\$SEED, PDB1-PDB11
 - ...
 - Cycle 21: PDB239-PDB250
 - Cycle 22: PDB251-PDB252
 - TOTAL: **15 hrs 10m 56s**
 - Median PDB upgrade: **38m 31s**
 - Mapped to single databases: **3m 36s per DB**


Performance Example 3: Linux Server 252 PDBs **-n 32**

- 32 core Server
 - 2 CPUs – 16 cores each
 - 252 PDBs
 - 252 GB RAM
- Upgrade:
 - `sga_target=40GB`
 - Flush Shared Pool operations removed
 - `catctl.pl -n 32 -N 2`
 - 17 cycles
 - Cycle 1: CDB\$ROOT
 - Cycle 2: PDB\$SEED, PDB1-PDB15
 - ...
 - Cycle 17: PDB239-PDB252
 - TOTAL: **13h 27m**
 - Median PDB upgrade: **46m 16s**
 - Mapped to single databases: **3m 11s per DB**

Performance Example 4: Linux Server 252 PDBs –n 32 -M

- 32 core Server
 - 2 CPUs – 16 cores each
 - 252 PDBs
 - 252 GB RAM

**33% Faster with
the –M option**

- Upgrade:
 - sga_target=40GB
 - Flush Shared Pool operations removed
 - catctl.pl -n 32 -N 2 -M
 - 17 cycles
 - Cycle 1: CDB\$ROOT
 - Cycle 2: PDB\$SEED, PDB1-PDB15
 - ...
 - Cycle 17: PDB239-PDB252
 - TOTAL: 9h 6m 18s
 - Median PDB upgrade: 27m 48s
 - Mapped to single databases: 2m 10s per DB
- 

Performance Example 5: Linux Server 252 PDBs **-n 64 -M**

- 32 core Server
 - 2 CPUs – 16 cores each
 - 252 PDBs
 - 252 GB RAM

**Increased
contention slows
down upgrade**

- Upgrade:
 - `sga_target=40GB`
 - Flush Shared Pool operations removed
 - `catctl.pl -n 64 -N 2 -M`
 - 17 cycles
 - Cycle 1: CDB\$ROOT
 - Cycle 2: PDB\$SEED, PDB1-PDB31
 - ...
 - Cycle 8: PDB224-PDB252
 - TOTAL: **15h 14m 34s**
 - Median PDB upgrade: **99m 30s**
 - Mapped to single databases: **3m 37s per DB**

Conclusion

- Using the –M option decreases total upgrade time at the expense of individual PDB availability
 - The –M option gives more relative benefit with larger numbers of PDBs per cycle
- More PDBs per cycle is generally better than more processes per PDB
 - This will decrease overall upgrade time, but individual PDB upgrades will take longer
 - At some point contention (dictionary, shared pool) will increase dramatically
- Keep no. of cycles as low as possible...to a point
 - **There is no magic formula**

Oracle Multitenant Database Upgrade Internals

- 1 Introduction
- 2 Database Upgrade News
- 3 Oracle Multitenant Overview
- 4 Multitenant Upgrade
- 5 Inside catctl.pl – and Options
- 6 Performance Figures
- 7 Wrap Up

The screenshot shows a blog post from Oracle OpenWorld 2013. The main heading is "Focus on Database Upgrade at OpenWorld 2013". The author is Mike Dietrich, a Consulting Member of Technical Staff at Oracle. The post discusses the upcoming Oracle OpenWorld conference in September and provides a "Focus on Database Upgrade" document listing sessions, demos, and hands-on labs. A table of sessions is provided below the text.

Session Title	Time	Location	Reference
General Session: Oracle Database 12c—Prepared for Clouds and Big Data	10:45 AM - 11:45 AM	Moscone North - Hall D	GEN8229
Consolidating Databases with Oracle Database 12c	12:15 PM - 1:15 PM	Moscone South - 102	CON8707
Different Ways to Upgrade, Migrate, and Consolidate with Oracle Database 12c	3:15 PM - 4:15 PM	Moscone South - 102	CON8176

Resources

- Download slides from:
 - <http://blogs.oracle.com/UPGRADE>

The screenshot shows a blog post from Oracle. At the top, there is a red banner with the text "Upgrade your Database - NOW!" and the Oracle logo. Below the banner, the main content area features a "Recent Posts" sidebar on the left, a central article titled "Incremental Statistics Collection improved in Oracle 12c" by Mike Dietrich, and an "About" section on the right. The article text describes a workshop in Asia. Below the text is a photograph of a busy airport terminal with people walking and overhead signs. The "About" section includes a profile picture of Mike Dietrich and his title as Senior Principal Technologist.



Resources

- Step-by-Step Description:

- https://blogs.oracle.com/UPGRADE/entry/upgrade_pdbs_everything_at_once1
- https://blogs.oracle.com/UPGRADE/entry/upgrade_pdbs_one_at_a



Hardware and Software Engineered to Work Together

ORACLE®