



Last updated: 08-SEP-2010

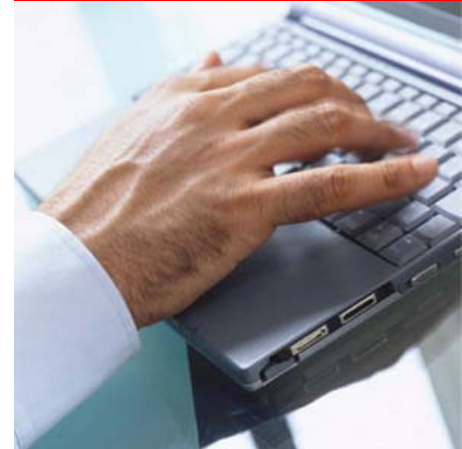
ORACLE®

Oracle Database 11.2 Upgrade Methods

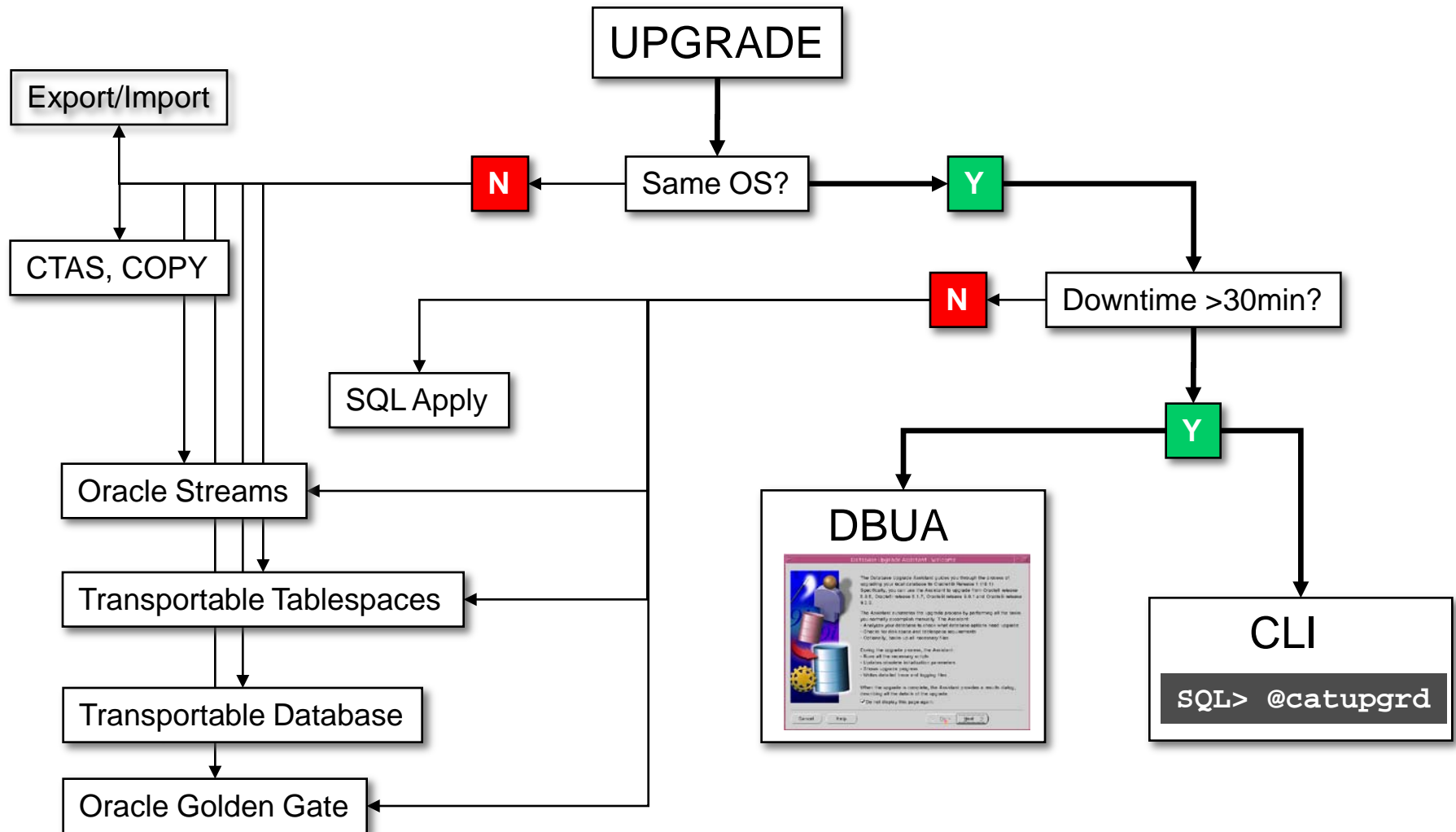
Roy F. Swonger
Database Upgrade & Utilities
ORACLE Corporation

Agenda

- Regular Upgrade Methods
- Post Upgrade Tasks
- Upgrade Alternatives
- Summary



Upgrade Alternatives



"Regular" Database Upgrade

- Upgrade duration is mainly dependent on the number of installed components
 - Completes usually in 20-90 minutes
 - No difference between DBUA and command line upgrade
 - This is **not a recommendation** to deinstall any components!!!

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
Oracle Expression Filter	00:00:18
Oracle Rule Manager	00:00:12
Gathering Statistics	00:04:53

Total Upgrade Time: 00:52:31

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

Database Upgrade Assistant (GUI)



- Features:
 - Graphically led upgrade
 - Lots of important checks
 - RAC *aware* - inclusion of all nodes
 - for RAC (almost) a must !!!
 - Offline Backup and Restore possible
 - ASM upgrade (until 11.1)
 - Oracle XE upgrade
 - Patch upgrades (as of 10.2.0.3)
- Logs:
 - `$ORACLE_HOME/cfgtoollogs/dbua`
- Documentation:
 - Oracle® Database Upgrade Guide
 - 11.1: http://download.oracle.com/docs/cd/B28359_01/server.111/b28300/toc.htm
 - 11.2: http://download.oracle.com/docs/cd/E11882_01/server.112/e10819/toc.htm

Database Upgrade Assistant (GUI)



- Silent mode:

- `$ dbua -help` shows all valid options

- See doc:

11.2:

http://download.oracle.com/docs/cd/E11882_01/server.112/e10819/upgrade.htm#UPGRD12405

- Example:

```
dbua -silent -sid dwh
-oracleHome /opt/oracle/product/RDBMS10g
-diagnosticDest /opt/oracle/diag
  -sysDBAUserName sys
  -sysDBAPassword manager
-recompile_invalid_objects true
  -degree_of_parallelism 4
-emConfiguration LOCAL
  -dbsnmpPassword manager
  -sysmanPassword manager
```

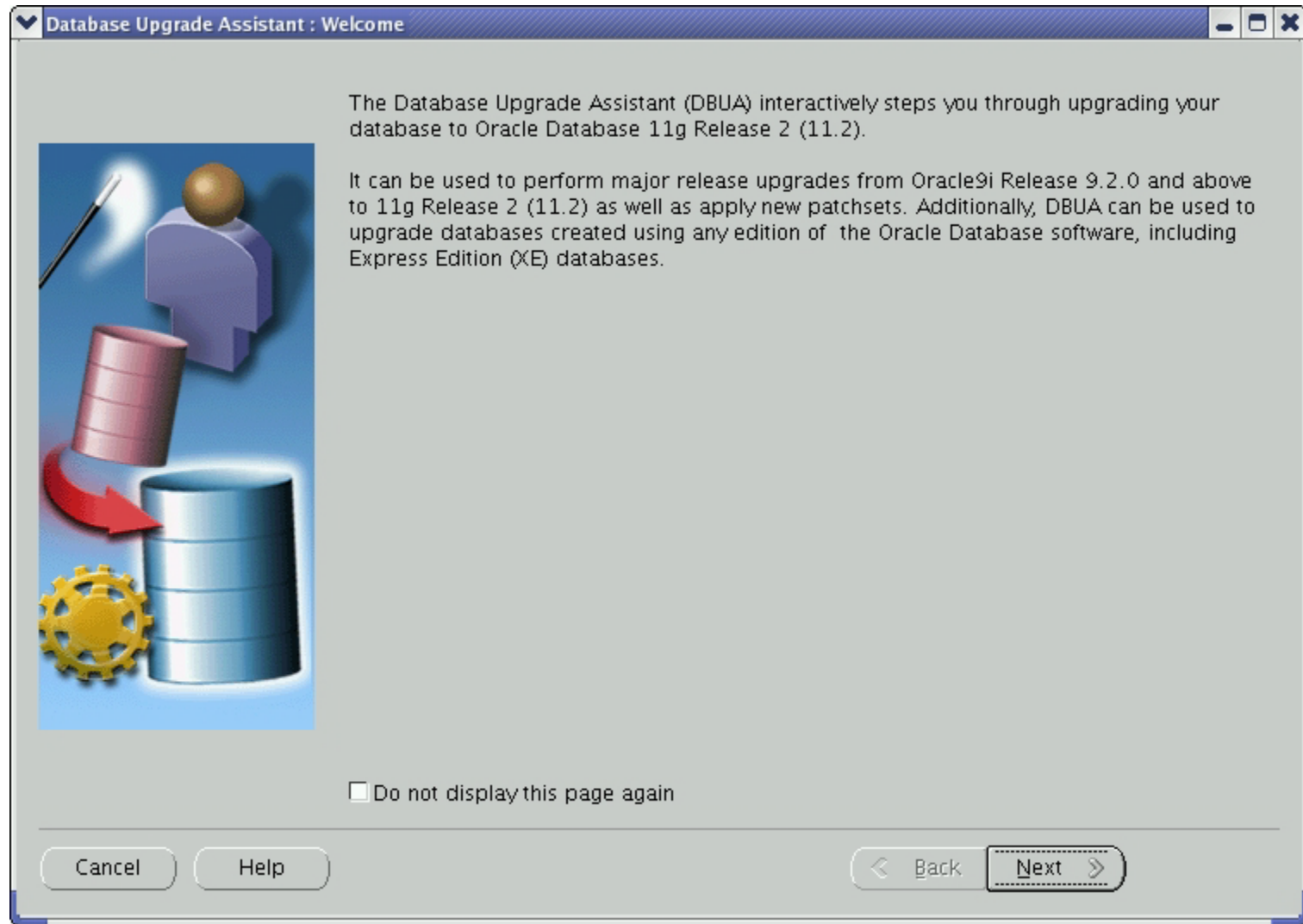
Database Upgrade Assistant (GUI)



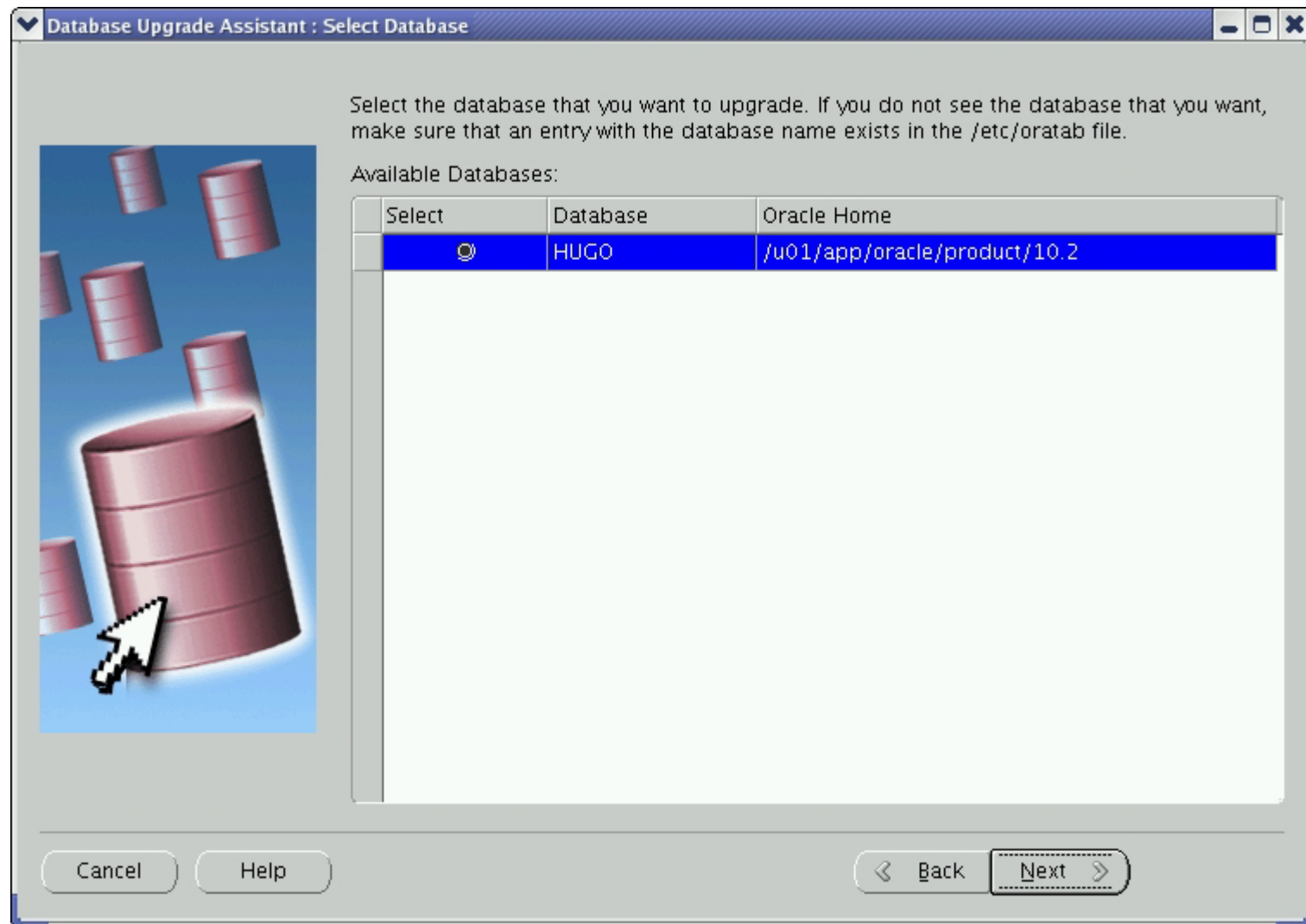
- Best Practice: Before you start DBUA
 - Run `$OH_11g/rdbms/admin/utlu112i.sql` in your current environment
 - Check especially the components status in DBA_REGISTRY
 - To remove (or reinstall) components manually:
[Note:472937.1](#) Information On Installed Database Components
[Note:753041.1](#) How to diagnose Components with NON VALID status

```
*****
Components: [The following database components will be upgraded or installed]
*****
--> Oracle Catalog Views           [upgrade] VALID
--> Oracle Packages and Types      [upgrade] VALID
--> JServer JAVA Virtual Machine   [upgrade] VALID
--> Oracle XDK for Java            [upgrade] VALID
--> Oracle XML Database            [upgrade] VALID
--> Oracle Java Packages           [upgrade] VALID
```

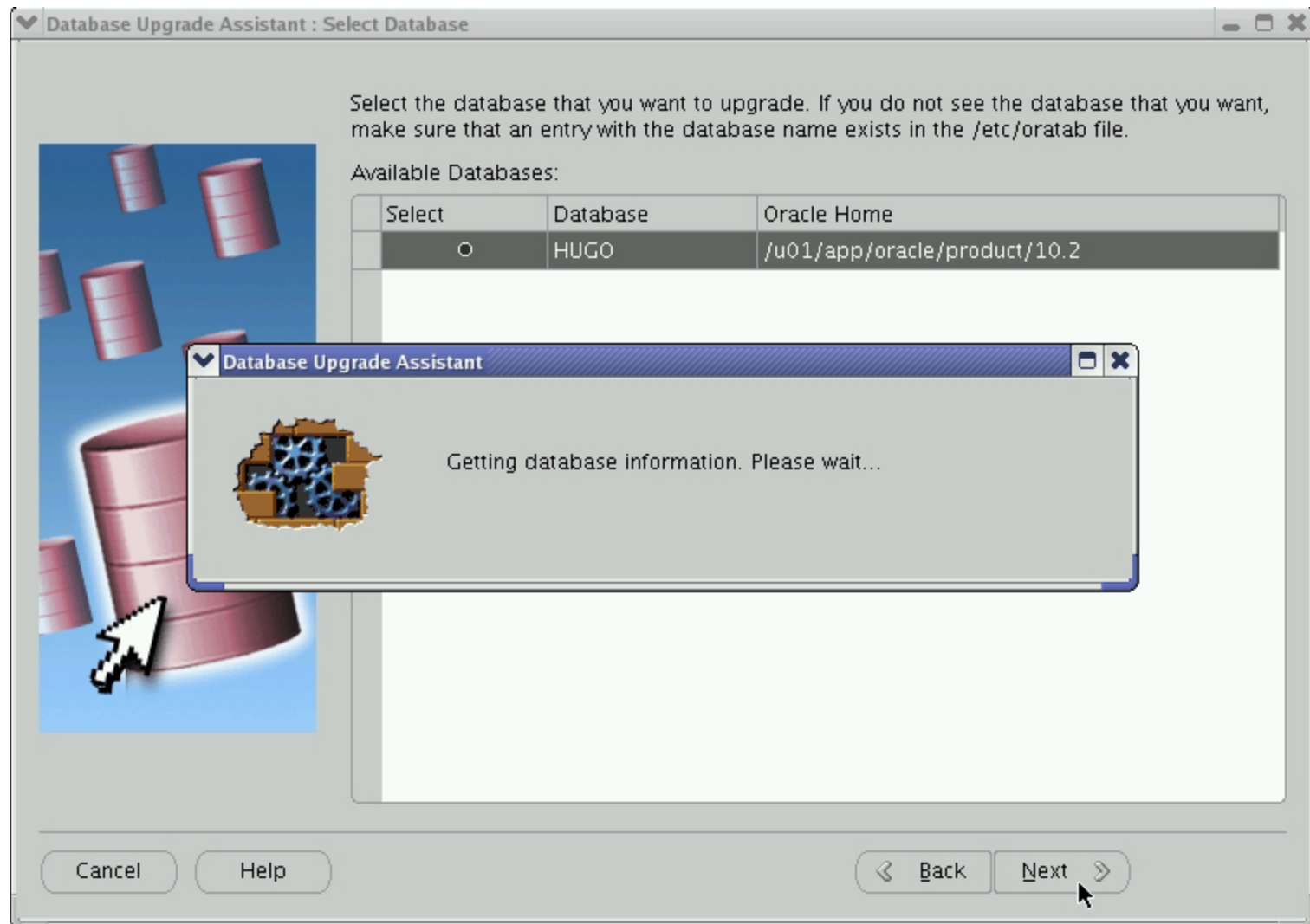
Database Upgrade Assistant (GUI)



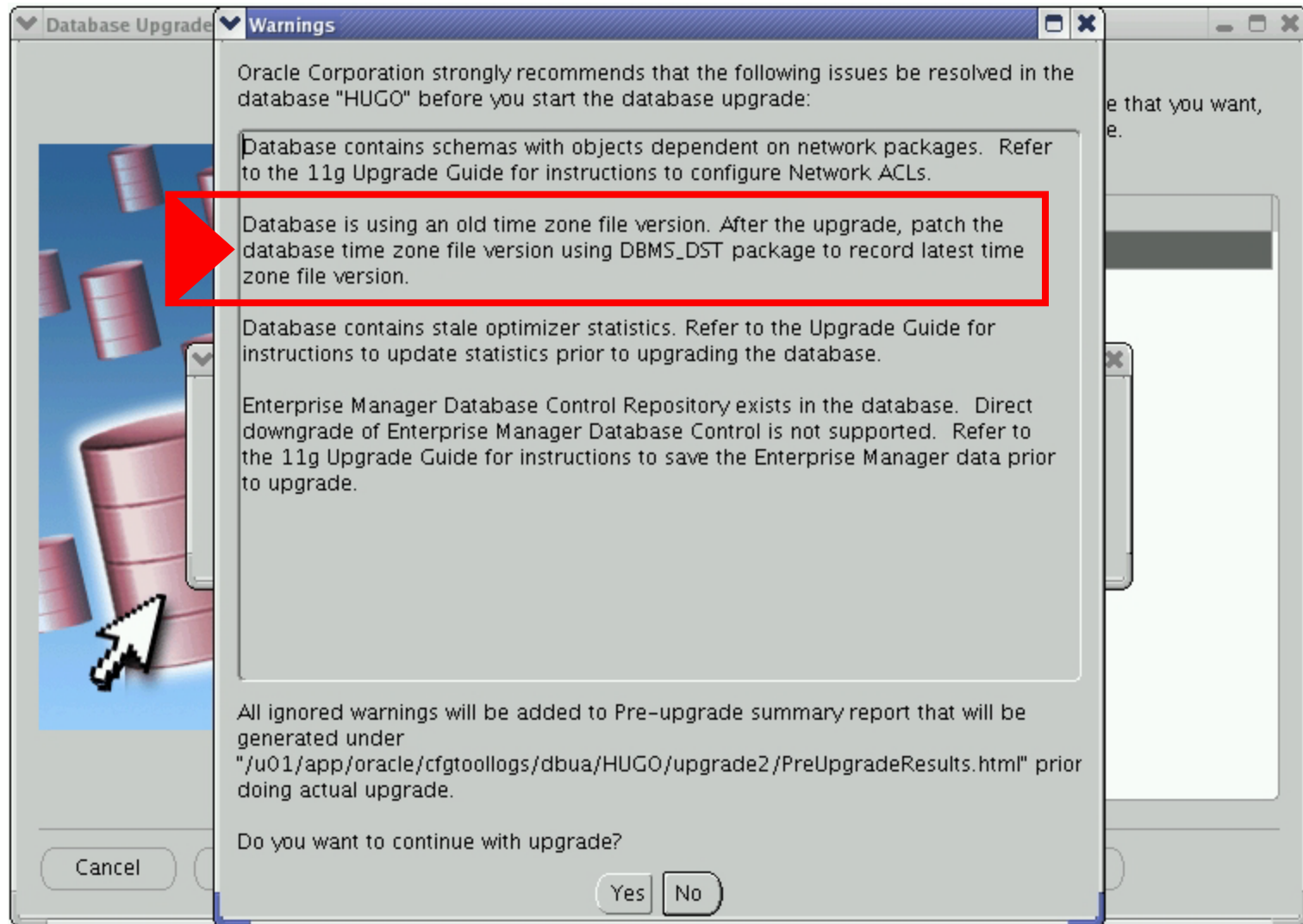
Database Upgrade Assistant (GUI)



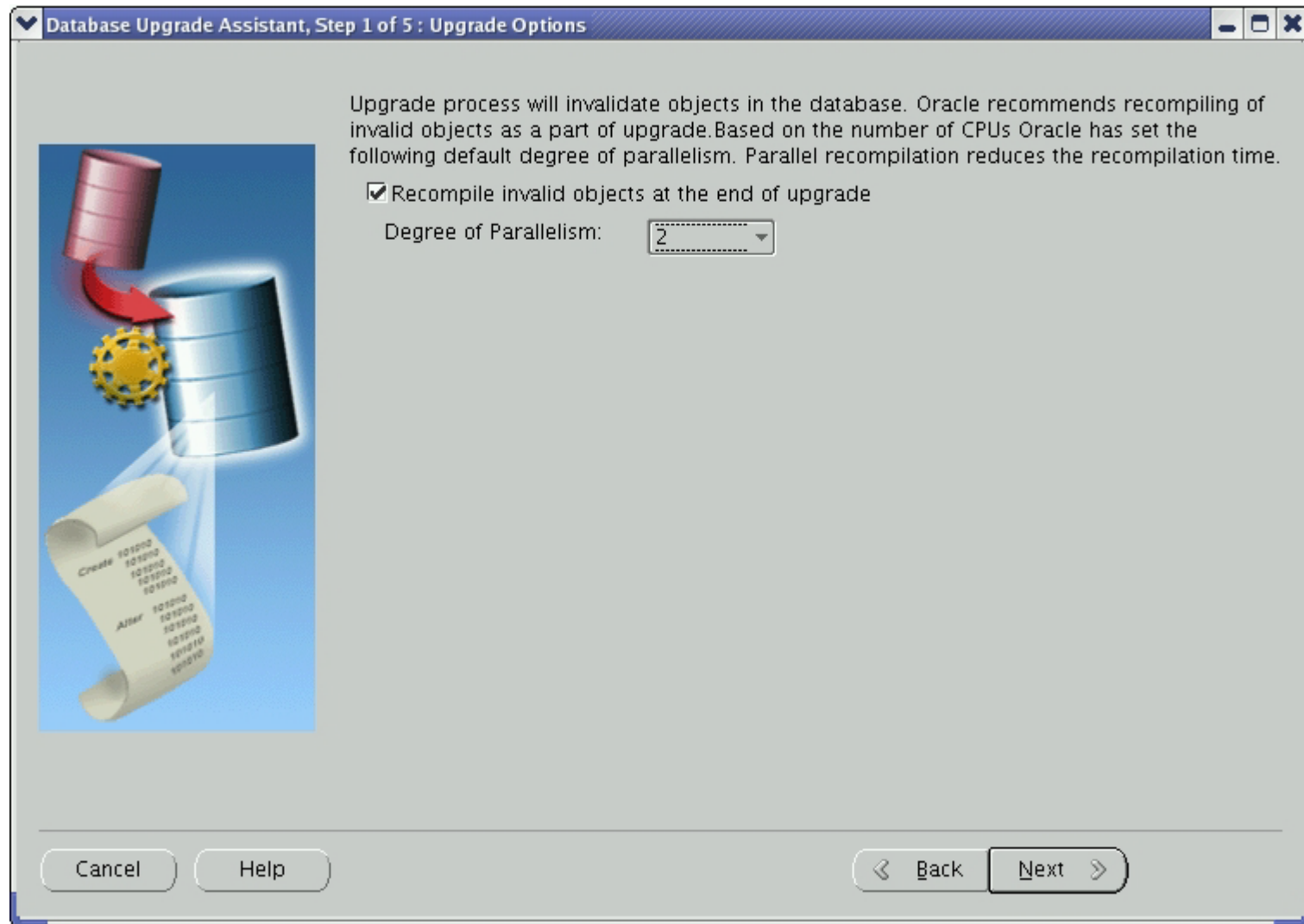
Database Upgrade Assistant (GUI)



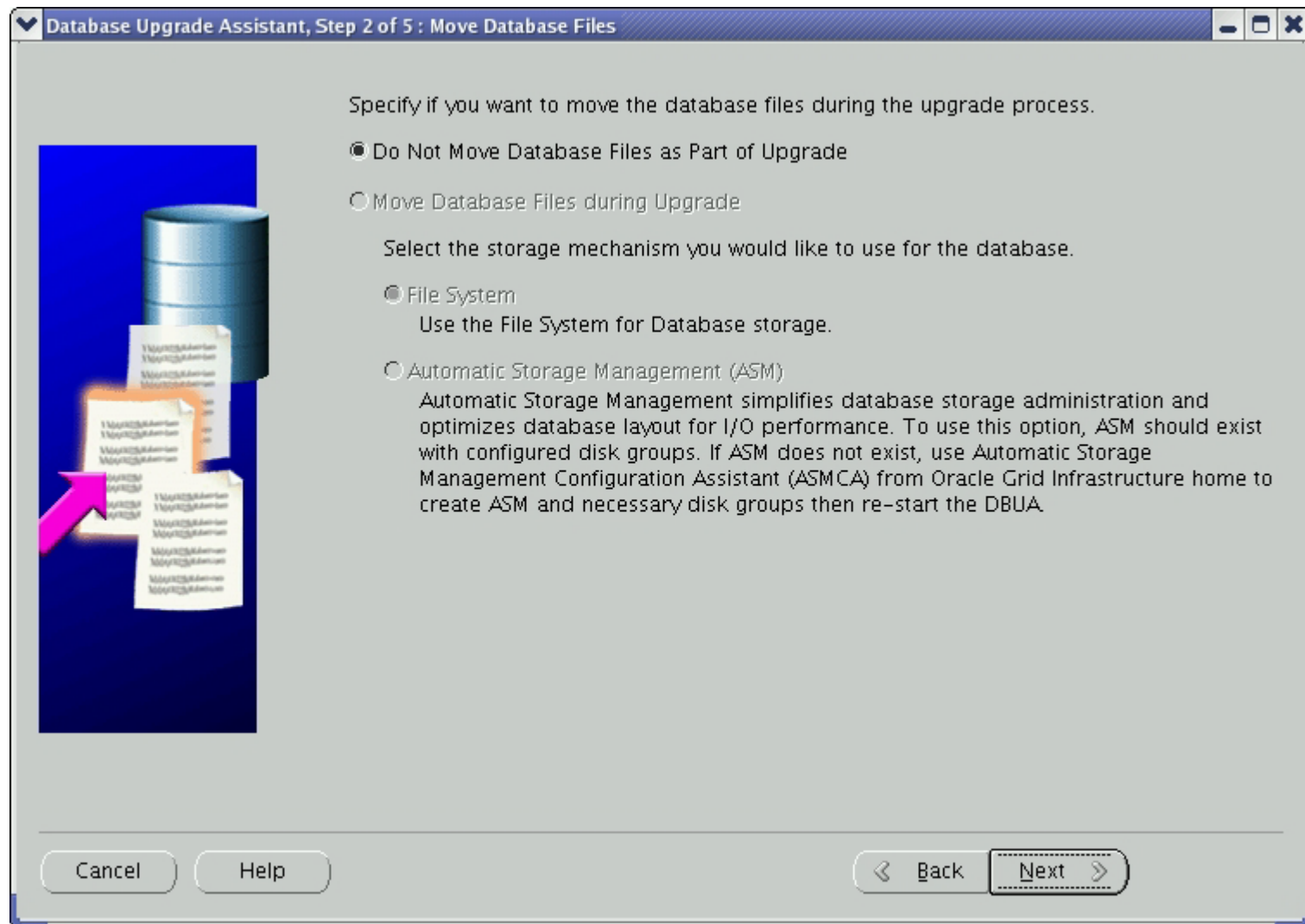
Database Upgrade Assistant (GUI)



Database Upgrade Assistant (GUI)



Database Upgrade Assistant (GUI)



Database Upgrade Assistant (GUI)

Database Upgrade Assistant, Step 3 of 4 : Recovery and Diagnostic Locations

Flash Recovery Area is an Oracle managed disk location used for storing backup and recovery related files. Oracle strongly recommends configuring a flash recovery area as it significantly enhances speed, reliability and manageability of the database recovery process.

☒ Specify Flash Recovery Area

Oracle recommends that the database files and recovery files be located on physically different disks for data protection and performance.

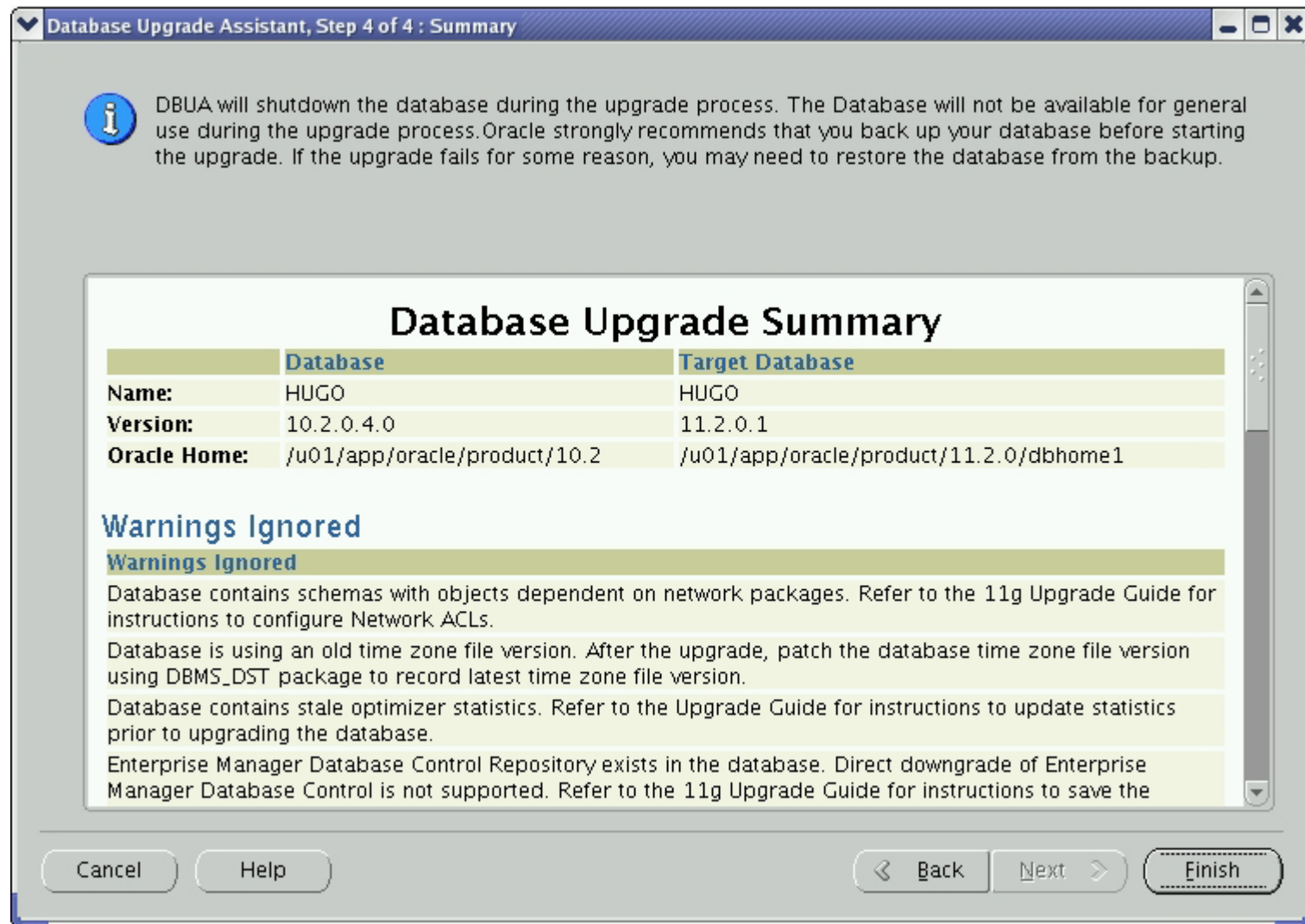
Flash Recovery Area:

Flash Recovery Area Size:

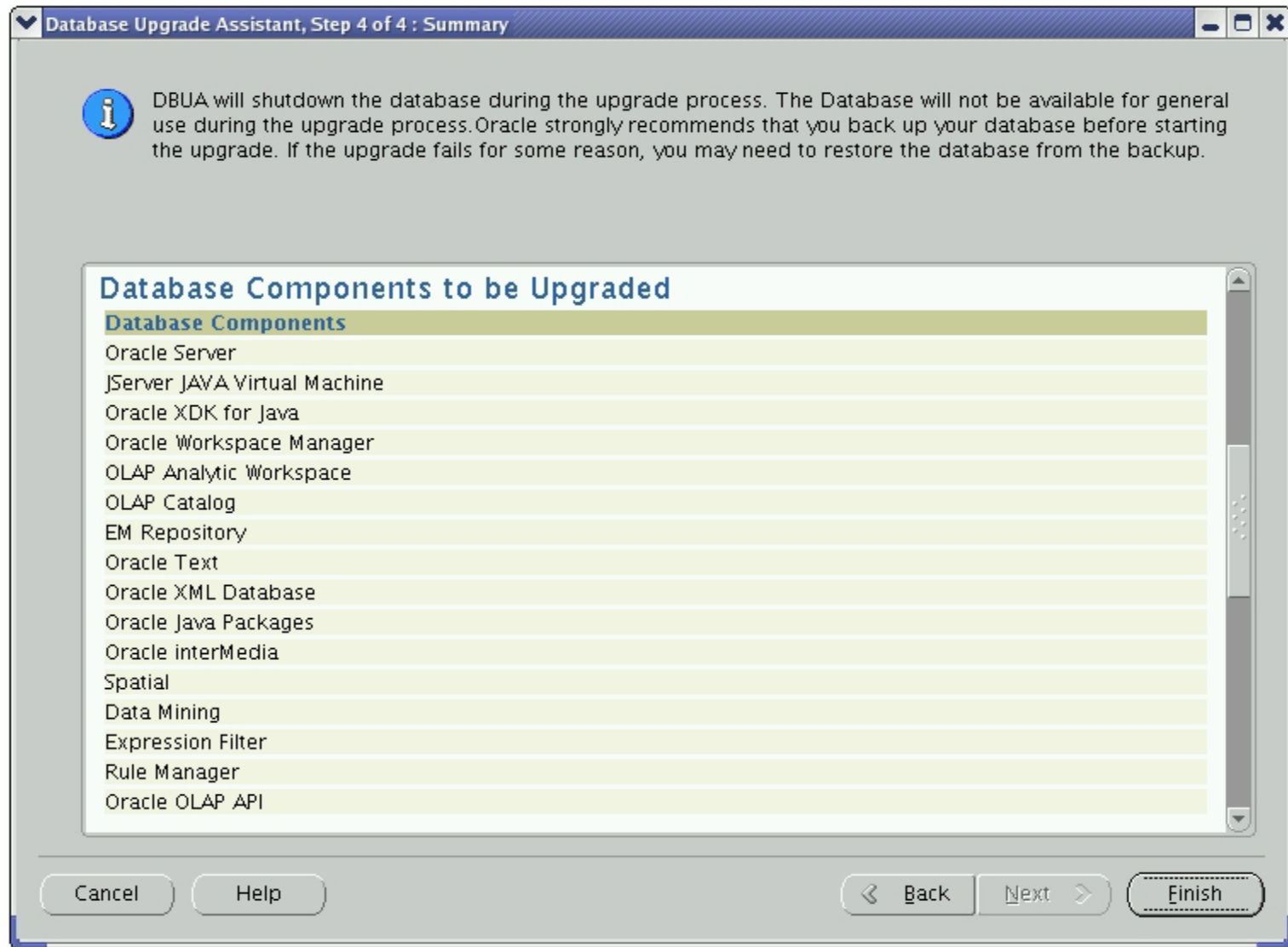
Diagnostic destination is the default location to store Oracle trace and diagnostic files. It replaces the initialization parameter settings for background dump destination, user dump destination and core dump destination from earlier releases.

Diagnostic Destination:

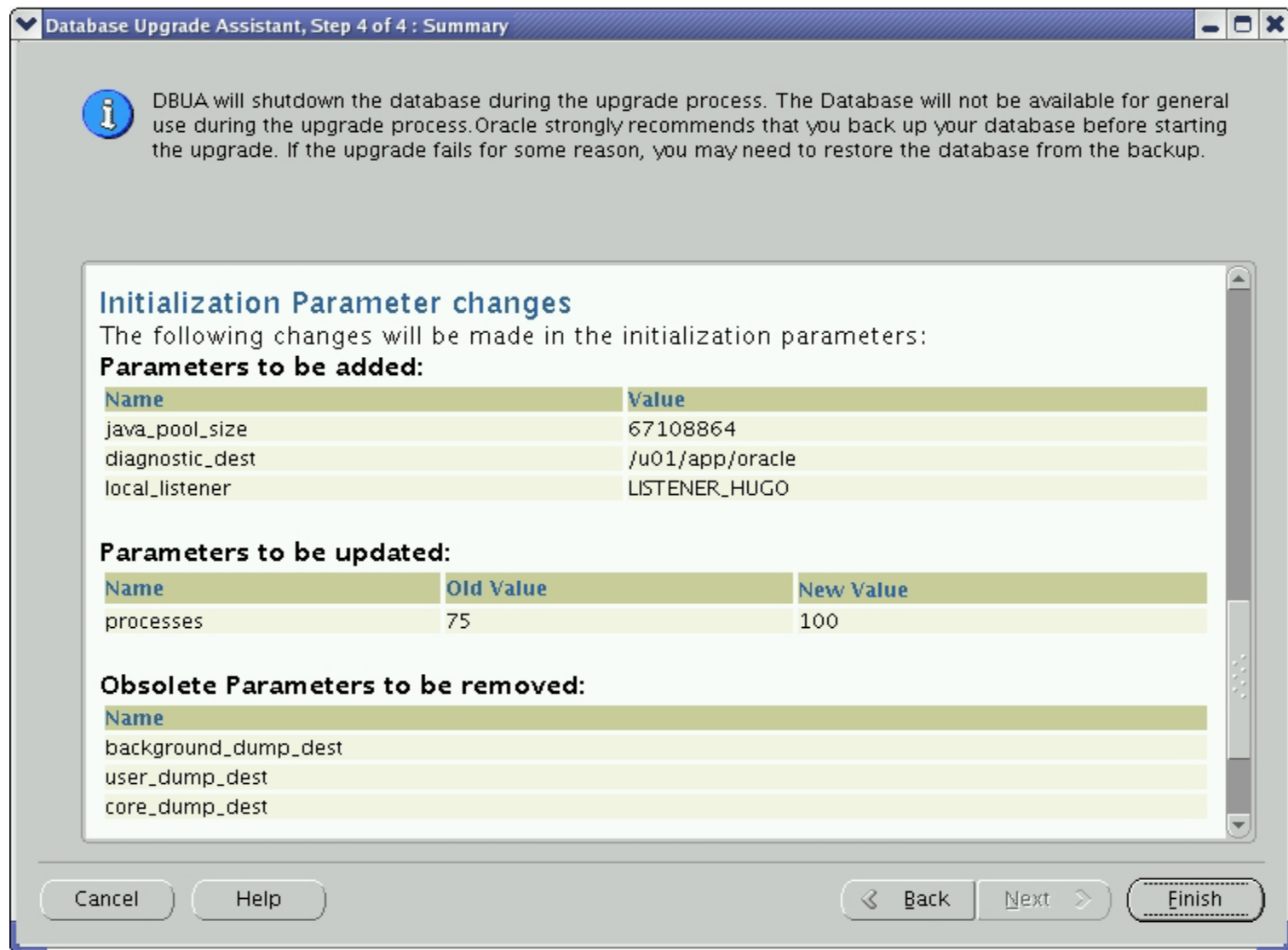
Database Upgrade Assistant (GUI)



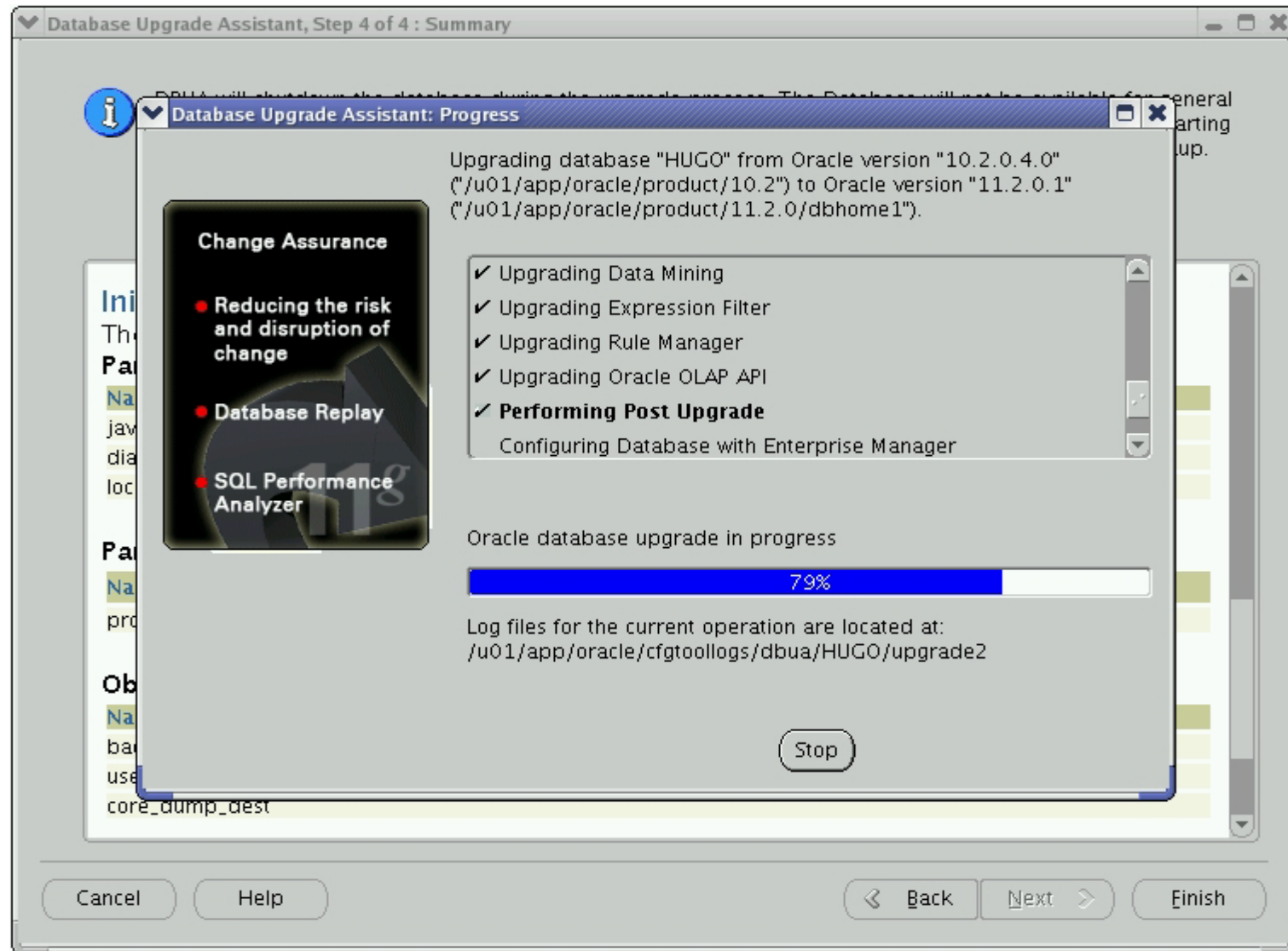
Database Upgrade Assistant (GUI)



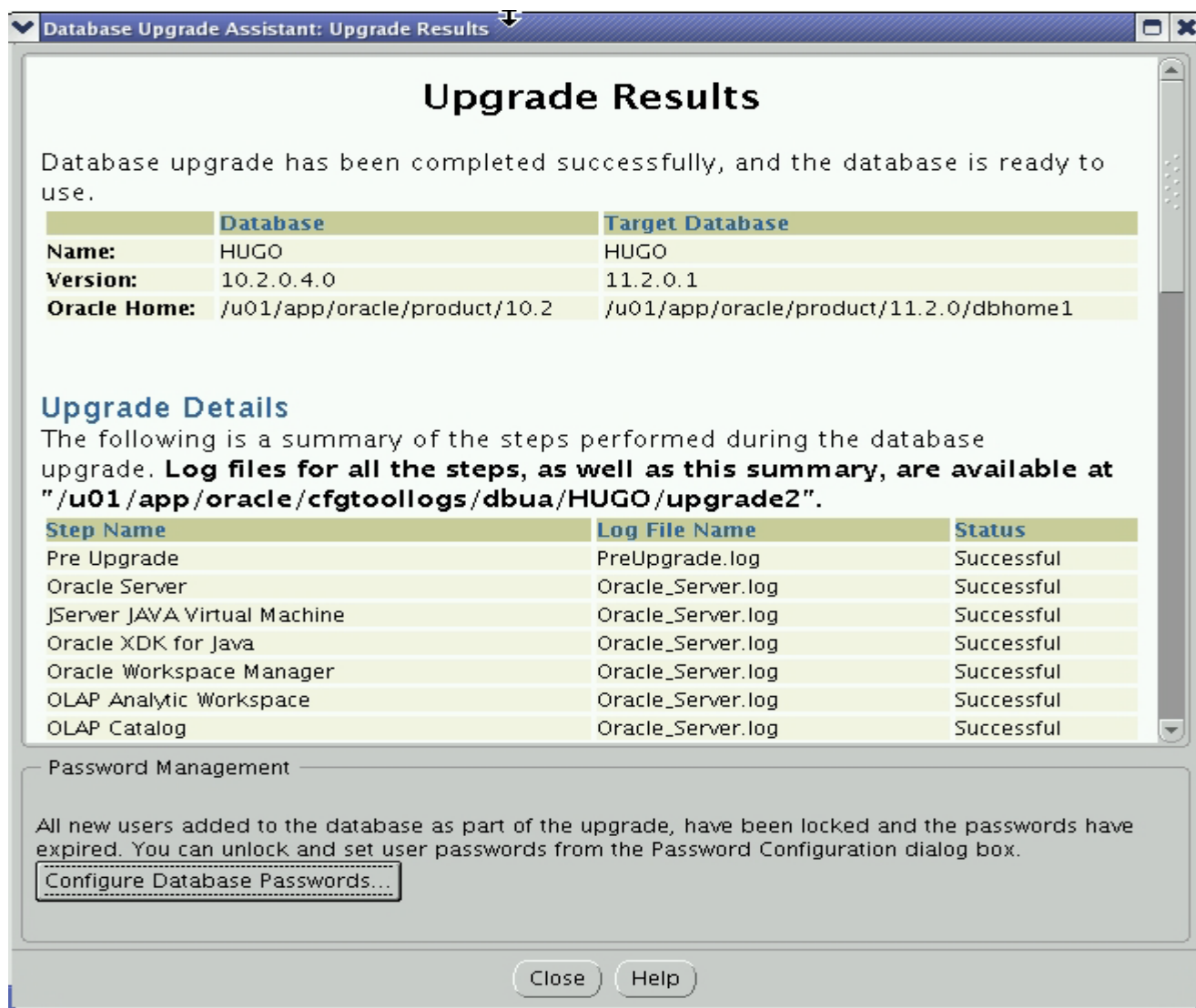
Database Upgrade Assistant (GUI)



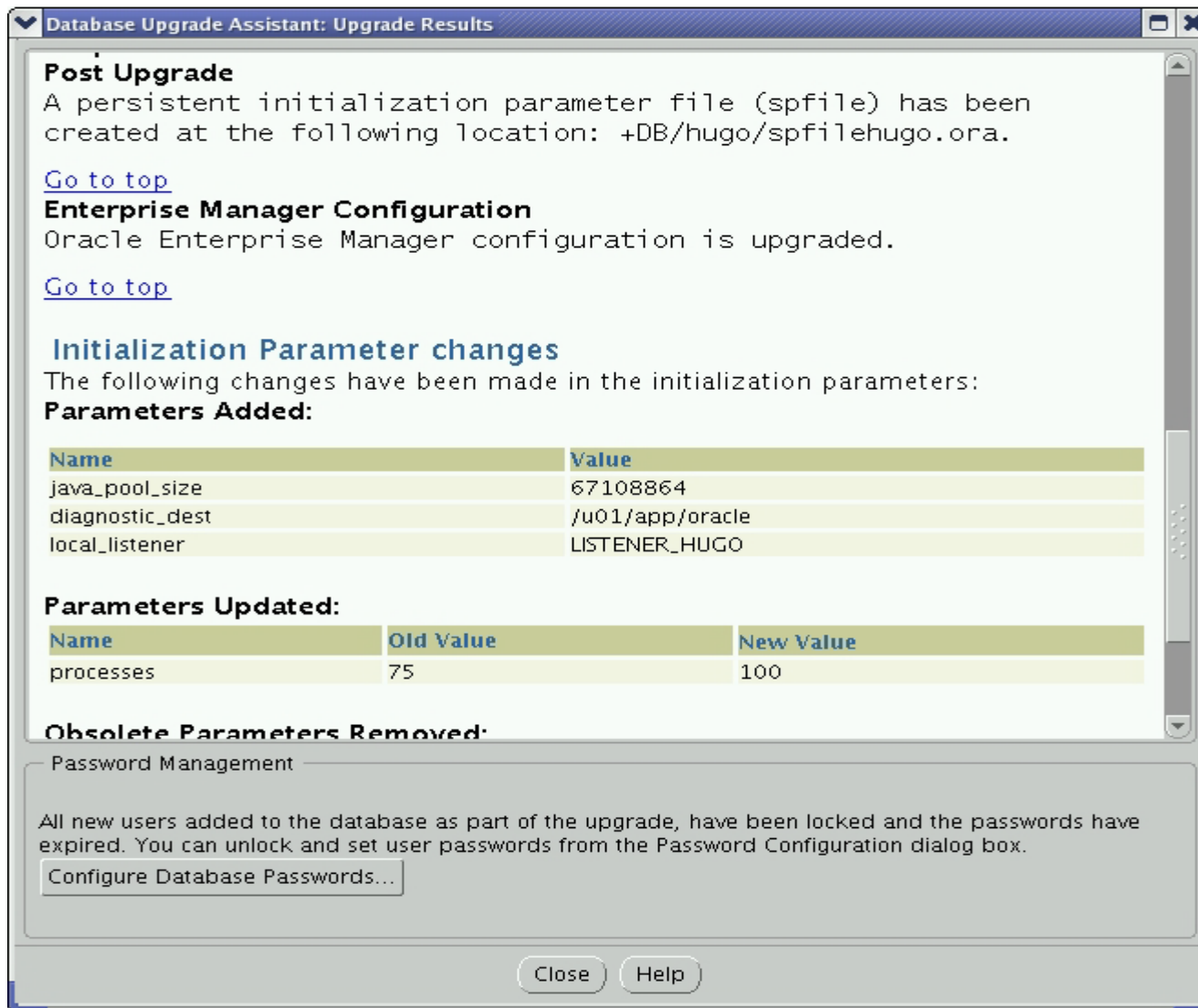
Database Upgrade Assistant (GUI)



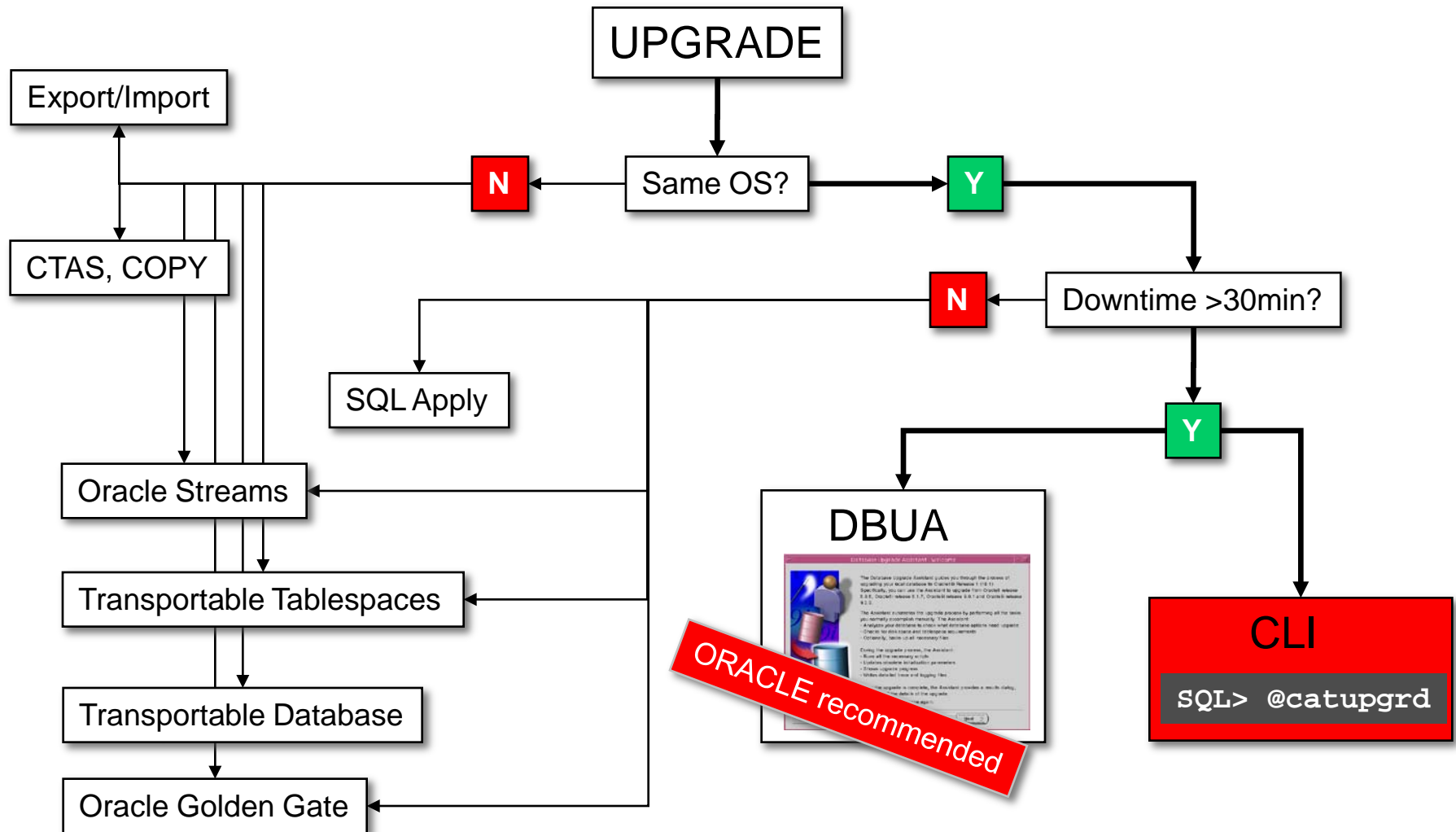
Database Upgrade Assistant (GUI)



Database Upgrade Assistant (GUI)

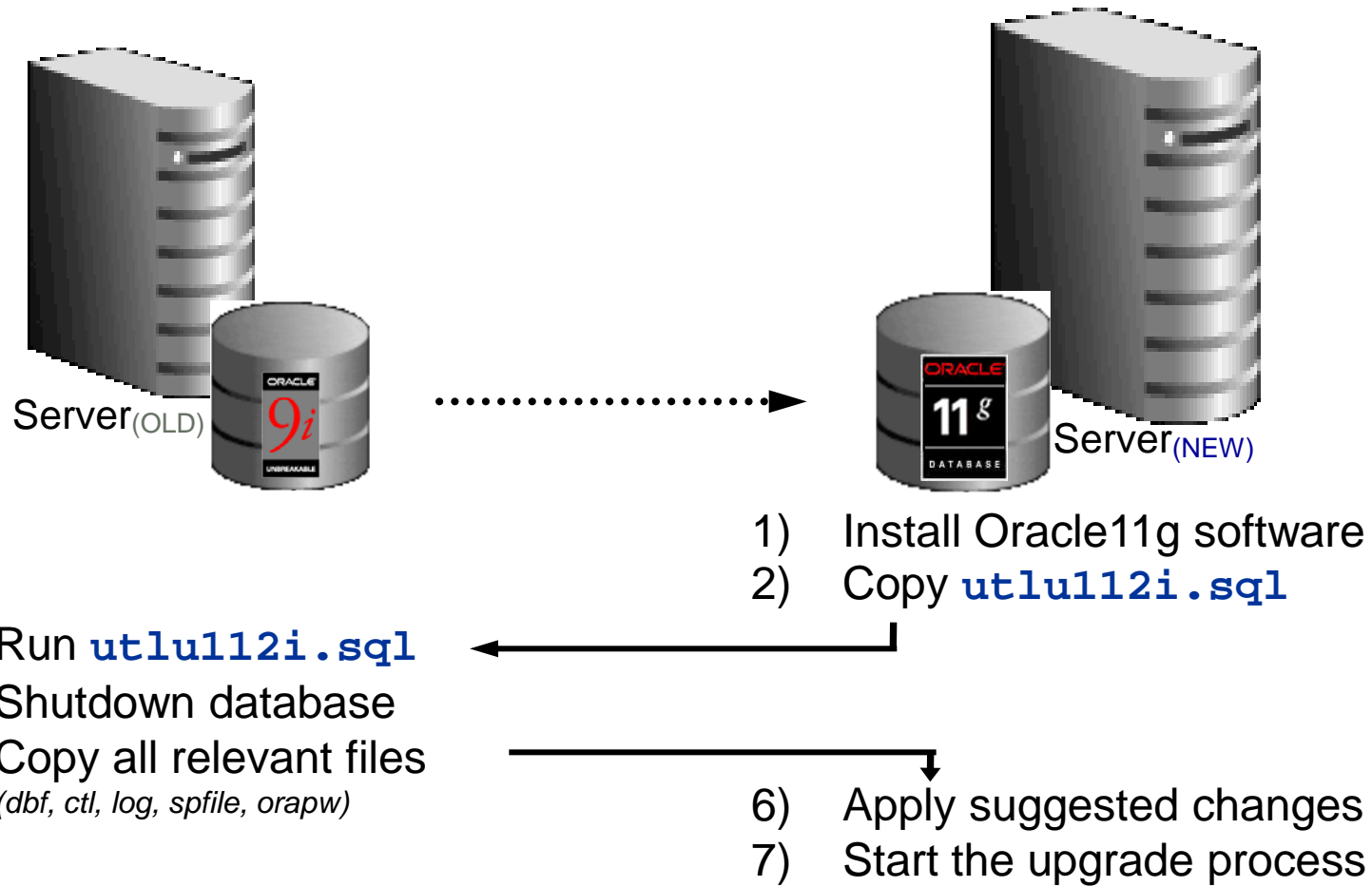


Upgrade Alternatives

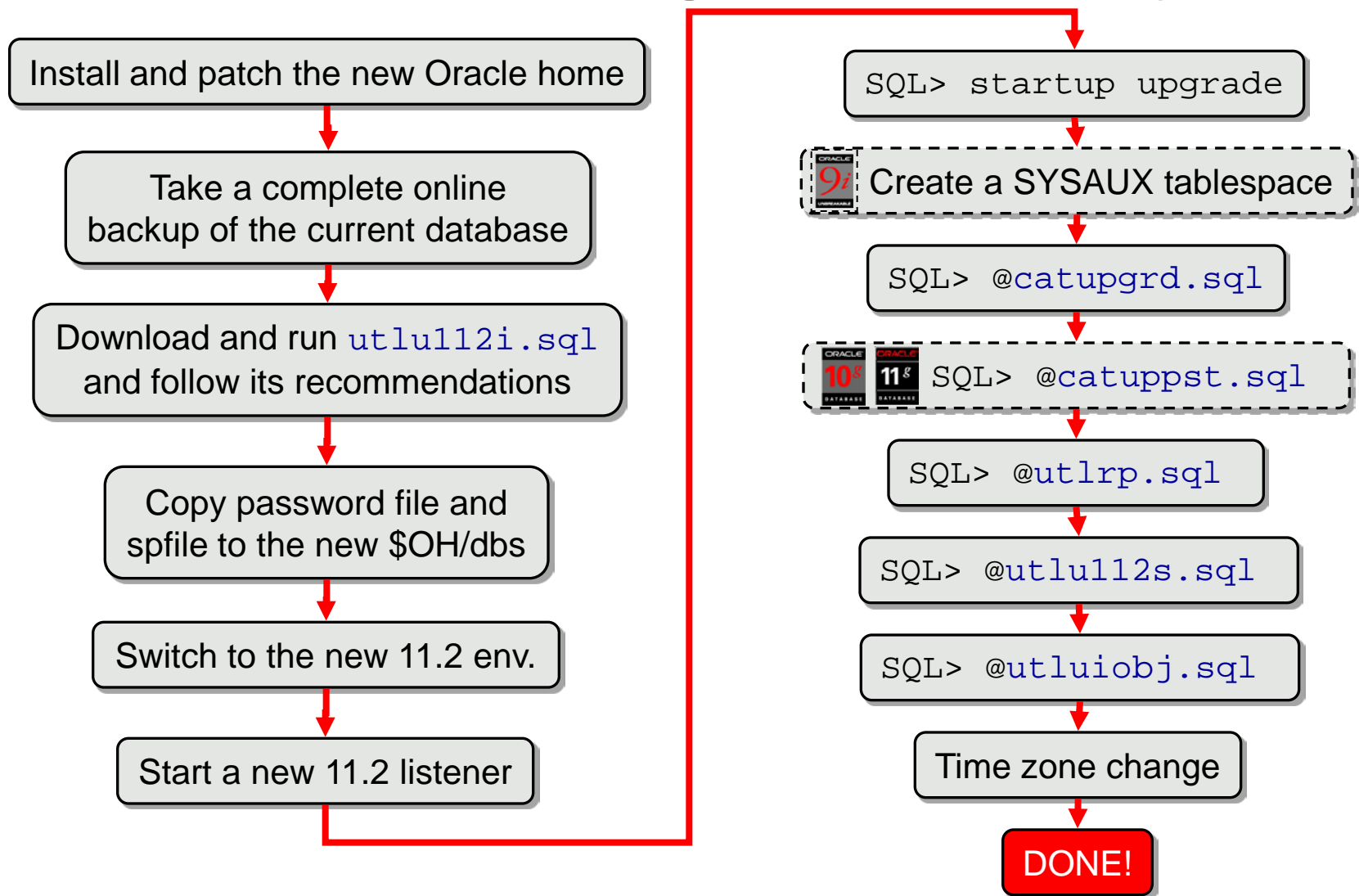


Command Line Upgrade

- Typical scenario: e.g. changing to a new server



Command Line Upgrade – Step-by-Step



Command Line Upgrade

Info

- Upgrade information script: `utlu112i.sql`
 - Run in the environment of the source database
 - Checks all init parameters and displays warnings for obsolete and deprecated parameters
 - Checks
 - Components
 - Tablespace SYSAUX
 - National Characterset
 - Timezone file version check
 - Cluster check

Command Line Upgrade

- Get the current version of **utlu_{nm}i.sql**
 - Download it now!
 - **Note:884522.1**

Coming From Version	Upgrade Target Version
9.2.0 (9.2.0.8 and beyond), 10.1.0, 10.2.0, 11.1.0	11gR2 - utlu112i.sql
9.2.0 (9.2.0.4 and beyond), 10.1.0,10.2.0	11gR1- utlu111i.sql
8.1.7, 9.0.1, 9.2.0 (9.2.0.4 and beyond), 10.1.0	10gR2 - utlu102i.sql

Command Line Upgrade

- `utlu112i.sql`: DB info

```
Oracle Database 11.2 Pre-Upgrade Information Tool      09-21-2009 22:33:20
```

```
*****
```

```
Database:
```

```
*****
```

```
--> name:          ORCL
--> version:        10.2.0.3.0
--> compatible:     10.2.0.3.0
--> blocksize:      8192
--> platform:       Linux IA (32-bit)
1 --> timezone file: V4
```

```
*****
```

```
Miscellaneous Warnings
```

```
*****
```

```
2 WARNING: --> Database is using a timezone file older than version 11.
.... After the release migration, it is recommended that DBMS_DST package
.... be used to upgrade the 10.2.0.3.0 database timezone version
.... to the latest version which comes with the new release.
```

- Timezone conversion should be done after the upgrade has completed
 - Recommended
 - Necessary if datatype `TIMESTAMP WITH TIMEZONE` is used

Command Line Upgrade

- `utlu112i.sql`: Tablespaces adequate size?

```
*****  
Tablespaces: [make adjustments in the current environment]  
*****  
3 --> SYSTEM tablespace is adequate for the upgrade.  
.... minimum required size: 543 MB  
.... AUTOEXTEND additional space required: 163 MB  
--> UNDOTBS1 tablespace is adequate for the upgrade.  
.... minimum required size: 233 MB  
.... AUTOEXTEND additional space required: 18 MB  
--> SYSAUX tablespace is adequate for the upgrade.  
.... minimum required size: 186 MB  
.... AUTOEXTEND additional space required: 66 MB  
--> TEMP tablespace is adequate for the upgrade.  
.... minimum required size: 61 MB  
.... AUTOEXTEND additional space required: 41 MB
```

Command Line Upgrade

- `utlu112i.sql`: Init parameter changes?

```
*****
Update Parameters: [Update Oracle Database 11.2 init.ora or spfile]
*****
WARNING: --> "java_pool_size" needs to be increased to at least 64 MB
.
*****
Renamed Parameters: [Update Oracle Database 11.2 init.ora or spfile]
*****
-- No renamed parameters found. No changes are required.
.
*****
Obsolete/Deprecated Parameters: [Update Oracle Database 11.2 init.ora or spfile]
*****
--> background_dump_dest          11.1          DEPRECATED    replaced by
"diagnostic_dest"
--> user_dump_dest                11.1          DEPRECATED    replaced by
"diagnostic_dest"
--> core_dump_dest                11.1          DEPRECATED    replaced by
"diagnostic_dest"
.
```

Command Line Upgrade

- [utlu112i.sql](#): Components and options?

```
*****
5 Components: [The following database components will be upgraded or installed]
*****
--> Oracle Catalog Views           [upgrade]  VALID
--> Oracle Packages and Types      [upgrade]  VALID
--> JServer JAVA Virtual Machine   [upgrade]  VALID
--> Oracle XDK for Java            [upgrade]  VALID
--> Oracle XML Database            [upgrade]  VALID
--> Oracle Java Packages           [upgrade]  VALID
```

- Annotation:

You'll have to install all options installed for the release you are upgrading from – otherwise some components can't be upgraded

- To remove (or reinstall) components manually:

[Note:472937.1](#) Information On Installed Database Components and Schemas

[Note.300056.1](#) Debug and Validate Invalid Objects

[Note:753041.1](#) How to diagnose Components with NON VALID status


[Note.733667.1](#) How to Determine if XDB is Being Used in the Database?

Command Line Upgrade

- Create Dictionary statistics
- Shutdown the database (**IMMEDIATE/NORMAL**)
- Adjust init parameters:
 - COMPATIBLE \geq 10.1.0
 - SGA_TARGET \geq 524MB (32-bit) ... \geq 748MB (64-bit)
 - PGA_AGGREGATE_TARGET \geq 25MB
 - LOG_ARCHIVE_FORMAT must contain %s, %t and %r
- Move init.ora/SPFILE and PWDsid.ora to their new location
- Create a new 11g-Listener (*use the NETCA*)
- Change environment to point to the new \$ORACLE_HOME

Command Line Upgrade

SQL> STARTUP UPGRADE;



```
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;  
Resource Manager disabled during database migration
```

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

- Suppresses unnecessary error messages like **ORA-00942: table or view does not exist** - thus logfiles will be easier to read and check

Command Line Upgrade

- Create tablespace SYSAUX (only if source is a 9i db) :

```
SQL> CREATE TABLESPACE sysaux  
      DATAFILE 'file' SIZE 500M  
      EXTENT MANAGEMENT LOCAL  
      SEGMENT SPACE MANAGEMENT AUTO  
      ONLINE;
```


Command Line Upgrade

- One upgrade script for all releases and all components:

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

- Useful:

```
SQL> SPOOL c:\temp\upgrade.log
```

Best Practice

- Database will be shutdown when script has been completed

Command Line Upgrade

- Post upgrade script: `catuppst.sql`
 - Only necessary when upgrading from ≥ 10.1
 - Located in `$/rdbms/admin`
 - Runs when database is started up in `normal` mode
 - Will update the following information:
 - Upgrade Automatic Workload repository (AWR) baseline information
 - Upgrade ADDM task metadata
 - Update Oracle Label security (OLS) policies

Command Line Upgrade

Best Practice

- Generate fixed object stats:

```
SQL> execute  
       dbms_stats.gather_fixed_objects_stats;
```

- Purpose at this stage is to make the next step more efficient

Command Line Upgrade

- Recompilation:
 - `utlrp.sql`
 - Calls `utlprp.sql` and determines the parallel degree for recompilation based on CPU cores
 - Recompiles all INVALID objects
 - Utilizes package `utl_recomp`
 - Re-enables functional indexes automatically
 - `utlprp.sql` can be called directly like:
 - `SQL> @utlprp 7`
 - This can be useful to minimize CPU usage

Command Line Upgrade

- Progress during recompilation :

1. Query returning the number of invalid **objects remaining**.

This number should decrease with time.

```
SELECT COUNT(*) FROM obj$ WHERE status IN (4, 5, 6);
```

2. Query returning the number of objects **compiled so far**.

This number should increase with time.

```
SELECT COUNT(*) FROM UTL_RECOMP_COMPILED;
```

3. Query showing **jobs created** by UTL_RECOMP.

```
SELECT job_name FROM dba_scheduler_jobs  
WHERE job_name like 'UTL_RECOMP_SLAVE_%';
```

4. Query showing UTL_RECOMP **jobs** that are **running**.

```
SELECT job_name FROM dba_scheduler_running_jobs  
WHERE job_name like 'UTL_RECOMP_SLAVE_%';
```

Command Line Upgrade



Status

- Post upgrade script: **utlu112s.sql**
 - Run against new database in 11g environment
 - Checks the upgrade results according to **DBA_REGISTRY**
 - Displays duration of the upgrade per component and in total

Command Line Upgrade

- Post upgrade script: **utlu112s.sql**

```
SQL> @?/rdbms/admin/utlu112s.sql
```

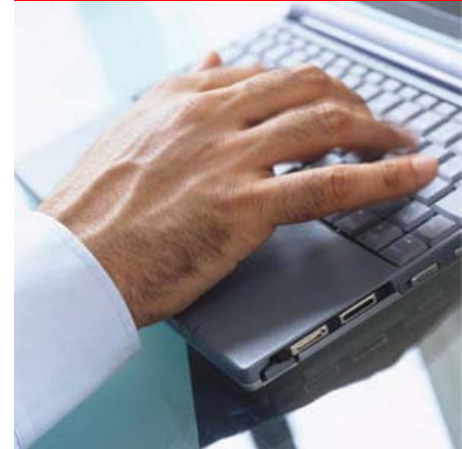
```
Oracle Database 11.2 Post-Upgrade Status Tool          10-07-2009 11:48:30
```

Component	Status	Version	HH:MM:SS
Oracle Server	VALID	11.2.0.1.0	00:24:32
JServer JAVA Virtual Machine	VALID	11.2.0.1.0	00:06:10
Oracle Workspace Manager	VALID	11.2.0.1.0	00:01:11
OLAP Analytic Workspace	VALID	11.2.0.1.0	00:00:44
OLAP Catalog	VALID	11.2.0.1.0	00:01:33
Oracle OLAP API	VALID	11.2.0.1.0	00:00:58
Oracle Enterprise Manager	VALID	11.2.0.1.0	00:15:19
Oracle XDK	VALID	11.2.0.1.0	00:06:11
Oracle Text	VALID	11.2.0.1.0	00:01:18
Oracle XML Database	VALID	11.2.0.1.0	00:07:43
Oracle Database Java Packages	VALID	11.2.0.1.0	00:00:42
Oracle Multimedia	VALID	11.2.0.1.0	00:09:57
Spatial	VALID	11.2.0.1.0	00:10:34
Oracle Expression Filter	VALID	11.2.0.1.0	00:00:23
Oracle Rules Manager	VALID	11.2.0.1.0	00:00:20
Gathering Statistics			00:11:31

```
Total Upgrade Time: 01:39:16
```

Agenda

- Regular Upgrade Methods
- Post Upgrade Tasks
- Upgrade Alternatives
- Summary



Post Upgrade Task - SPFILE

- Always **create an editable init.ora** from the current SPFILE after the upgrade has been finished
- SPFILE is:
 - It's a binary file!!!
 - Default since Oracle 9.0
 - It simply exists after using DBUA or DBCA
 - Parameter can be changed by:

```
SQL> alter system set PARAMETER=VALUE scope=both;
```

or:

```
SQL> create pfile from spfile;  
Now edit init.ora with an editor:  
SQL> startup force pfile=initDB.ora  
SQL> create spfile from pfile;
```

Post Upgrade Task – Timezone

Only in 11g Release 2

- Adjust timezone data in the database to **DST V11** or higher:

```
startup upgrade
exec dbms_dst.begin_upgrade(new_version => 11);
shutdown immediate;

startup;
set serveroutput on;
declare
  num_of_failures number;
begin
  dbms_dst.upgrade_database(num_of_failures);
  dbms_output.put_line(num_of_failures);
  dbms_dst.end_upgrade(num_of_failures);
  dbms_output.put_line(num_of_failures);
end;
/
```

- For more information see the Globalization Doc:

http://download.oracle.com/docs/cd/E11882_01/server.112/e10729/ch4datetime.htm#NLSPG261

Post Upgrade Task - Workload Statistics

- Gather **system statistics** during a regular workload period - otherwise non-appropriate values for the CBO will be used:

```
SQL> EXECUTE dbms_stats.gather_system_stats('start');
```

<< Run it for several hours – does not generate overhead!!! >>

```
SQL> EXECUTE dbms_stats.gather_system_stats('stop');
```

```
SQL> select pname NAME, pval1 VALUE, pval2 INFO
       from aux_stats$;
```

NAME	VALUE	INFO
-----	-----	-----
STATUS		COMPLETED
DSTART		04-03-2010 12:30
DSTOP		05-03-2010 12:30
FLAGS	1	
CPUSPEEDNW	2498,65	
IOSEEKTIM	11,405	
IOTFRSPEED	25595,605	
...		

Post Upgrade Task – Fixed Table Statistics

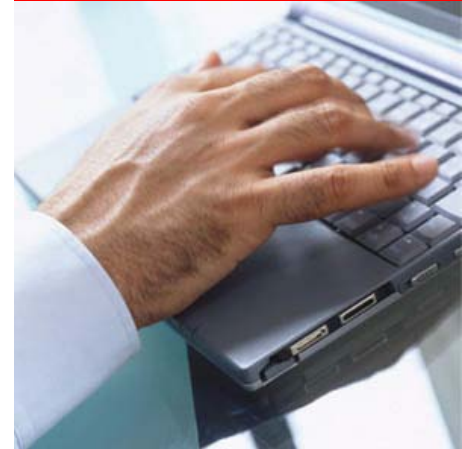
- Create **fixed table statistics** directly after catuppst.sql has completed:

```
SQL> execute  
       dbms_stats.gather_fixed_objects_stats;
```

- Create fixed table statistics again after a week with regular production workload
- Thereafter, this task should be needed only a few times per year

Agenda

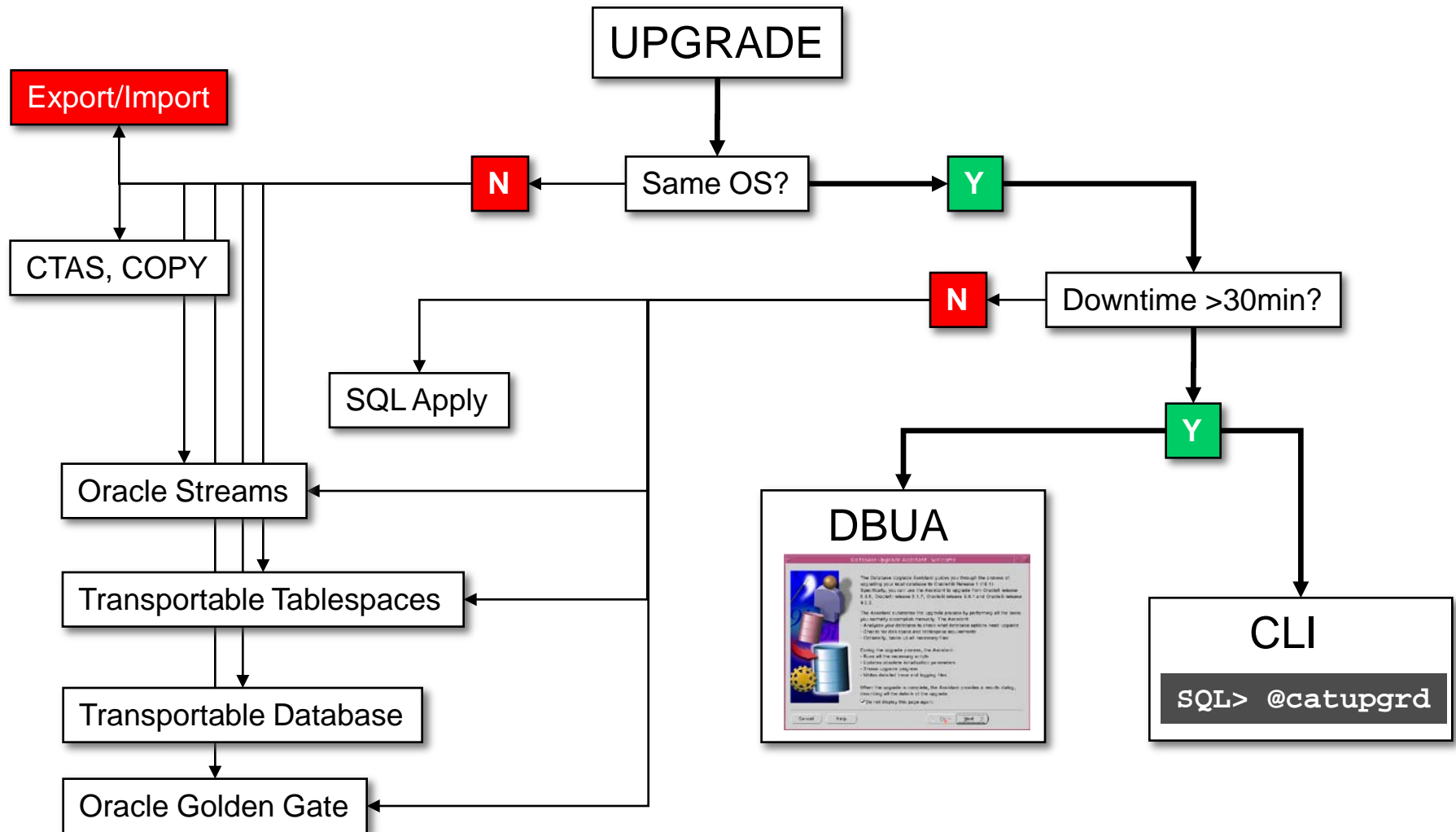
- Regular Upgrade Methods
- Post Upgrade Tasks
- Upgrade Alternatives
- Summary



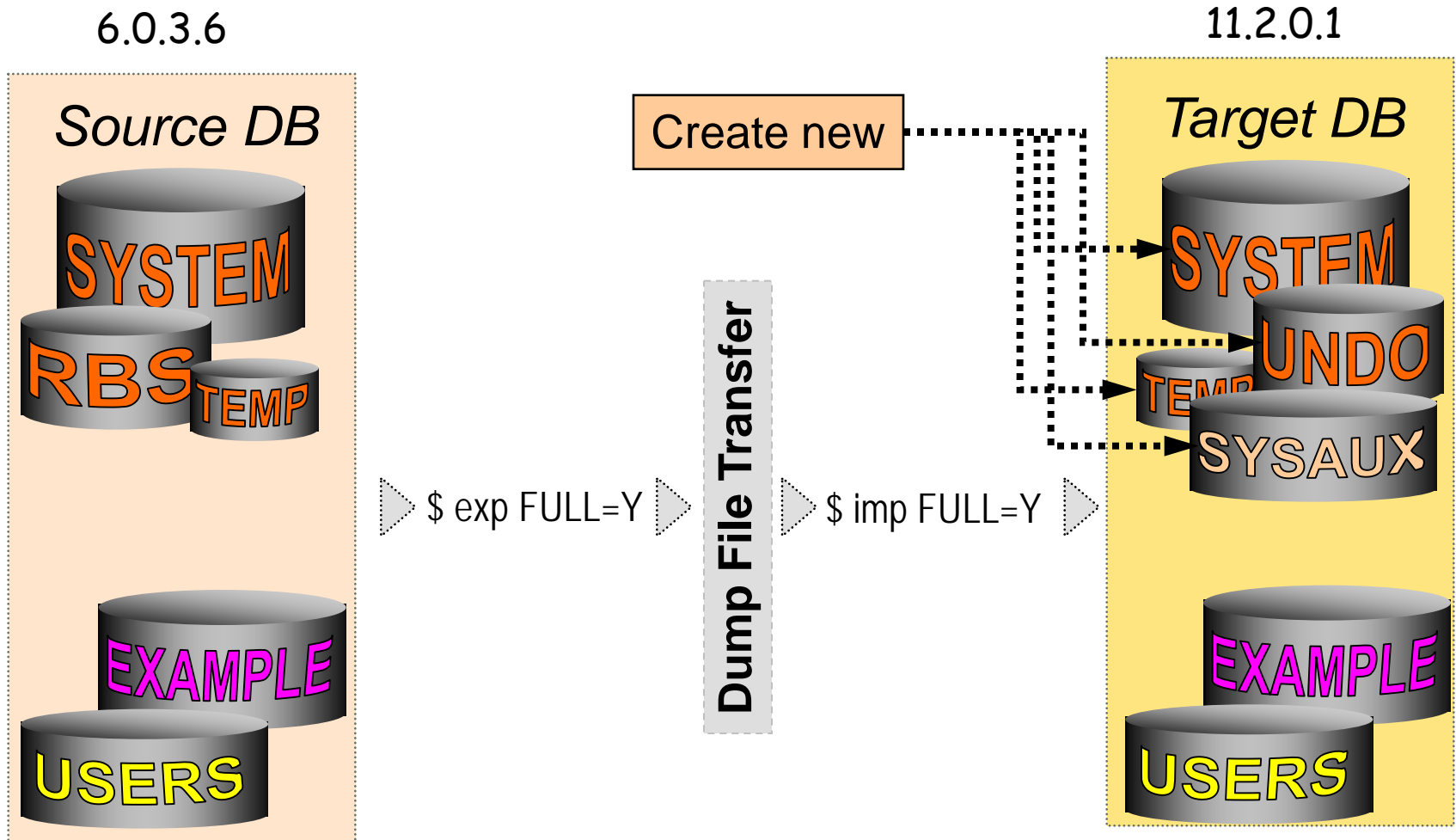
Upgrade Alternatives

- Platform Migration methods
- Minimal downtime methods
 - What does "minimal downtime" really mean?
 - 12 hours?
 - 60 minutes?
 - 5 minutes?
 - Less?
 - No downtime at all?

Upgrade Alternatives



Export - Import



Export - Import

- All purpose
- Import of all versions \geq Oracle V5 possible
- "exp" is *not supported* in 11g anymore
 - But the utility is still there and can be used
 - "imp" is still supported for importing older dumpfiles
- Not really fast but well known and reliable
 - Relation between amount of data and runtime
- Necessary and helpful :
 - Changing the database charactersets
 - Use Scanner Utility **CSscan** before altering the DB Character set
Note: 123670.1
 - Changing the OS platform
 - Schema consolidation
 - Non-direct-upgrade supported releases such as 8.0.3

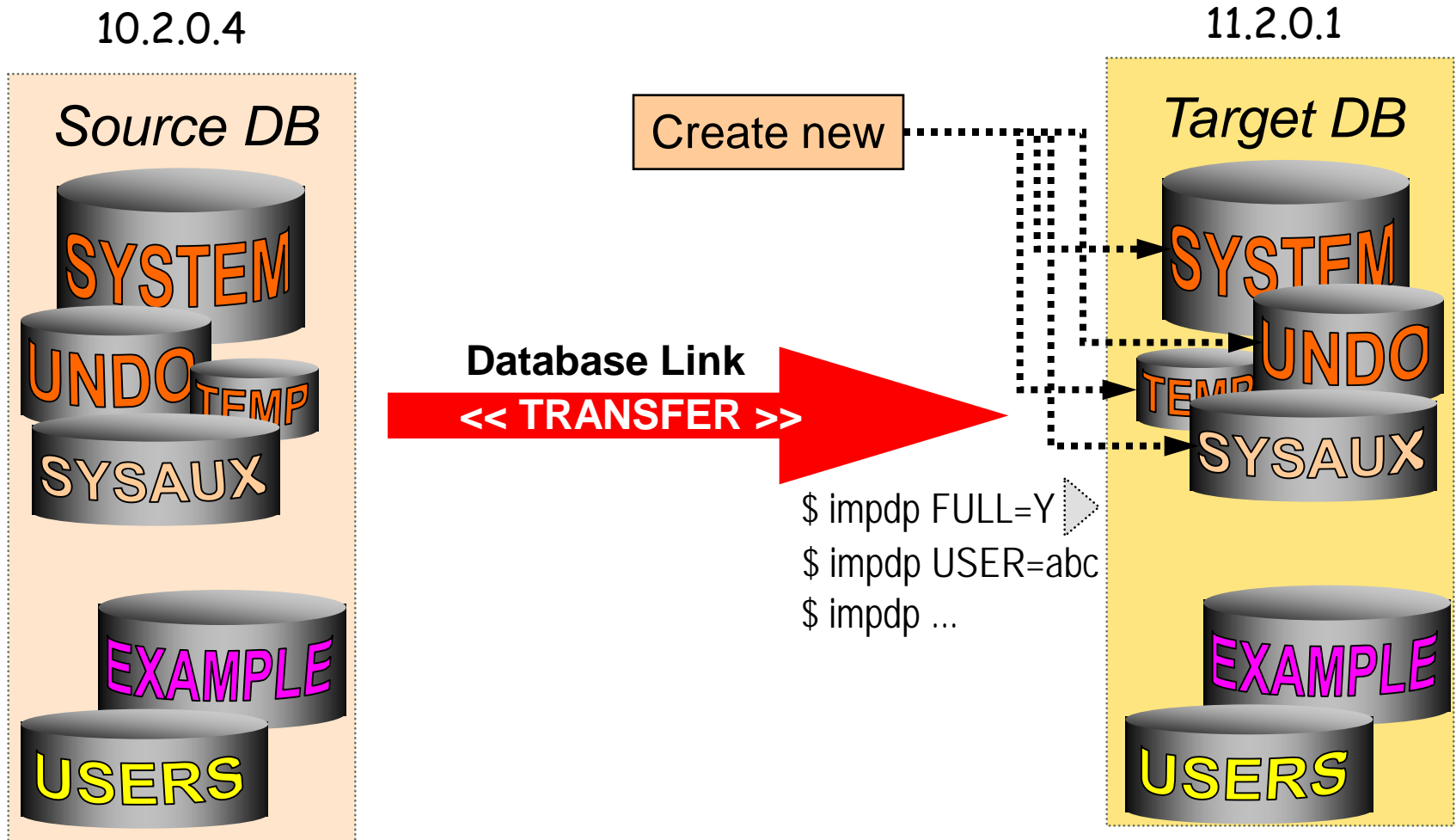
Export - Import

- Hints and tricks
 - Transfer dump files always in **BINARY** mode
 - Do full database export always as user **SYSTEM**
 - GRANTs on SYS's objects have to be exported separately
 - Import takes approximately 3x times as long as Export
 - Export always with the lowest involved database version
 - Import always with **imp** of target database
 - See also: [Note:286775.1](#)
 - Export performance
 - **DIRECT=Y** ... bypasses SQL-Layer, but no conversions!
 - Parallelize export by dividing into logical independent chunks of data
 - Import performance
 - Increase **BUFFER**
 - **INDEXES=N** ... build indexes later in parallel ... **INDEXFILE=...**
 - Parameter **COMMIT_WRITE=NOWAIT (10g)** or **COMMIT_WAIT=NOWAIT (11g)** during import

Data Pump

- The "new" `exp/imp` - since Oracle Database 10g
 - Faster than `exp/imp`
 - Powerful concept and more capabilities than `exp/imp`
 - EXCLUDE
 - COMPRESS=ALL
 - SQL with WHERE clause
 - Compatibility and version changes: [Note:553337.1](#)

Data Pump with NETWORK_LINK

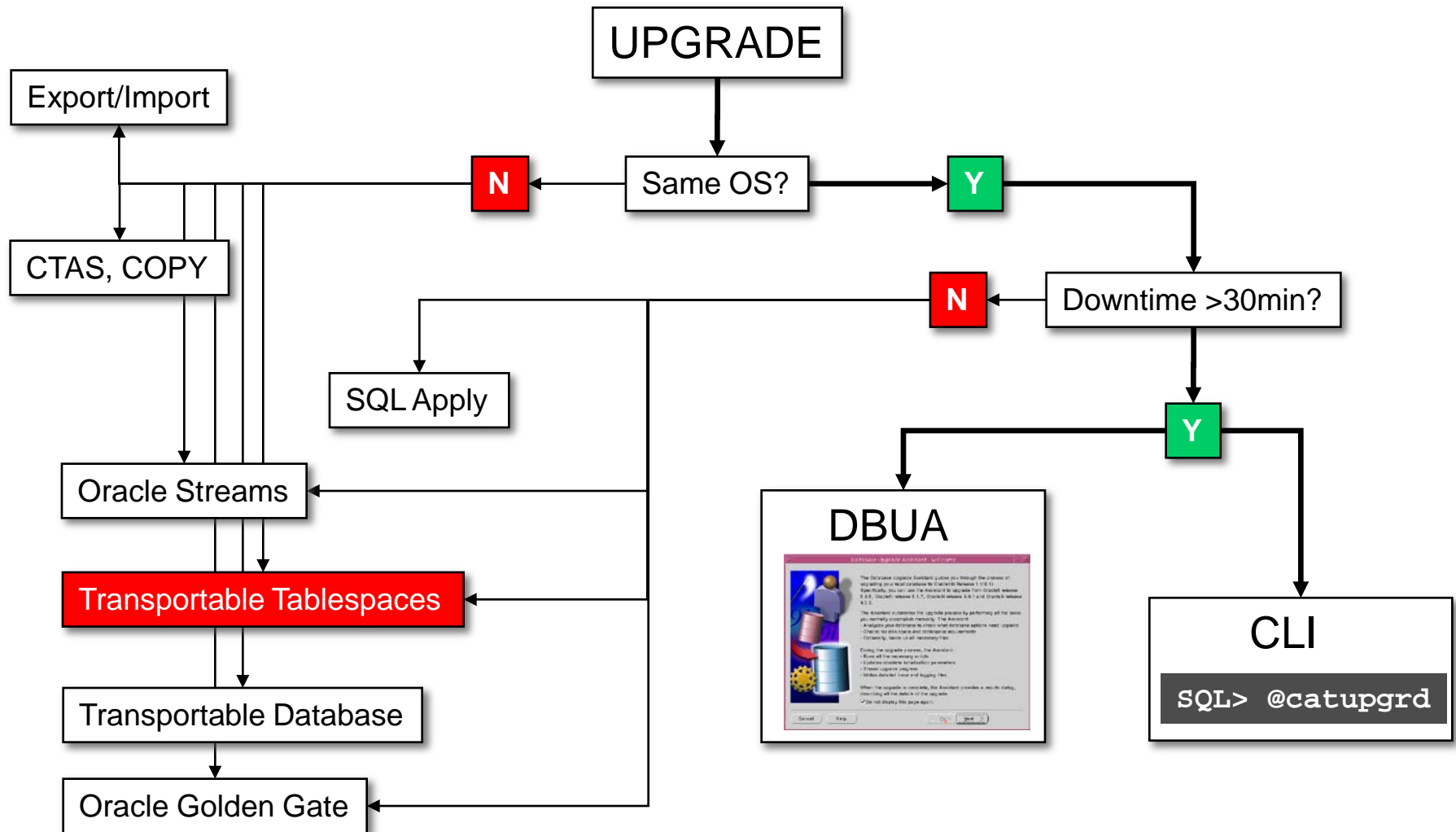


Data Pump

- Data Pump cross database link
 - Parameter: NETWORK_LINK
 - Run `impdp` on the target system
 - No `expdp` necessary
 - Does not work with LONG/LONG RAW and object types
 - No disk-IO and no dump file transfer will be performed
 - Limitation: network bandwidth
 - Example:

```
$ impdp system/pw NETWORK_LINK=mydblink FULL=Y
```

Upgrade Alternatives

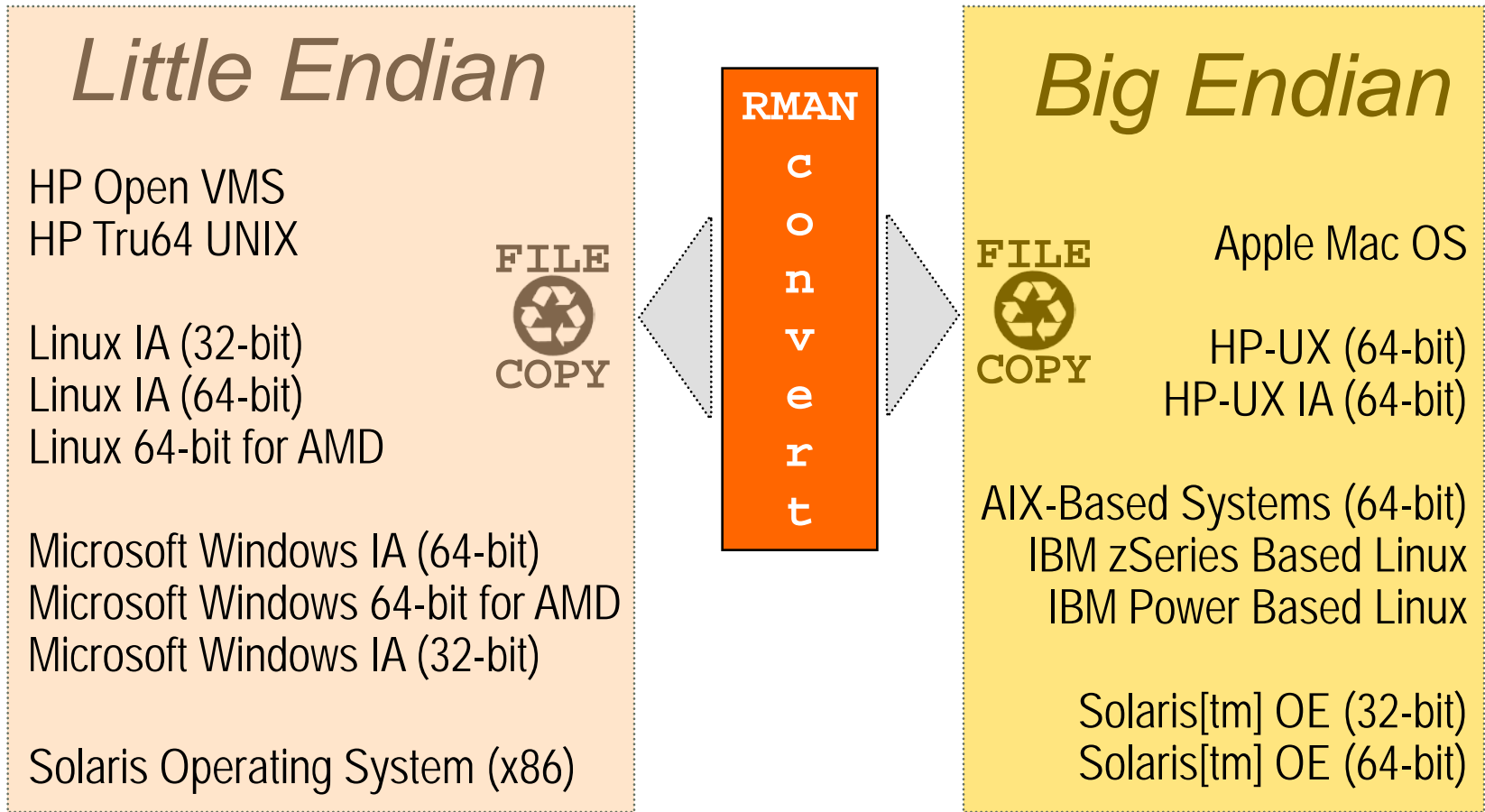


Transportable Tablespaces

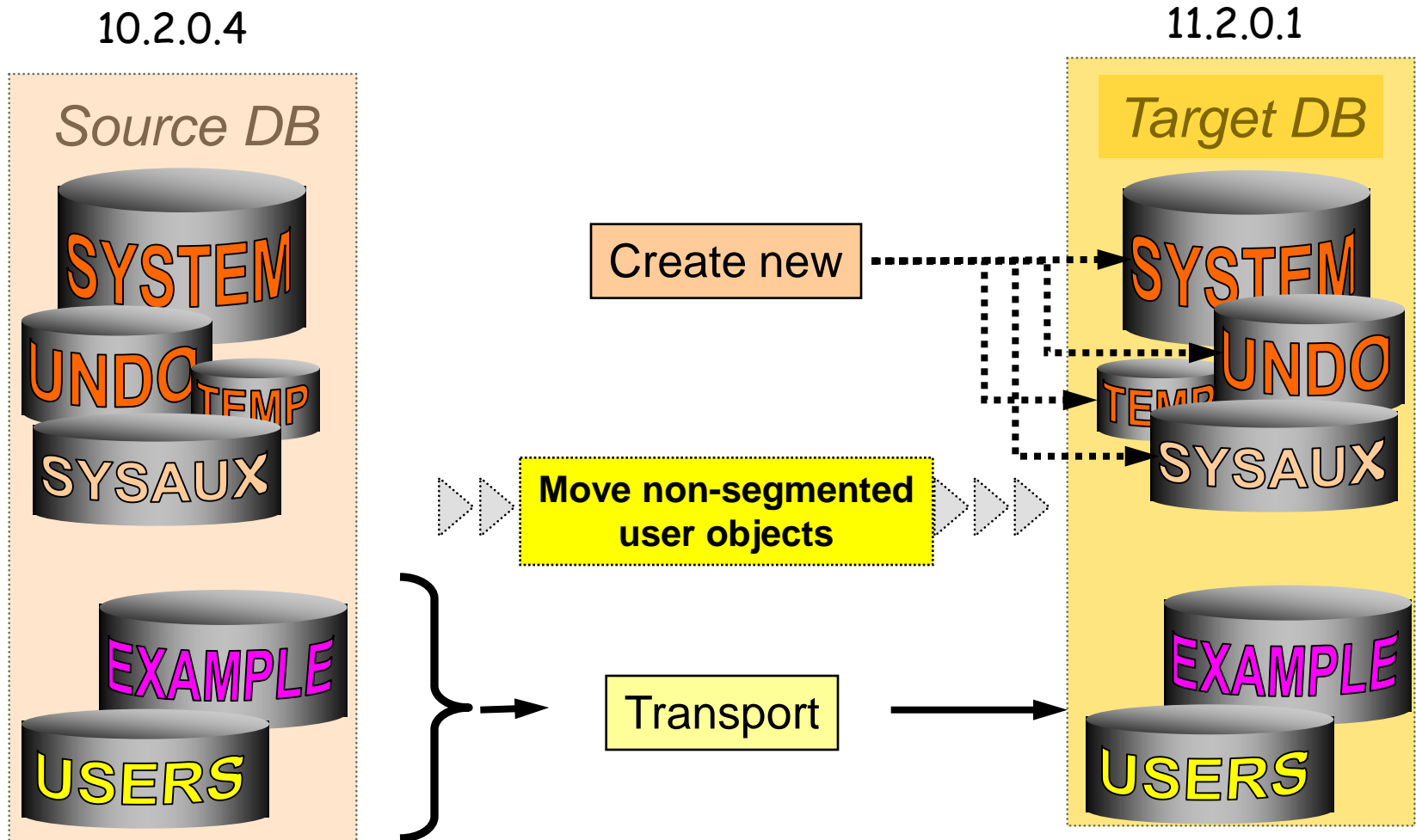
- Concept:
 - Create an "empty" database in the new environment
 - Plug in all data tablespaces from source to target database
 - SYSTEM+SYSAUX tablespaces can't be transported
 - Additional steps necessary to move views, synonyms etc.
 - "Possibly" very fast upgrade
 - Complexity could be constraining
 - Works cross-platform and cross-Endianness since Oracle Database 10g

Transportable Tablespaces

- TTS x-platform (v\$transportable_platform):



Transportable Tablespaces



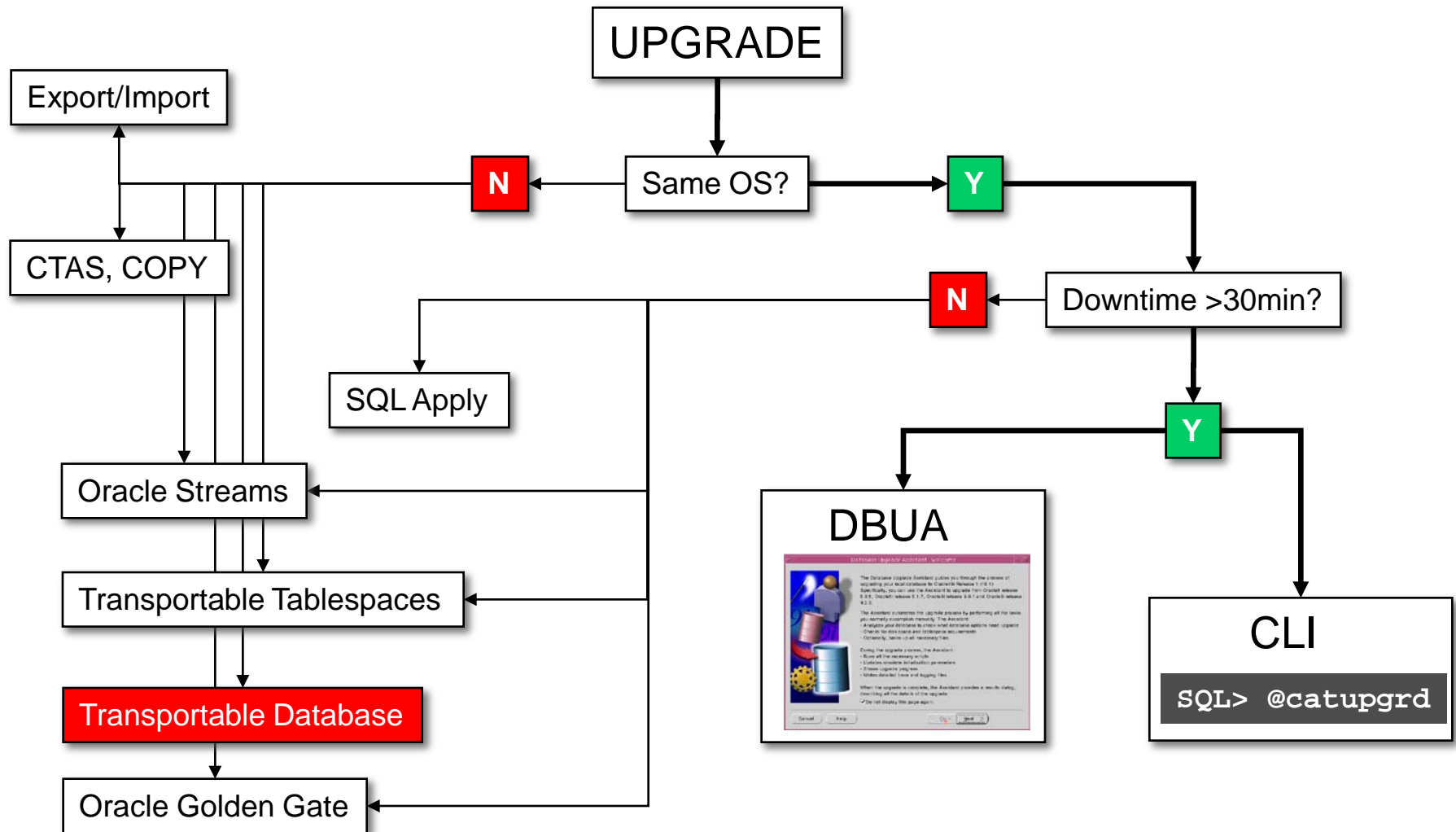
Upgrade Alternatives

- Transportable Tablespaces - 3 ways
 - The "brutal" way
 - Full exp/imp with ROWS=N
 - The "smart" way
 - Generate scripts
 - String concatenation with || ...
 - DBMS_METADATA
 - The "very smart" way
 - RMAN clone with SKIP TABLESPACES option
- BUT: Take care especially of sequences!!

Transportable Tablespaces

- Tips & Tricks
 - Talk as early as possible to the application development if TTS will be your upgrade strategy
 - Less complex design is the requirement for fast TTS
 - Use a Physical Standby as transport system
 - Fallback possibility to the old system
 - If you don't move datafiles:
 - Tablespaces can be mounted from both databases simultaneously as long as they are READ ONLY
 - As soon as a tablespace will be set READ WRITE on one database it'll be "lost" for the other one

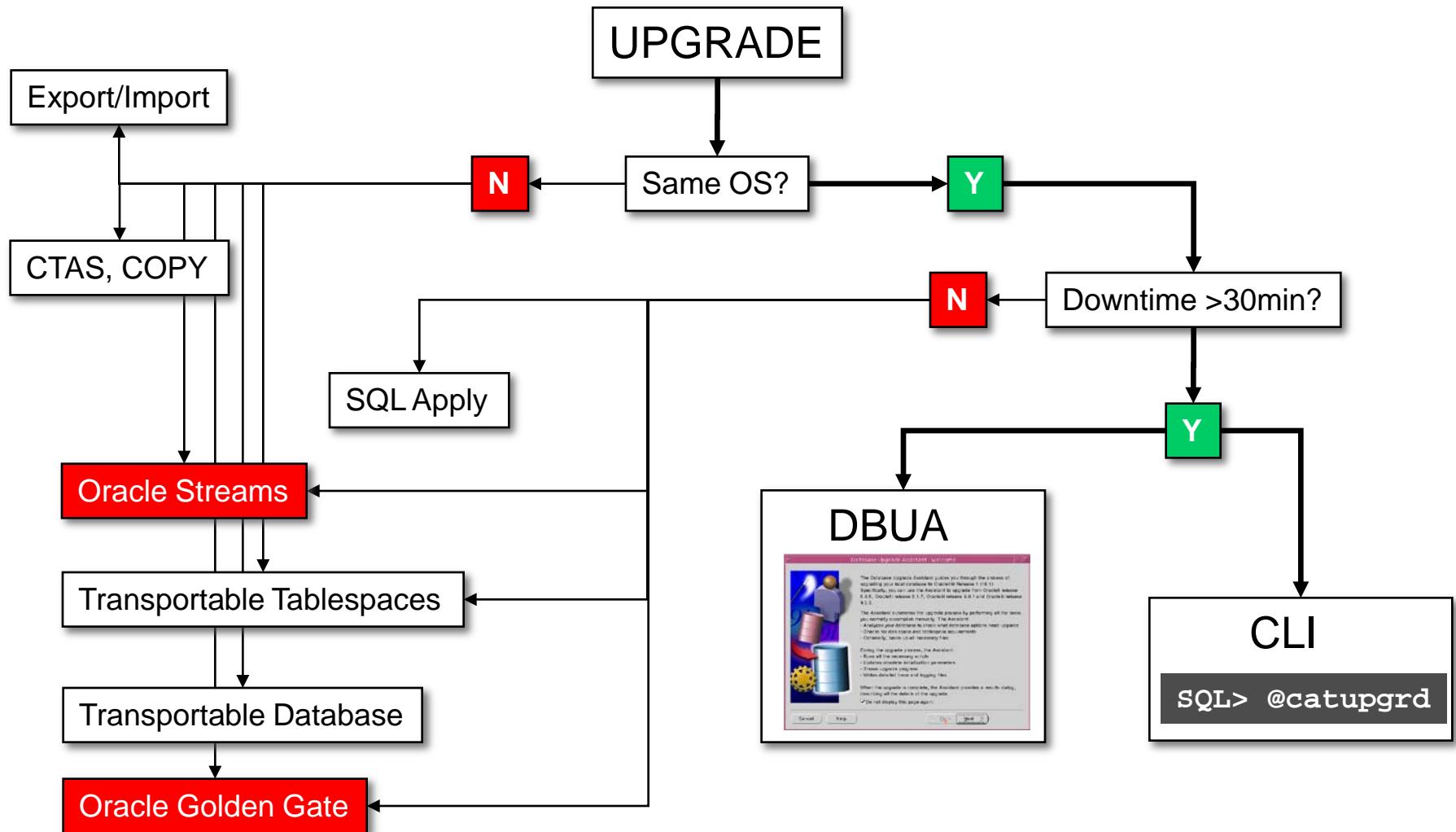
Upgrade Alternatives



Transportable Database

- Feature since Oracle Database 10g Release 2
 - Cross-platform
 - Unfortunately not cross-Endianness!!!
 - With RMAN in an automated way
 - Database must be switched to READ ONLY mode
 - Datafiles must be converted with RMAN into target format
 - RMAN CONVERT DATABASE command
 - Either on the source or the target system – in most cases completes faster on the target system
 - Not a real minimal downtime concept
 - But very comfortable for migrations within one Endianness group

Upgrade Alternatives



Oracle Streams

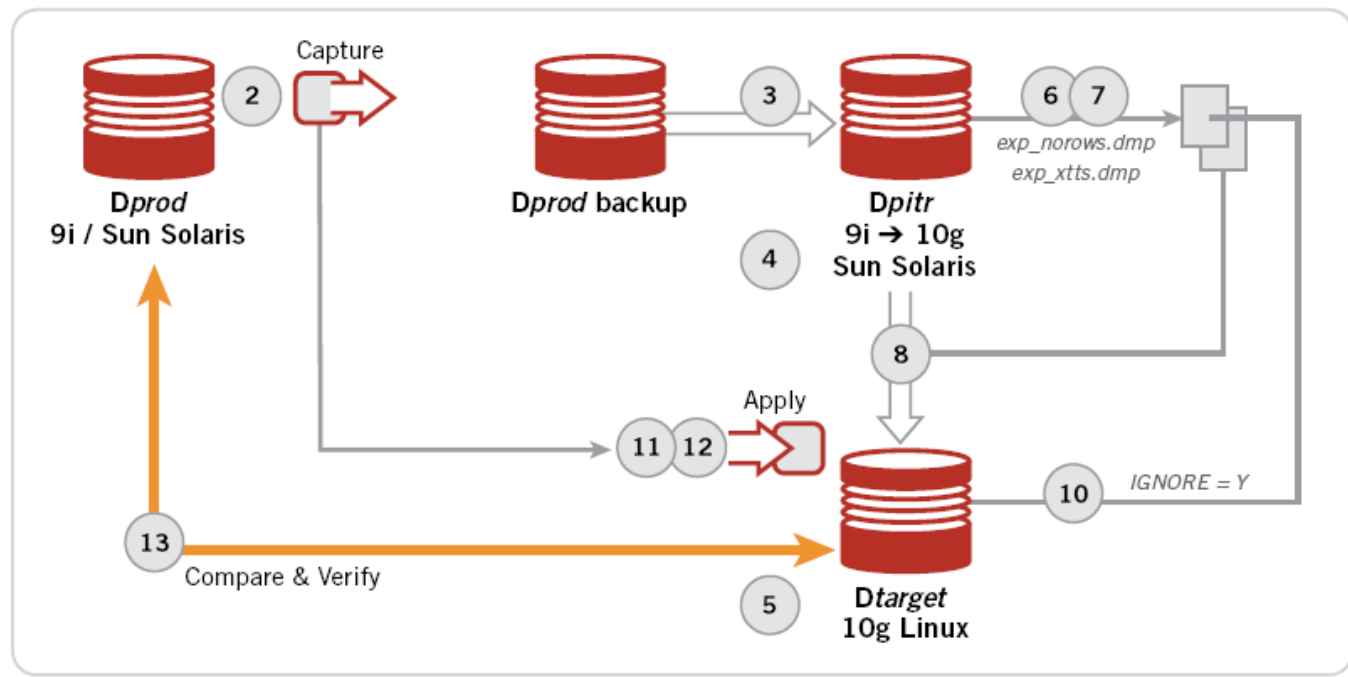
- Concept
 - Build up a copy of your database and upgrade it
 - Synchronize it with the source database
 - Downtime:
 - Just reconnecting the clients
 - Cross platform
 - Cross version since Oracle 9iR2
 - Some effort necessary to set it up
 - Fallback possible
 - Logminer
 - Datatype restrictions
 - Performance
- How to:

Oracle® Streams Concepts and Administration: Appendix D

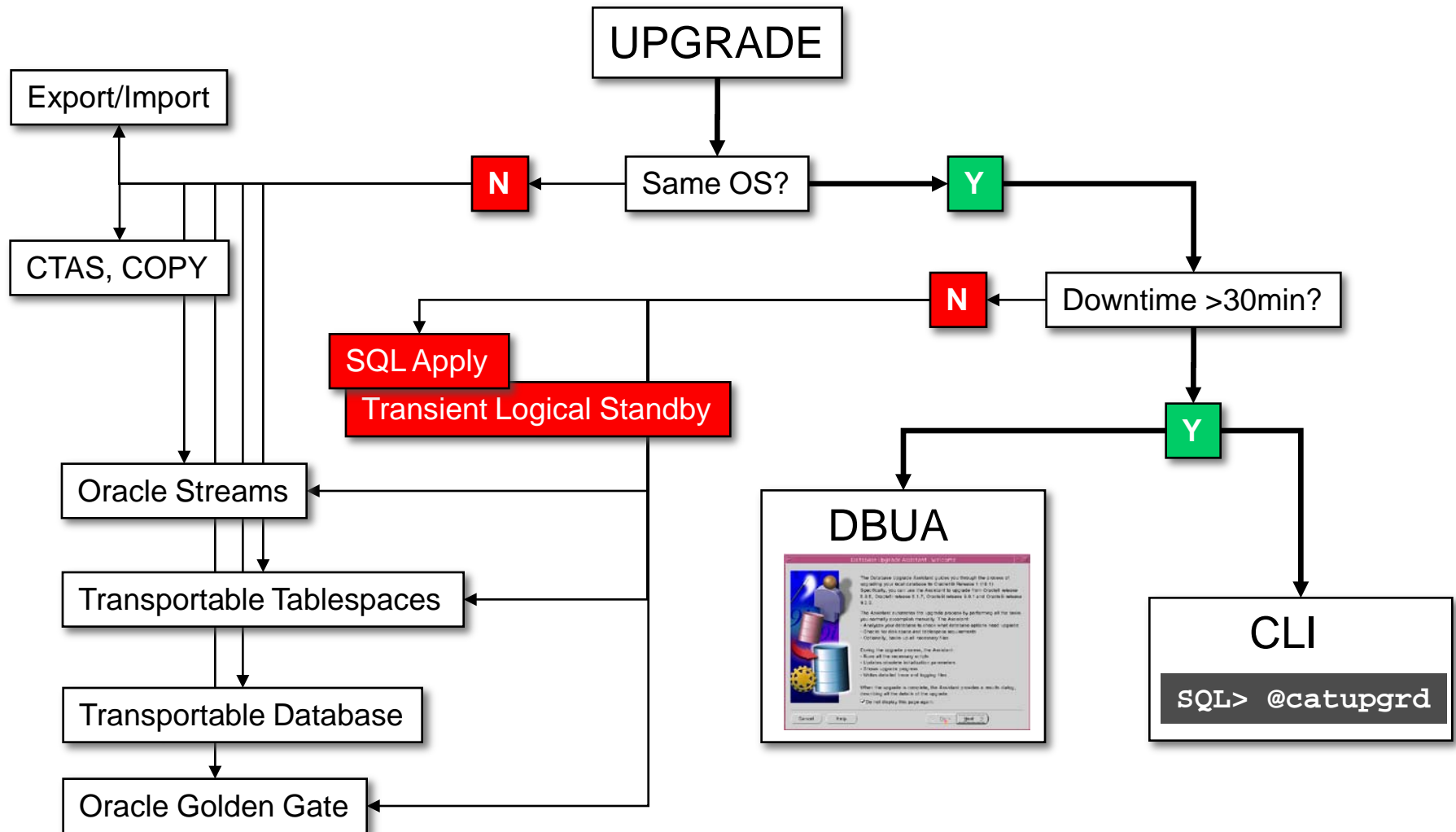
http://download.oracle.com/docs/cd/E11882_01/server.112/e10704/ap_strmnt.htm#CIHJBIAA

Oracle GoldenGate

- Concept
 - Create a copy of your database with Transportable Tablespaces
 - GoldenGate CDG mechanism for synchronization
 - <http://www.goldengate.com/>



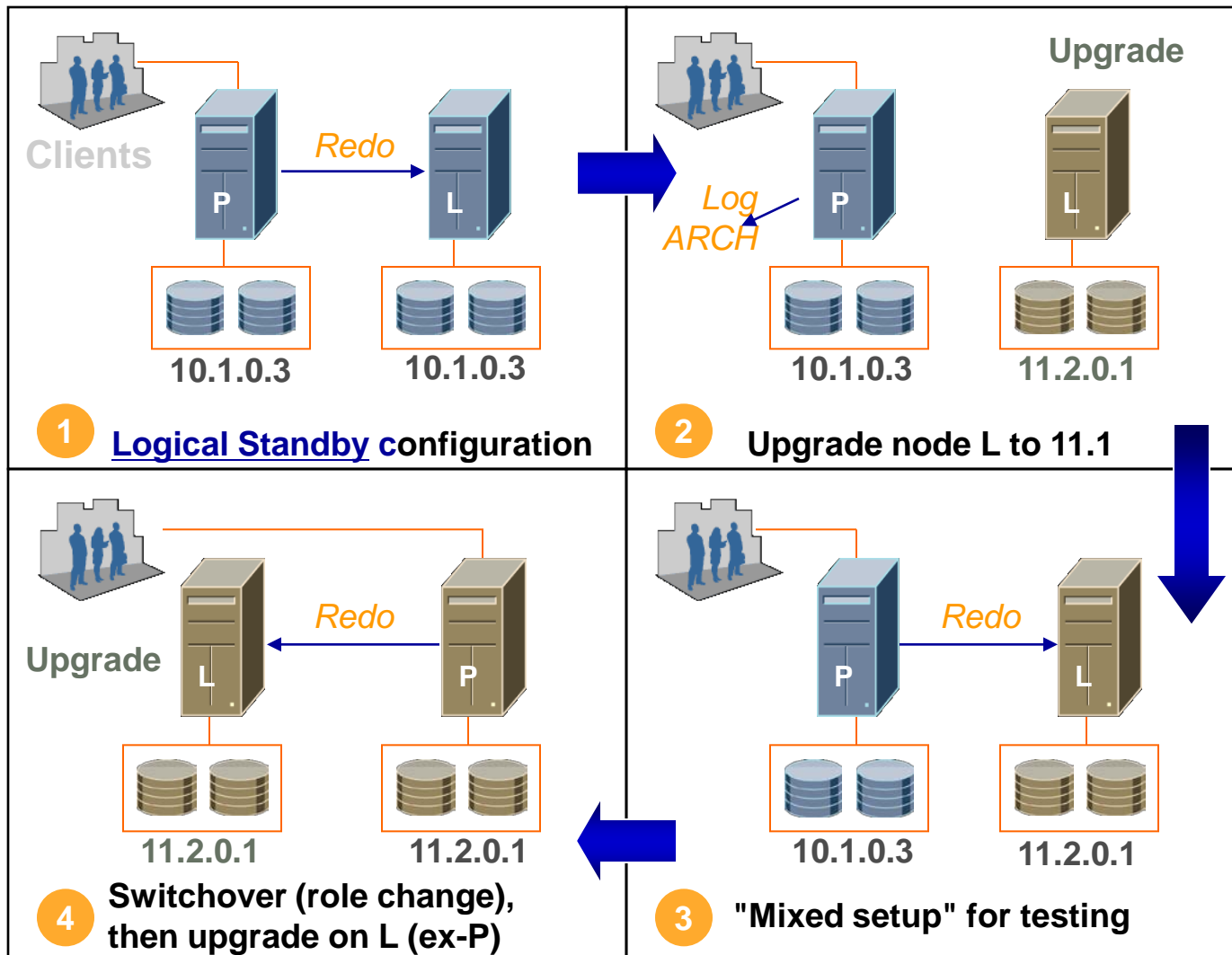
Upgrade Alternatives



Logical Standby with Oracle Data Guard

- Concept:
 - Build up a Physical Standby database
 - Convert the Physical Standby into a Logical Standby
 - Upgrade the Logical Standby database
 - Switchover – Standby will be production system now
 - Then: Upgrade of the former production database
 - Eventually: Switchover to the original roles
 - Downtime less 2 minutes
 - BUT:
 - No OS change possible
 - Logminer has known restrictions

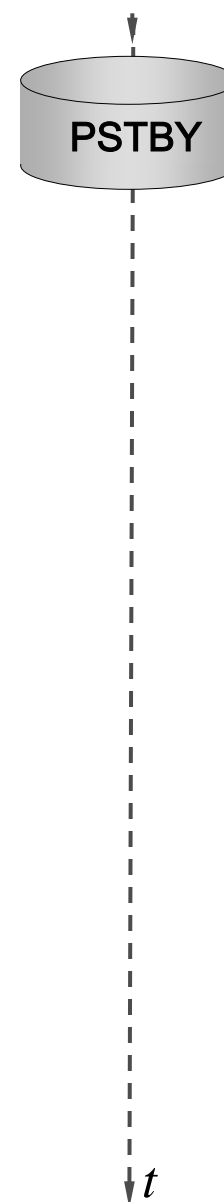
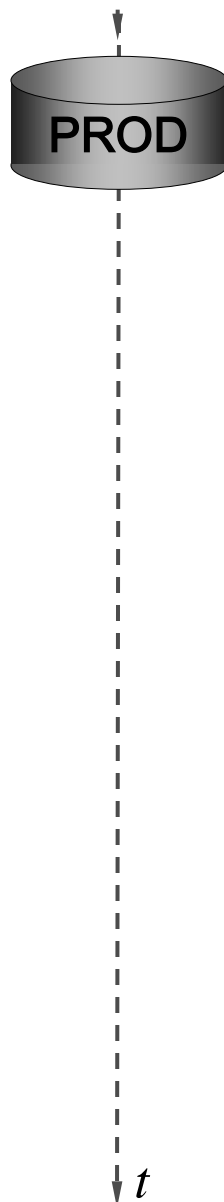
Logical Standby with Oracle Data Guard



Transient Logical Standby

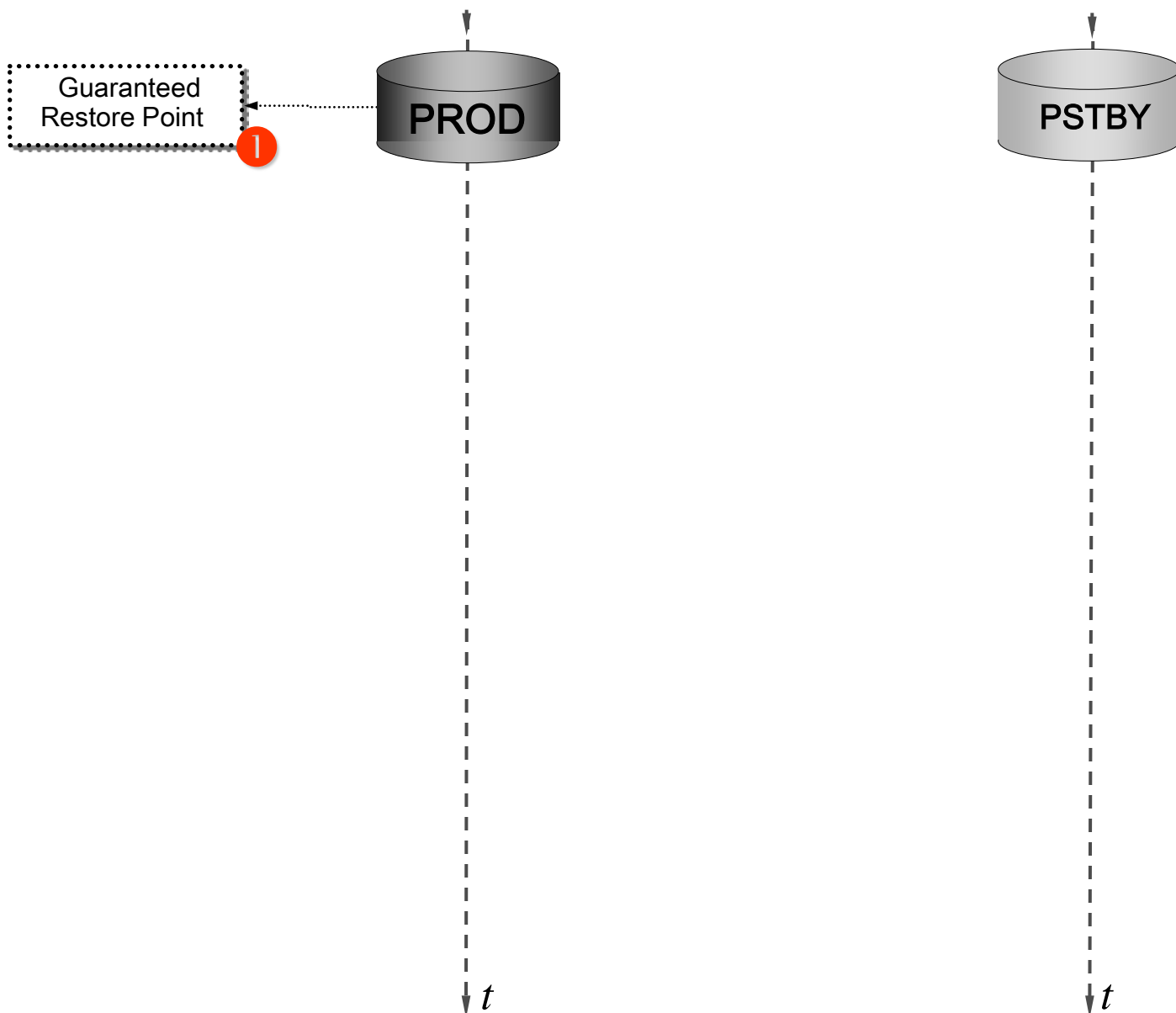
- Concept:
 - Build up a Physical Standby database
 - Convert the Physical Standby into a Logical Standby
 - Upgrade the Logical Standby database
 - Switchover – Standby will be production system now
 - Then: Flashback the former production database
 - Convert it into a Physical Standby
 - Upgrade just by log apply
 - Eventually: Switchover to the original setup
 - Works pretty straight forward with Oracle Database 11g
- Find shell scripts in [Note:949322.1](#)

1

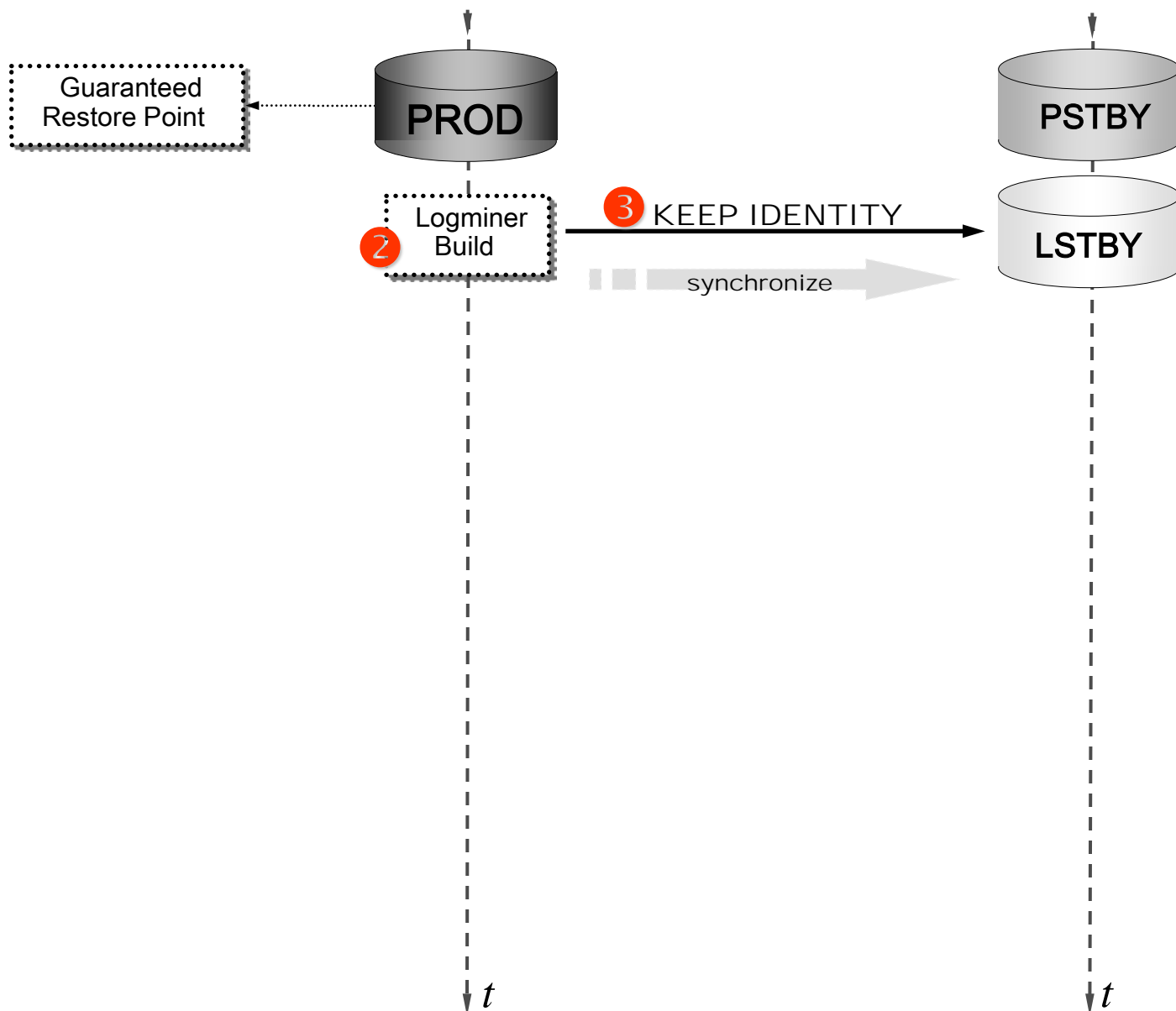


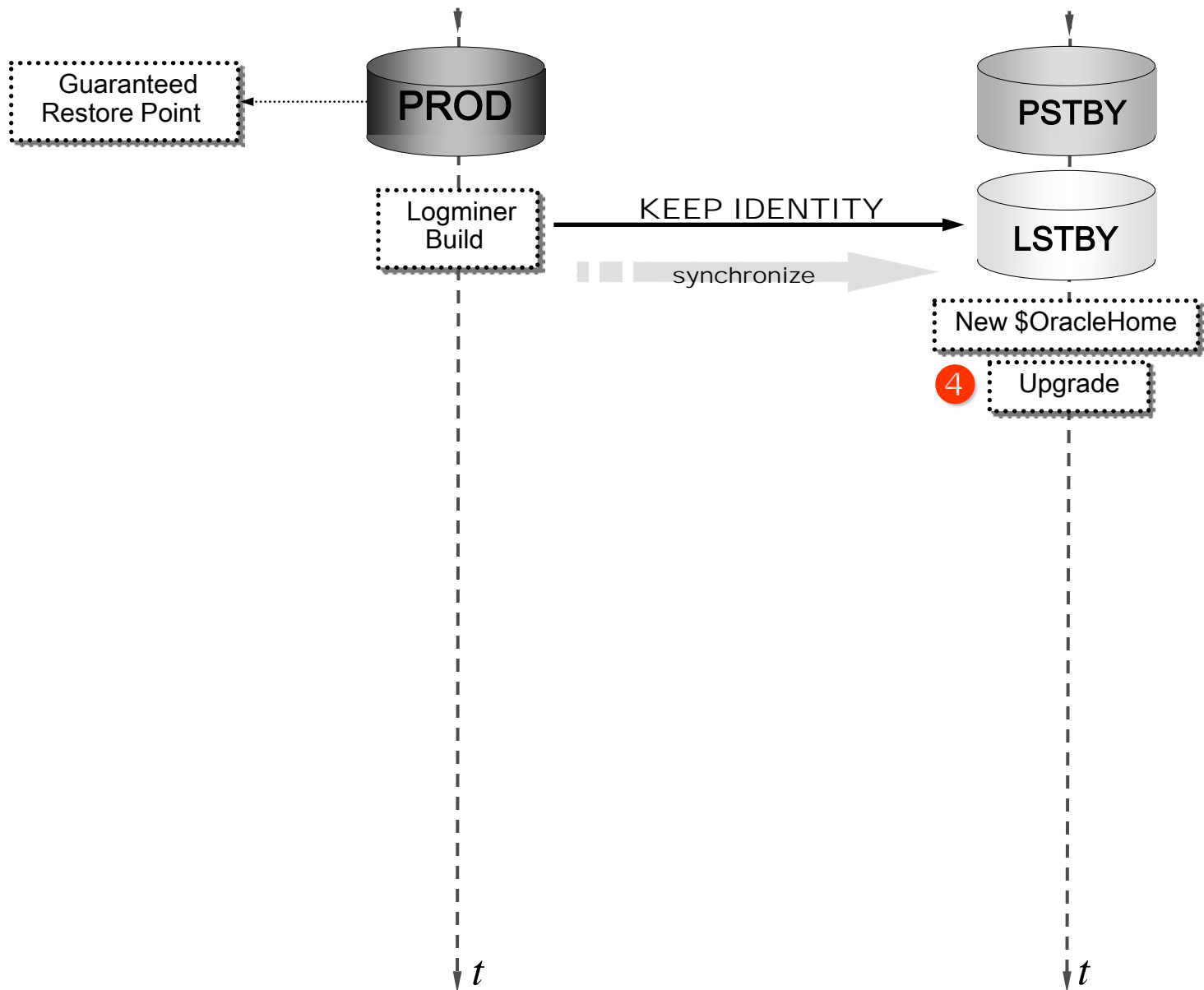
ORACLE®

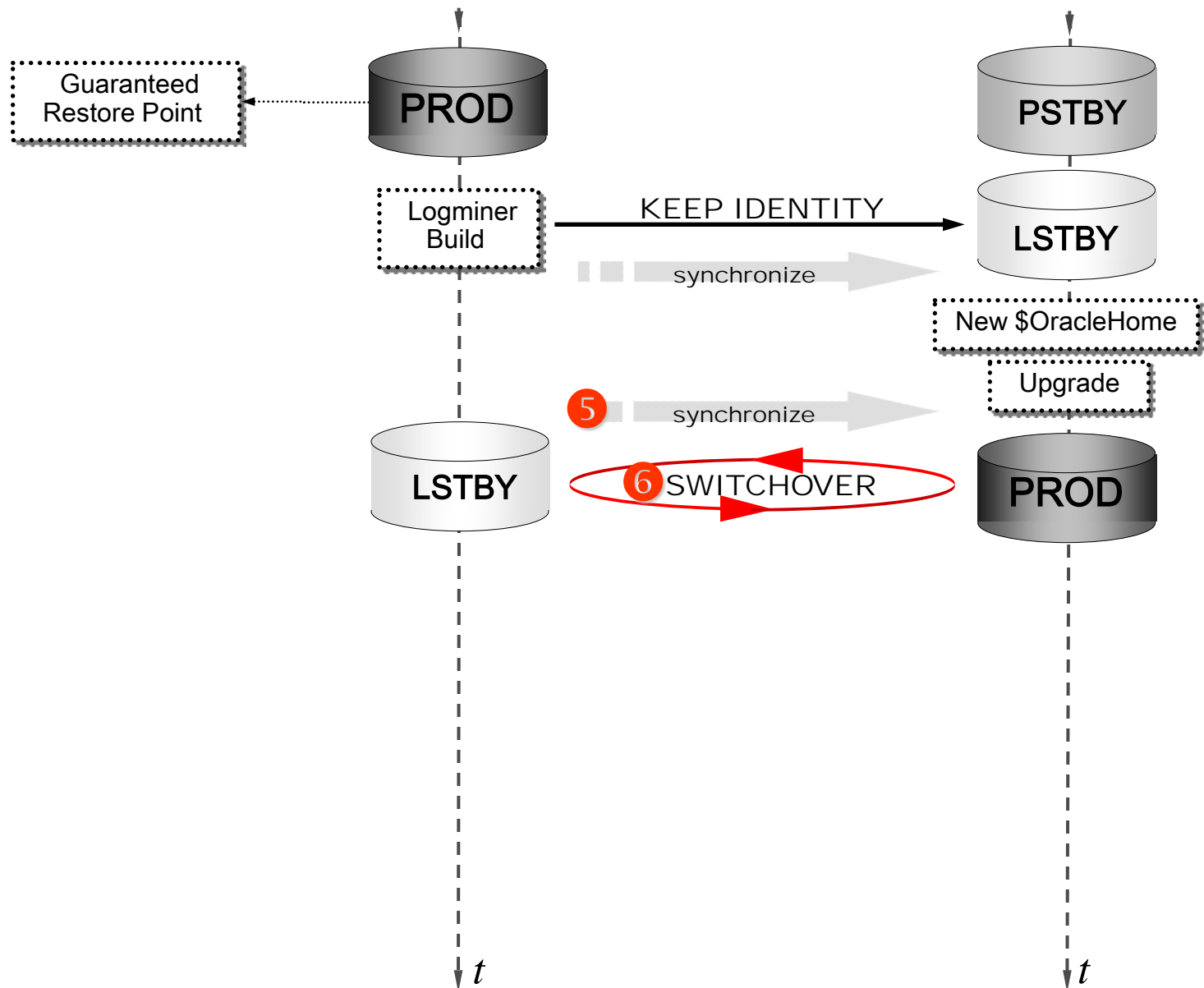
2

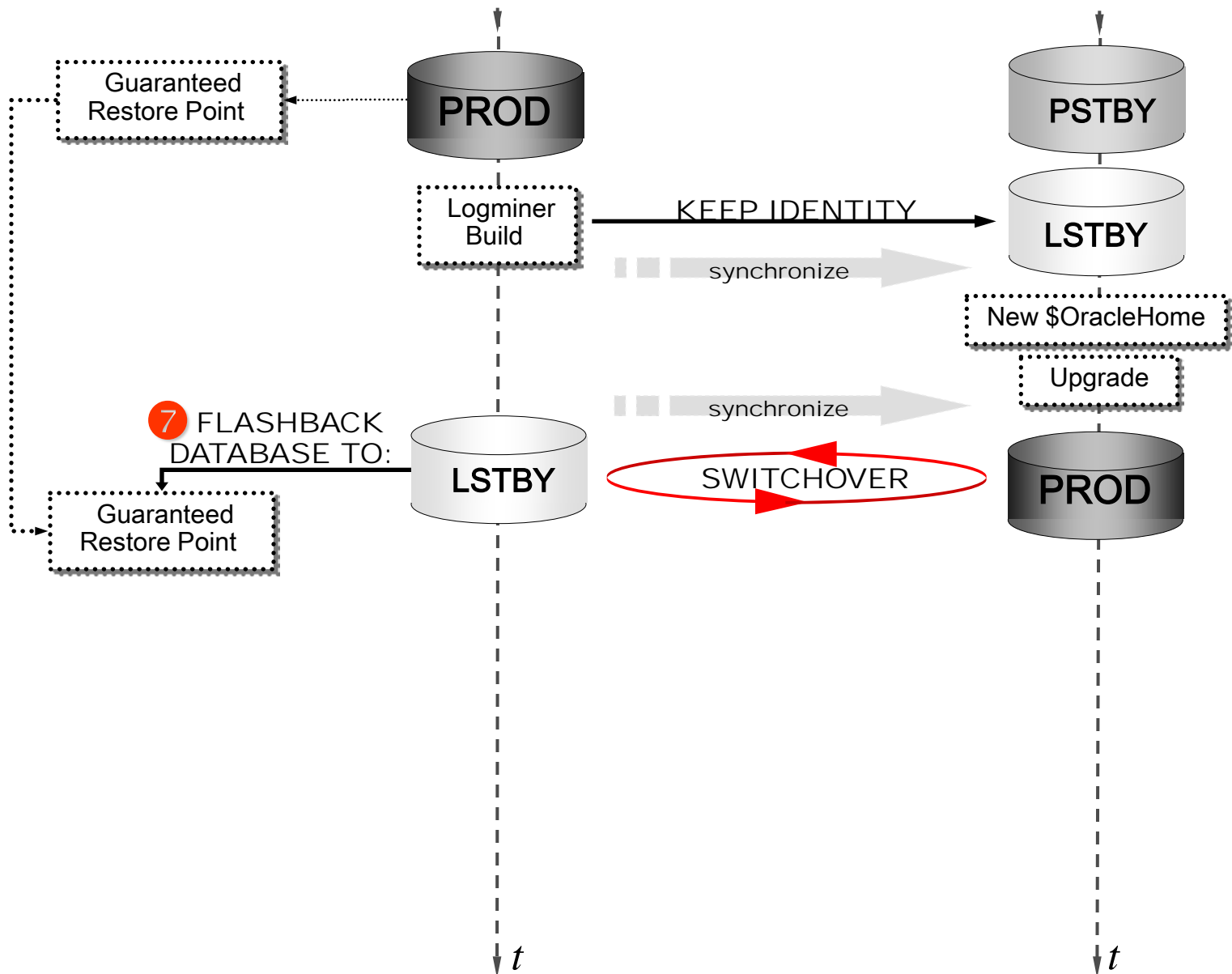


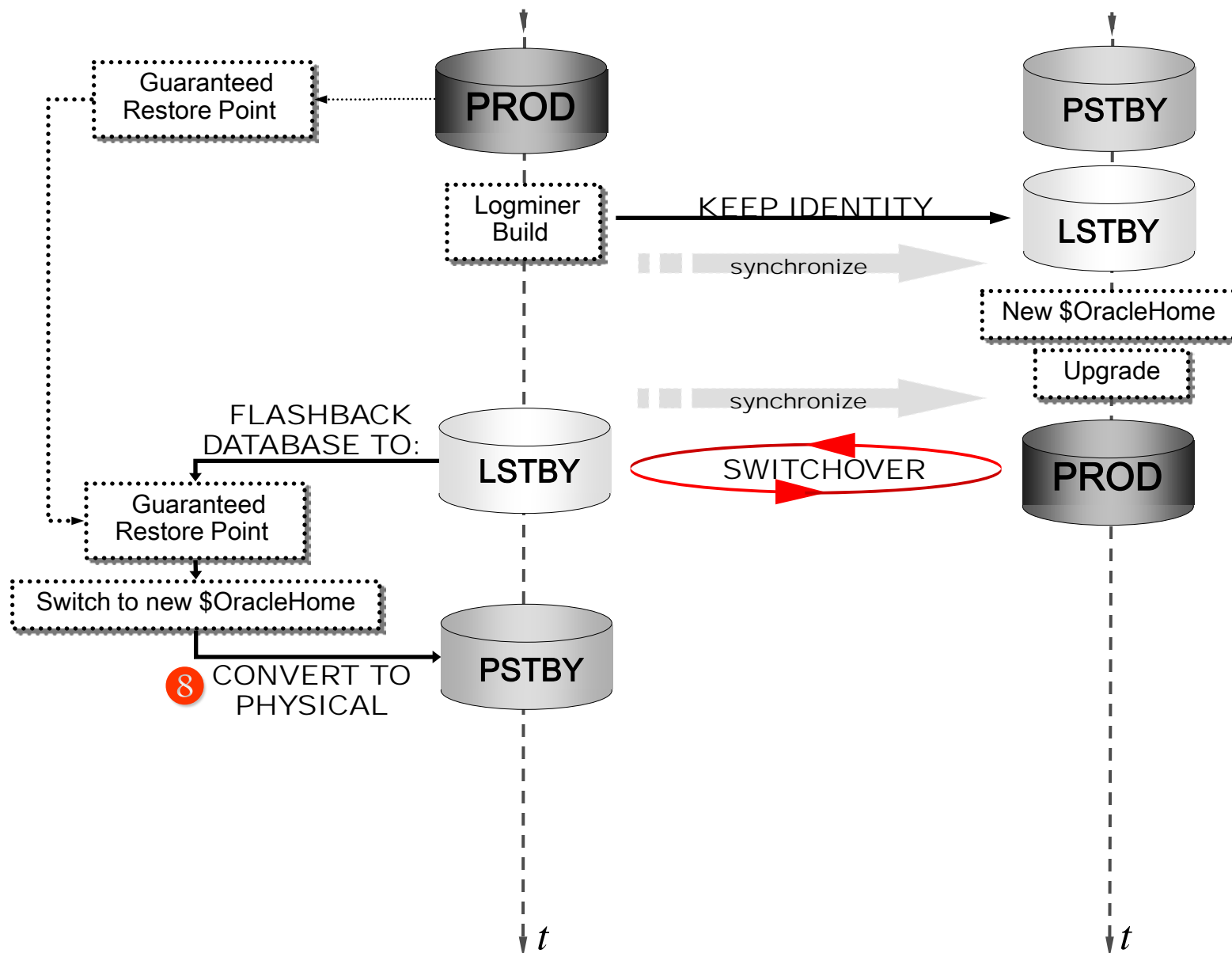
ORACLE®

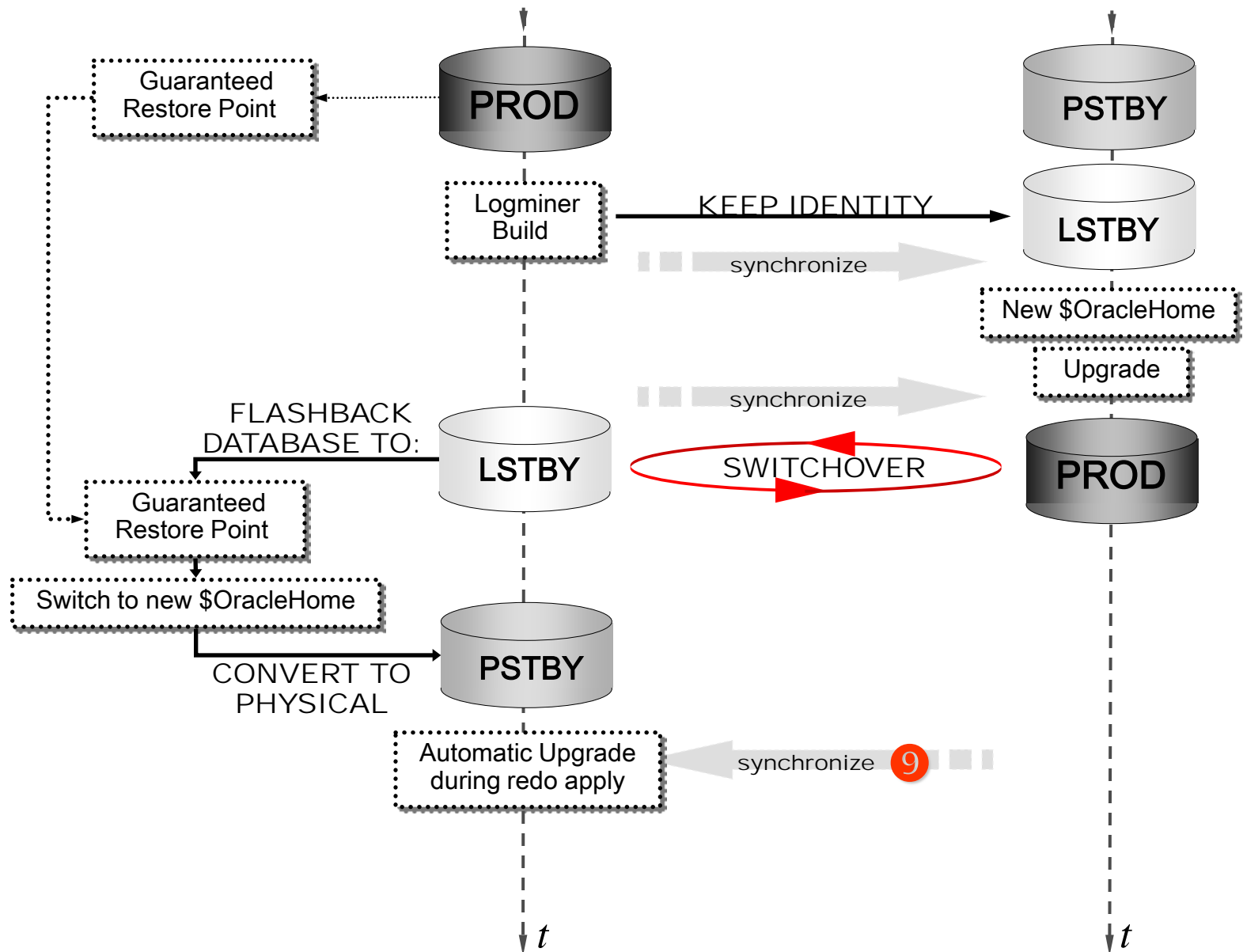


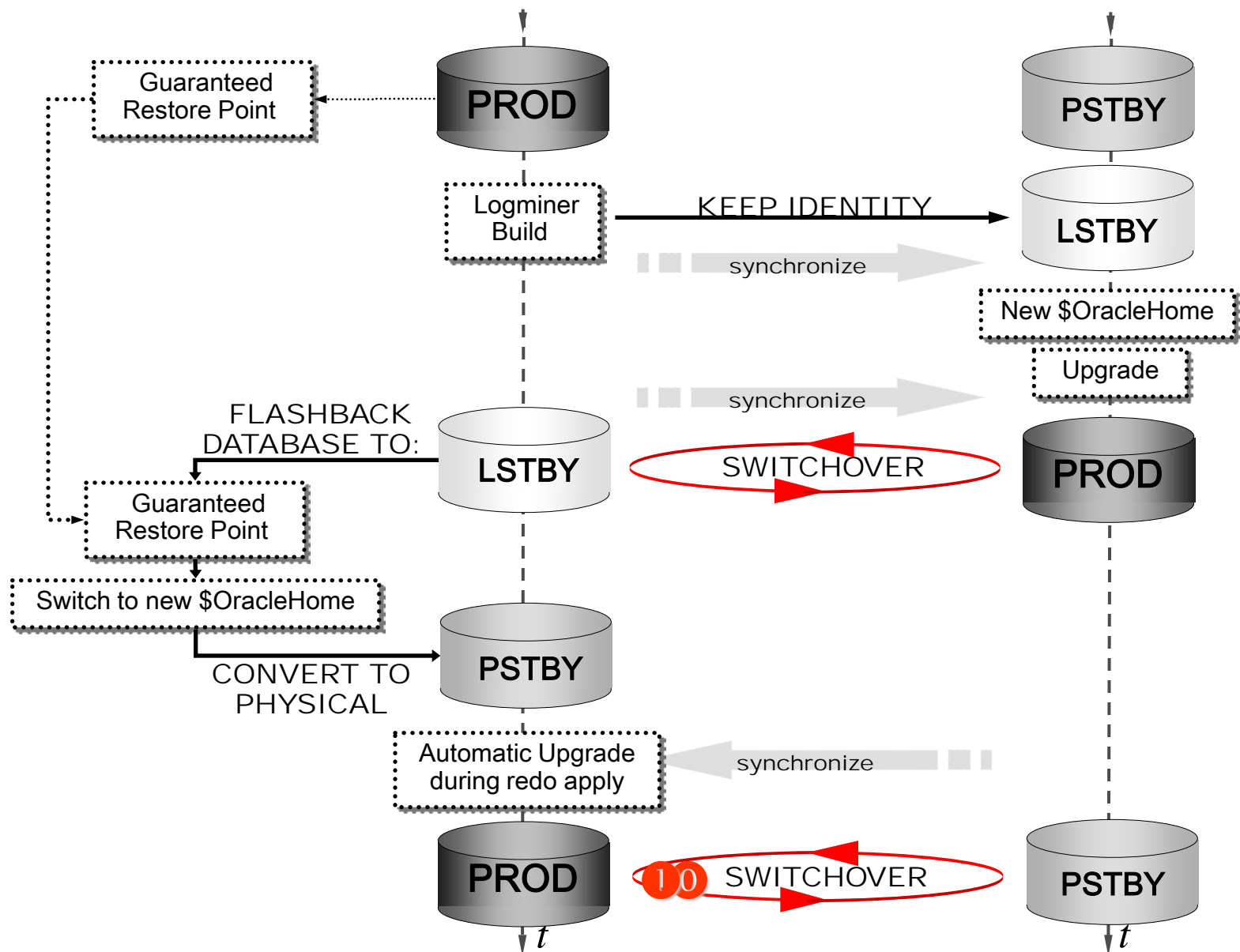






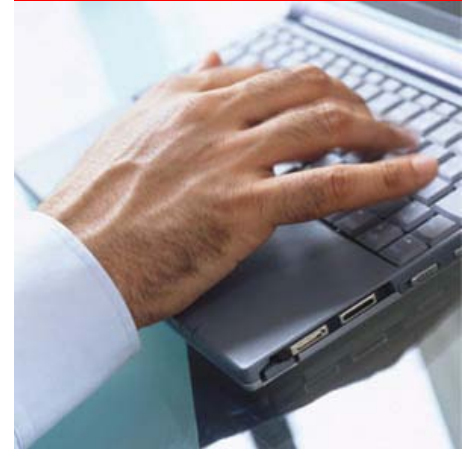






Agenda

- Regular Upgrade Methods
- Post Upgrade Tasks
- Upgrade Alternatives
- Summary



When to Choose the DBUA




- Can afford 30 – 90 minutes average downtime
- Operating system remains the same
- GUI is preferred over manual command line interface
 - Automatically performs useful pre-upgrade checks
 - Less error-prone / less manual effort
- Existing database is at least 9.2.0.8
- Note: especially useful for RAC databases
- Consideration:
 - Source and target Oracle Homes must be on the same system
 - Cannot be re-run if an error is encountered mid-upgrade

When to Choose Command-Line

```
SQL> spool upgrade.log  
SQL> @catupgrd.sql
```

- Can afford 30-90 minutes average downtime
- Manual command-line interface is preferred over GUI
- Existing database is at least 9.2.0.8
- Migrating to a new hardware platform with same OS
- Consideration
 - Cannot migrate to a different operating system architecture
 - More manual steps required
 - Potential for errors due to typos, missed details

When to Choose an Alternative Method

- Alternative methods include
 - Original exp/imp or Data Pump expdp/impdp)
 - Oracle Streams or Oracle Golden Gate
 - Data Guard (SQL Apply)
 - Transportable Tablespaces, Transportable Database
 - Moving data via CREATE TABLE AS SELECT or other techniques
 - Alternative methods **must** be used when
 - Moving to a different operating system platform (32- and 64-bit versions of an OS are considered “the same platform” in this case
 - Upgrading from a release older than 9.2.0.8
 - Alternative methods **may** be a good option when
 - Minimal downtime (<30 minutes) required or desired
 - Re-organizing database storage or schemas
- 
- The diagram shows a central box labeled 'DIBUA' with a red 'WARNING' banner. It is connected to several other components: 'CTAB COPY' at the top left, 'SQL Apply' at the top, 'Oracle Streams' and 'Transportable Tablespaces' in the middle left, 'Oracle Golden Gate' at the bottom left, 'Data Guard' at the top right, and 'DirectWrite VTL' at the middle right. Arrows indicate the flow of data or integration between these components.



Upgrade Summary

- Choosing an upgrade method depends on:
 - Database environment
 - Amount of downtime that is acceptable
 - DBA's knowledge and tolerance for complexity
- If possible, using the DBUA is the recommended method for simplicity and ease-of-use
- Always create an online backup with RMAN
- Please remember:
Upgrade has never been easier - but you still have to test!!!
- 11g R2 is a stable database release so go for it!



Q&A

**Questions
Answers**

carol.palmer@oracle.com

ORACLE®