

Infrastructure at your Service.

Rolling Upgrade with Oracle 12c: A piece of cake!



About me

Hervé Schweitzer

CTO

Delivery Manager

Certified Oracle Master 11g

Mobile +41 79 693 43 67

herve.schweitzer@dbi-services.com

www.dbi-services.com



Who we are dbi services

Experts At Your Service

- > 35 specialists in IT infrastructure
- > Certified, experienced, passionate

Based In Switzerland

- > 100% self-financed Swiss company
- > More than CHF 4 mio. in sales

Leading In Infrastructure Services

- > Over 80 customers in CH, D, & F
- > 30 SLAs **dbi FlexService** signed



Agenda

1. Introduction
2. Rolling Upgrade concept
3. Upgrade from 12cR1 to 12cR1
4. Core messages

Introduction

Few words about replication



Involvement & Infrastructure
at your service.

Introduction

Logical replication concept

All DML (data changes) are converted to a row-by-row operation based on change vectors

Each row in a table must be uniquely identified

- > Primary key - unique key
- > Or all table columns are used

Replication is stopped if a conflict occurs

- > Conflict must be resolved manually (out of sync)

Can contain objects which don't exist in the source database

Only a subset of tables can be replicated

Introduction

Oracle Logical standby concept

On source database

- > LogMiner dictionary is built into the redo data

On target database

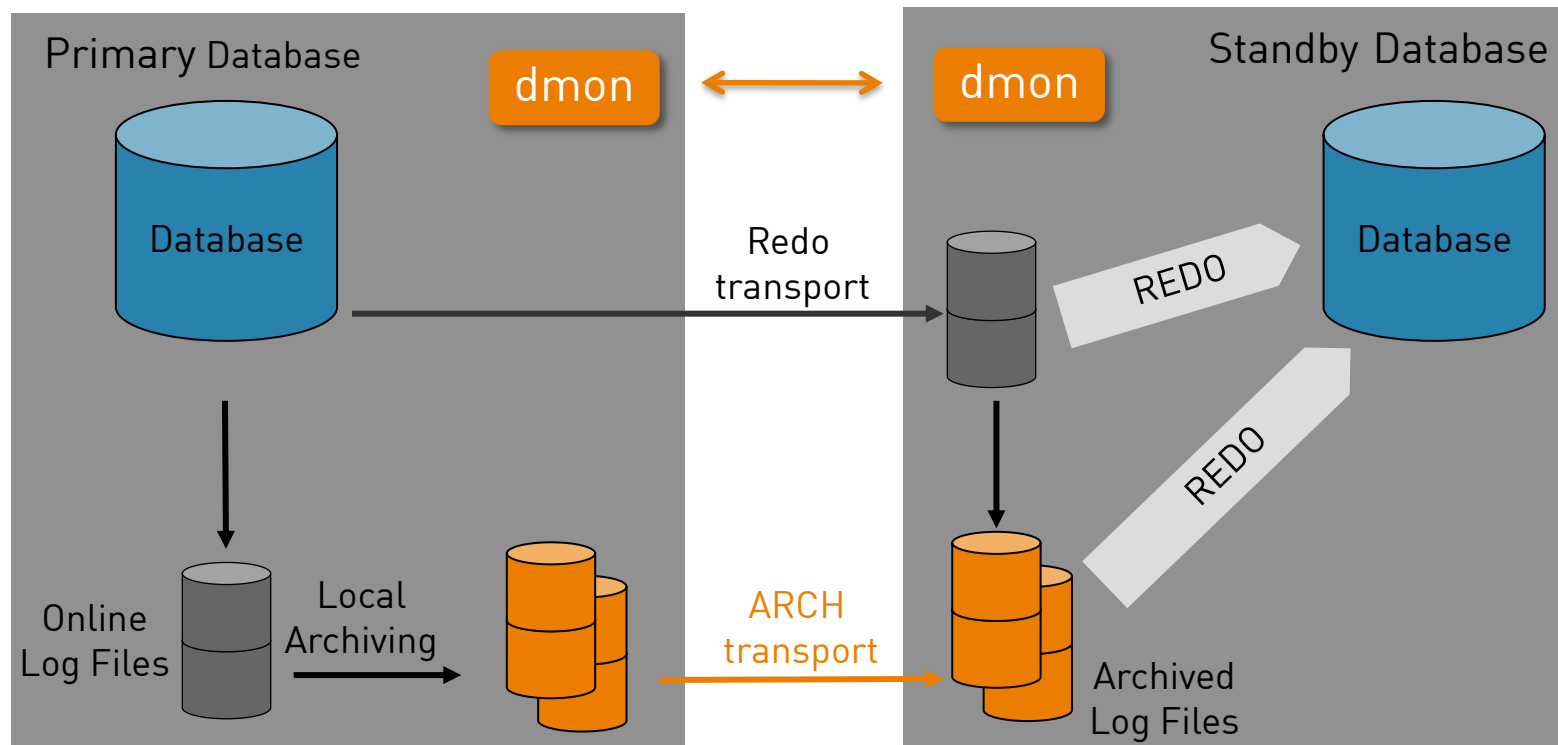
- > Oracle LogMiner re-construct the SQL statement based on the redo log contents

Only usable for Oracle-to-Oracle replication on identical platforms

Introduction

Physical standby database

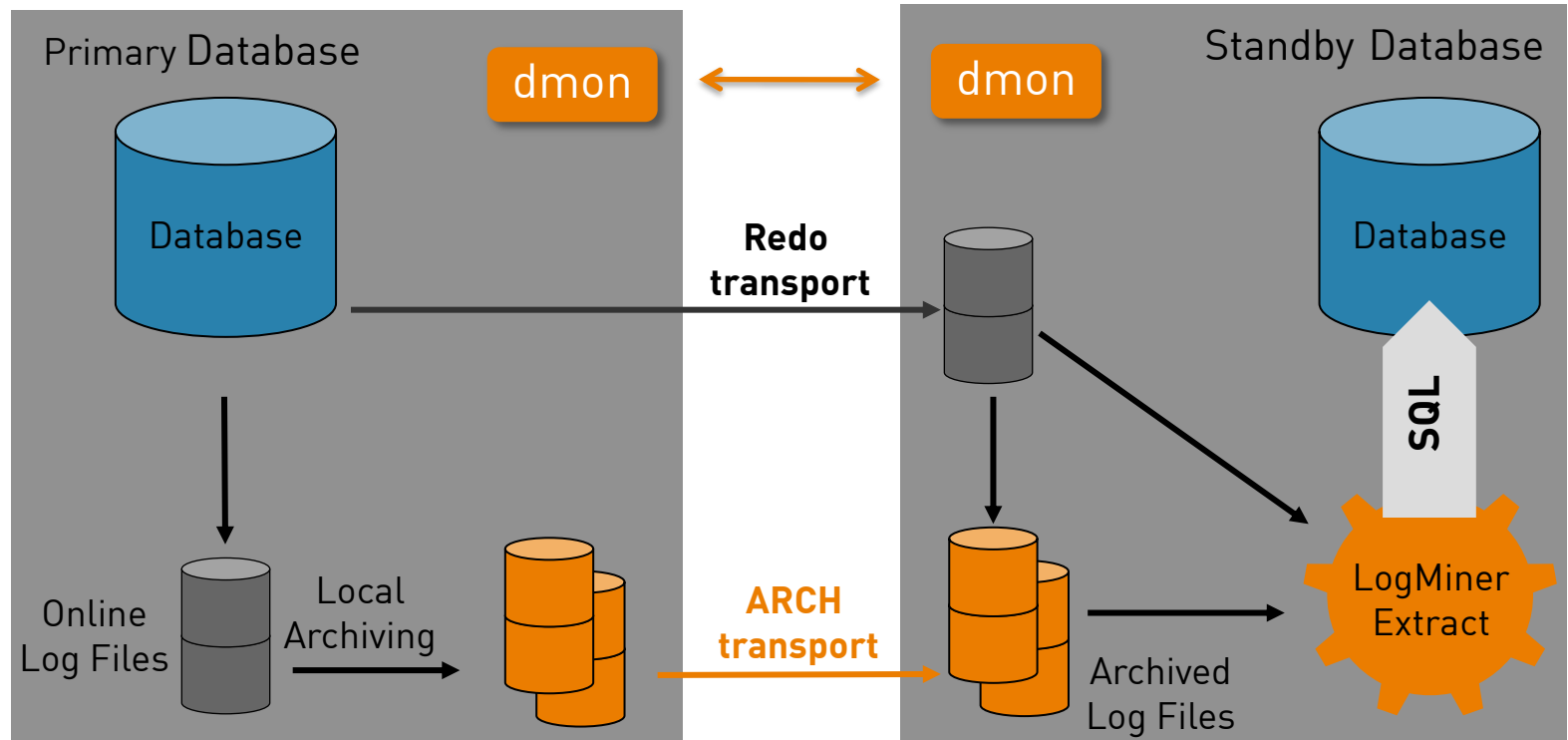
Apply the redo over the recovery process from the transmitted redo or archive log files



Introduction

Logical standby database

Based on LogMiner, extracts SQL statements from the transmitted redo or archive log files and applies them into the logical standby database



Introduction

Unsupported on logical replication

Oracle Logical Standby does not support below object types

- > BFILE
- > ROWID, UROWID
- > Collections (including VARRAYs and nested tables)
- > Objects with nested tables and REFs
- > The following Spatial types are not supported:
 - > MDSYS.SDO_GEORASTER
 - > MDSYS.SDO_TOPO_GEOMETRY

Lots of DDL commands are not replicated

- > Documentation Data Guard Concepts and Administration
 - > C.12 Skipped SQL Statements on a Logical Standby Database

Introduction

Features of logical standby database

Usable for maintenance tasks

- > Database upgrade with near Zero downtime
- > Adding partitioning to non-partitioned tables
- > Changing BasicFiles LOBs to SecureFiles LOBs
- > Changing XMLType stored as CLOB to XMLType stored as binary XML
- > Altering tables to become OLTP-compressed

Rolling Upgrade concept



Rolling Upgrade concept

With Oracle 11gR2

Since Oracle 10.1 logical standby database exists, which allows Rolling Upgrade for database upgrade

Rolling Upgrade allows **near zero downtime** database upgrade

Lots of manual intervention for the setup, configuration and operation

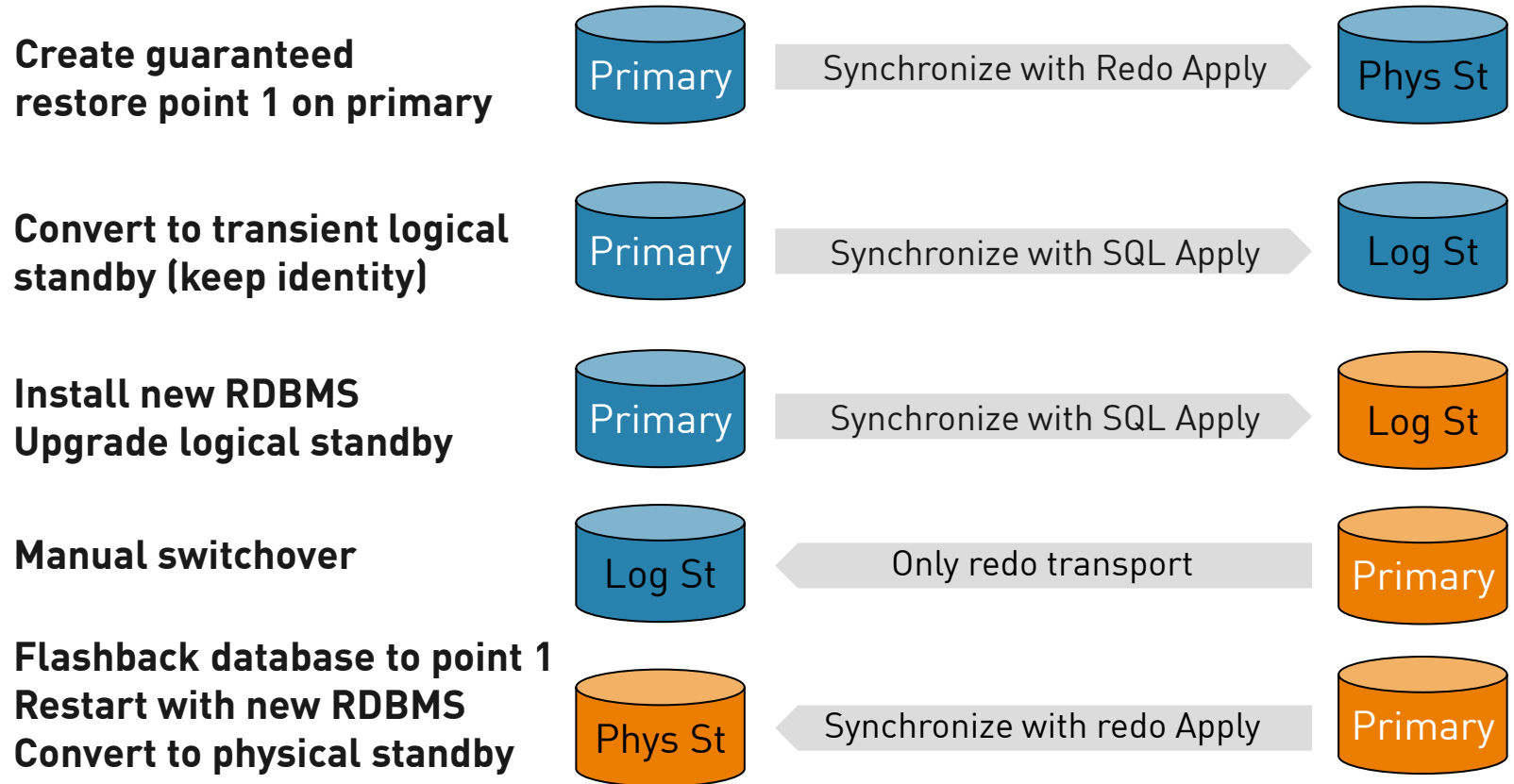
Does not support Data Guard Broker configuration

Rolling Upgrade can be implemented with unsupported object types

- > it needs more attention and no changes are possible on unsupported objects during the upgrade

Rolling Upgrade concept With Oracle 11gR2

Upgrade plan from 11.2.0.1 to 11.2.0.3 with Rolling Upgrade



Rolling Upgrade concept

New package with Oracle 12cR1

DBMS_ROLLING usage overview

Stages	Actions
Specification	<code>dbms_rolling.init_plan</code>
	<code>dbms_rolling.set_parameter</code>
Compilation	<code>dbms_rolling.build_plan</code>
Execution	<code>dbms_rolling.start_plan</code>
	Upgrade logical standby database manually (change OH, catupgrd)
	<code>dbms_rolling.switchover</code>
	Upgrade former primary database manually (change OH, restart)
	<code>dbms_rolling.finish_plan</code>
	<code>dbms_rolling.destroy_plan</code>

Rolling Upgrade concept

New package with Oracle 12cR1

DBMS_ROLLING with Data Guard Broker usage overview

Stages	Actions
Specification	<code>dbms_rolling.init_plan</code>
	<code>dbms_rolling.set_parameter</code>
DG BROKER	<code>dgmgrl> disable configuration</code>
Compilation	<code>dbms_rolling.build_plan</code>
Execution	<code>dbms_rolling.start_plan</code>
	Upgrade logical standby database manually(change OH, catupgrd)
	<code>dbms_rolling.switchover</code>
	Upgrade former primary database manually(change OH, catupgrd)
	<code>dbms_rolling.finish_plan</code>
DG BROKER	<code>dgmgrl> enable configuration</code>
	<code>dbms_rolling.destroy_plan</code>

Upgrade from 12cR1 to 12cR1

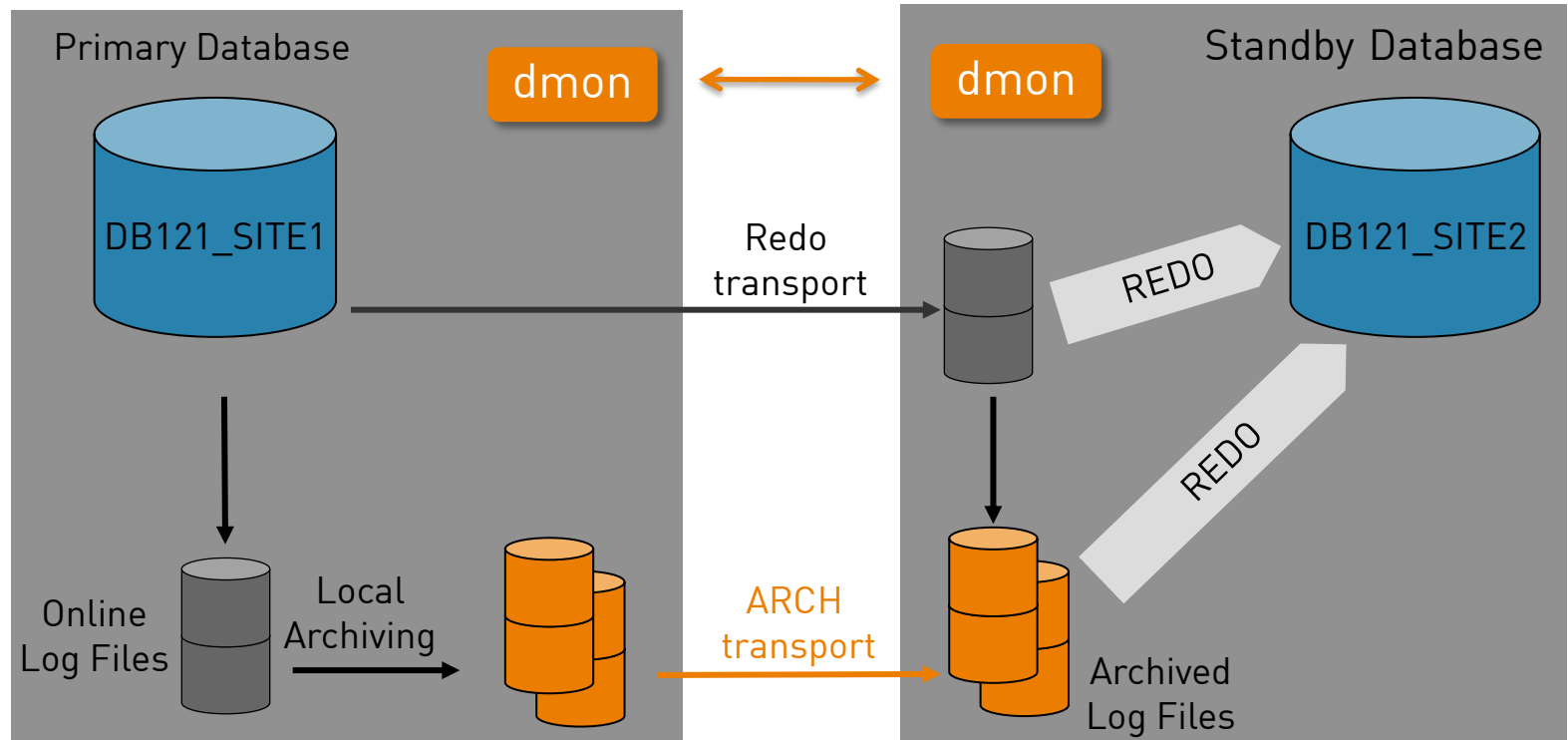
Ingenuity & Infrastructure
at your service.



Upgrade from 12cR1 to 12cR1

Overview

This upgrade will be done without software update, because currently only one version of 12c exist.



Upgrade from 12cR1 to 12cR1 Specification

Start Rolling Upgrade plan building

```
SQL> exec DBMS_ROLLING.INIT_PLAN(future_primary=>'DB121_SITE2');  
PL/SQL procedure successfully completed.
```

dba_rolling_parameters data dictionary view is updated

```
SQL> SELECT scope, name, curval FROM dba_rolling_parameters;  
SCOPE          NAME          CURVAL  
-----  
DB121_SITE1   INVOLVEMENT   FULL  
DB121_SITE1   MEMBER        NONE  
DB121_SITE2   INVOLVEMENT   FULL  
DB121_SITE2   MEMBER        TRAILING  
              ACTIVE_SESSIONS_TIMEOUT 3600  
              ACTIVE_SESSIONS_WAIT   0  
              BACKUP_CONTROLFILE    rolling_change_backup.f  
.  
.  
.  
31 rows selected.
```

Upgrade from 12cR1 to 12cR1

Specification

Procedure to set specific parameter

```
SQL> exec DBMS_ROLLING.SET_PARAMETER(  
    scope IN VARCHAR2,    -- DB_UNIQUE_NAME for local parameter  
    name  IN VARCHAR2,  
    value IN VARCHAR2);
```

Reset log_level to default value for the DBMS_ROLLING PL/SQL package

- > Default is ERROR
- > Other existing values are INFO, FULL, DEBUG

```
SQL> exec DBMS_ROLLING.SET_PARAMETER (name=>'LOG_LEVEL',value=>NULL);
```

Upgrade from 12cR1 to 12cR1

Disable Data Guard Broker configuration

Data guard broker configuration must be disabled during rolling upgrade operations

```
DGMGRL> show configuration
Configuration - DB121

Protection Mode: MaxAvailability
Databases:
DB121_SITE1 - Primary database
DB121_SITE2 - Physical standby database

Fast-Start Failover: DISABLED
Configuration Status:
SUCCESS

DGMGRL> disable configuration
Disabled.
```

Upgrade from 12cR1 to 12cR1

Compilation

Build the upgrade plan on primary database

```
SQL> exec DBMS_ROLLING.BUILD_PLAN;  
PL/SQL procedure successfully completed.
```

Verify successful upgrade plan creation

```
SQL> SELECT instid id, target, phase, description  
       FROM   dba_rolling_plan;
```

ID	TARGET	PHASE	DESCRIPTION
1	DB121_SITE1	START	Verify database is a primary
2	DB121_SITE1	START	Verify MAXIMUM PROTECTION is disabled
3	DB121_SITE2	START	Verify database is a physical standby
4	DB121_SITE2	START	Verify physical standby is mounted
5	DB121_SITE1	START	Verify spfile file exists and is modifiable
7	DB121_SITE1	START	Verify DG Broker configuration is disabled
9	DB121_SITE1	START	Verify flashback database is enabled
.	.	.	.

64 rows

Upgrade from 12cR1 to 12cR1

Compilation

If the below errors occurs, your standby database must be set to the «MOUNT» mode, but the message misses to mention the standby database 😞

```
SQL> exec DBMS_ROLLING.BUILD_PLAN;

ERROR at line 1:
ORA-45438: database is not in mounted mode
ORA-06512: at "SYS.DBMS_ROLLING", line 16
```

Error message is visible in the data dictionary view

```
SQL> SELECT * FROM dba_rolling_events;
EVENT_TIME          MESSAGE
-----
23-SEP-13 05.41.54.529592 PM detected invalid open mode on
DB121_SITE2: expected (MOUNTED) actual
(OPEN READ ONLY)
```

Upgrade from 12cR1 to 12cR1

Compilation

Steps of the different phases

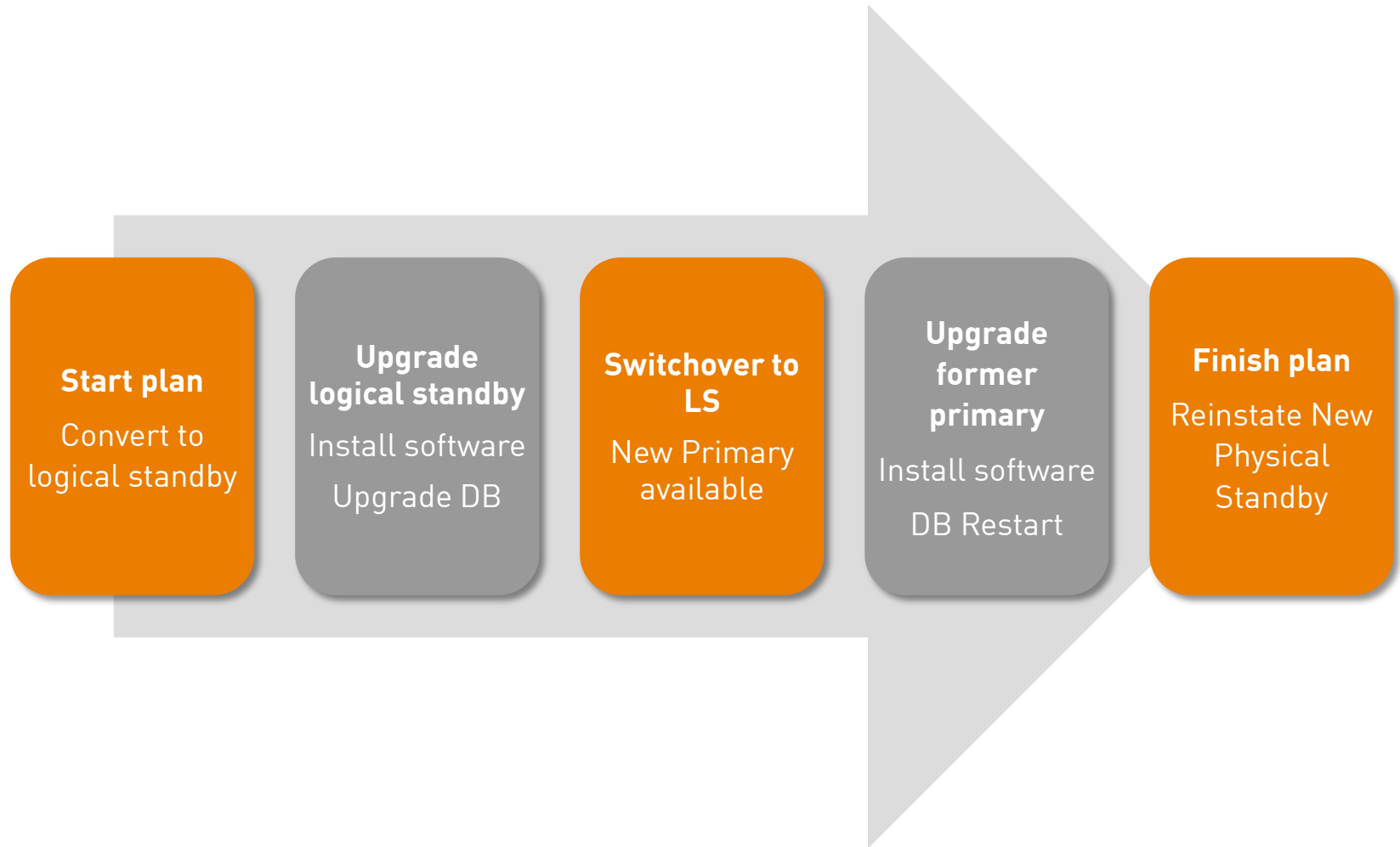
```
SQL> SELECT target,phase, count(*)
      FROM dba_rolling_plan
      GROUP BY target,phase;
```

TARGET	PHASE	COUNT (*)
-----	-----	-----
DB121_SITE1	START	13
DB121_SITE2	START	23
DB121_SITE1	SWITCH	5
DB121_SITE2	SWITCH	8
DB121_SITE1	FINISH	10
DB121_SITE2	FINISH	5

64 rows

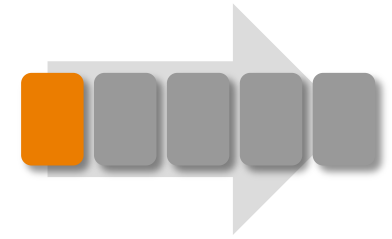
Upgrade from 12cR1 to 12cR1

Execution plan overview



Upgrade from 12cR1 to 12cR1

Start plan



Create guaranteed restore point on both databases

Convert physical standby to transient logical standby

```
SQL> exec DBMS_ROLLING.START_PLAN;  
ORA-45415: instruction execution failure  
ORA-06512: at "SYS.DBMS_ROLLING", line 80
```

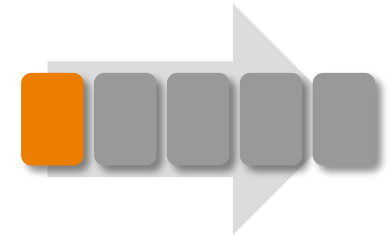
```
SQL> SELECT instid,source,target,phase,status  
       FROM   dba_rolling_plan;
```

INSTID	SOURCE	TARGET	PHASE	STATUS
7	DB121_SITE1	DB121_SITE1	START	SUCCESS
8	DB121_SITE1	DB121_SITE2	START	SUCCESS
9	DB121_SITE1	DB121_SITE1	START	ERROR
10	DB121_SITE1	DB121_SITE2	START	PENDING

If an error occur check data dictionary view dba_rolling_plan

Upgrade from 12cR1 to 12cR1

Start plan



For detailed error information

```
SQL> SELECT instid, target, phase, description
       FROM   dba_rolling_plan
       WHERE  instid IN (9,27);
```

INSTID	TARGET	PHASE	DESCRIPTION
9	DB121_SITE1	START	Verify flashback database is enabled
27	DB121_SITE1	START	Execute dbms_logstdby.build

Additional information is also available in the alert.log file

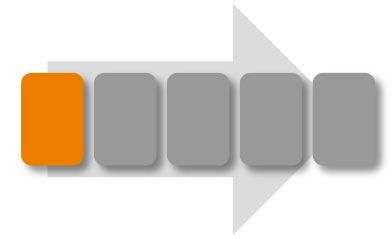
```
oracle@srv01:/ [DB121] tail -f alert_DB121.log
RTS(4020): failed on instruction 27 from plan revision 1
Dumping error context:
ORA-65024: Pluggable database is not open.
ORA-06512: at "SYS.DBMS_LOGMNR_INTERNAL", line 7214
```

All pluggable databases must be opened

Flashback database must be activated on both databases

Upgrade from 12cR1 to 12cR1

Start plan - Check



Check if all phase='START' have the SUCCESS status

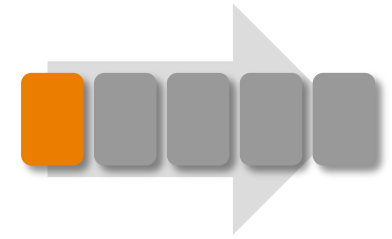
```
SQL> exec DBMS_ROLLING.START_PLAN;  
PL/SQL procedure successfully completed.
```

```
SQL> SELECT      status, count(*)  
      FROM      dba_rolling_plan  
      WHERE     phase='START'  
      GROUP BY  status;
```

STATUS	COUNT (*)
-----	-----
SUCCESS	36

Upgrade from 12cR1 to 12cR1

Start plan - Check

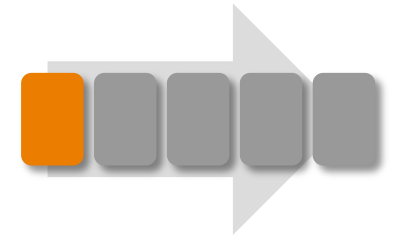


Verify that standby database is successfully converted

```
oracle@srv02:/u00/app/oracle/[DB121] DB121
***** dbi services Ltd. *****
STATUS                : OPEN
DB_UNIQUE_NAME        : DB121_SITE2
OPEN_MODE              : READ WRITE
LOG_MODE               : ARCHIVELOG
DATABASE_ROLE         : LOGICAL STANDBY
FLASHBACK_ON          : YES
FORCE_LOGGING         : NO
VERSION                : 12.1.0.1.0
*****
```

Upgrade from 12cR1 to 12cR1

Start plan - Check



Check if replication is running on logical standby database

```
SQL> SELECT file_name, sequence#, dict_begin, dict_end, applied
FROM dba_logstdby_log ORDER BY sequence#;
```

FILE_NAME	SEQ#	DICT_BEGIN	DICT_END	APPLIED
-----	-----	-----	-----	-----
/.../DB121_SITE2/foreign_archivelog/DB121_SITE1/...	29	YES	YES	CURRENT
/.../DB121_SITE2/foreign_archivelog/DB121_SITE1/...	30	NO	NO	CURRENT
/.../DB121_SITE2/foreign_archivelog/DB121_SITE1/...	31	NO	NO	NO

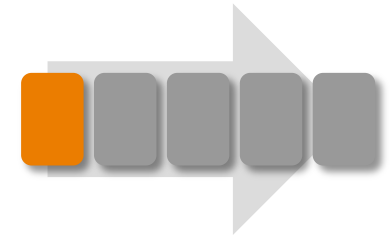
Replication initialization to load the dictionary, can take up to minutes

```
SQL> SELECT type, high_scn, status FROM v$logstdby_process;
```

TYPE	STATUS
-----	-----
COORDINATOR	ORA-16115: 60% of LogMiner dictionary loading is done
READER	ORA-16127: stalled waiting for additional transactions to be applied
BUILDER	ORA-16117: processing
PREPARER	ORA-16116: no work available

Upgrade from 12cR1 to 12cR1

Start plan - Check



Pluggable databases must be opened manually

```
ORA-16331: container "PDB1" is not open
SQL> alter pluggable database PDB1 open;
```

In some case recovery must be restarted

```
SQL> alter database start logical standby apply immediate;
```

To verify if both database are synchronous

> On Primary

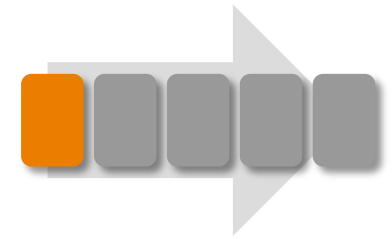
```
SQL> archive log list
Current log sequence          34
```

> On Standby

```
SQL> SELECT type,status FROM v$logstdby_process;
TYPE          STATUS
-----
READER        ORA-16242: Processing log file (thread# 1, sequence# 34)
BUILDER       ORA-16119: building transaction at SCN 0x0000.002386cc
```

Upgrade from 12cR1 to 12cR1

Start plan - Check



Some commands to handle collisions

> Check failed transaction

```
SQL> SELECT      xidusn,xidslt,xidsqn,status,event,status_code,commit_scn
FROM            dba_logstdby_events
WHERE          commit_scn IS NOT NULL
ORDER BY      commit_scn;
```

> How skip failed transaction, in help of the above information

```
SQL> execute DBMS_LOGSTDBY.SKIP_TRANSACTION (4,11,1520);
```

> Restart replication

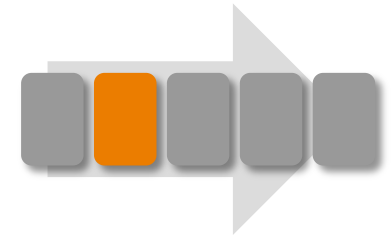
```
SQL> alter database start logical standby apply immediate;
```

> Check if the replication processes are running

```
SQL> SELECT type,high_scn,status FROM v$logstdby_process;
```


Upgrade from 12cR1 to 12cR1

Manually upgrade logical standby



Not possible because no newer software version exists

```
SQL> shutdown immediate
-- set new ORACLE_HOME
SQL> startup upgrade
SQL> @catupgrd.sql
```

Only several replication tests with the pluggable database PDBDB121 have been performed

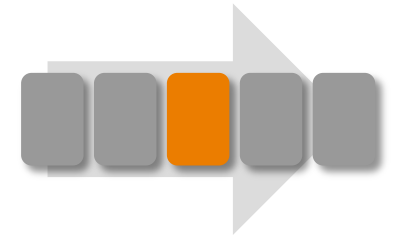
- > Update on schemas HR and SCOTT successful
- > “CREATE TABLE AS SELECT” can’t be replicated

```
HR> CREATE TABLE test AS SELECT * FROM countries;
```

- > «CREATE TABLE» is supported but not «CREATE TABLE AS SELECT» ☹

Upgrade from 12cR1 to 12cR1

Switchover to logical standby



Start switchover command on current primary

```
SQL> execute DBMS_ROLLING.SWITCHOVER;  
PL/SQL procedure successfully completed.
```

Follow the switchover on data dictionary view dba_rolling_plan

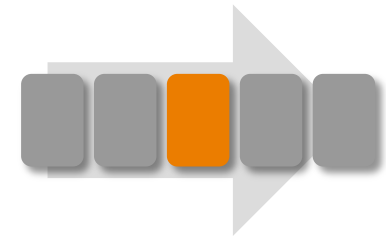
Check database status after switchover is finished

```
oracle@srv01:/ [rdbms121] DB121  
***** dbi services Ltd. *****  
STATUS                : OPEN  
DB_UNIQUE_NAME        : DB121_SITE1  
OPEN_MODE              : READ WRITE  
LOG_MODE               : ARCHIVELOG  
DATABASE_ROLE         : LOGICAL STANDBY  
FLASHBACK_ON          : YES  
*****
```

```
oracle@srv02:/ [rdbms121] DB121  
***** dbi services Ltd. *****  
STATUS                : OPEN  
DB_UNIQUE_NAME        : DB121_SITE2  
OPEN_MODE              : READ WRITE  
LOG_MODE               : ARCHIVELOG  
DATABASE_ROLE         : PRIMARY  
FLASHBACK_ON          : YES  
*****
```

Upgrade from 12cR1 to 12cR1

Switchover to logical standby



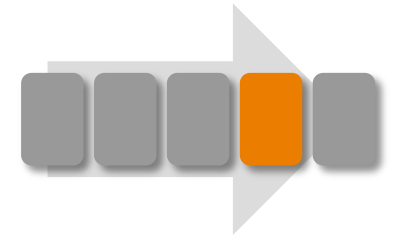
Verification of the switchover steps on old primary

```
SQL> SELECT instid id,target,status,description
       FROM   dba_rolling_plan
       WHERE  phase='SWITCH';
```

ID	TARGET	STATUS	DESCRIPTION
37	DB121_SITE2	SUCCESS	Verify database is in OPENRW mode
38	DB121_SITE2	SUCCESS	Record completion of user upgrade of DB121_SITE2
39	DB121_SITE2	SUCCESS	Scan LADs for presence of DB121_SITE1 destination
40	DB121_SITE2	SUCCESS	Test if DB121_SITE1 is reachable using TNS service
41	DB121_SITE1	SUCCESS	Enable log file archival to DB121_SITE2
42	DB121_SITE2	SUCCESS	Start logical standby apply
43	DB121_SITE2	SUCCESS	Wait until apply lag has fallen below 600 seconds
44	DB121_SITE1	SUCCESS	Log post-switchover instructions to events table
45	DB121_SITE1	SUCCESS	Switch database to a logical standby
46	DB121_SITE2	SUCCESS	Wait until end-of-redo has been applied
47	DB121_SITE2	SUCCESS	Switch database to a primary
48	DB121_SITE1	SUCCESS	Enable compatibility advance despite presence of GRPs
49	DB121_SITE1	SUCCESS	Synchronize plan with new primary

Upgrade from 12cR1 to 12cR1

Manually Upgrade former primary



Shutdown the former primary database

Install new software

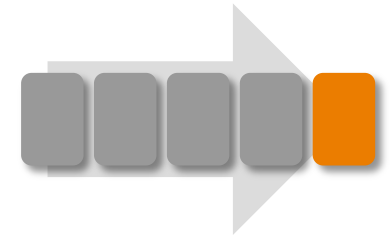
> not possible because no newer software version exist

Restart the former primary database

```
SQL> shutdown immediate
-- set new ORACLE_HOME
SQL> startup mount
```

Upgrade from 12cR1 to 12cR1

Finish plan



Flashback the former primary database

Convert the former primary database to physical standby

Restart recovery process

Remove the guaranteed restore point

This command will be started on new primary !

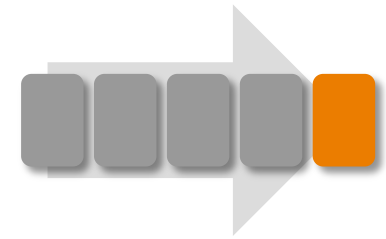
```
SQL> execute DBMS_ROLLING.FINISH_PLAN;
BEGIN DBMS_ROLLING.FINISH_PLAN; END;

ERROR at line 1:
ORA-45438: database is not in mounted mode
ORA-06512: at "SYS.DBMS_ROLLING", line 36

SQL> execute DBMS_ROLLING.FINISH_PLAN;
-- is blocked
```

Upgrade from 12cR1 to 12cR1

Finish plan - workarround



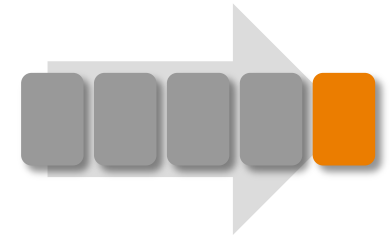
Check current rolling upgrade status

```
SQL> SELECT instid id,target,status,description
       FROM   dba_rolling_plan
       WHERE  phase='FINISH'
/
```

ID	TARGET	STATUS	DESCRIPTION
50	DB121_SITE1	SUCCESS	Verify only a single instance is active
51	DB121_SITE1	SUCCESS	Verify database is mounted
52	DB121_SITE1	SUCCESS	Flashback database
53	DB121_SITE1	SUCCESS	Convert into a physical standby
54	DB121_SITE2	SUCCESS	Verify database is open
55	DB121_SITE2	SUCCESS	Save the DBID of the new primary
56	DB121_SITE2	SUCCESS	Save the logminer session start scn
57	DB121_SITE1	PENDING	Wait until transient logical redo branch has been regist.
58	DB121_SITE1	PENDING	Start media recovery
59	DB121_SITE1	PENDING	Wait until recovery has started on transient...
60	DB121_SITE1	PENDING	Wait until upgrade redo has been recovered
61	DB121_SITE1	PENDING	Prevent compatibility advance
63	DB121_SITE1	PENDING	Drop guaranteed restore point DBMSRU_INITIAL

Upgrade from 12cR1 to 12cR1

Finish plan - workarround



The reason for the error is:

- > the log_archive_dest_2 is not configured on new primary

And therefore:

- > the new physical standby did not received the archive log files

I tried to fix the blocked procedure problem on new primary issuing the command :

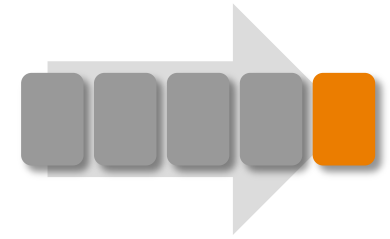
```
SQL> alter system set log_archive_dest_2 = 'service="DB121_SITE1"',
'ASYNCR NOAFFIRM delay=0 optional compression=disable max_failure=0
max_connections=1 reopen=300 db_unique_name="DB121_SITE1"
net_timeout=30',
'valid_for=(online_logfile,all_roles) '
SCOPE=BOTH;

SQL> alter system archive log current;
```

Bingo ! The DBMS_ROLLING.FINISH_PLAN procedure has successfully completed

Upgrade from 12cR1 to 12cR1

Enable DGB configuration



Configuration was disabled with primary database on SITE1

But will be re-enabled with primary database on SITE2

Is not a problem anymore 😊

```
DGMGRL> connect sys/*****@DB121_SITE2

DGMGRL> enable configuration
Enabled.

DGMGRL> show configuration
Configuration - DB121
  Protection Mode: MaxAvailability
  Databases:
    DB121_SITE2 - Primary database
    DB121_SITE1 - Physical standby database
  Fast-Start Failover: DISABLED
  Configuration Status:
SUCCESS
```


Core messages

Insight & Infrastructure
at your service.



Core messages

Pro & contra

- > Easy to use with clear information delivered
- > Good solution for near-to-zero Oracle database upgrade



- > Different character set are not supported
- > Oracle ActiveGuard licence required
- > Only usable starting the first patch set of 12cR1 release
- > Manual upgrade required from the initial 12c release
- > During upgrade no DG broker available ☹️ to help analyzing replication problems



Core messages

Rolling Upgrade vs other Oracle solutions

	Physical Standby Database	GoldenGate	Rolling Upgrade	Multitenant upgrade with Unplug-plug
Oracle upgrade with less than 5 minutes downtime	No	Yes	Yes	Yes
Oracle upgrade with less than 1 minute downtime	No	Yes	Yes	No

Any questions? Please do ask.

Hervé Schweitzer

CTO

Delivery Manager

Mobile +41 79 963 43 67

Herve.schweitzer@dbi-services.com

www.dbi-services.com



We look forward to working with you!