

# 100TB Migrations Keep the Downtime low



Mike Dietrich  
Master Product Manager  
Database Upgrades and Migrations  
Oracle Corporation

ORACLE®

Updated: 17-MAY-2017

Copyright © 2017 Oracle and/or its affiliates. All rights reserved. | Migrate Very Large Databases with Smaller Downtime

\$> whoami



**Mike Dietrich**

Master Product Manager  
Database Upgrades & Migrations



MikeDietrichDE



<https://MikeDietrichDE.com>

**6 years**

RDBMS Core & Mission Critical Support

**5.5 years**

Technology Presales for DataGuard, Upgrades

**+9 years**

**ST Upgrade Development Team**

**50%**

Reference  
Projects

**50%**

Workshops  
Worldwide


**+ x%**

Development  
Work

# Database Upgrade Blog - Slides

- <https://MikeDietrichDE.com/>

**Upgrade your Database – NOW!**  
Mike Dietrich's Oracle Database Upgrade Blog



Blog Slides

## Slides Download Center

[Edit](#)

This page will be refreshed to a more user-friendly look&feel soon.

### Comprehensive

- [Upgrade, Migrate & Consolidate to Oracle Database 12.2 & Cloud](#)  
*Updated: 26-FEB-2017*
- [Upgrade, Migrate & Consolidate to Oracle Database 12c](#)  
*Refreshed 3-DEC-2016*
- [Why you need to upgrade NOW!](#)

### Recent Posts

[Hands-On Lab available: Upgrade to Oracle Database 12.2.0.1](#)

[Collaborate Conference 2017 – Upgrade "Performance" Talk + Oracle Database 12.2 Hands-On Lab](#)

[Multiple hop upgrades? Execute the matching preupgrade scripts for each hop](#)

[Oracle Database 12.2.0.1 for Windows available](#)



# Database Upgrade Blog – Hands-On Lab

- <https://MikeDietrichDE.com>

## Upgrade your Database – NOW!

Mike Dietrich's Oracle Database Upgrade Blog



Blog Slides **Hands-On Lab**

### Hands-On Lab

#### Hands On Lab – Oracle Database 12.2.0.1

- Hands On Lab  
Upgrade, Migrate, Consolidate to Oracle Database 12.2.0.1  
Uploaded: March 29, 2017
- Hands-On Lab Instructions  
Uploaded: March 29, 2017
- Known Issues:
  - HOL Part 1

#### Recent Posts

Hands-On Lab available: Upgrade to Oracle Database 12.2.0.1  
Collaborate Conference 2017 – Upgrade "Performance" Talk + Oracle Database 12.2 Hands-On Lab  
Multiple hop upgrades? Execute the matching preupgrade scripts for each hop  
Oracle Database 12.2.0.1 for Windows available



# Concept Transportable Tablespaces





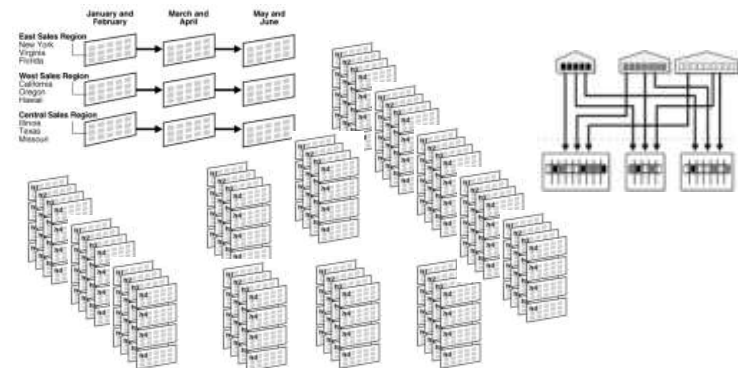
# Transportable Tablespaces Pros and Cons

## ■ Pro

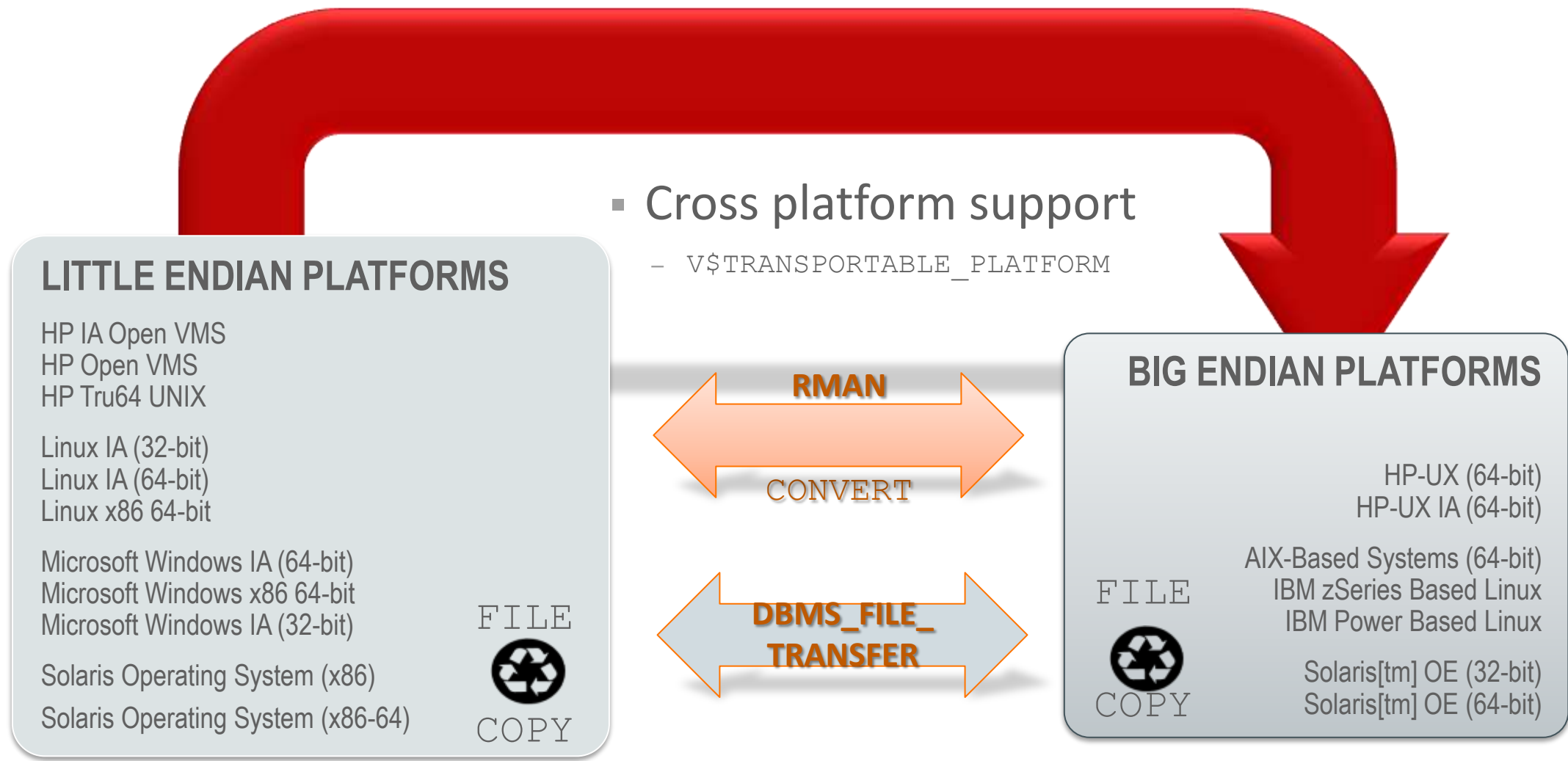
- Potentially very fast
  - Copying large files can be faster than exporting & importing everything
  - No need to rebuild indexes
- Cross platform since Oracle 10g
- Decrease copy/convert time by using RMAN Incremental Backups
- Proven solutions for EBS and other apps available

## ■ Con

- SYSTEM/SYSAUX can't be transported
- Complexity is your enemy
  - Too many objects to rebuild
    - Views, synonyms, sequences ...
    - **Simple is better for fast TTS!!!**
  - Too many objects in tablespaces slow down meta expdp/impdp
    - **(Sub)partitions**, partitioned indexes ...



# Concept Transportable Tablespaces xTTS



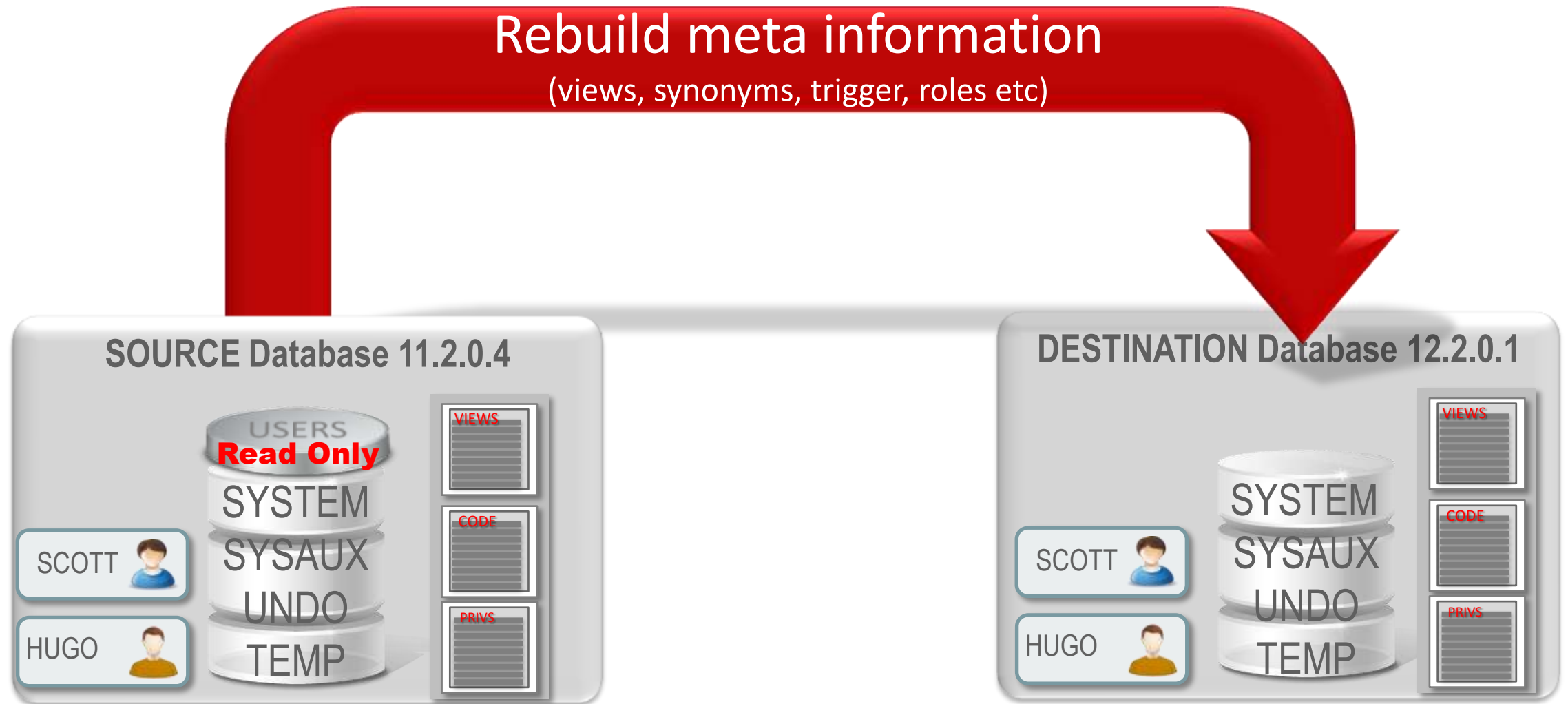
# Concept Transportable Tablespaces xTTS

- TTS cross platform
  - RMAN creates a file copy
  - Can be done on source or target system
    - Use the faster storage
  - Takes approximately the same amount of time as a backup and requires staging space
  - Multiple channels can be used
  - Example:


```
RMAN> CONVERT TABLESPACE users,example  
      TO PLATFORM 'Linux IA (32-bit)'  
      FORMAT='/stage/transport_linux/%U';
```
  - DBMS\_FILE\_TRANSFER converts implicitly and does not require staging



# Upgrade/Migration: Transportable Tablespaces



# Possible options

- Moving meta information
  - The “**brute force**” approach
    - Data Pump 



```
expdp/impdp CONTENT=METADATA_ONLY
```

- The “**smart**” approach
  - DBMS\_METADATA



```
SELECT DBMS_METADATA.GET_DDL('SYNONYM', SYNONYM_NAME,  
OWNER) FROM all_synonyms where owner='PUBLIC' and  
table_owner not in ('SYS');
```

# Real World Checkpoint



## Customer

### ■ Fuji Xerox Singapore

## Project

## Constraints

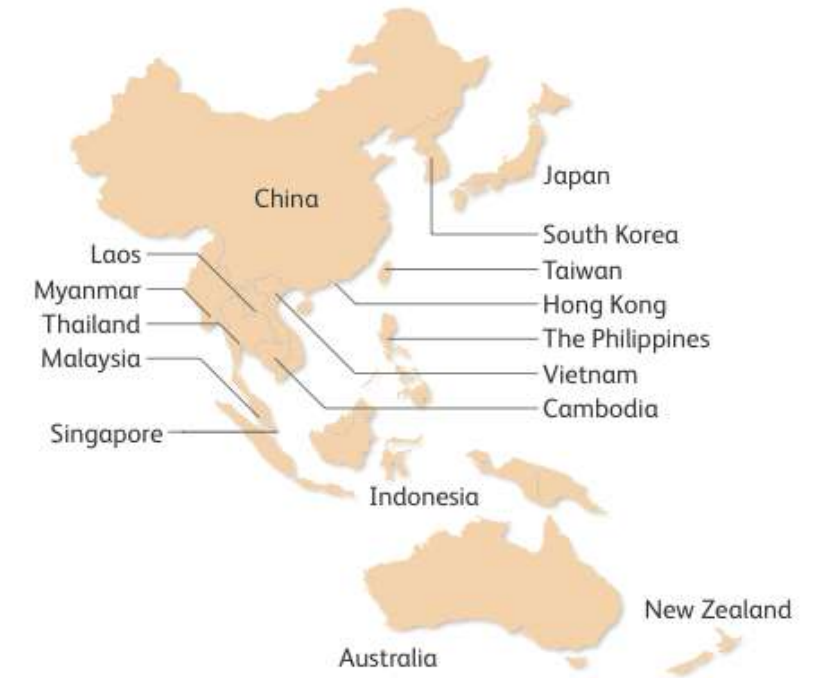
## Preparation

## Migration

## Success?

## Remarks

- Headquartered in Tokyo, locations throughout the Asia-Pacific region
- Global leader in document services and communications
- Over ¥1 trillion annual revenue
- 45,000+ employees



# Real World Checkpoint



Customer

Project

Constraints

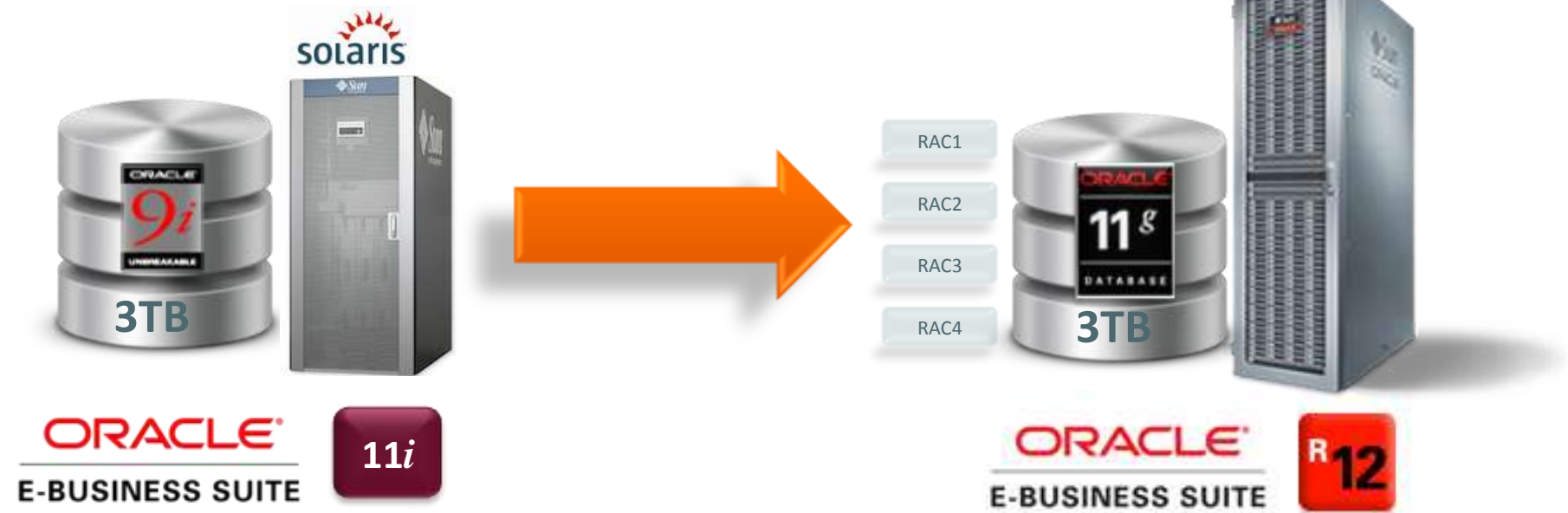
Preparation

Migration

Success?

Remarks

- Upgrade and migrate Oracle E-Business Suite database and applications
  - Multi-language environment



# Real World Checkpoint



## Customer

## Project

## Constraints

## Preparation

## Migration

## Success?

## Remarks

- Database hosts information from multiple countries in a single EBS instance
  - Includes nine different alphabets
- OS and Endian Conversion
- Coordination of EBS and DB upgrades and patching
- Single 1Gbit network card on source system
- No testing impact on PROD allowed
- Initial migration testing showed **7+ days** of downtime

# Real World Checkpoint



Customer

Project

Constraints

Preparation

Migration

Success?

Remarks

- Database Migration Options
  - ☒ exp/imp: too slow
  - ☒ Data Pump, xTTS: requires 10g or newer
  - ☑ **Decision: upgrade DB, then use xTTS**
- Test plan
  - Multiple test runs to understand and tune the process
  - Copy of production environment to avoid any impact on business operations during testing
- Worked with third-party SI, Oracle ACS and Oracle Development early in the process



# Real World Checkpoint

## Customer

- Detailed migration planning

## Project

## Constraints

## Preparation

## Migration

## Success?

## Remarks

[illegible]

# Real World Checkpoint



Customer

Project

Constraints

Preparation

Migration

Success?

Remarks

- Addressed network speed issues
  - Reduced file copy time from **9 hours to 4 hours**
    - Added network cards to source system (total 4 x 1Gbit)
    - Parallel scripts to copy data files from source to target
- Identified and applied helpful patches on source and target systems
- Tuned parameters and parallelism for EBS upgrade
- Analyzed and tuned post-upgrade performance on target system

# Real World Checkpoint



Customer

Project

Constraints

Preparation

**Migration**

Success?

Remarks

1. Install target environment
2. Upgrade source DB to 11g on Solaris
3. Migrate across platform using xTTS
4. Upgrade EBS to R12 on Exadata



Cross-platform Transportable  
Tablespaces



# Real World Checkpoint



## Customer

- **YES:** went live in Spring 2013

## Project

- Some EBS actions required or desirable prior to the DB upgrade

## Constraints

## Preparation

- E.g. Patch to improve performance on the `DR$PENDING` table

## Migration

- Just a few post-upgrade DB tuning steps needed

## Success?

- Re-registered services to fix load imbalance in RAC
- Found and fixed a few recommended parameter settings that had been missed

## Remarks

- And some EBS tuning as well
  - Increased number of JVM to accommodate more users
  - Modified Forms OC4J Container values to improve navigation

# Real World Checkpoint



Customer

Project

Constraints

Preparation

Migration

Success?

Remarks

- **Database upgrade** is often only **part of the project**



- The more current your source version, the less work is involved in an upgrade or migration
- Tune your application, not just your database!
- MOS Note:
  - [Using Transportable Tablespaces to Migrate Oracle E-Business Suite Release 12.0 or 12.1 Using Oracle Database 12c Release 1 Enterprise Edition \(Doc ID 1945814.1\)](#)





# The biggest pain points of TTS?

- Copy and convert a large database
- Rebuild all the meta information



# TTS Pain Points

## ■ Size

– Solution:

### **RMAN Incremental Backups**

- PERL scripts in [MOS Note:1389592.1](#) and in [MOS Note: 2005729.1](#)
- Source: 10.2.0.3 or newer
- Target: 11.2.0.4 or newer

## ■ Complexity

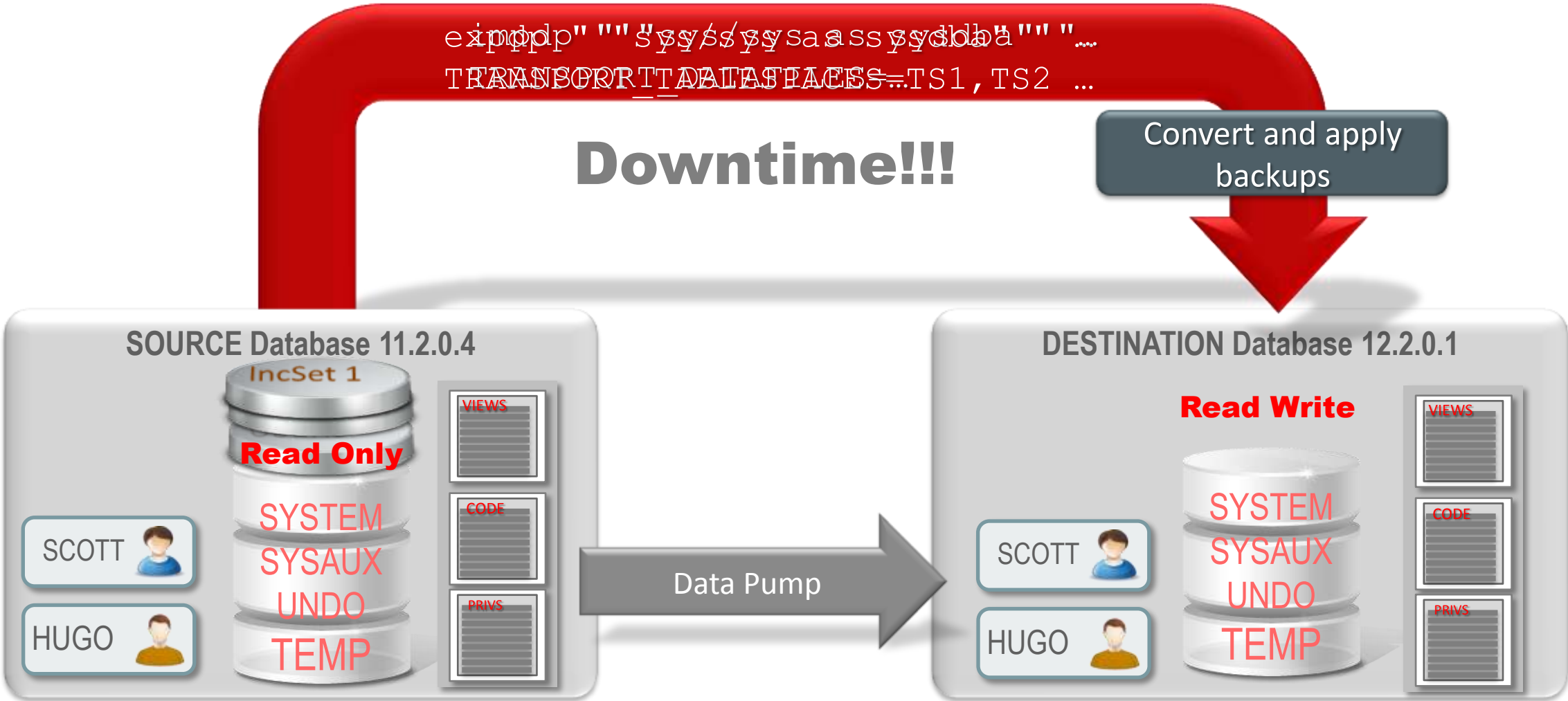
– Solution:

### **Full Transportable Export/Import**

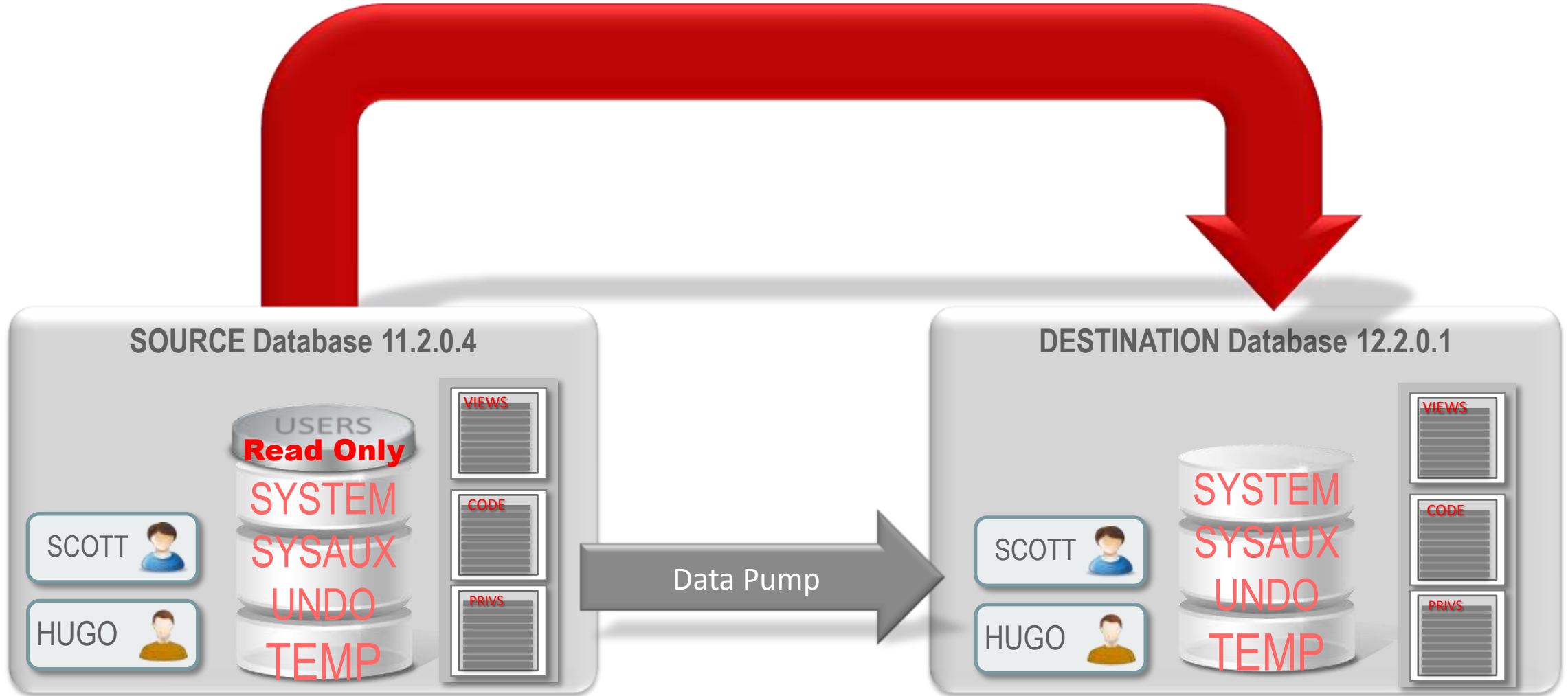
- Data Pump feature allows One Command Migration
- Source: 11.2.0.3 or newer
- Target: 12.1.0.1 or newer



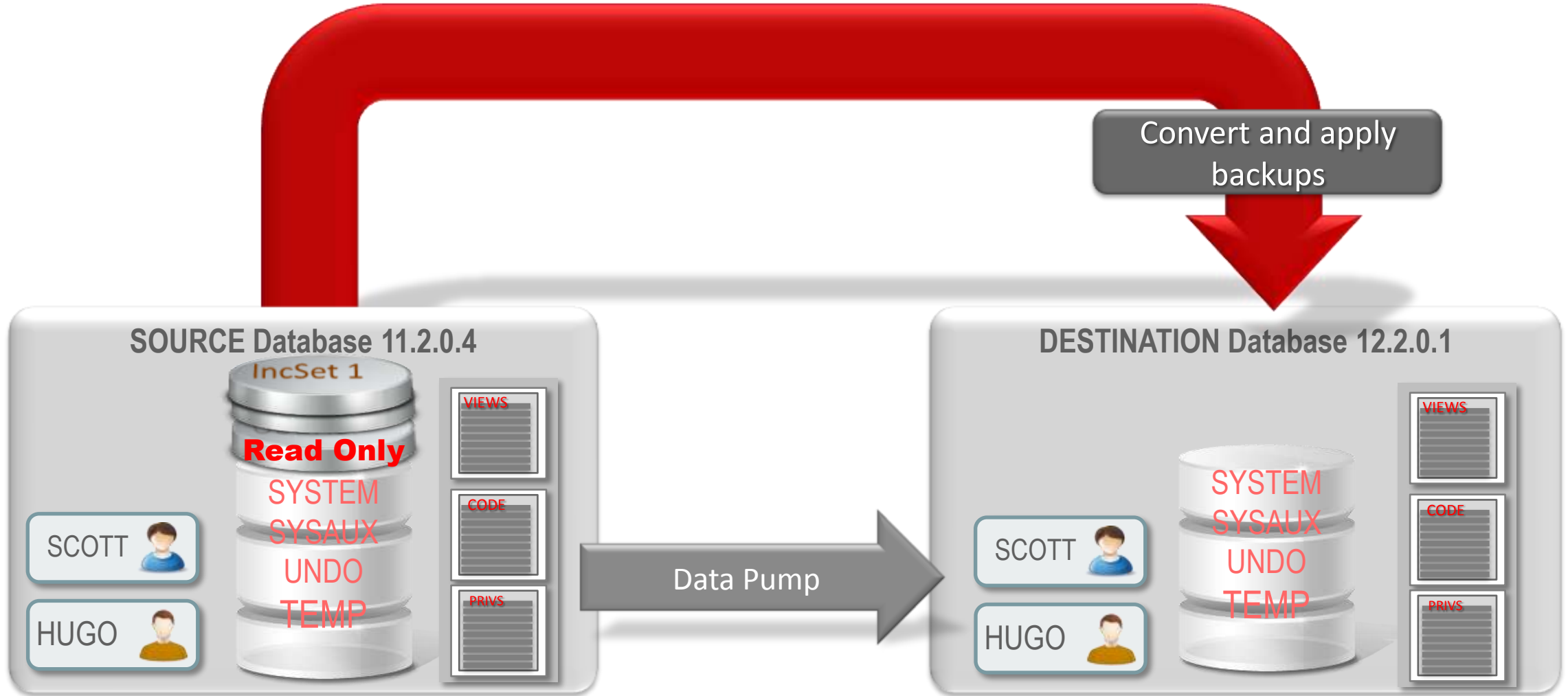
# Pure Transportable Tablespace with Incremental Backups



# Full Transportable Export/Import with Copies



# Full Transportable Export/Import with Backups



A man in a dark jacket and glasses is captured mid-air, jumping over a crowd barrier. He is looking back over his shoulder. The background is a large, dense crowd of people, many of whom are cheering and raising their hands. The scene appears to be at a sports event or a large gathering.

# Let's do it

**Full Transportable Export/Import  
using RMAN Incremental Rolled Forward Backups**



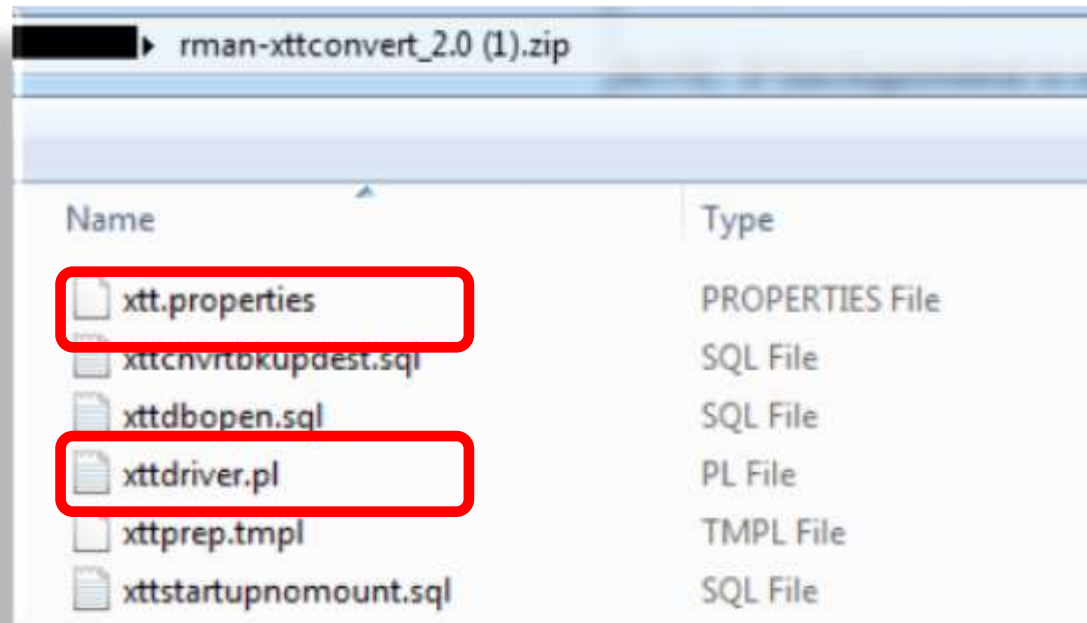
# Overview - Phases

- Phase 1 - Initial Setup phase
- Phase 2 - Prepare phase
- Phase 3 - Roll Forward phase
- Phase 4 - Final Incremental Backup
- Phase 5 - Transport Phase: Import all Metadata
- Phase 6 - Validate the Transported Tablespaces
- Phase 7 - Cleanup



# Phase 1 - Initial Setup phase

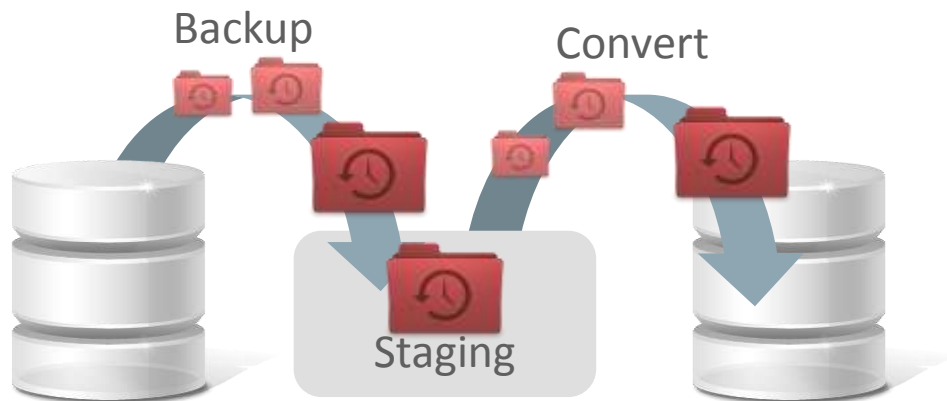
- Download the PERL scripts
  - For 11g source: [MOS Note: 1389592.1](#)
  - For 12c source: [MOS Note: 2005729.1](#)
  - Key scripts:



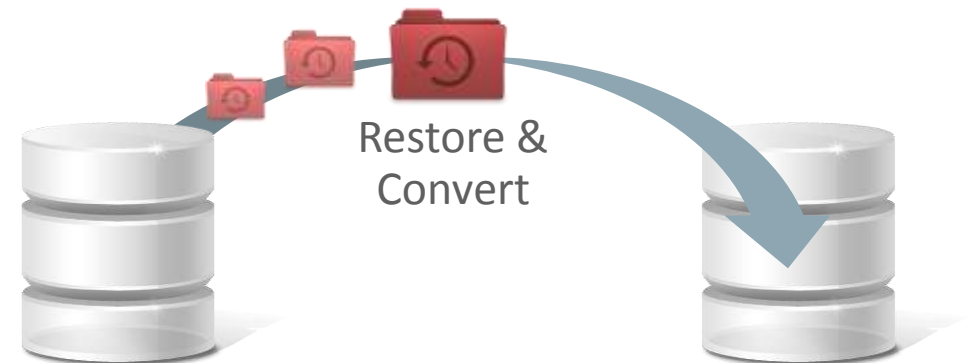
- Extract them to: `/home/oracle/xtt`

# Phase 1 - Initial Setup phase

- Choose the best method
  - **RMAN backup / convert**
    - Requires staging space for CONVERT
    - `xttdriver.pl -p` and `-c`



- **DBMS\_FILE\_TRANSFER**
  - 2TB limitation per file
  - Does not require staging space
  - CONVERT happens implicitly
  - `xttdriver.pl -S` and `-G`



# Phase 1 - Initial Setup phase

- Create a destination database
  - For **Full Transportable Export/Import**:
    - SourceDB must be 11.2.0.3 or higher
    - DestDB must be 12.1.
  - COMPATIBLE equal or higher
  - Identical database character sets
  - Identical national character sets
  - Identical time zone versions

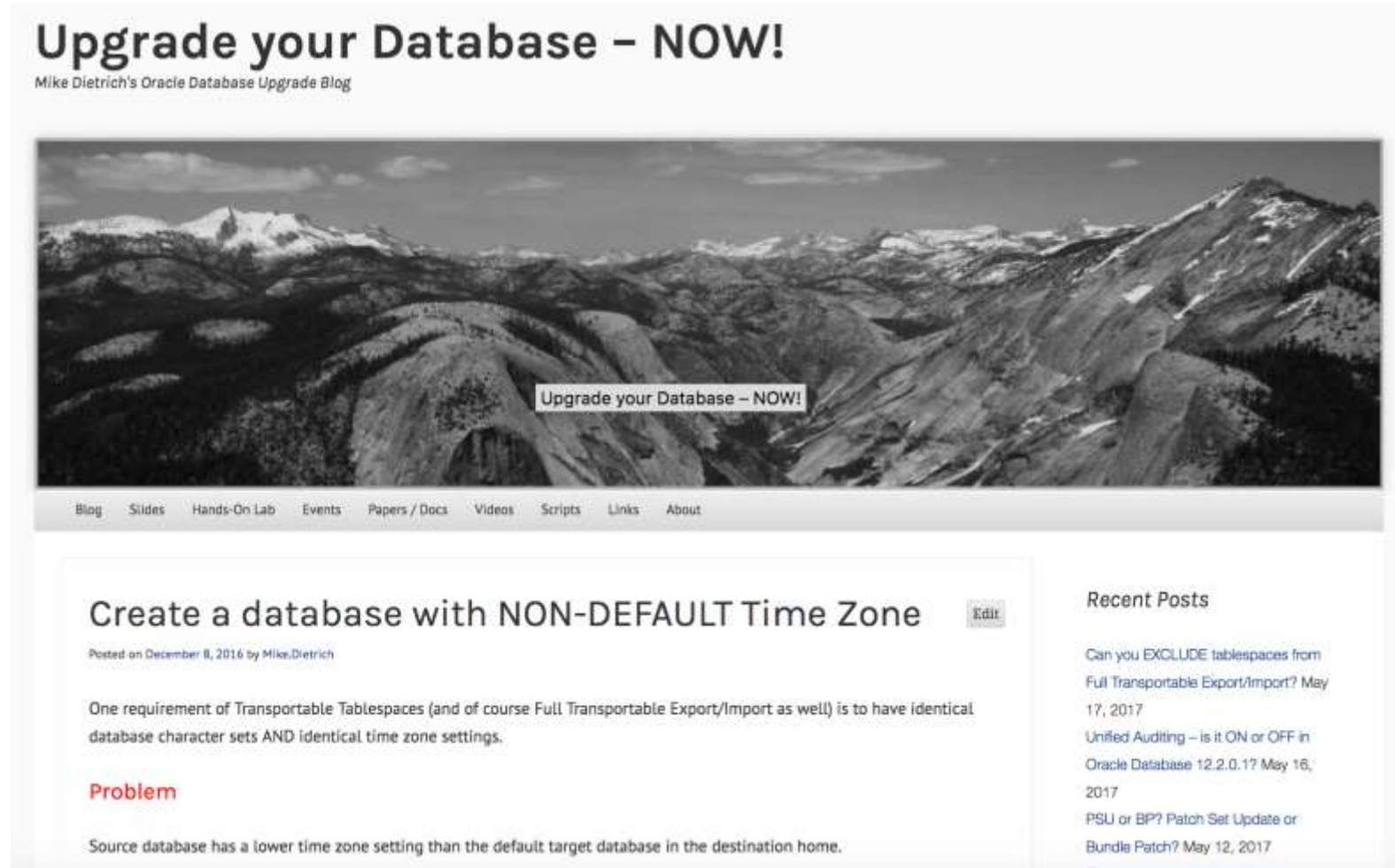
Oracle Database Release	Default Time Zone Version
10.2.0.3, 10.2.0.4, 10.2.0.5	DST V4
11.1.0.6 , 11.1.0.7	DST V4
11.2.0.1	DST V11
11.2.0.2 , 11.2.0.3, <b>11.2.0.4</b>	DST V14
12.1.0.1, <b>12.1.0.2</b>	DST V18
<b>12.2.0.1</b>	DST V26
Most recent interim patch: See <a href="#">MOS Note:412160.1</a>	DST V29

# Things you REALLY need to be aware of

- The MOS Notes don't talk about "full" database migrations and miss most or all TTS/FTEX steps
- The PERL scripts are supported in all directions except for Windows:
  - Although preferred destination system is Linux (either 64-bit Oracle Linux or a certified version of RedHat Linux), this procedure can be used with other Unix based operating systems. However, any non-Linux operating system must be on 11.2.0.4
- Very large (BIGFILE tablespaces) files >16TB
  - Data Pump issues
  - File system limits
- DBMS\_FILE\_TRANSFER can be very fast but has a 2TB limit

# How to create a database with a lower TZ version?

- <https://mikedietrichde.com/2016/12/08/create-a-database-with-non-default-time-zone/>



# Phase 1 - Initial Setup phase

- Identify tablespaces to be transported
- Configure:

## xtt.properties

```
## Tablespaces to transport
## =====
tablespaces=TS1,TS2

## Source database platform ID
## =====
platformid=13

## Source system file locations
## =====
## Location where datafile copies are created
## during the "-p prepare" step.
dfcopydir=/oracle/DQ1/rman_stage

## backupformat
## -----
## Location where incremental backups are created.
backupformat=/oracle/DQ1/rman_stage
```

```
## Destination system file locations
## =====
## Location where datafile copies are placed by the user
## when they are transferred manually from source system.
stageondest=/oracle/DQ1/rman_stage

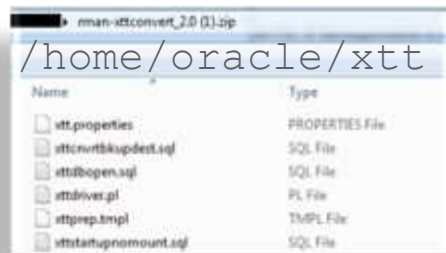
## storageondest
## -----
## Location where the converted datafile copies will be
## written during the "-c conversion of datafiles" step.
## This is the final location of the datafiles
## where they will be used by the destination database.
storageondest=/oracle/DQ1/sapdata50

## backupondest
## -----
## Location where converted incremental backups
## on the destination system will be written during
## the "-r roll forward datafiles" step.
backupondest=/oracle/DQ1/rman_stage_incr
```

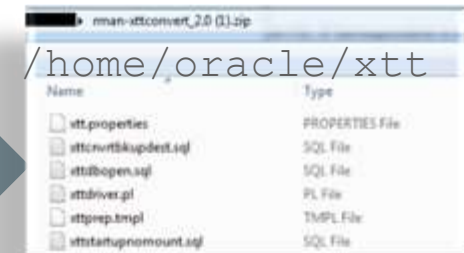


# Phase 1 - Initial Setup phase

- Enable block change tracking in source database  
`ALTER DATABASE ENABLE BLOCK CHANGE TRACKING USING FILE '<name>' REUSE;`
- Copy all xtt-scripts to the destination host
- Set TMPDIR=/home/oracle/xtt on both hosts



Copy xtt scripts including modified xtt.properties



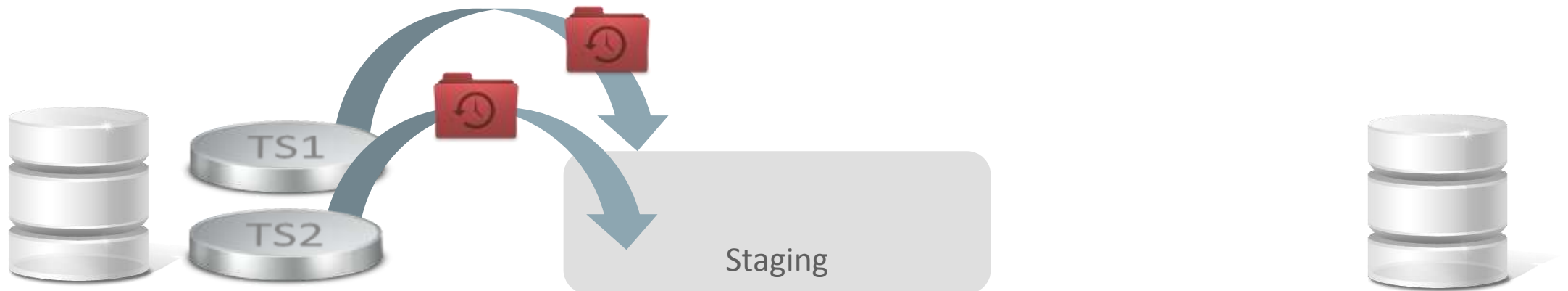
# Phase 2 - Prepare phase

- Create data file copies on **source**

– [oracle@**source**] \$ \$ORACLE\_HOME/perl/bin/perl **xttdriver.pl -p**

– Creates the following files used later:

- xttpplan.txt
- rmanconvert.cmd

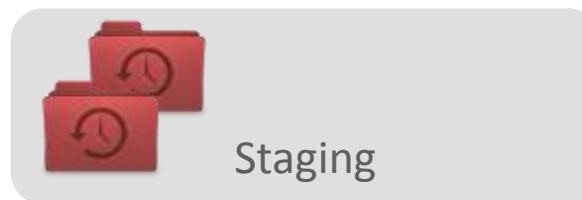


# Phase 2 - Prepare phase

- Transfer files to **destination** host
  - Not necessary if your staging location is available to the destination host (NFS etc)
  - `xtt.properties: dfcopydir = stageondest`

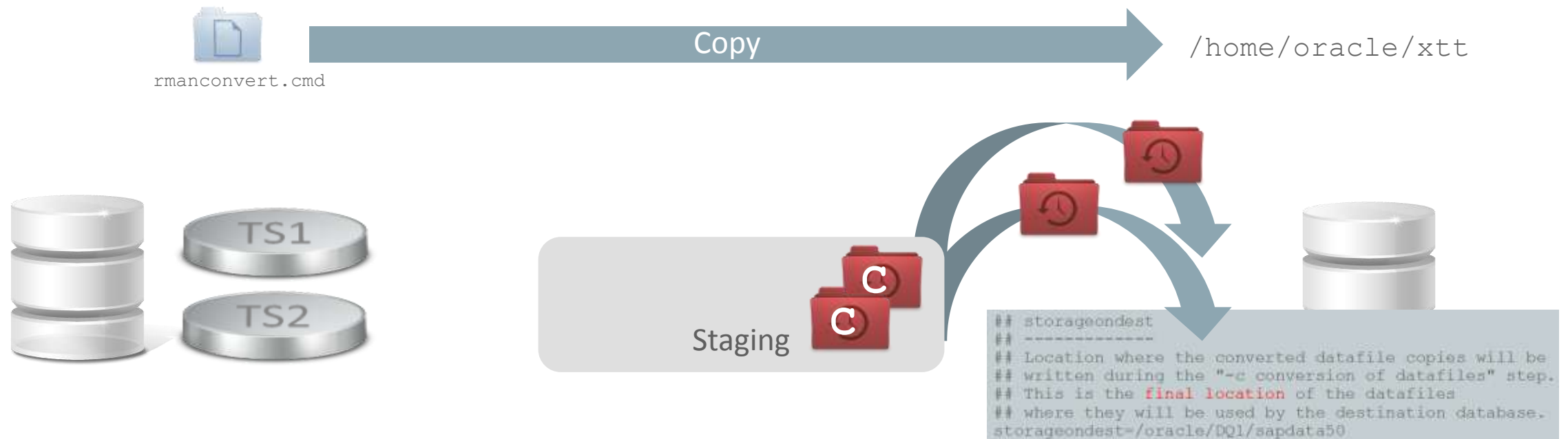
```
## Source system file locations
## =====
## Location where datafile copies are created
## during the "-p prepare" step.
dfcopydir=/oracle/DQ1/rman_stage
```

```
## Destination system file locations
## =====
## Location where datafile copies are placed by the user
## when they are transferred manually from source system.
stageondest=/oracle/DQ1/rman_stage
```



## Phase 2 - Prepare phase

- Copy `rmanconvert.cmd` to destination
- Convert the data file copies and write them to `storageondest`
  - `[oracle@dest]$ $ORACLE_HOME/perl/bin/perl xttdriver.pl -c`



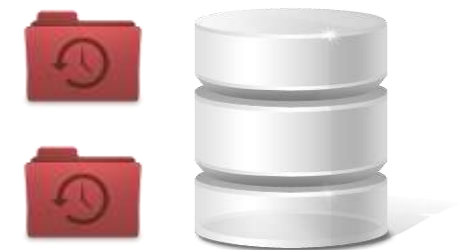
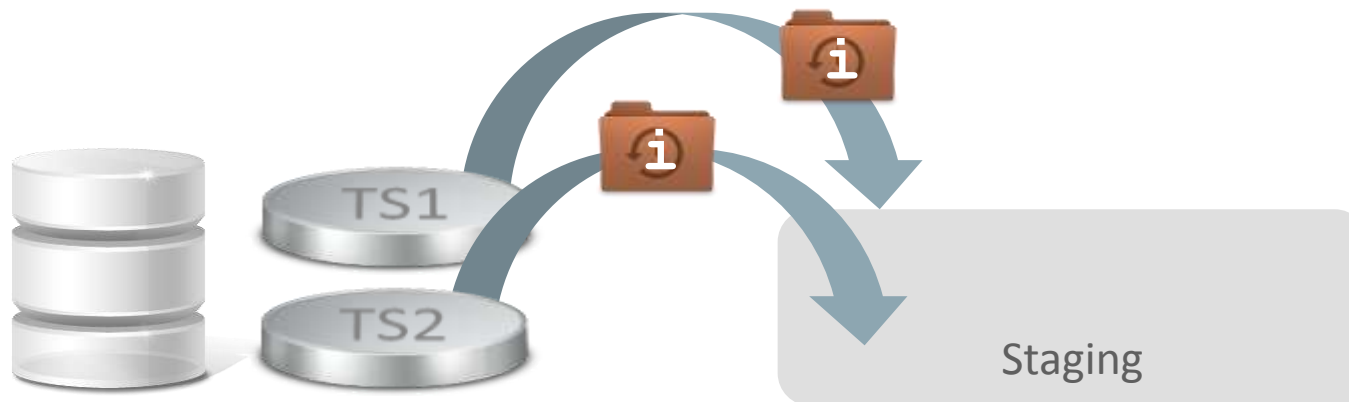
# Phase 3 - Roll Forward phase

- Create incremental backups on **source**

– [oracle@**source**] \$ \$ORACLE\_HOME/perl/bin/perl **xttdriver.pl -i**

– Creates the following files used later:

- tsbkupmap.txt
- incrbackups.txt [not necessary here due to NFS mount]

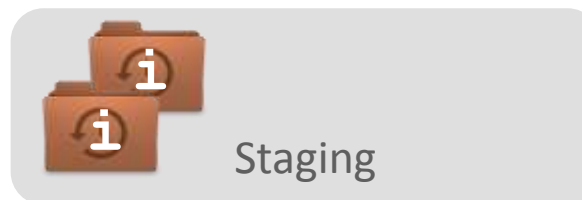


# Phase 3 - Roll Forward phase

- Transfer incremental backups to **destination** host
  - Not necessary if your staging location is available to the destination host (NFS etc)
  - `xrt.properties: backupformat = stageondest`

```
## backupformat
## -----
## Location where incremental backups are created
backupformat=/oracle/DQ1/rman_stage
```

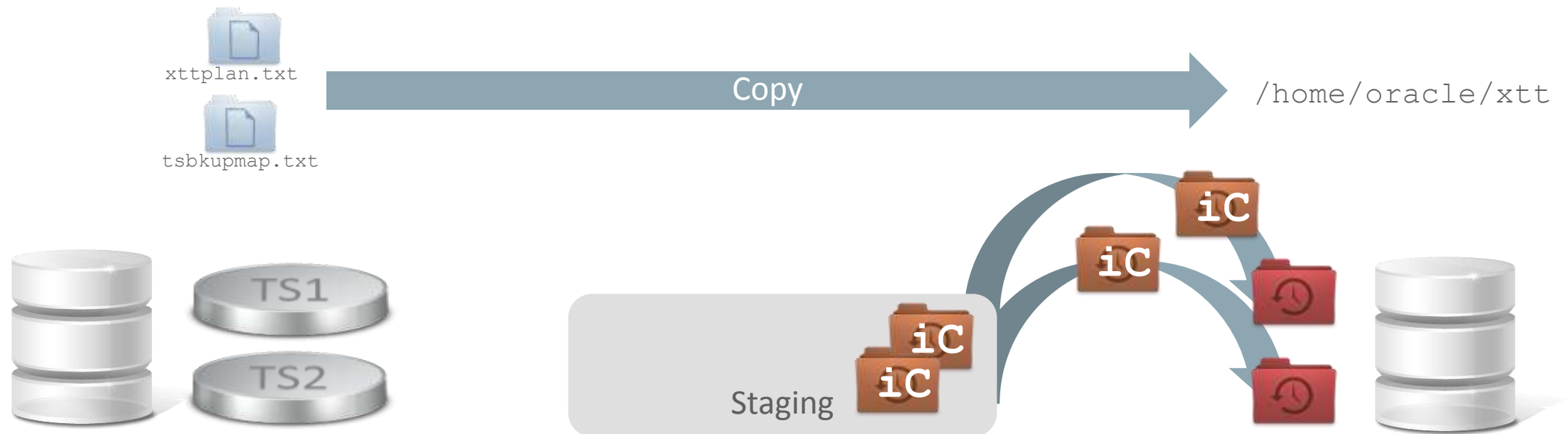
```
## Destination system file locations
## =====
## Location where datafile copies are placed by the user
## when they are transferred manually from source system.
stageondest=/oracle/DQ1/rman_stage
```





# Phase 3 - Roll Forward phase

- Copy `xttpplan.txt` and `tsbkupmap.txt` to destination
- Convert the inc backups and merge them into tablespace files on storage on dest
  - `[oracle@dest]$ $ORACLE_HOME/perl/bin/perl xttdriver.pl -r`



# Phase 3 - Roll Forward phase

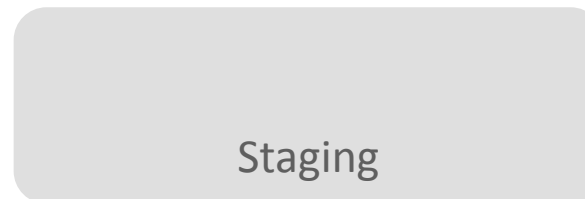
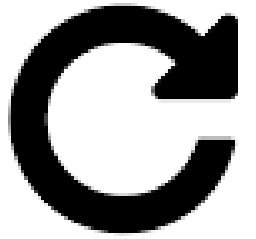
- Record FROM\_SCN on **source** for next incremental backup

- `[oracle@source]$ $ORACLE_HOME/perl/bin/perl xttdriver.pl -s`
- Writes it into `xttplan.txt`



# Phase 3 - Roll Forward phase

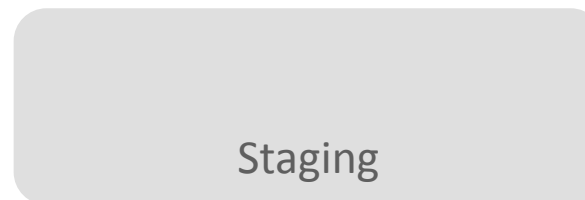
- Repeat entire Phase 3 as often as necessary
  - Increase of frequency will decrease file sizes



# Phase 4 - Final Incremental Backup

- Set tablespaces read/only – **Downtime!**

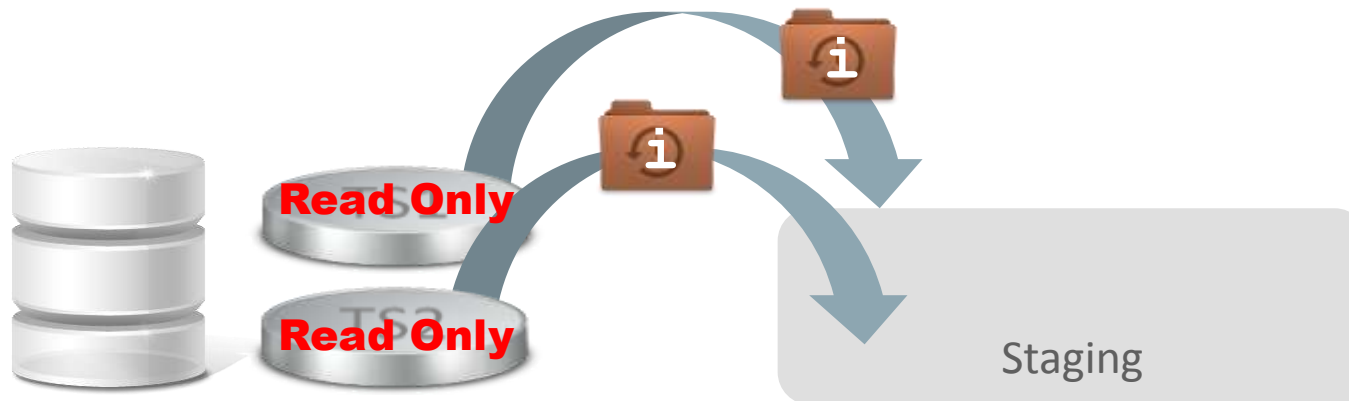
```
SQL:SOURCEDB> alter tablespace TS1 read only;  
SQL:SOURCEDB> alter tablespace TS2 read only;
```



# Phase 4 - Final Incremental Backup

- Create **final** incremental backup on **source**

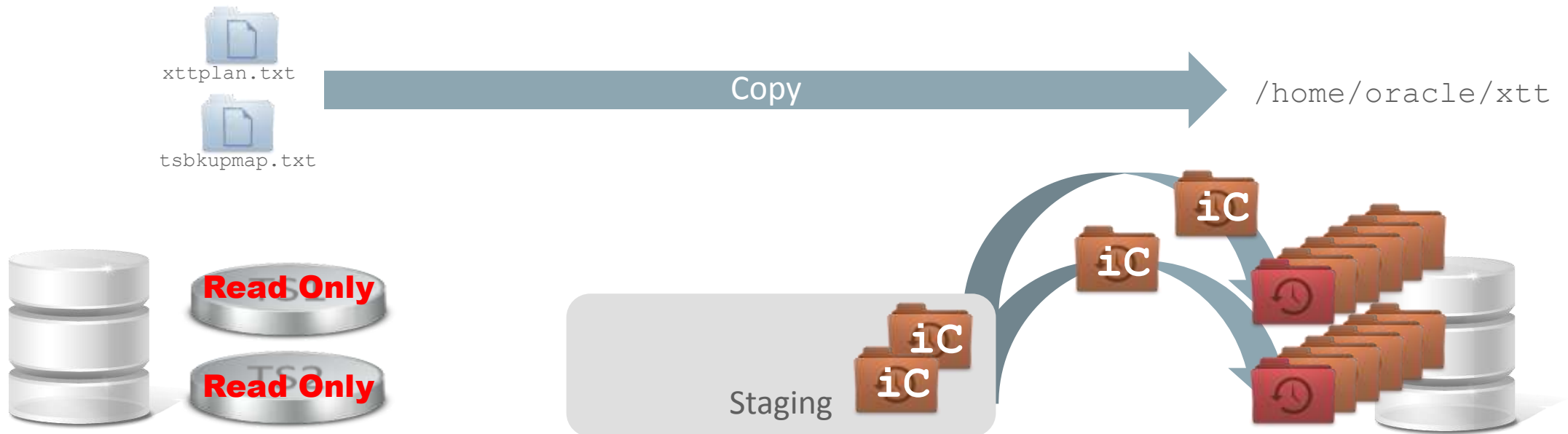
```
[oracle@source]$ $ORACLE_HOME/perl/bin/perl xttdriver.pl -i
```



# Phase 4 - Final Incremental Backup

- Copy `xttpplan.txt` and `tsbkupmap.txt` to **destination**
- Convert **final** inc backups and merge them into tablespace files

```
[oracle@dest]$ $ORACLE_HOME/perl/bin/perl xttddriver.pl -r
```





# Phase 5 - Transport Phase: Import all Metadata

- Prepare destination database for **Full Transportable Export/Import**

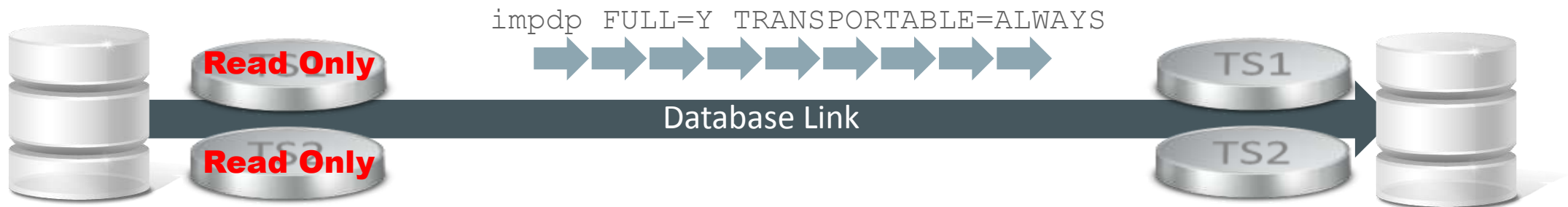
- SQL:DESTDB> CREATE DIRECTORY ftex\_dir AS '/home/oracle/dp';
- SQL:DESTDB> GRANT READ, WRITE ON DIRECTORY ftex\_dir TO mike;
- SQL:DESTDB> CREATE PUBLIC DATABASE LINK v112 USING 'v112';



# Phase 5 - Transport Phase: Import all Metadata

- Start **Full Transportable Export/Import**

```
[oracle@dest]$ impdp mike/passwd@v121 NETWORK_LINK=v112  
FULL=Y TRANSPORTABLE=ALWAYS [VERSION=12]  
METRICS=Y EXCLUDE=STATISTICS  
LOGTIME=ALL LOGFILE=ftex_dir:v112fullimp.log  
TRANSPORT_DATAFILES='/oracle/DQ1/sapdata50/ts1.dbf'  
TRANSPORT_DATAFILES='/oracle/DQ1/sapdata50/ts2.dbf'
```



# Phase 6 - Validate the Transported Tablespaces

- Validate transported tablespaces

- RMAN> validate tablespace TS1, TS2 check logical;



# Phase 7 - Cleanup

- Set tablespaces on **source** read/write
- Cleanup all files created for this process
- Cleanup staging area if not done already



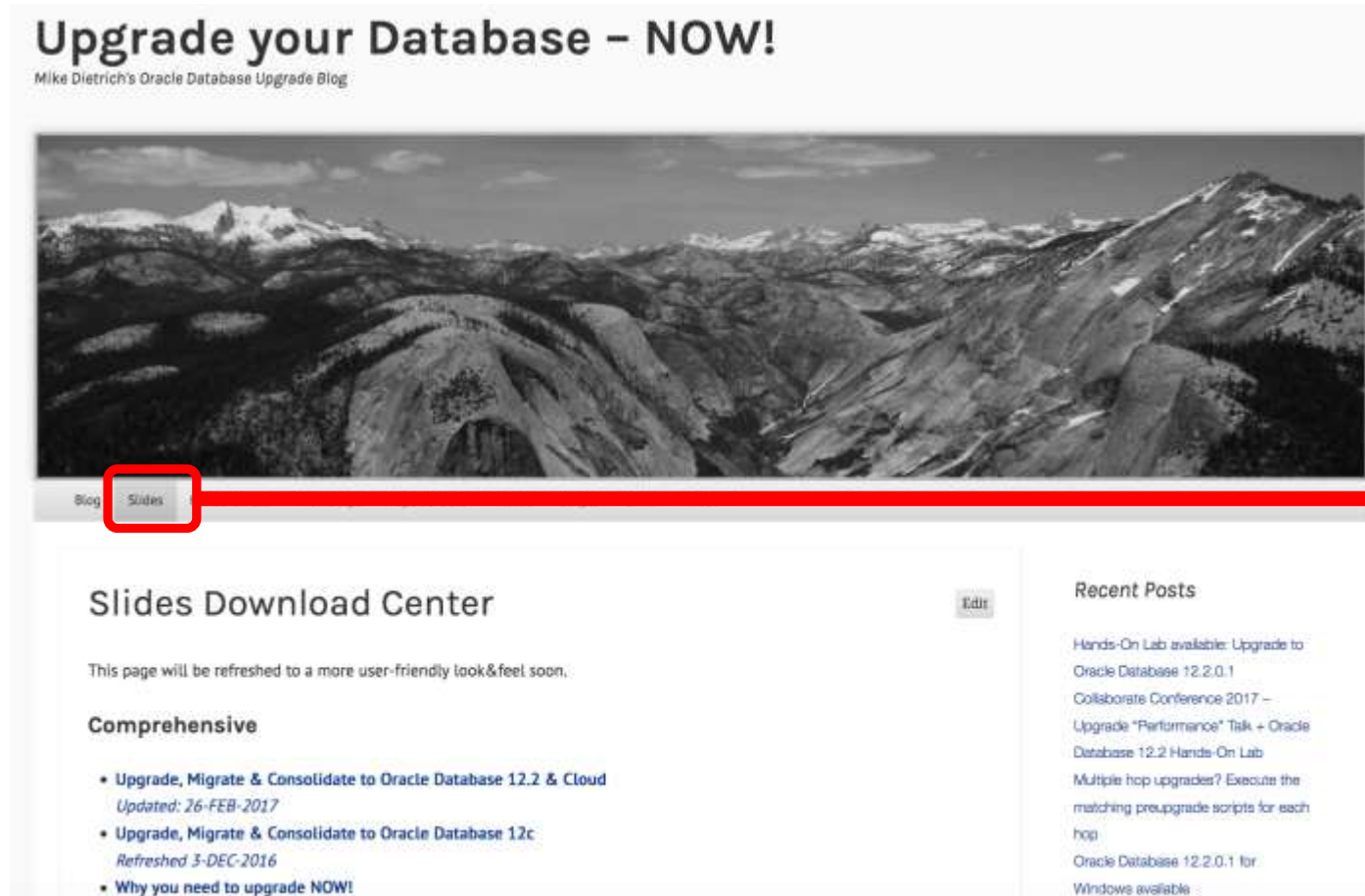
A photograph of three FC Barcelona players celebrating a goal. They are wearing their blue and red home kit with 'QATAR AIRWAYS' on the front. The player in the center is Neymar, and the player on the right is Lionel Messi. They are all running with their arms outstretched and mouths open in celebration. The background shows a stadium with spectators and a goal net.

# Yes, we made it 😊

**It looks a bit complicated at first sight but once you tried it, it may be the game changer**

# Database Upgrade Blog

- <https://MikeDietrichDE.com>





# Integrated Cloud

## Applications & Platform Services

ORACLE®