

The Oracle logo, consisting of the word "ORACLE" in a red, sans-serif, uppercase font. It is positioned in the upper left corner of the slide, partially overlaid by a decorative graphic of orange and red 'x' marks and a dark blue wavy shape.

ORACLE

Photo by [jean_wimmerlin](#) on [Unsplash](#)

Migrating the Beast

MakelT, May 2025

Oracle

DBAs

run the world





DANIEL OVERBY HANSEN

Distinguished Product Manager
Database Upgrade, Migrations & Patching



dohdatabase



@dohdatabase.com



<https://dohdatabase.com>

Find Slides and Much More on Our Blogs



MikeDietrichDE.com

Mike.Dietrich@oracle.com



dohdatabase.com

Daniel.Overby.Hansen@oracle.com



DBArj.com.br

Rodrigo.R.Jorge@oracle.com



AlexZaballa.com

Alex.Zaballa@oracle.com

Web Seminar

Episode 16

(replaces Episode 1 from Feb 2021)

Oracle Database Release and Patching Strategy for 19c and 23c

115 minutes – May 10, 2023

Slides



Episode 17

From SR to Patch – Insights into the Oracle Database Development process

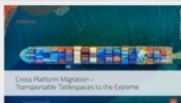
55 minutes – June 22, 2023



NEW Episode 18

Cross Platform Migration – Transportable Tablespaces to the Extreme

145 min – February 22, 2024



Episode 2

AutoUpgrade to Oracle Database 19c

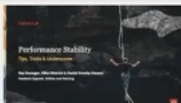
115 minutes – Feb 20, 2021



Episode 3

Performance Stability, Tips and Tricks and Underscores

120 minutes – Mar 4, 2021



Episode 4

Migration to Oracle Multitenant



Recorded Web Seminars

<https://MikeDietrichDE.com/videos>

More than 35 hours of technical content,
on-demand, anytime, anywhere





ANDREAS GROETZ

Oracle DBA Tech Lead
Entain Services Austria GmbH

Entain is one of the world's largest sports betting and gaming groups. Leveraging the power of the Entain Platform, they bring moments of excitement into their customers lives through more than 30 iconic brands such as bwin, Coral, Ladbrokes and many more.

Entain operates on over 140 licenses across 40+ territories and employs over 29,000 talented workforce. Entain is listed on the London Stock Exchange and is a constituent of the FTSE 100 Index.

Σntain



Ladbrokes



BET**MGM**

sportingbet

CORAL 
GET CLOSER TO THE ACTION

bwin 

EUROBET 

SuperSport

party p♦ker

Foxy BINGO

Challenges

What is special, what makes it so complex?

Migration Challenges



SPARC SuperCluster



ZDLRA



Exadata X9M Extreme Flash

Migration Challenges

180TB
size



SPARC SuperCluster



ZDLRA



Exadata X9M Extreme Flash

Migration Challenges

15TB
redo/day



SPARC SuperCluster



ZDLRA



Exadata X9M Extreme Flash

Migration Challenges



SPARC SuperCluster



Exadata X9M Extreme Flash



5 Physical Standby DBs

Local, and in different region, 2500km away

Constraints



Limiting factors, and other things to know



Photo by Mihaly Koles on Unsplash

Up to 15TB redo/day is beyond what Oracle GoldenGate will be able to synch

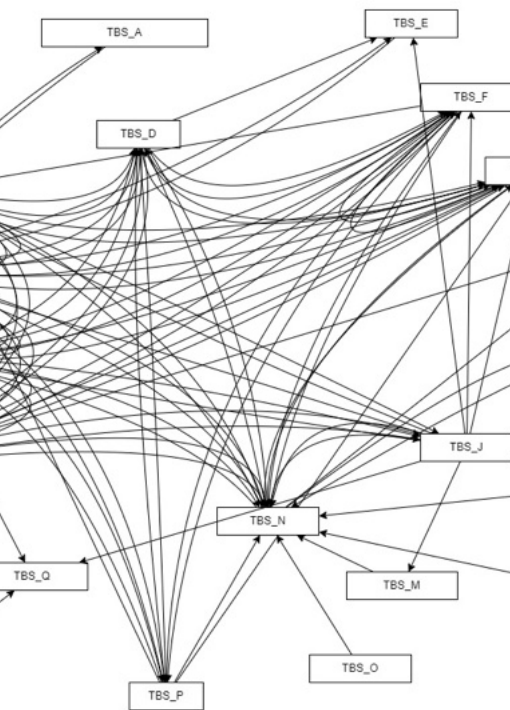
The system is highly active 24 x 7 x 365



Photo by Dave Hoefler on Unsplash

Very large database,
very constrained downtime

- 180+ TB database size
- 5-6 TB growth/month
- Every minute of downtime costs \$\$\$



Migrating tablespaces upfront or separately **definitely** not an option

- Way too many cross-dependencies
- Tablespaces aren't isolated



Photo by Linh Ha on Unsplash

Every complex Oracle data type you can imagine is used

- XML binary types
- Nested partitioned tables
- Evolved object types



Photo by [Masaki Komori](#) on [Unsplash](#)

Tight downtime window

- Dry run: 2 hours outage approved
 - Tablespace read-only
 - Full Transportable Export
- Live migration: 13 hours approved



Photo by Moreanella Breton on Unsplash

The available **Oracle V4 PERL** migration scripts would have been worked technically, but ...

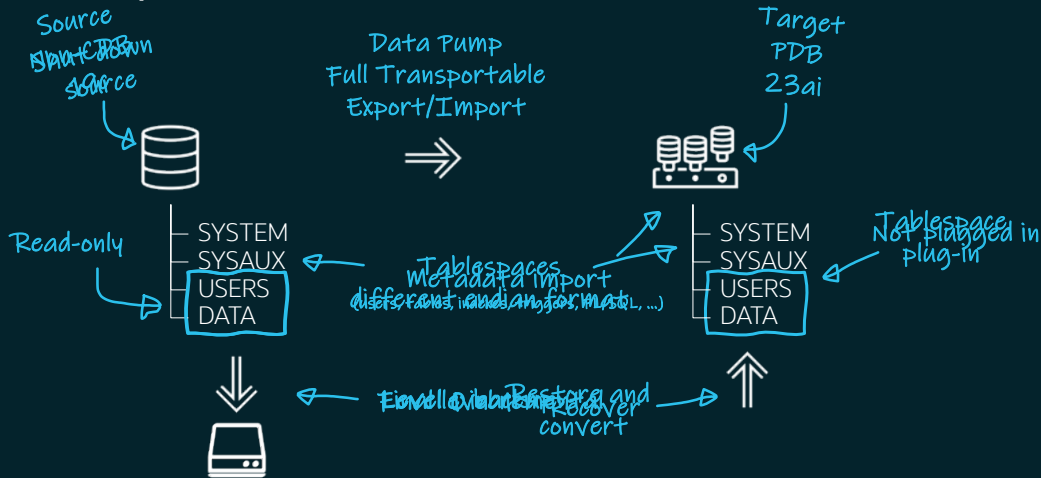
- No section-size backup support
- No standby backup support
- No selective PDB migration support

Migration



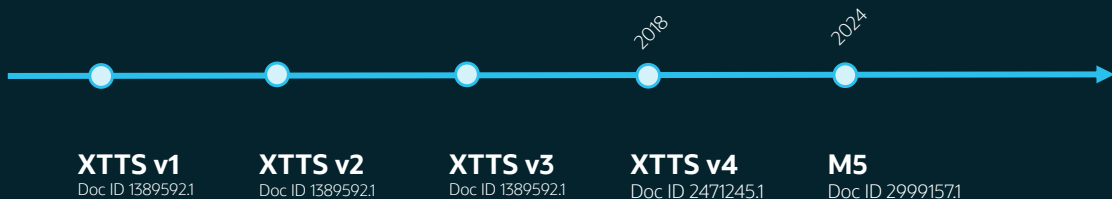
Typically, we use **Full Transportable Export/Import** for large cross-endian migrations

Concept



Scripts for Incremental Backup Automation

Backup / restore / recover



- No multisection backups
- No encrypted tablespaces
- Inefficient parallelism
- Incomplete multitenant support

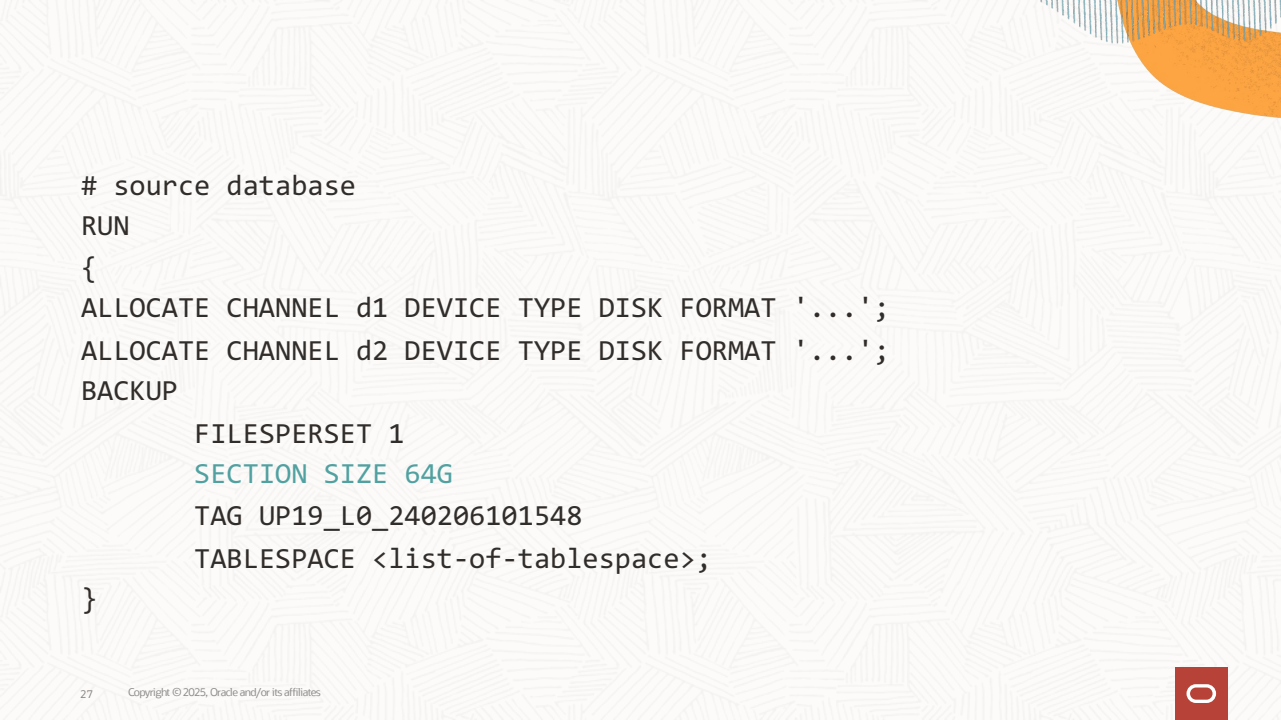


M5 is the **next-generation** cross-platform transportable tablespace procedure

- New RMAN functionality combined with Full Transportable Export/Import
- Doc ID [29991571](#)

M5 Migration Script

The new migrations scripts superseding the V4 PERL scripts



```
# source database
RUN
{
  ALLOCATE CHANNEL d1 DEVICE TYPE DISK FORMAT '...';
  ALLOCATE CHANNEL d2 DEVICE TYPE DISK FORMAT '...';
  BACKUP
    FILESPERSET 1
    SECTION SIZE 64G
    TAG UP19_L0_240206101548
    TABLESPACE <list-of-tablespace>;
}
```

```
# source database
RUN
{
  ALLOCATE CHANNEL d1 DEVICE TYPE DISK FORMAT '...';
  ALLOCATE CHANNEL d2 DEVICE TYPE DISK FORMAT '...';
  BACKUP
    FILESPERSET 1
    SECTION SIZE 64G
    TAG UP19_L0_240206101548
    TABLESPACE <list-of-tablespace>;
}
```

```
# target database
RUN
{
  ALLOCATE CHANNEL DISK1 DEVICE TYPE DISK FORMAT '...';
  ALLOCATE CHANNEL DISK2 DEVICE TYPE DISK FORMAT '...';
  RESTORE ALL FOREIGN DATAFILES TO NEW FROM BACKUPSET
    '<backup-set-1>',
    '<backup-set-2>',
    ...
    '<backup-set-n>'
};
```


Benefits

M5 procedure supports:

- Encrypted tablespaces
- Multisection backups
- Migrating multiple databases into the same CDB simultaneously
- Compressed backup sets
- Better parallelism

Requirements

- Source and target database must
 - be 19.18.0 or higher
 - use Data Pump Bundle Patch



Always use the latest version of M5 script

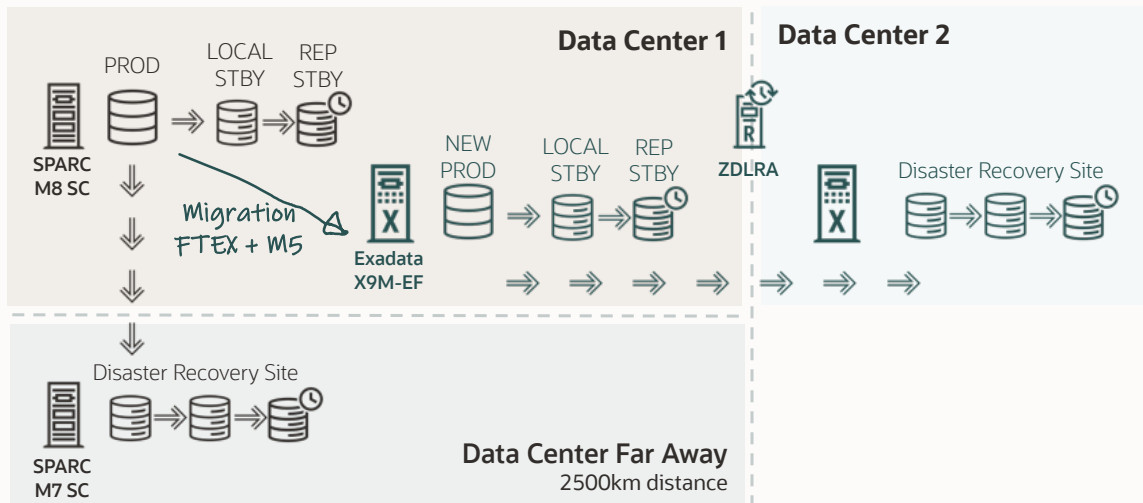
- Download from Doc ID [29991571](#)



Use Block Change Tracking for faster incremental backups

- Check the License Guide for details

Migration Plan



M5 Workflow

Configure

Level 0

Level 1

Outage

Final
Backup

Final
restore

Export
Import



SPARC
SuperCluster



Local standby

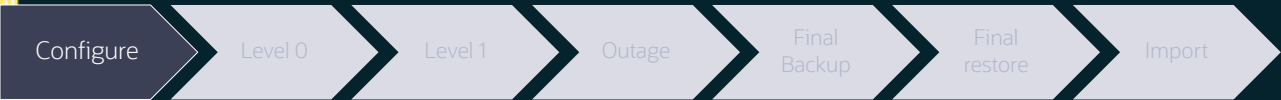


ZDLRA



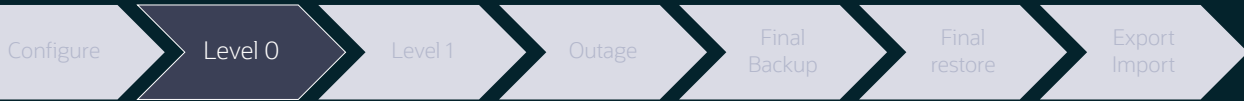
Exadata
X9M Extreme Flash

M5 Workflow



- Download M5 script from Doc ID [2999157.1](#)
- Configure shared NFS
- Edit `dbmig_ts_list.txt`
- Edit `dbmig_driver.properties`
- Create new, empty target database

M5 Workflow



SPARC
SuperCluster

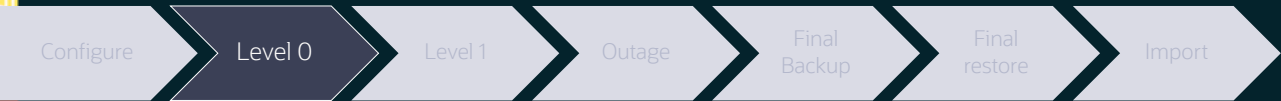


ZDLRA



Exadata
X9M Extreme Flash

M5 Workflow



- Start initial level 0 backup
 - Use driver script `dbmig_driver_m5.sh` `L0`
- Driver script creates a restore script
 - Restore using `restore_L0_<source_sid>_<timestamp>.cmd`
- Check logs

M5 Workflow

Configure

Level 0

Level 1

Outage

Final
Backup

Final
restore

Export
Import



SPARC
SuperCluster

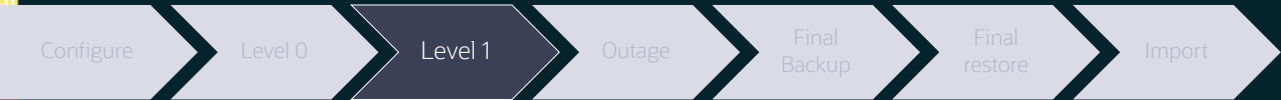


ZDLRA



Exadata
X9M Extreme Flash

M5 Workflow



- Start level 1 incremental backup
 - Use driver script `dbmig_driver_m5.sh` [L1](#)
- Driver script creates a restore script
 - Restore using `restore_L1_<source_sid>_<timestamp>.cmd`
- Check logs
- Repeat as often as desired

M5 Workflow

Configure

Level 0

Level 1

Outage

Final
Backup

Final
restore

Export
Import



SPARC
SuperCluster

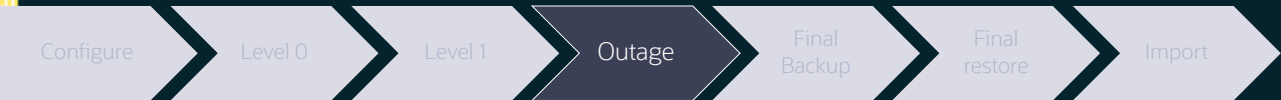


ZDLRA



Exadata
X9M Extreme Flash

M5 Workflow



- Maintenance window begins
- Read-only sessions can still use the database

M5 Workflow

Configure

Level 0

Level 1

Outage

Final
Backup

Final
restore

Export
Import



SPARC
SuperCluster

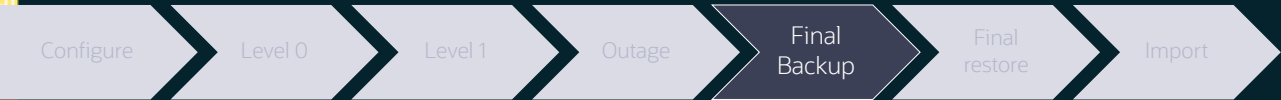


ZDLRA



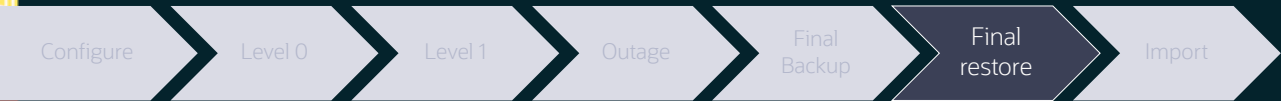
Exadata
X9M Extreme Flash

M5 Workflow



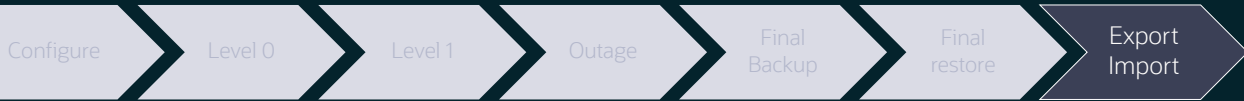
- Start final level 1 incremental backup
 - Use driver script `dbmig_driver_m5.sh` [L1F](#)
 - Sets tablespaces read-only
 - Performs level 1 incremental backup
 - Starts Data Pump full transportable export
- Optionally, shuts down source database

M5 Workflow



- Driver script created a restore script
 - Restore using `restore_L1F_<source_sid>_<timestamp>.cmd`
- Check logs

M5 Workflow



SPARC
SuperCluster

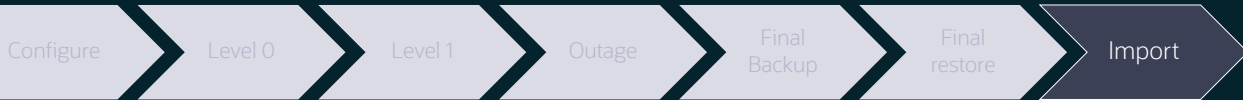


ZDLRA



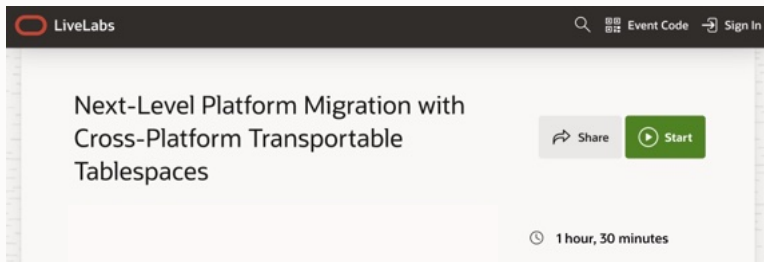
Exadata
X9M Extreme Flash

M5 Workflow



- Copy Data Pump dump file to *DATA_PUMP_DIR*
- Use import driver script in test mode
 - Start `impdp.sh <dump_file> <restore_log> test`
- Check generated parameter file
 - Use `impdp.sh <dump_file> <restore_log> run`
- Check Data Pump log file

Wanna try it out?



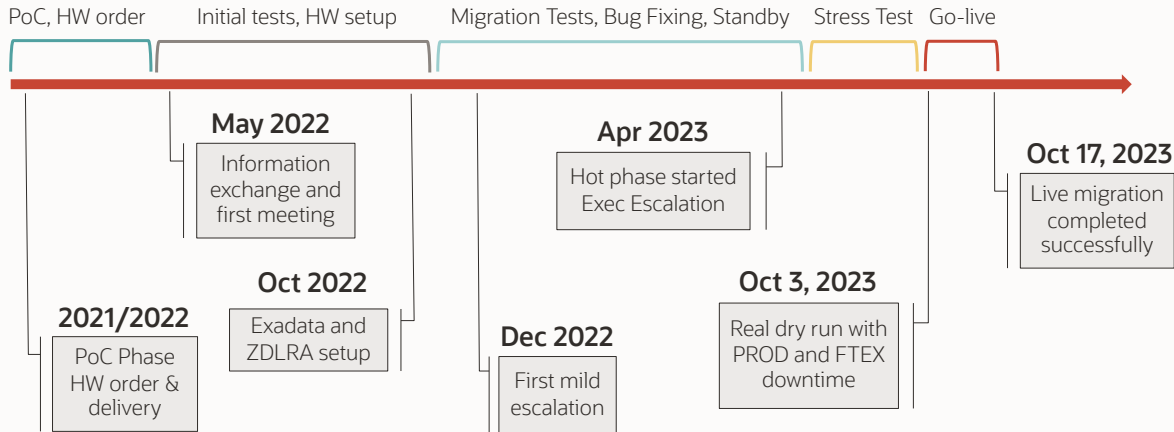
Oracle LiveLabs – Run the lab just inside your browser!

The Project Plan



Timelines and the Run Book

Overall Project Timeline



Key to Success: Runbook

Complex projects absolutely require a **detailed runbook**

ID	Task	Status	Responsible Primary Person	Responsible Secondary Person	Predecessor	Start Time (CEST)	Duration (hh:mm)	End Time (CEST)	Start Time (IST)	End Time (IST)	Actual Start Time (CEST)	Actual Duration	Actual End Time (CEST)	Comments - Blocker
----	------	--------	----------------------------------	------------------------------------	-------------	----------------------	---------------------	--------------------	------------------------	-------------------	--------------------------------	--------------------	------------------------------	-----------------------

- This run book covered over 200 individual tasks

A screenshot of a complex project runbook table. The table has many columns, including ID, Task, Status, Responsible Primary/Secondary Person, Predecessor, Start Time (CEST), Duration (hh:mm), End Time (CEST), Start Time (IST), End Time (IST), Actual Start Time (CEST), Actual Duration, Actual End Time (CEST), and Comments - Blocker. The rows are color-coded in green, blue, and yellow, representing different task statuses or categories. The table is densely packed with text and numbers, showing a high level of detail for project management.

51 Copyright © 2025, Oracle and/or its affiliates

How many people were involved?



6 DBAs



Managers



2 data center
engineers



4 product
engineering



4 DB developer



War Room



2-3 infrastructure



2 monitoring,
communication



4 compliance



2 network



10 testing

Migration Issues



Some of them...

Where we started ...



PoC first FTEX **export**:

01-OCT-21 05:32:36.275: Job "SYSTEM"."SYS_EXPORT_FULL_01" successfully completed at
Fri Oct 1 05:32:36 2021 elapsed **0 04:25:22**

PoC first FTEX **import**:

05-OCT-21 01:48:59.534: Job "SYSTEM"."SYS_IMPORT_FULL_01" completed with **103000**
error(s) at Tue Oct 5 01:48:59 2021 elapsed **3 18:34:09**



The way forward ...

80 SRs opened and solved in various areas

Migration
ZDLRA
Standby
Infrastructure

18 one-off patches

5 merges

Daily calls with Oracle, countless evening / night / weekend hours

Many areas required special attention....

- Optimizer Statistics
- Scheduler jobs
- Resource Manager
- Cross Schema objects
- AQ
- Evolved Types/partitioned nested tables
- Binary XML
- Standby DBs
- ...



Issue 1 | Long Running Metadata Import

Fix applied to remedy export errors

- BUG 34201281 - MERGE ON DATABASE RU 19.12.0.0.0 OF 33963454 34052641

Result:

- Now Full Transportable import alone took over 6 days (!!)

```
08-JUN-22 16:21:17.887: W-1 Processing object type DATABASE_EXPORT/.../PROCACT_INSTANCE
14-JUN-22 18:56:58.813: W-1      Completed 108 PROCACT_INSTANCE objects in 527737 seconds
...
14-JUN-22 19:15:50.016: Job "SYSTEM"."SYS_IMPORT_TRANSPORTABLE_01" completed with 316 error(s)
                        at Tue Jun 14 19:15:49 2022 elapsed 6 06:45:38
```

Issue 1 | Long Running Meta Import

Long running action identified via tracing:

```
UPDATE "POSTMAN"."T_MAIL_LOG"  
SET "C_CVAR"=SYS_REMAP_XMLTYPE("C_CVAR")
```

- 300+ million rows

Issue in internal package DBMS_CSX_INT

- Fast merge of XMLTYPE is not happening as expected
 - *Reason:* Incorrect internal check query
- Tokens between source and target are not identical
 - *Reason:* Different Endianness

Issue 1 | Long Running Meta Import

Solution:

- Use workaround from MOS Note: 2309649.1 in UPGRADE mode
 - [MOS Note: 2309649.1 - How to Migrate Large Amount of Binary XML Data between Databases](#)

```
25-JUL-22 12:28:40.813: Job "SYSTEM"."SYS_IMPORT_TRANSPORTABLE_01" completed  
with 317 error(s) at Mon Jul 25 12:28:40 2022 elapsed 0 04:33:03
```


Issue 2 | Metadata API and Nested Tables

Full transportable import errors out for a **nested partitioned table**

```
PLS-00172: string literal too long
```

```
ORA-39151: Table "DBA_XY"."X_GAMES" exists.
```

```
All dependent metadata and data will be skipped due to table_exists_action
```

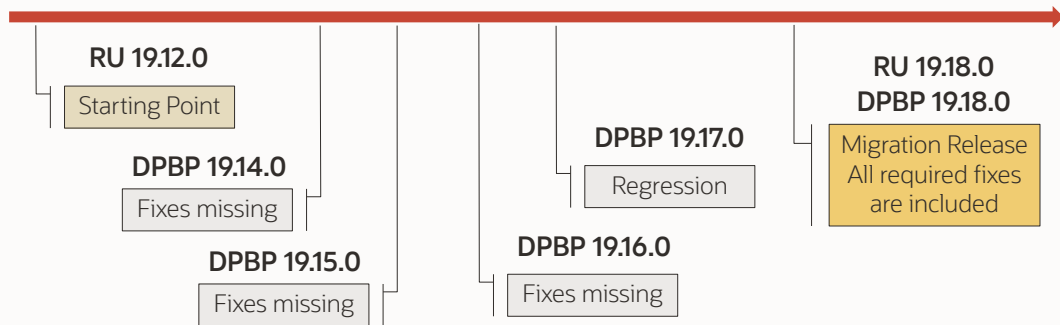
Root cause was a string overflow in the Metadata API

- Data Pump creates the index, and then alters it – here the overflow happened
- Side effect was a misleading error message

Issue 3 | Data Pump Bundle Patch

Many of the TTS and Metadata fixes got included into Data Pump Bundle

- [Data Pump Recommended Proactive Patches For 19.10 and Above \(Doc ID 2819284.1\)](#)



Issue 4 | Evolved Object Types

Evolved TYPES can lead to Data Pump errors during transportable import:

```
ORA-39083: Object type TABLE:"APPUSER"."CARS" failed to create with error:  
ORA-39218: type check on object type "APPUSER"."CAR_TYPE" failed  
ORA-39216: object type "APPUSER"."CAR_TYPE" hashcode or version number mismatch
```

Further Information:

- [Blog Post: Understand why Data Pump errors with evolved types](#)



Using evolved types in table definitions

--Create a new type. The type is now version 1

--Use the type in a table

```
CREATE TYPE CAR_INFO_TYPE IS OBJECT (model VARCHAR2(40));
```

```
CREATE TABLE CARS (id number, car_info car_info_type);
```

```
INSERT INTO CARS VALUES (1, car_info_type('Volvo V90'));
```

The type is now evolving



--Make a change to the type. The type is now version 2

```
ALTER TYPE CAR_INFO_TYPE ADD ATTRIBUTE horsepower NUMBER CASCADE NOT INCLUDING TABLE DATA;
```

```
INSERT INTO CARS VALUES (2, car_info_type('BMW 530i', 250));
```

Existing data is not updated



--Make another change to the type. The type is now version 3

```
ALTER TYPE CAR_INFO_TYPE ADD ATTRIBUTE color VARCHAR2(20) CASCADE NOT INCLUDING TABLE DATA;
```

```
INSERT INTO CARS VALUES (3, car_info_type('Hyundai Sonata', 160, 'Black'));
```

Evolved Types

```
SELECT * FROM CARS
```



CARS	
1	car_info_type v1: Volvo V90
2	car_info_type v2: BMW 530i, 250
3	car_info_type v3: Hyundai Sonata, 160, Black



DICTIONARY	
car_info_type v1	model
car_info_type v2	model, horsepower
car_info_type v3	model, horsepower, color



Data Pump recreates types during
Full Transportable Export/Import

Evolved Types

- To avoid **data corruption**,
Data Pump must recreate the exact same type evolution in target database
- Due to **implementation restrictions**,
it is not always possible to recreate the exact same type evolution
- In such situations, to avoid corruption,
Data Pump reports ORA-39218 or ORA-39216 on **import**



Evolved Types | Possible Solutions

- 1 Conventional Data Pump export
- 2 Manually recreate `type` in target database with matching evolution
- 3 Recreate `type` without evolution before export

[Blog post](#) with details

Issue 5 | Advanced Queueing

Source database

`<queue_table_name>`
`AQ$_<queue_table_name>_E`
`AQ$_<queue_table_name>_I`
`AQ$_<queue_table_name>_T`
`AQ$_<queue_table_name>_F`
`AQ$_<queue_table_name>_C`
`AQ$_<queue_table_name>_D`
`AQ$_<queue_table_name>_G`
`AQ$_<queue_table_name>_H`
`AQ$_<queue_table_name>_L`
`AQ$_<queue_table_name>_P`
`AQ$_<queue_table_name>_S`
`AQ$_<queue_table_name>_V`

Queue table

Queue
infrastructure

Target database

`<queue_table_name>`
`AQ$_<queue_table_name>_E`
`AQ$_<queue_table_name>_I`
`AQ$_<queue_table_name>_T`
`AQ$_<queue_table_name>_F`

Issue 5 | Advanced Queueing

Queue tables and underlying objects may change during import

- COMPATIBLE **during creation** of queue tables matters
- COMPATIBLE **during import** matters as well
- [MOS Note: 2291530.1 - Understanding How AQ Objects Are Exported And Imported](#)
- [Blog post: Changing data types in queue tables during import](#)

Options:

- Recreate the queue tables with "old" COMPATIBLE setting
- Benefit from new COMPATIBLE setting and test the application



Take into account when comparing source and target databases' object count

- Understanding How Advanced Queueing (AQ) Objects Are Exported And Imported (Doc ID [2291530.1](#))



Data Pump does not start queues

- Manually start queues after migration
- Use `DBMS_AQADM.START_QUEUE`

Issue 6 | Default Tablespaces

Due to a security fix export wants to write into the user's default tablespace

- Bug 27692190
- But default tablespaces are read-only while full transportable export runs

Workaround:

- Change default tablespace for all users to SYSTEM
- Full Transportable Export
- Full Transportable Import
- Revert default tablespaces back to original in target (and source)

Issue 7 | Exporting Statistics

Exporting statistics is slow using
`DBMS_STATS.EXPORT_SCHEMA_STATS`

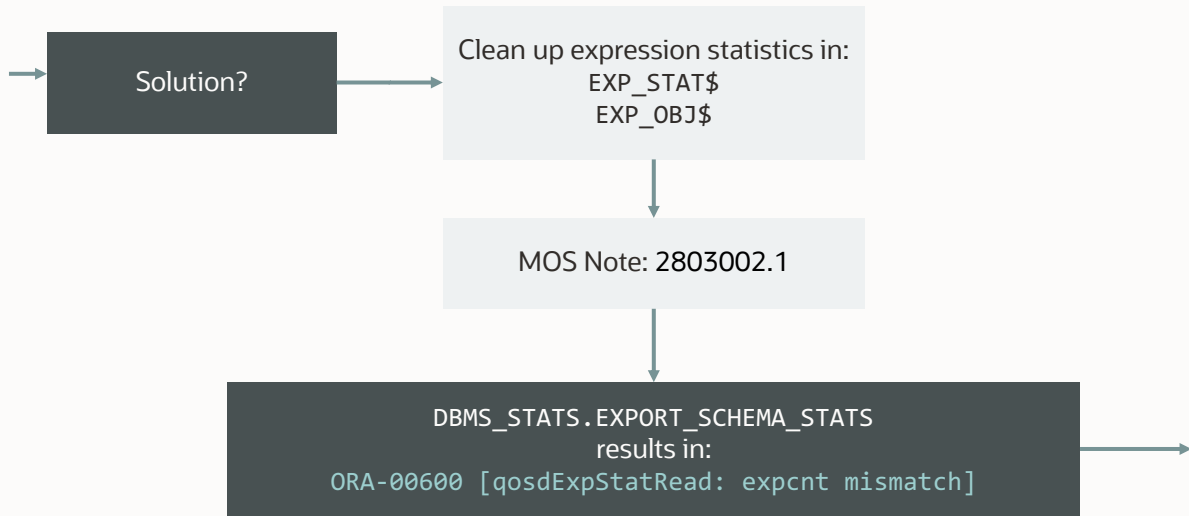


10046 trace reveals long runtime on:
`EXP_STAT$`
`EXP_OBJ$`

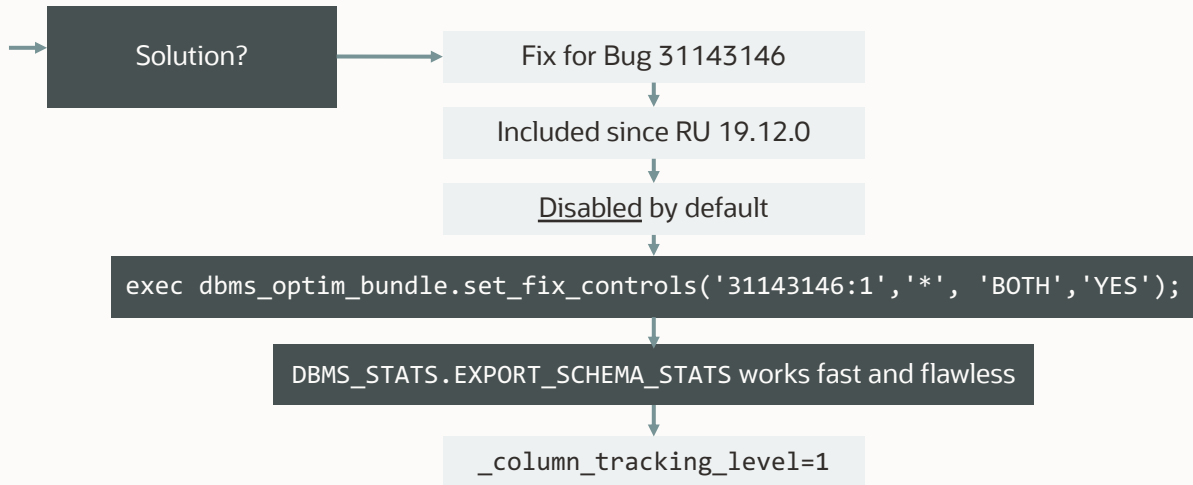
Expression Statistics for Auto-Indexing

Default: `_column_tracking_level=53;`

Issue 7 | Exporting Statistics



Issue 7 | Exporting Statistics



Issue 8 | Auditing

Tablespaces containing auditing tables **can't be set read-only**

Data Pump always unloads the audit records into the dump file

- Huge audit trail will lead to a huge dump file and longer outage

Options:

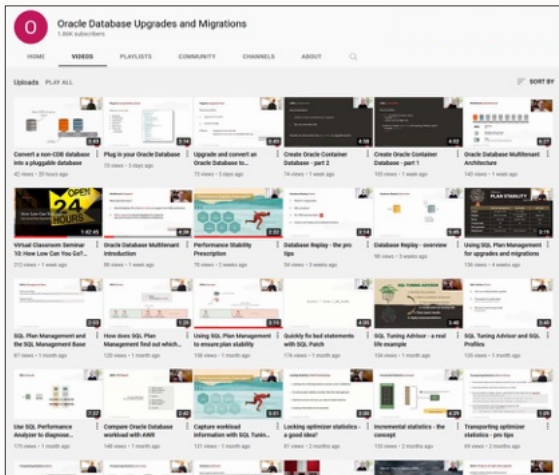
- Export audit records, and eventually import them afterwards
- Archive audit records, purge the audit trail

Key Learnings



- 1 Practice, practice, practice
- 2 Create a detailed runbook
- 3 Work together

YouTube | Oracle Database Upgrades and Migrations



<https://www.youtube.com/@upgradenow>

- 300+ videos
- New videos every week
- No marketing
- No buzzword
- All tech



Thank You

