Data Pump Extreme – Deep Dive

Virtual Classroom Series

Photo by Fikri Rasyid on Unsplash






ROY SWONGER
Vice President
Database Upgrade, Utilities & Patching

royfswonger

@royfswonger






MIKE DIETRICH
Distinguished Product Manager
Database Upgrade and Migrations

mikedietrich

@mikedietrichde

https://mikedietrichde.com






DANIEL OVERBY HANSEN
Senior Principal Product Manager
Cloud Migrations

dohdatabase

@dohdatabase

https://dohdatabase.com






RODRIGO JORGE
Senior Principal Product Manager
Database Patching and Upgrade

rodrigoaraujorge

@rodrigojorgedba

https://dbarj.com.br/en






BILL BEAUREGARD
Senior Principal Product Manager
Data Pump and SQL Loader

william-beauregard-3053791






VENKATESH SANGAM
Senior Director
Database Utilities

vsangam

Webinar | Get The Slides

https://MikeDietrichDE.com/slides

105 minutes – Feb 4, 2021

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

120 minutes – Mar 16, 2021

Episode 5

Migration Strategies – Insights, Tips and Secrets

120 minutes – Mar 25, 2021

Episode 6

Move to the Cloud – Not only for techies

115 minutes – Apr 8, 2021

Episode 7

Cool Features – Not only for DBAs

110 minutes – Jan 14, 2021

Episode 8

Database Upgrade Internals – and so much more

110 minutes – Feb 11, 2021

Episode 9

Performance Testing Using the Oracle Cloud for Upgrades and Migrations

90 minutes – May 19, 2021

NEW Episode 10

How Low Can You Go? Minimal Downtime Upgrade Strategies

100 minutes – Oct 26, 2021

Recorded Web Seminars

https://MikeDietrichDE.com/videos

deep dive

DATA PUMP

with development

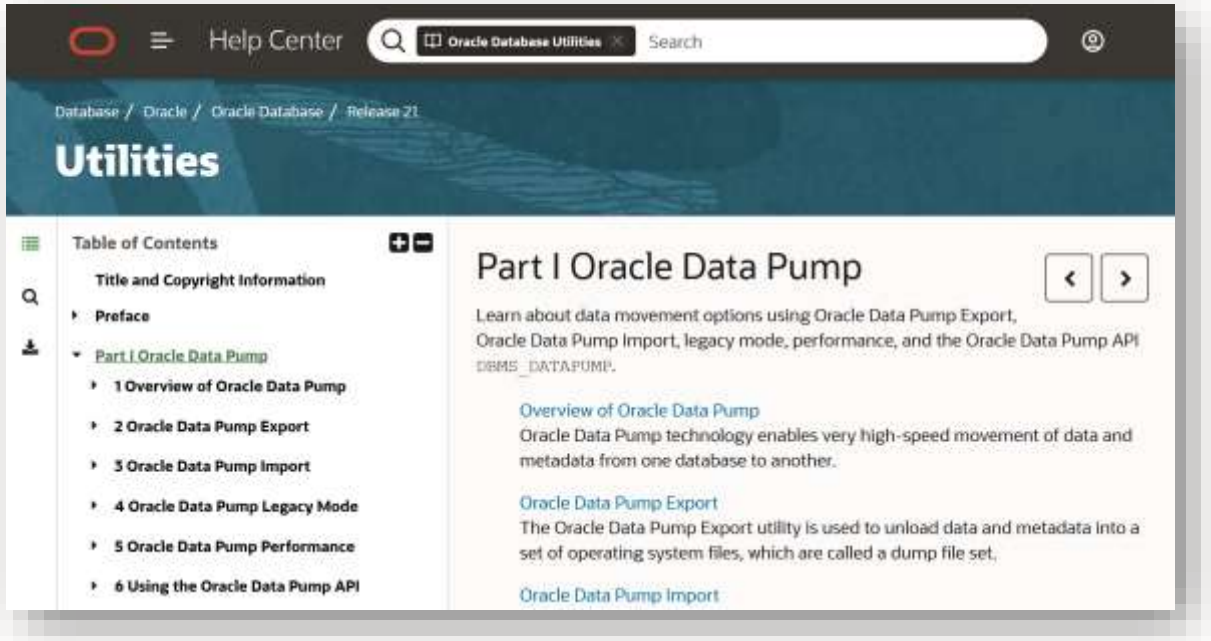
"Oracle Data Pump technology enables very high-speed movement of data and metadata from one database to another."

Oracle Database Utilities 19c

Data Pump | Documentation

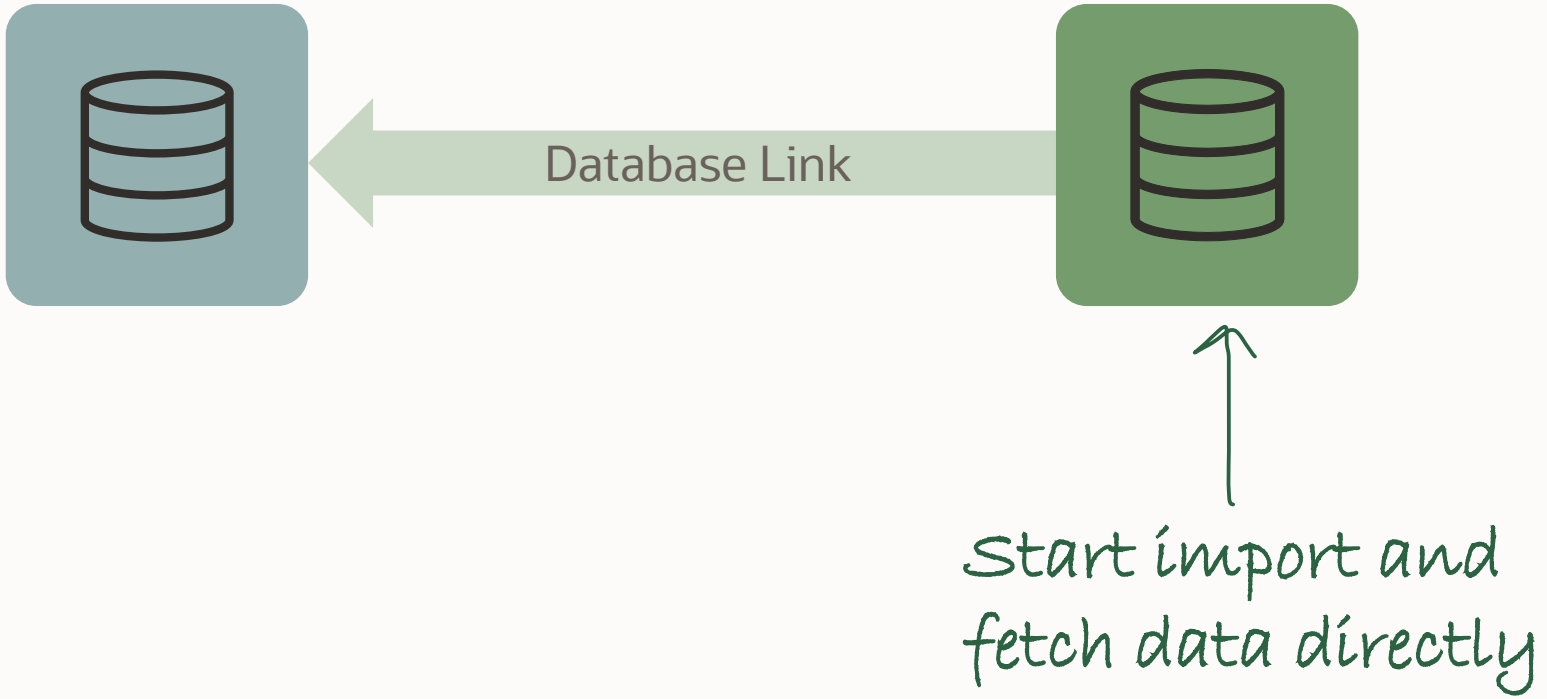
Oracle Database 19c – Utilities Guide

Oracle Database 21c – Utilities Guide



Data Pump | Dump File

Data Pump | Dump File



Data Pump | Mode Comparison

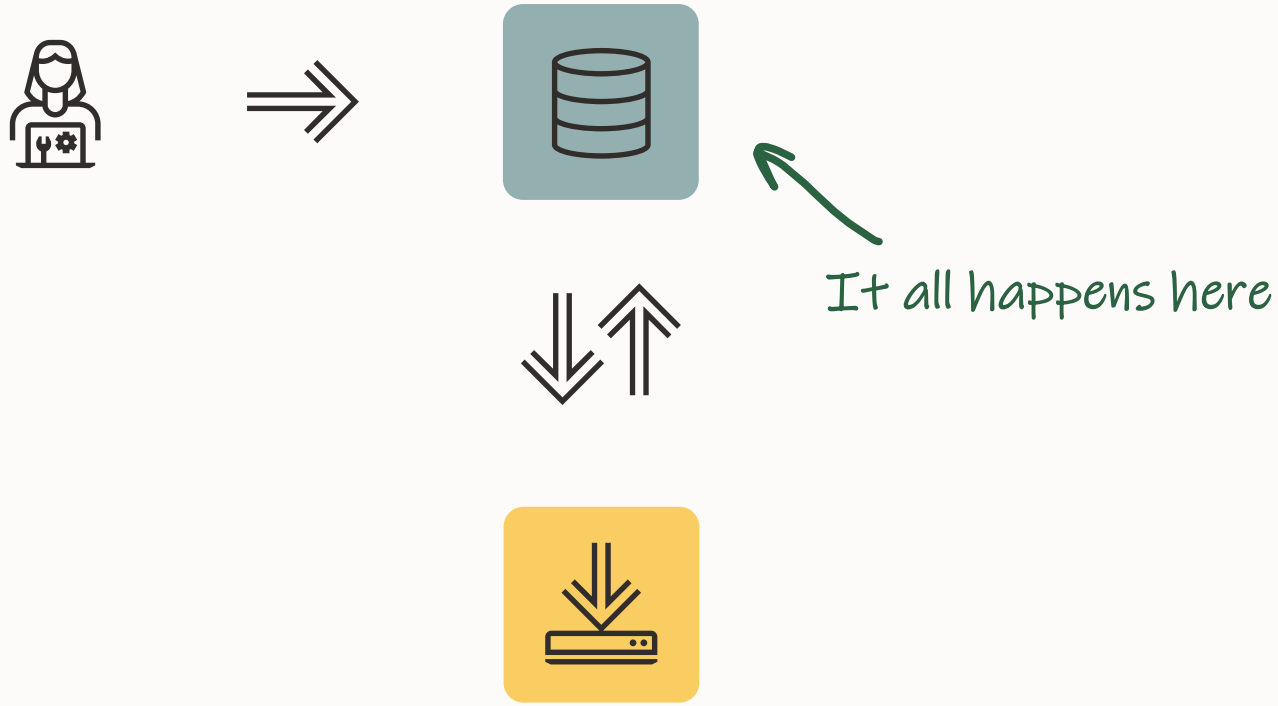
DUMP FILE	NETWORK
Requires access to file system	SQL*Net connectivity
Requires disk space for dump files	No extra disk space needed
Full functionality	Restricted functionality

Pro tip: Read more about how Data Pump moves data

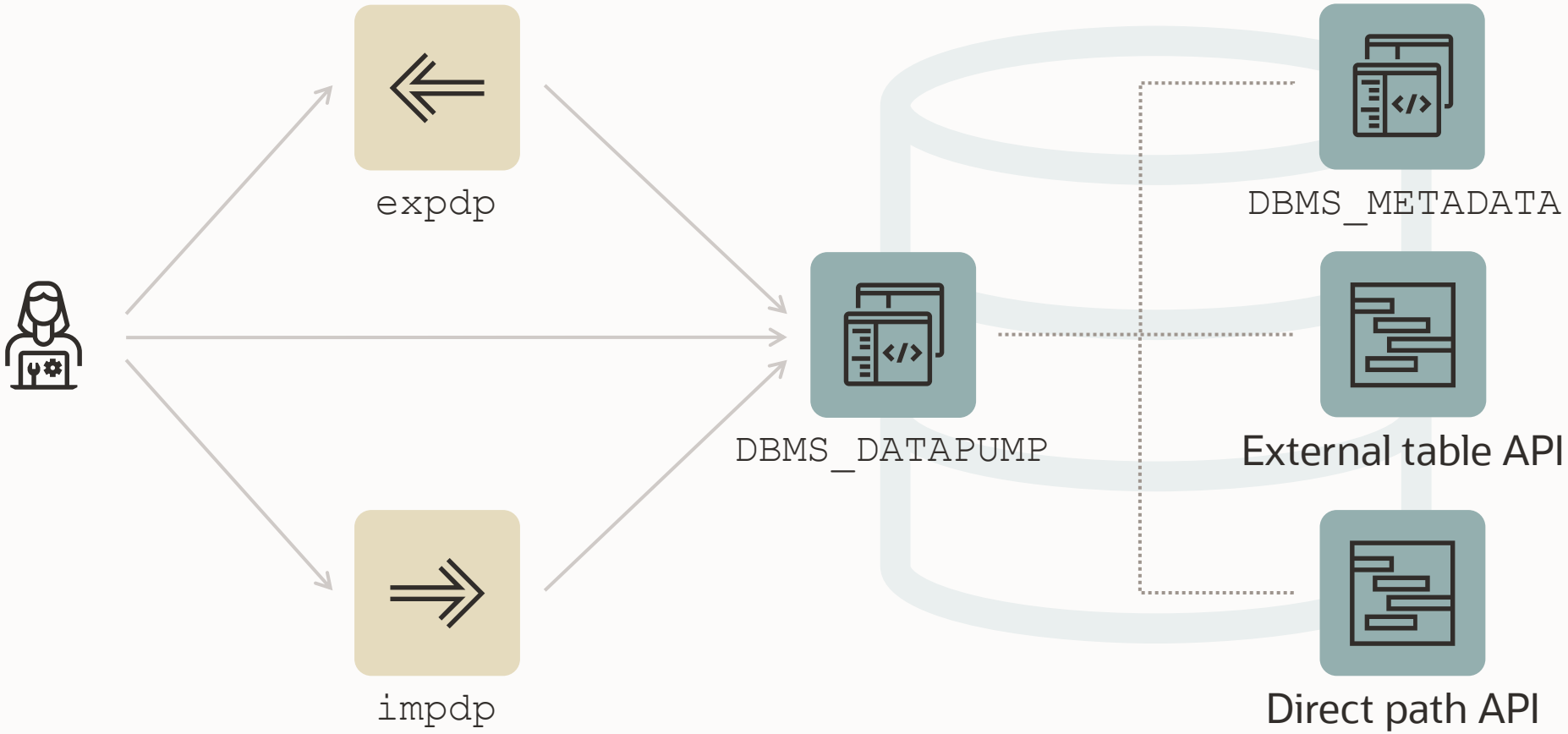
Data Pump | Architecture

Data Pump is server-based,

not client-based



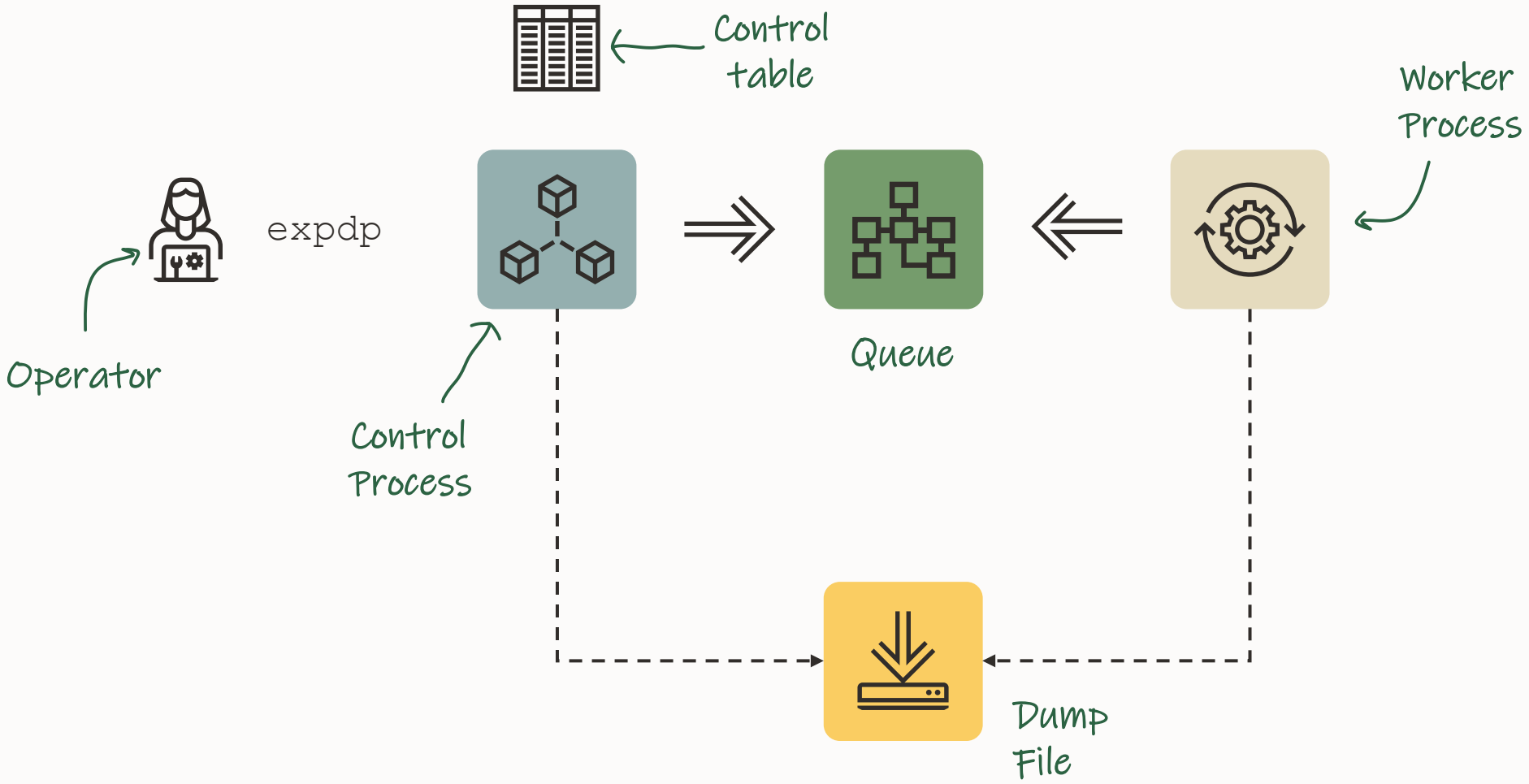
Data Pump | Architecture



Data Pump | Architecture Block Diagram

Clients 	`Expdp/impdp` & Interactive commands		Apps & services: ZDM, ODM, SQLcl, OEM…
APIs 	`DBMS_DATAPUMP`	`DBMS_METADATA`	`DBMS_TTS`
Engine 	Data Pump Engine		
Access Methods & Drivers 	External Tables via `ORACLE_DATAPUMP` Driver	Direct Path `DPAPI` API	Conventional Path (SQL)
Transport media	Dumpfile set 		Network Mode Import 

Data Pump | Architecture



Data Pump | Architecture

Control Table

A regular heap table containing:

- Job info and parameters
- Current status
- Object information
- Index into the dump files
- Enables restarts

Data Pump | Architecture

Control Table

- Control table is dropped upon successful completion of a job
- Optionally, kept using `KEEP_MASTER=Y`
- Can be queried like any other table
- Table written to the dump file set
- First object imported

Data Pump | Architecture

```
$ expdp dpuser/oracle schemas=app keep_master=y
```


Data Pump | Architecture

```
SQL> select name, value_t from dpuser.sys_export_schema_01;

NAME                   VALUE_T
---------------------- ----------------------------------------
SYS_EXPORT_SCHEMA_01   DB19.LOCALDOMAIN
LOG_FILE_DIRECTORY     DATA_PUMP_DIR
LOG_FILE_NAME          export.log
CLIENT_COMMAND         dpuser/******** schemas=app keep_master=y
SCHEMA_LIST            'APP'
SCHEMA_EXPR            IN ('APP')
COMPRESSION            METADATA_ONLY
COMPRESSION_ALGORITHM  BASIC
DATA_ACCESS_METHOD     AUTOMATIC
.
.
.
```

Data Pump | Unloading and loading

Data Files

Used for transportable tablespace

Direct Path

Only metadata is unloaded into/loaded from dumpfile

External Tables

Data remains in data files

Insert as Select

Conventional Path

Pro tip: Cross-endian data migration requires data files are converted

Data Pump | Unloading and loading

Data Files

Direct Path

External Tables

Insert as Select

Conventional Path

Unloads from / load into data files directly

Circumvents SQL layer

Fast

Not usable in all situations

Pro tip: Data Pump automatically selects the best unload/load method

Data Pump | Unloading and loading

Data Files

Direct Path

External Tables

- Use SQL layer to unload to / load from external table
- Can use `APPEND` hint for faster load
- Very good parallel capabilities
- Dump file format similar to direct path

Insert as Select

Conventional Path

Pro tip: Data unloaded with Data Pump is not compatible with a regular external table (`CREATE TABLE ... ORGANIZATION EXTERNAL ...`)

Data Pump | Unloading and loading

Data Files

Direct Path

External Tables

Insert as Select

Used by network link imports only

Will disable use of direct path

Conventional Path

Not very common

Data Pump | Unloading and loading

Data Files

Direct Path

External Tables

Insert as Select

Conventional Path

- Used as last resort
- Slower
- Import only

Data Pump | Unloading and loading

```
ACCESS_METHOD=[AUTOMATIC | DIRECT_PATH | EXTERNAL_TABLE | CONVENTIONAL_PATH | INSERT_AS_SELECT]
```

Pro Tip: Current table stats will help Data Pump make the right choice

Data Pump | Unloading and loading

```
$ expdp dpuser/oracle schemas=app metrics=y
```


Data Pump | Metadata

A category of metadata is described by an object path

Examples:

```
TABLE
TABLE/INDEX
TABLE/STATISTICS/TABLE_STATISTICS
TABLE/TRIGGER
```

You can get a full list of object paths from these views:

```
DATABASE_EXPORT_OBJECTS
SCHEMA_EXPORT_OBJECTS
TABLE_EXPORT_OBJECTS
```


Data Pump | Metadata

```
SQL> select object_path, comments from schema_export_objects;

OBJECT_PATH                    COMMENTS
------------------------------ --------------------------------------------------
ALTER_FUNCTION                 Recompile functions
ALTER_PACKAGE_SPEC             Recompile package specifications
ALTER_PROCEDURE                Recompile procedures
ANALYTIC_VIEW                  Analytic Views
AQ                             Advanced Queuing
ASSOCIATION                    Statistics type associations
ATTRIBUTE_DIMENSION            Attribute Dimensions
AUDIT_OBJ                      Object audits on the selected tables
CLUSTER                        Clusters in the selected schemas and their indexes
CLUSTERING                     Table clustering
.
.
.
```

Use `INCLUDE` or `EXCLUDE` to add or remove a specific category of metadata

`INCLUDE` and `EXCLUDE` are mutually exclusive. Now, from Oracle Database 21c they can be combined

Data Pump | Metadata

Some metadata has dependencies.

Example: Excluding a table will also exclude

- Indexes
- Constraints
- Grants
- Triggers
- And the like upon that table

Example: Excluding an index will also exclude

- Statistics on that index

Getting Started

Data Pump

Data Pump | Getting Started

Privilege

Directory

Streams Pool

Mode

Parameter File

Consistency

You can export your own schema.

In addition, two predefined roles:

- `DATAPUMP_EXP_FULL_DATABASE`
- `DATAPUMP_IMP_FULL_DATABASE`

Pro tip: These roles are powerful - use caution when granting them

Data Pump | Getting Started

Privilege
Directory
Streams Pool
Mode
Parameter File
Consistency

Don't use `SYS AS SYSDBA`

> **Caution:** Do not start Export as `SYSDBA`, except at the request of Oracle technical support. `SYSDBA` is used internally and has specialized functions; its behavior is not the same as for general users.

Database 19c, Utilities Guide

Data Pump | Getting Started

Privilege

Directory

Streams Pool

Mode

Parameter File

Consistency

Tablespace quota

- For create control table

Example of a Data Pump user

```
SQL> create user dpuser identified by oracle;

SQL> grant datapump_exp_full_database to dpuser;
SQL> grant datapump_imp_full_database to dpuser;

SQL> alter user dpuser quota unlimited on users;
```

Data Pump | Getting Started

- Privilege
- **Directory**
- Streams Pool
- Mode
- Parameter File
- Consistency

Needed to store dump files and log files

```
$ mkdir /home/oracle/dp

SQL> create directory dpdir as '/home/oracle/dp';
SQL> grant read, write on directory dpdir to dpuser;
```

If you don't specify a directory, the default is `DATA_PUMP_DIR`

```
SQL> select directory_path
     from   dba_directories
     where  directory_name='DATA_PUMP_DIR';
```


Data Pump | Getting Started

- Privilege
- Directory
- **Streams Pool**
- Mode
- Parameter File
- Consistency

Ensure STREAMS_POOL_SIZE is at a reasonable value

```
SQL> select current_size/1024/1024 as current_size_mb
     from   v$sga_dynamic_components
     where  component='streams pool';
```

Typically, in the range of 64M to 256M is adequate

```
SQL> alter system set streams_pool_size=256m scope=both;
```

Pro tip: Read about how other parameters affect Data Pump

Data Pump | Getting Started

- Privilege
- Directory
- Streams Pool
- **Mode**
- Parameter File
- Consistency

What do you want to export or import?

- Full
- Schema
- Table
- Tablespace
- Transportable Tablespace

Pro tip: Read more about export and import modes in the documentation

Data Pump | Getting Started

- Privilege
- Directory
- Streams Pool
- Mode
- **Parameter File**
- Consistency

You can specify Data Pump options in two ways:

On command line

```
$ expdp dpuser schemas=app directory=dp_dir
```

In parameter file - recommended

```
$ cat export.par
schemas=app
directory=dp_dir

$ expdp dpuser parfile=export.par
```

Data Pump | Getting Started

- Privilege
- Directory
- Streams Pool
- Mode
- **Parameter File**
- Consistency

Recommended to use parameter files

- Certain characters must be escaped
- These characters vary by operating system
- Escape characters vary by operating system
- Avoid typing long commands

Data Pump | Getting Started

- Privilege
- Directory
- Streams Pool
- Mode
- Parameter File
- **Consistency**

Default, consistent on a "per-table-basis"

Export starts at SCN 100

Table *A* exported at SCN 110

Table *B* exported at SCN 125

Table *C* exported at SCN 137

Table *D* exported at SCN 142

Data Pump | Getting Started

- Privilege
- Directory
- Streams Pool
- Mode
- Parameter File
- **Consistency**

Optionally, make export completely consistent

Export starts at SCN 100

Table *A* exported at SCN 100

Table *B* exported at SCN 100

Table *C* exported at SCN 100

Table *D* exported at SCN 100

Data Pump | Getting Started

- Privilege
- Directory
- Streams Pool
- Mode
- Parameter File
- **Consistency**

Consistent as of timestamp

```
$ expdp dpuser ... flashback_time=systimestamp
```

Consistent as of SCN

```
$ expdp dpuser ... flashback_scn=nnn
```

Legacy mode

```
$ expdp dpuser ... consistent=y
```

Requires UNDO

Pro tip: Export from your standby database

Data Pump | Getting Started

Watch on YouTube

deep dive

DATA PUMP

with development

Data Pump | Best Practices

Statistics

Diagnostics

Parallel

LOBs

Dump files

Compression

Checksum

Multitenant

Ensure dictionary statistics are current

- Before export
- After import

```
SQL> begin
        dbms_stats.gather_schema_stats('SYS');
        dbms_stats.gather_schema_stats('SYSTEM');
     end;
```

Pro tip: You can also use `GATHER_DICTIONARY_STATS`

Data Pump | Best Practices

Statistics

Diagnostics

Parallel

LOBs

Dump files

Compression

Checksum

Multitenant

Exclude optimizer statistics from export

```
#Regular export
expdp ... exclude=statistics

#Transportable tablespaces
expdp ... exclude=table_statistics,index_statistics
```

After import:

- Gather statistics in target database
- Or transport statistics using `DBMS_STATS`

Data Pump | Best Practices

- Statistics
- **Diagnostics**
- Parallel
- LOBs
- Dump files
- Compression
- Checksum
- Multitenant

Always include diagnostics in logfile

```
expdp ... logtime=all metrics=yes

impdp ... logtime=all metrics=yes
```

If you need help, we will always ask
for a logfile with these parameters set

Data Pump | Best Practices

No diagnostics

```
Processing object type SCHEMA_EXPORT/TABLE/TABLE
Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA
. . imported "METAL"."ALBUMS"                            988.8 KB   28069 rows
. . imported "METAL"."BANDS"                             3.444 MB   37723 rows
. . imported "METAL"."REVIEWS"                           66.47 MB   21510 rows
```

All diagnostics

```
16-OCT-20 17:26:57.158: Processing object type SCHEMA_EXPORT/TABLE/TABLE
16-OCT-20 17:26:58.262: Startup took 1 seconds
16-OCT-20 17:26:58.264: Startup took 1 seconds
16-OCT-20 17:26:59.082:      Completed 3 TABLE objects in 1 seconds
16-OCT-20 17:26:59.082:      Completed by worker 1 1 TABLE objects in 1 seconds
16-OCT-20 17:26:59.082:      Completed by worker 2 1 TABLE objects in 0 seconds
16-OCT-20 17:26:59.082:      Completed by worker 3 1 TABLE objects in 0 seconds
16-OCT-20 17:26:59.313: Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA
16-OCT-20 17:27:01.943: . . imported "METAL"."ALBUMS"    988.8 KB   28069 rows in 2 seconds using external_table
16-OCT-20 17:27:03.778: . . imported "METAL"."BANDS"     3.444 MB   37723 rows in 2 seconds using external_table
16-OCT-20 17:27:12.644: . . imported "METAL"."REVIEWS"   66.47 MB   21510 rows in 13 seconds using external_table
```

Data Pump | Best Practices

- Statistics
- Diagnostics
- **Parallel**
- LOBs
- Dump files
- Compression
- Checksum
- Multitenant

Use parallel to speed up the exports and imports

```
expdp ... parallel=n

impdp ... parallel=n
```

Degree of parallelism

- Cloud — Number of OCPUs
- On-prem (x86-64) — CPU cores x 2
- On-prem (other) — Depends

Pro tip: Hyperthreaded cores does not count

Data Pump | Best Practices

- Statistics
- Diagnostics
- **Parallel**
- LOBs
- Dump files
- Compression
- Checksum
- Multitenant

For exports

- For optimal performance, use multiple dump files
- Enables concurrent write to dump files

For imports

- Number of dump files does not affect parallel import
- Each worker process needs shared access to read from dump files

Data Pump | Best Practices

- Statistics
- Diagnostics
- Parallel
- **LOBs**
- Dump files
- Compression
- Checksum
- Multitenant

> *SecureFiles is the default storage mechanism for LOBs starting with Oracle Database 12c, and Oracle strongly recommends SecureFiles for storing and managing LOBs, rather then BasicFiles. BasicFiles will be deprecated in a future release.*

Database SecureFiles and Large Objects Developer's Guide

Data Pump | Best Practices

- Statistics
- Diagnostics
- Parallel
- **LOBs**
- Dump files
- Compression
- Checksum
- Multitenant

Always transform LOBs to SecureFile LOBs

```
impdp ... transform=lob_storage:securefile
```

It is faster to import LOBs,
when they are transformed to SecureFile LOBs

BasicFile LOBs do not allow parallel DML

Data Pump | Best Practices

Importing as BasicFiles

```
... imported "SCHEMA"."TABLE"      31.83 GB   681025 rows in 804 seconds using direct_path
```

Importing as SecureFiles

```
... imported "SCHEMA"."TABLE"      31.83 GB   681025 rows in 261 seconds using external_table
```


Data Pump | Best Practices

- Statistics
- Diagnostics
- Parallel
- LOBs
- **Dump files**
- Compression
- Checksum
- Multitenant

Always export to multiple files

```
expdp ... dumpfile=mydump%L.dmp
```

Optionally, limit individual file size

```
expdp ... dumpfile=mydump%L.dmp filesize=5G
```

Pro tip: You can also use the old `%U` format

Data Pump | Best Practices

- Statistics
- Diagnostics
- Parallel
- LOBs
- Dump files
- **Compression**
- Checksum
- Multitenant

Use compression to speed up your export

```
compression=all
compression_algorithm=medium
```

Requires Advanced Compression Option

Data Pump | Best Practices

- Statistics
- Diagnostics
- Parallel
- LOBs
- Dump files
- **Compression**
- Checksum
- Multitenant

Compression algorithms

BASIC:	The same algorithm used in previous versions. Good compression, without severely impacting on performance
LOW:	For use when reduced CPU utilization is a priority over compression ratio
MEDIUM:	Recommended option. Similar characteristics to BASIC, but uses a different algorithm
HIGH:	Maximum available compression, but more CPU intensive

Data Pump | Best Practices

- Statistics
- Diagnostics
- Parallel
- LOBs
- Dump files
- **Compression**
- Checksum
- Multitenant

Real-life examples - 12.2 EBS Database export

	FILE SIZE MB	RATIO	TIME
NONE	5.500	1,0	4m 54s
ALL BASIC	622	8,9	4m 58s
ALL LOW	702	7,8	5m 24s
ALL MEDIUM	567	9,7	4m 55s
ALL HIGH	417	13,2	5m 13s

	FILE SIZE MB	RATIO	TIME
NONE	5.800	1,0	2m 33s
ALL BASIC	705	8,2	3m 03s
ALL LOW	870	6,6	8m 11s
ALL MEDIUM	701	8,2	3m 01s
ALL HIGH	509	11,3	12m 16s

Data Pump | Best Practices

- Statistics
- Diagnostics
- Parallel
- LOBs
- Dump files
- **Compression**
- Checksum
- Multitenant

Default compression type

```
compression=metadata_only
```

Only metadata information is compressed.

Does not require Advanced Compression Option

Pro tip: Importing a compressed dump file does not require a license for ACO

Data Pump | Best Practices

- Statistics
- Diagnostics
- Parallel
- LOBs
- Dump files
- Compression
- **Checksum**
- Multitenant

What can happen to a dump file when it is transferred?

- Tampering
- Corruption

NEW IN 21c

Data Pump | Best Practices

- Statistics
- Diagnostics
- Parallel
- LOBs
- Dump files
- Compression
- **Checksum**
- Multitenant

Data Pump can calculate checksum on export

```
expdp ... checksum_algorithm=sha384
```

Verify dump file integrity on import

```
impdp... verify_only=yes


impdp ... verify_checksum=yes
```

NEW IN 21c

Data Pump | Best Practices

Alternatively, calculate checksum yourself

```
[oracle@hol]$ md5sum metal*.dmp



5edf66ed92086b4f69580fc27b75f662  metal_01.dmp
59eb25ff2a0f648c051a9212e0861979  metal_02.dmp
29951a56abe074d9151c27728d88e9eb  metal_03.dmp
c8860e7a71e74f8013068240b598c116  metal_04.dmp
0d05d258e4b501c657cd9490b7e48715  metal_05.dmp
1e367394a31e2ce45d2aeb6a3d4f9507  metal_06.dmp
9c276aa580c0e57c0829f274d04d15de  metal_07.dmp
0d560d0ce57c47425424e17604d8ec49  metal_08.dmp
```

```
SQL> SELECT object_name, checksum
     FROM DBMS_CLOUD.LIST_OBJECTS(
          '<credential_name>',
          '<location_uri>');

metal_01.dmp 5edf66ed92086b4f69580fc27b75f662
metal_02.dmp 59eb25ff2a0f648c051a9212e0861979
metal_03.dmp 29951a56abe074d9151c27728d88e9eb
metal_04.dmp c8860e7a71e74f8013068240b598c116
metal_05.dmp 0d05d258e4b501c657cd9490b7e48715
metal_06.dmp 1e367394a31e2ce45d2aeb6a3d4f9507
metal_07.dmp 9c276aa580c0e57c0829f274d04d15de
metal_08.dmp 0d560d0ce57c47425424e17604d8ec49
```

- Windows: `Get-FileHash *.dmp -Algorithm MD5`
- Errors manifests as `ORA-31693 ORA-29913 ORA-29104`

Data Pump | Best Practices

- Statistics
- Diagnostics
- Parallel
- LOBs
- Dump files
- Compression
- Checksum
- **Multitenant**

Avoid *noisy-neighbour* if resources are insufficient

Restrict number of concurrent Data Pump jobs in a PDB

```
SQL> alter system set max_datapump_jobs_per_pdb=2
     container=all;
```

Restrict the parallel degree in a Data Pump job

```
SQL> alter system set max_datapump_parallel_per_job=2
     container=all;
```

deep dive

DATA PUMP

with development

Photo by Nicolas Messifet on Unsplash

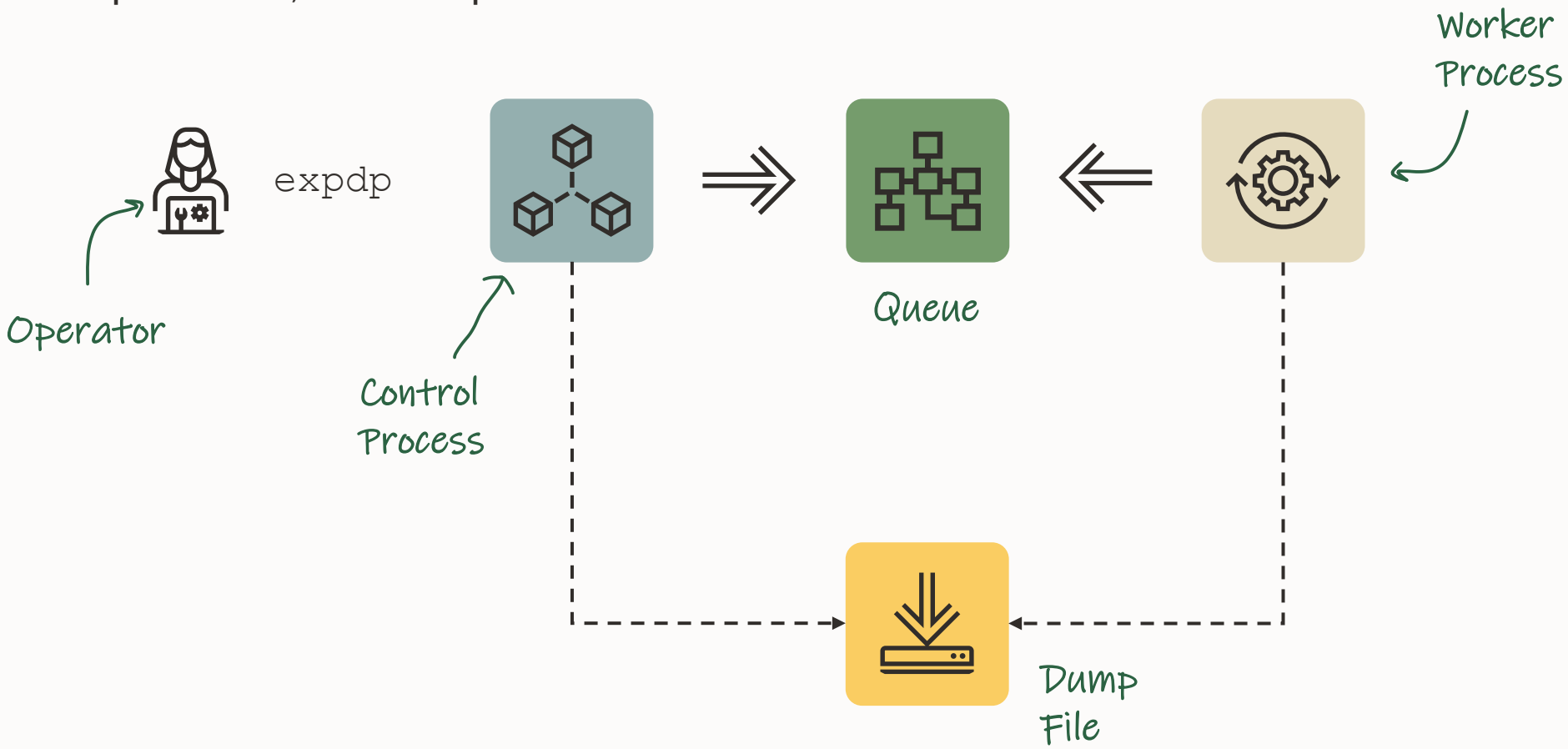
Everything Parallel

Export

Parallel | Control and Worker process

If you the default or `PARALLEL=1`

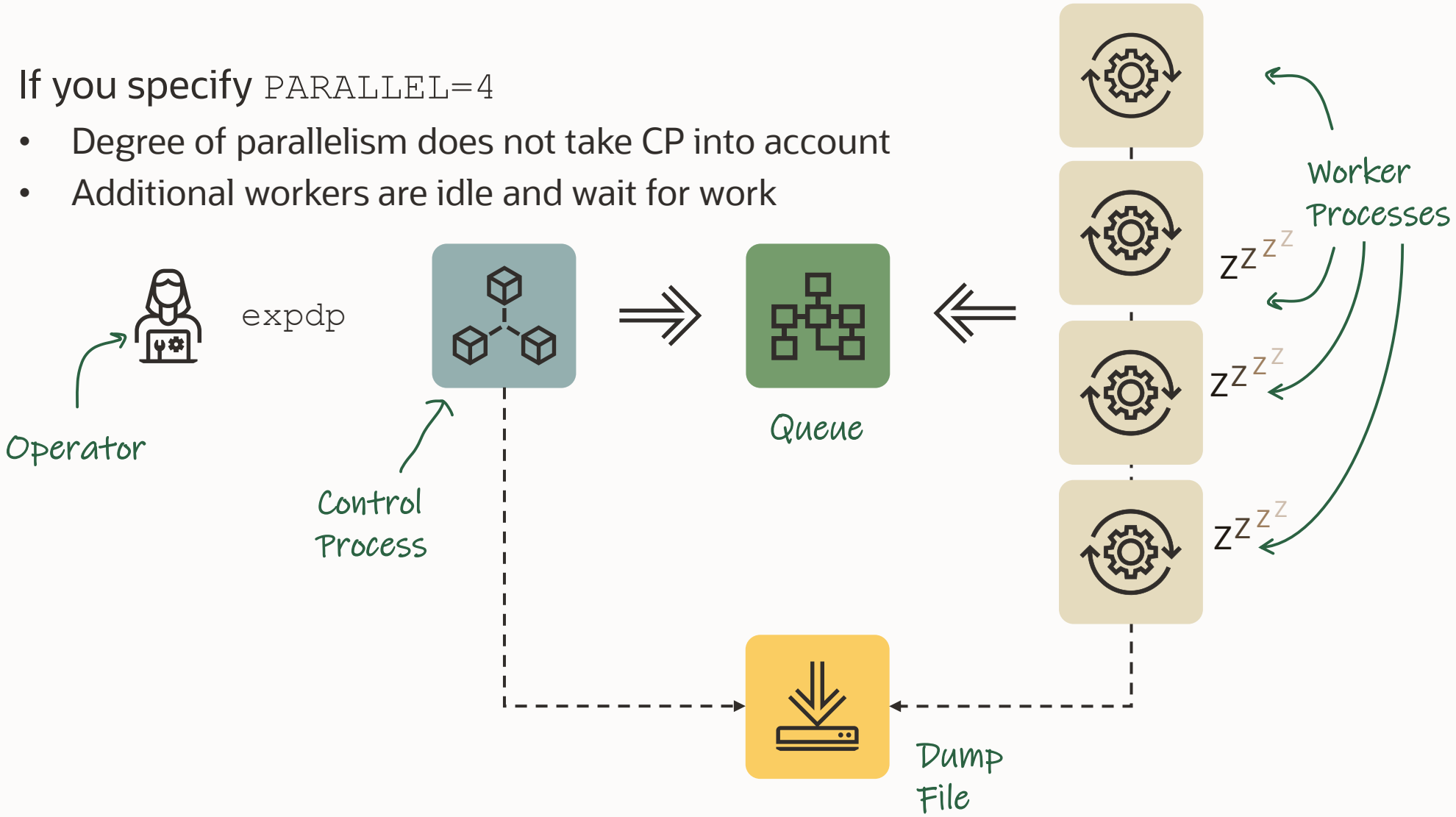
- 2 processes, 1 control process and 1 worker



Parallel | Degree of Parallelism

If you specify `PARALLEL=4`

- Degree of parallelism does not take CP into account
- Additional workers are idle and wait for work



Data Pump: Parallel Operations

Overview

- A job starts w/ a minimum of 2 processes: a Control Process (CP) + 1 Worker

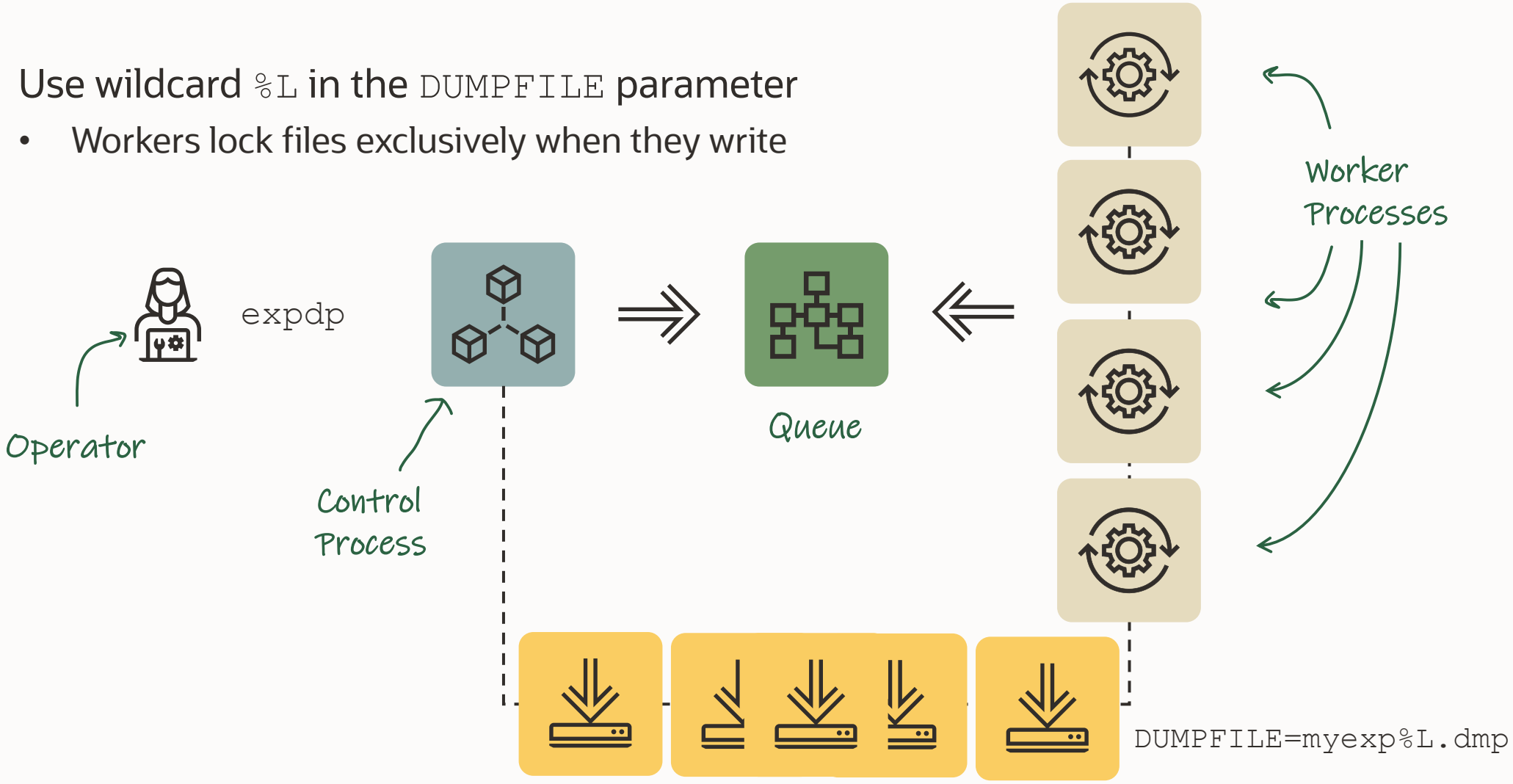
- The user can specify a Degree of Parallelism (DOP) for the Data Pump job
  - DOP = maximum number of ACTIVE parallel processes (Active workers + PQ processes)
    - DOP does not include the CP or shadow process
    - Any additional workers are idle and wait for work
    - Data Pump will only start the number of threads needed to complete the task

- Control Process: Verifies parameters & job description, controls & assigns work items to workers

Parallel | expdp - Multiple Dump Files

Use wildcard `%L` in the `DUMPFILE` parameter

- Workers lock files exclusively when they write

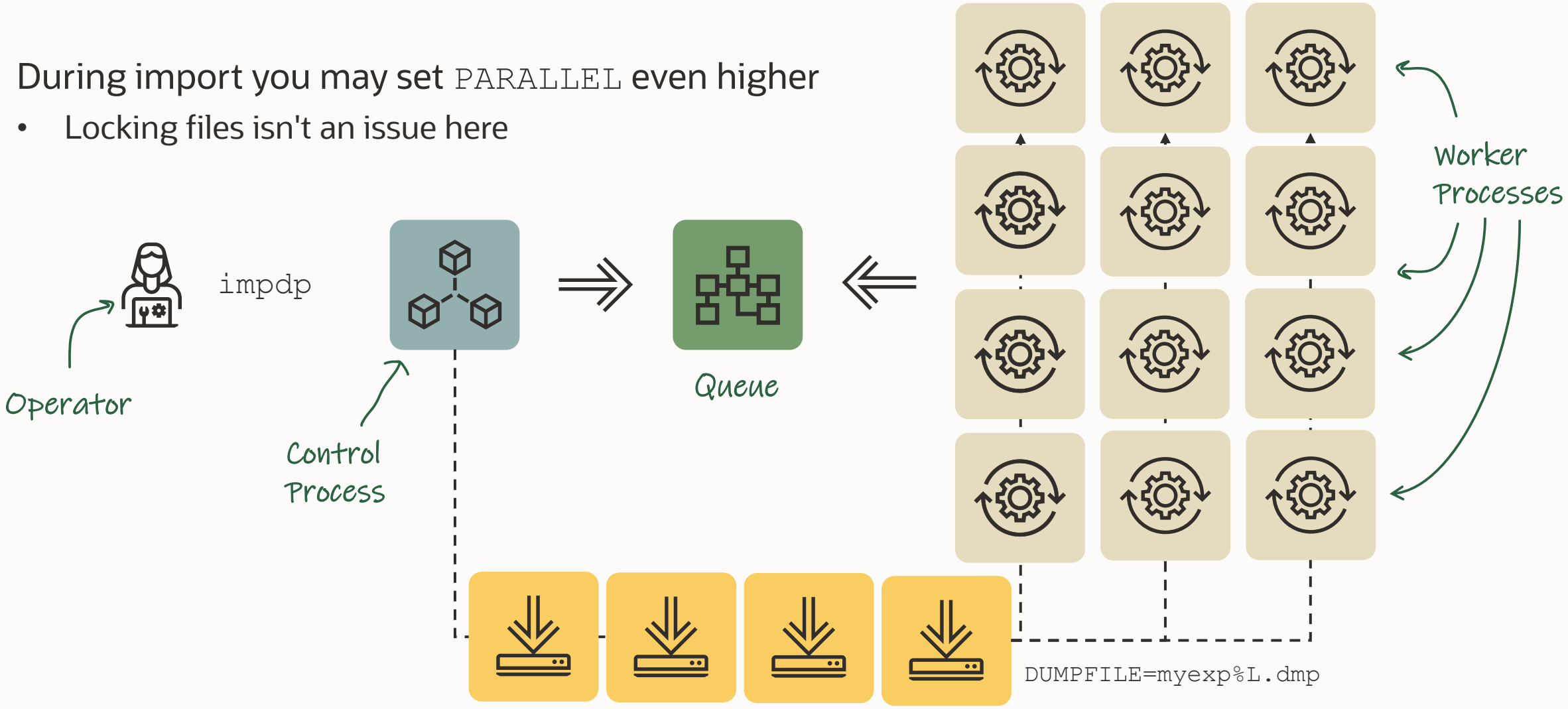


Use `PARALLEL` together with wildcard `%L` in the `DUMPFILE` parameter to create multiple dump files

Parallel | impdp - Multiple Dump Files

During import you may set `PARALLEL` even higher

- Locking files isn't an issue here

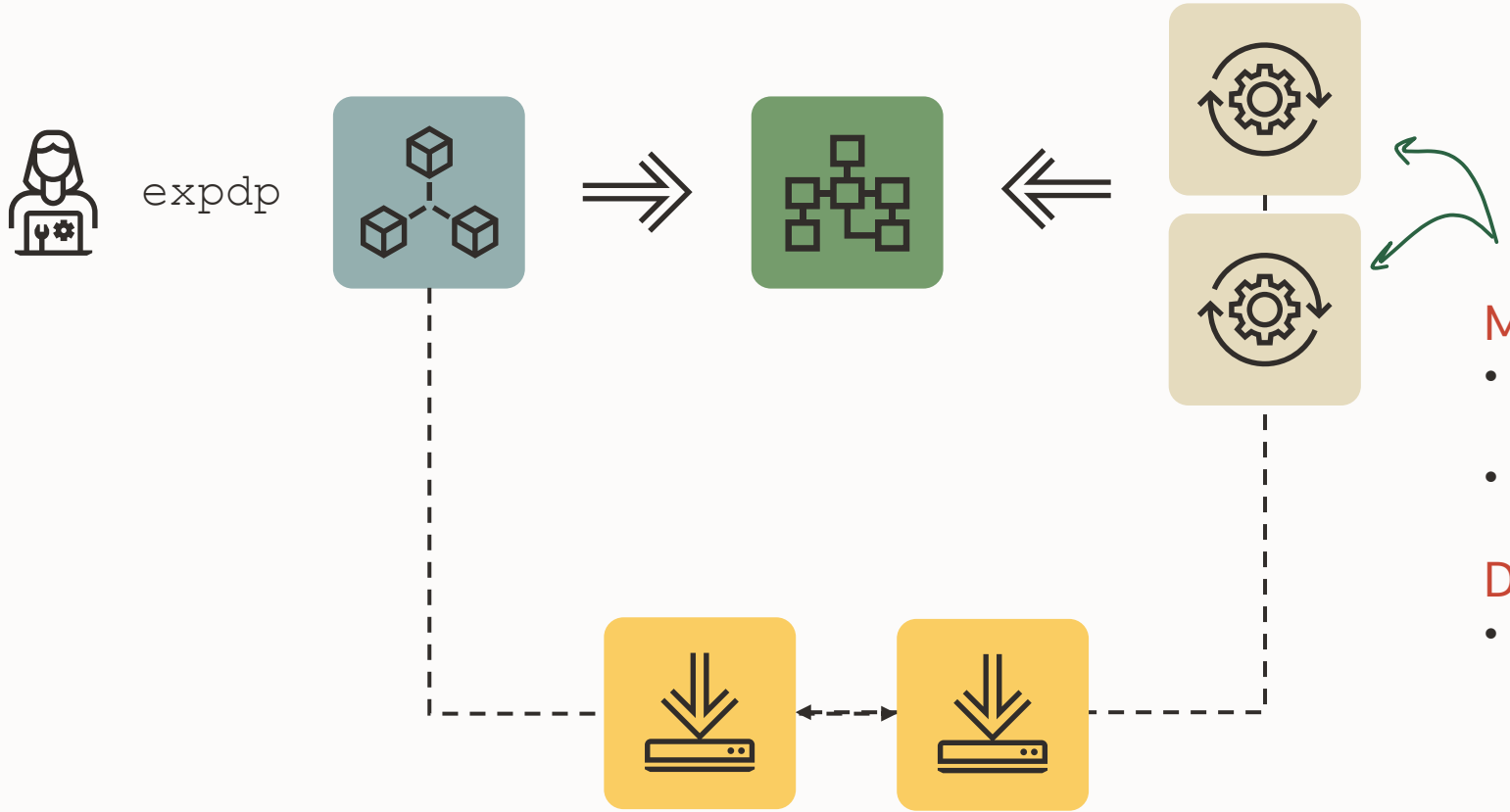


You forgot to set `PARALLEL`?
You can change it at runtime

What are Data Pump workers doing?
And how gets work assigned to each worker?

Parallel | Data Pump Worker: expdp

What does a worker process do during `expdp`?



Metadata

- Collect object metadata with `DBMS_METADATA`
- Write metadata to dumpfile as XML

Data

- Unload data via the Data Layer

Parallel | How export works

- Export (and network mode import) work in multiple phases

- Metadata
  - Each object path is a metadata work unit
  - Workers begin collecting metadata for export
  - Metadata unloaded in any order with parallel worker processes

- Data
  - Metadata for `TABLE_DATA` items contain a size estimate that drives order of export
  - Data work items are scheduled as worker processes become available.

Pro Tip: current dictionary and object statistics are crucial to data pump export performance!

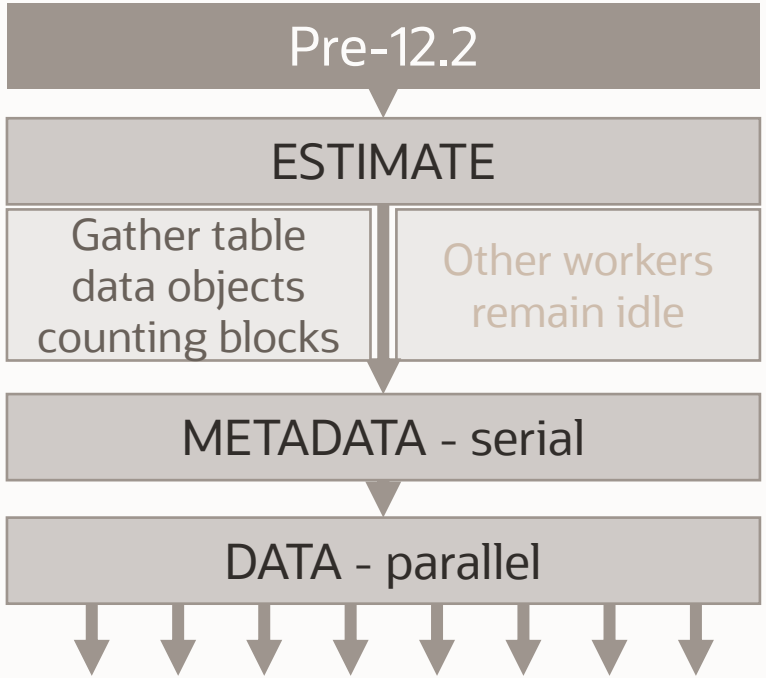
Parallel | How is work assigned to parallel Worker processes?

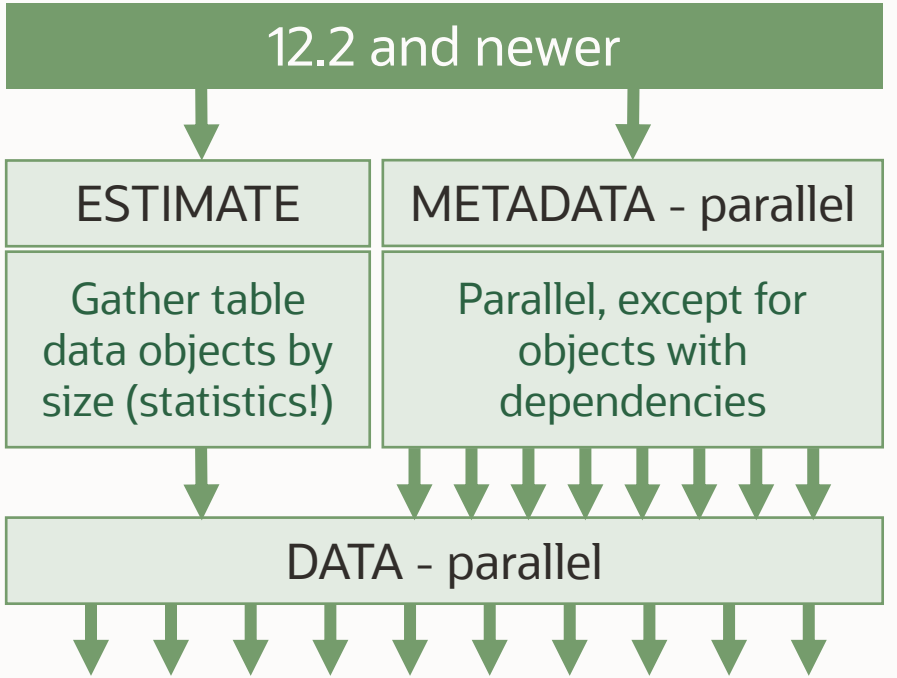
- `TABLE_DATA` work unit is:
  - Subpartition for subpartitioned tables
  - Partition for partitioned tables
  - Table for non-partitioned tables
  - Specify all data for the table as one work unit, regardless of partitioning w/
    `DATA_OPTIONS=GROUP_PARTITION_TABLE_DATA`

- The metadata work unit is a single object path

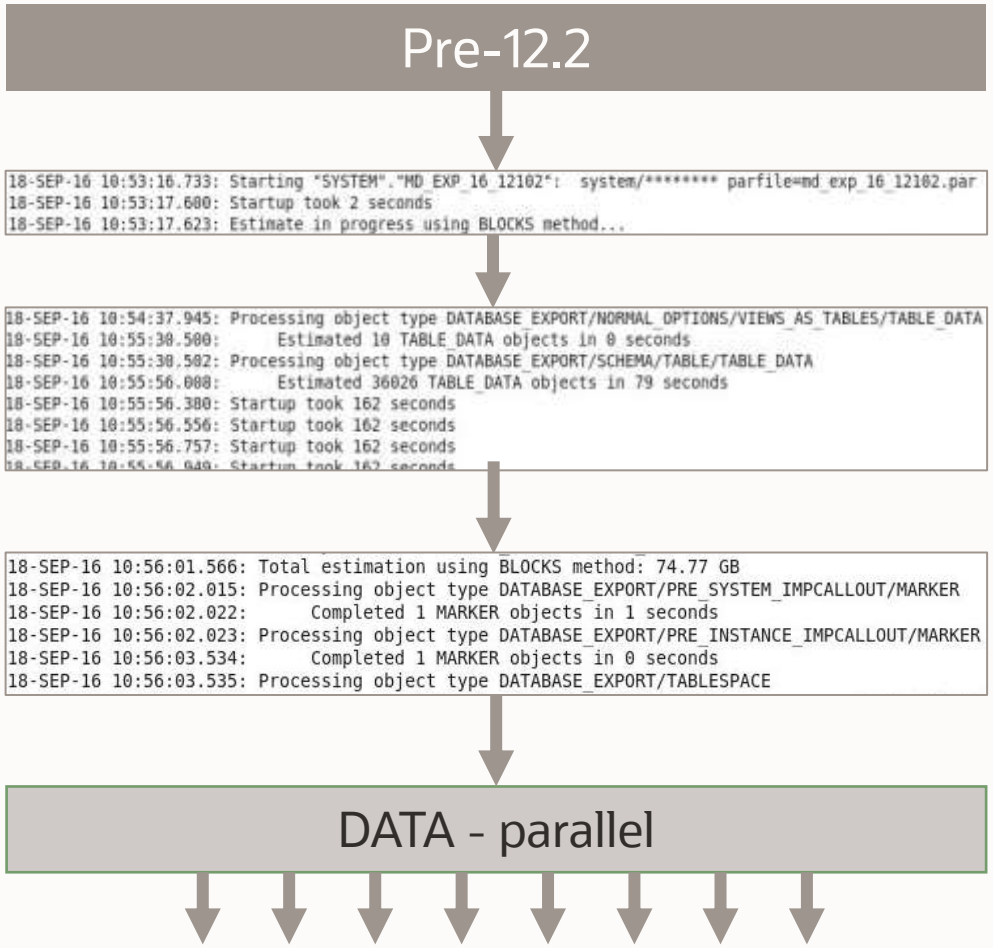
Parallel | Metadata Export - Comparison

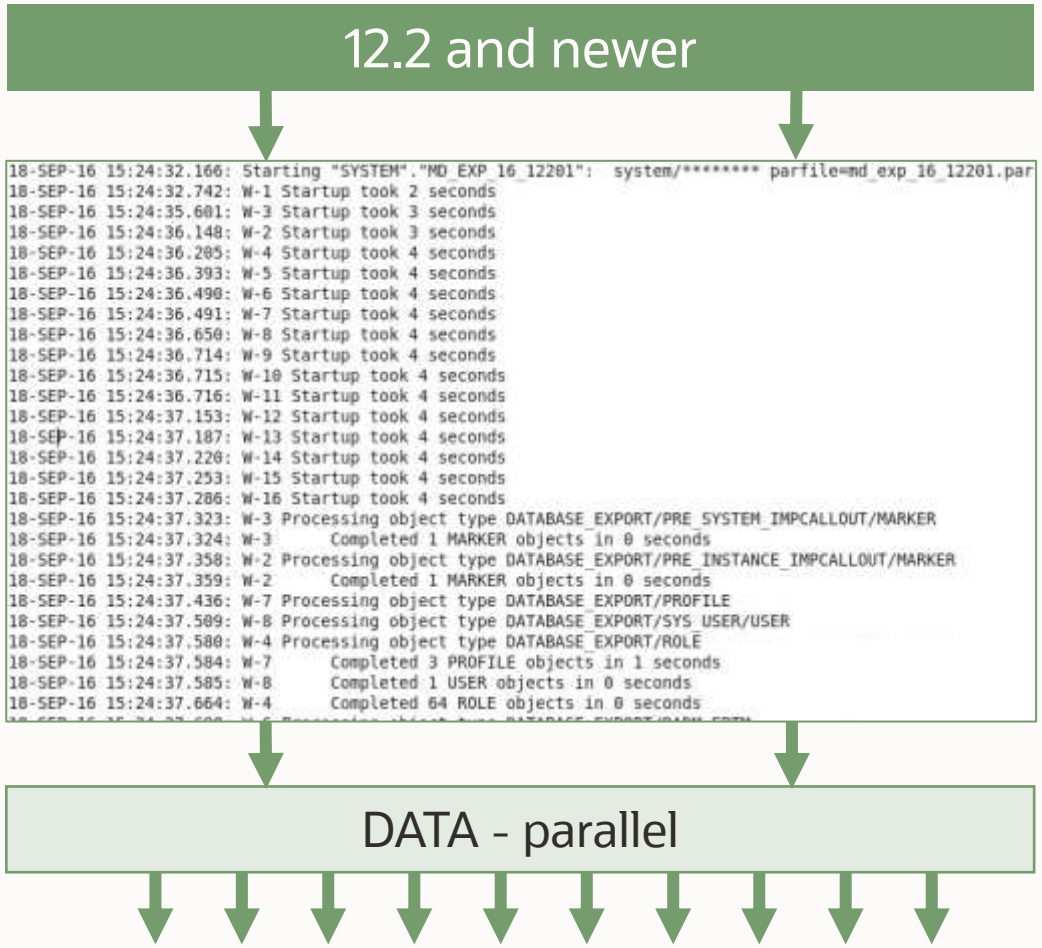
- Since 12.2: Metadata export happens concurrently with estimate phase for table data
- Most metadata & data objects are exported in parallel when `PARALLEL=2` or greater





Parallel | Metadata Export - Logfiles





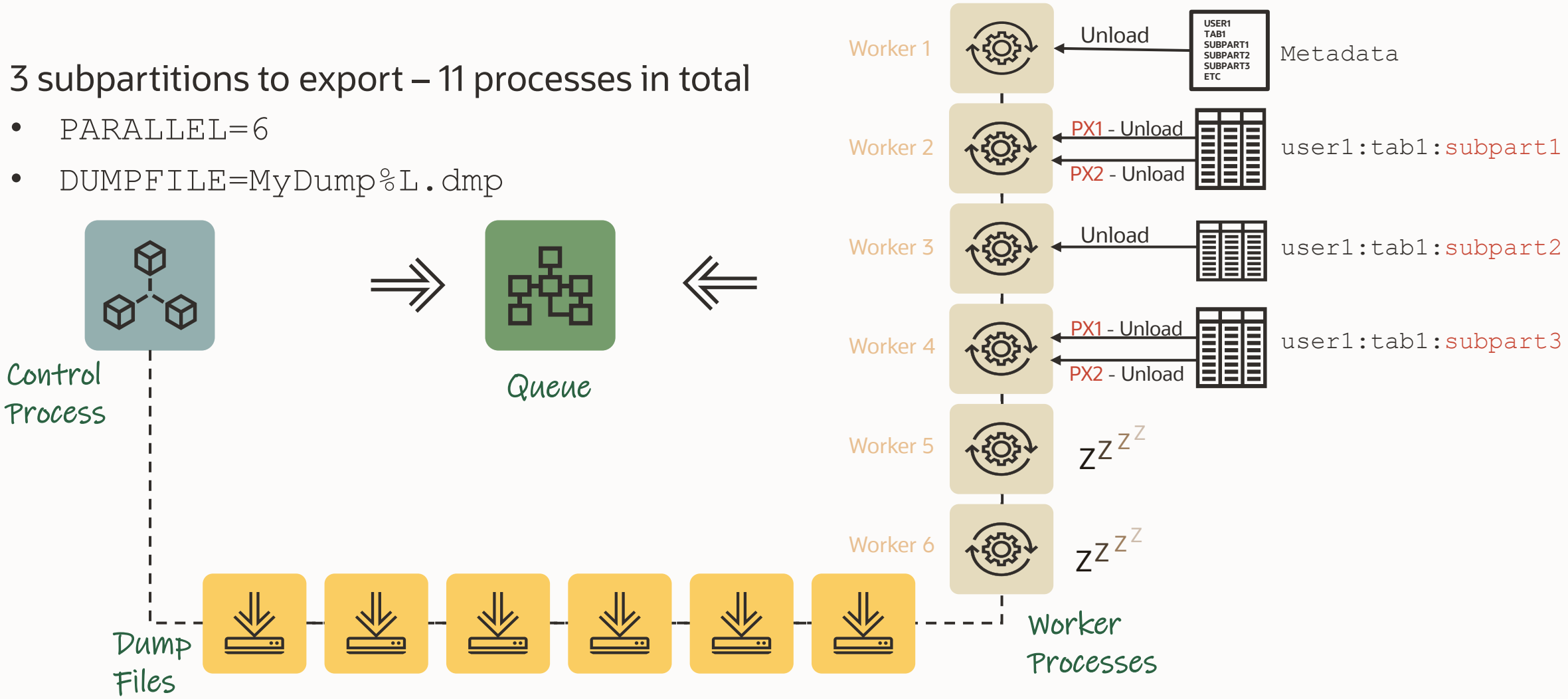
Pro Tip

Ensure statistics on objects and SYS/SYSTEM are current

Parallel | Export Example

3 subpartitions to export – 11 processes in total

- `PARALLEL=6`
- `DUMPFILE=MyDump%L.dmp`



Data Pump: 2 kinds of Parallelism for Dumpfiles

- Inter-partition parallelism
  - For tables with partitions below a particular threshold in size
  - One connection and worker per partition or sub partition

- Parallel query (PQ) intra-partition parallelism
  - Invoked for larger tables or partitions
  - Worker process is the query coordinator, and not included in the DOP limit

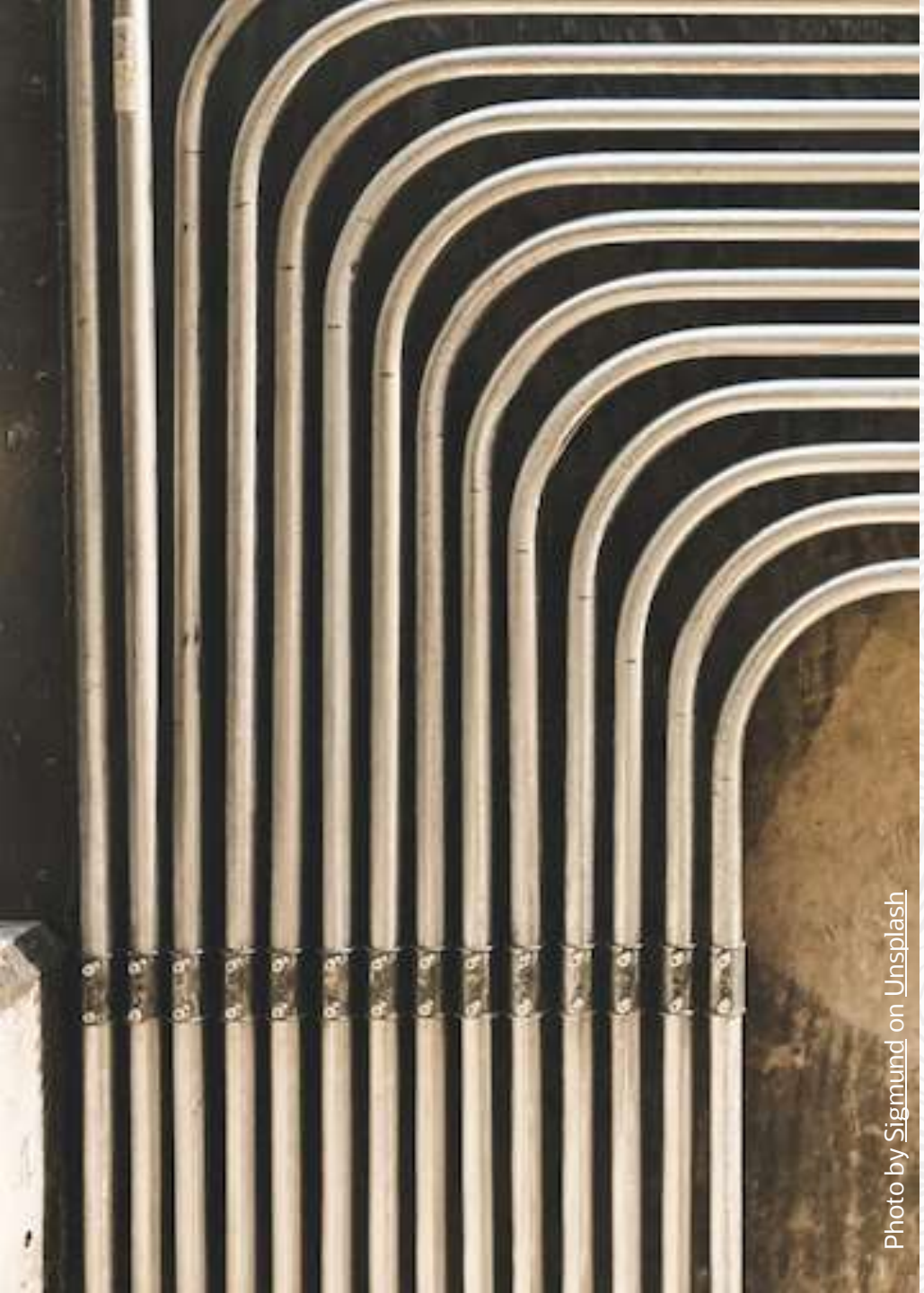


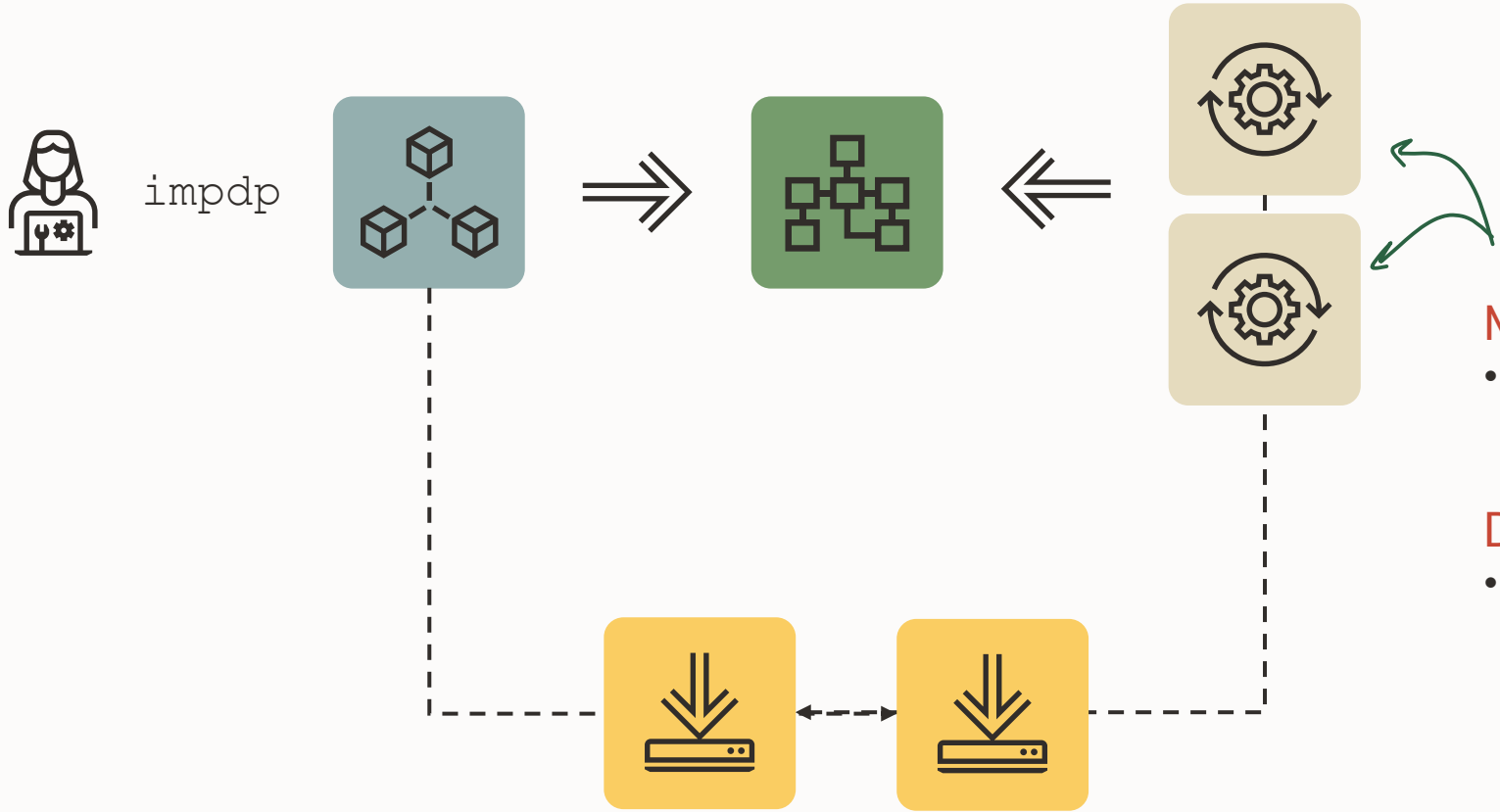
Photo by Sigmund on Unsplash

Everything Parallel

Import

Parallel | Data Pump Worker: impdp

What does a worker process do during `impdp`?



Metadata

- Convert metadata from XML into DDL and PL/SQL using `DBMS_METADATA`

Data

- Load data via the Data Layer

Parallel | How import works

- No estimate phase
- Reading from Control Table in the dump file
  - Ordering happens strictly by object path
- Metadata (except indexes and package bodies)
  - Mostly in parallel
- Data
  - Loading in parallel with multiple workers and maybe PQ processes
- Remaining metadata
  - Indexes and package bodies

Pro Tip: If you have very large indexes, a manual index built-up via script may be better

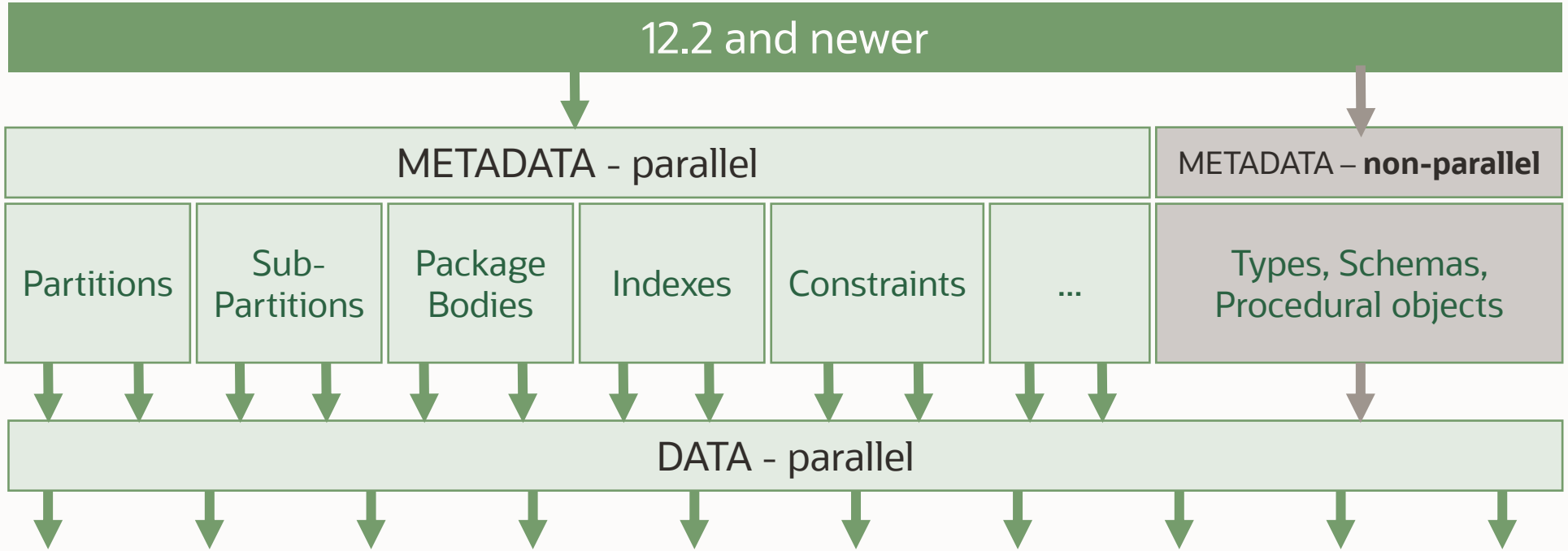
Pro Tip: MOS Note: 1288037.1 - How To Relate DataPump PARALLEL Parameter To Parallel Query Slaves

Parallel metadata import even works when export wasn't done in parallel e.g., in a release before Oracle 12.2

Parallel | Metadata Import

Since 12.2: Metadata import happens concurrently

- Most metadata & data objects are imported in parallel when `PARALLEL=2` or greater

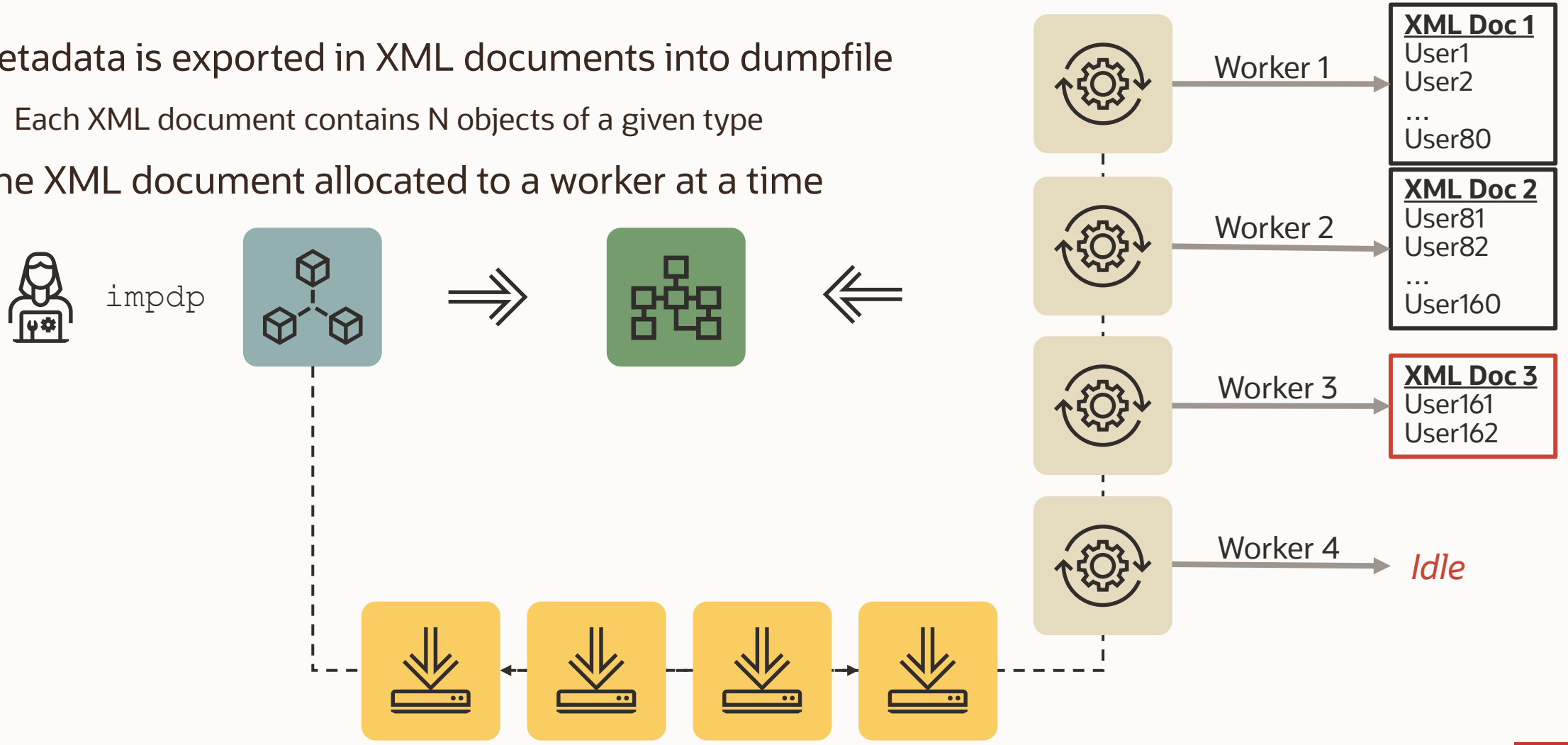


Parallel | Metadata Import - Internals

Metadata is exported in XML documents into dumpfile

- Each XML document contains N objects of a given type

One XML document allocated to a worker at a time



Parallel | Metadata Import - Logfiles

Comparison with `PARALLEL=8` for 27586 object grants - `METRICS=Y`

50 sec — Pre-12.2

```
15-SEP-16 13:56:16.317: Processing object type DATABASE_EXPORT/SCHEMA/SEQUENCE/GRANT/OWNER_GRANT/OBJECT_GRANT
15-SEP-16 13:57:06.374:      Completed 27586 OBJECT_GRANT objects in 50 seconds
```

14 sec — 12.2 and newer

```
15-SEP-16 11:59:35.190: W-7 Processing object type DATABASE_EXPORT/SCHEMA/SEQUENCE/GRANT/OWNER_GRANT/OBJECT_GRANT
15-SEP-16 11:59:49.304: W-4      Completed 27586 OBJECT_GRANT objects in 10 seconds
15-SEP-16 11:59:49.304: W-4      Completed by worker 1 3426 OBJECT_GRANT objects in 10 seconds
15-SEP-16 11:59:49.304: W-4      Completed by worker 2 3440 OBJECT_GRANT objects in 10 seconds
15-SEP-16 11:59:49.304: W-4      Completed by worker 3 3440 OBJECT_GRANT objects in 10 seconds
15-SEP-16 11:59:49.304: W-4      Completed by worker 4 3440 OBJECT_GRANT objects in 9 seconds
15-SEP-16 11:59:49.304: W-4      Completed by worker 5 3440 OBJECT_GRANT objects in 10 seconds
15-SEP-16 11:59:49.304: W-4      Completed by worker 6 3520 OBJECT_GRANT objects in 10 seconds
15-SEP-16 11:59:49.304: W-4      Completed by worker 7 3440 OBJECT_GRANT objects in 10 seconds
15-SEP-16 11:59:49.304: W-4      Completed by worker 8 3440 OBJECT_GRANT objects in 10 seconds
```

Parallel | Performance Comparison Example

EBS test database

- Import time in seconds
- `PARALLEL=32`

Object Type	*Count*	*Elapsed (sec) 11.2.0.4*	*Elapsed (sec) 12.2.0.1*
OBJECT_GRANT (owner)	27,586	49	22
SYNONYM	43,254	105	44
TYPE	4,364	108	110
PROCACT_SCHEMA	606	198	175
TABLE	33,164	923	248
OBJECT_GRANT (table)	358,649	541	157
INDEX	53,190	**6721**	272
PACKAGE	53,217	424	54
VIEW	34,690	538	184
PACKAGE BODY	52,092	1363	959

Photo by Jason Smith on Unsplash

Everything Parallel?

No parallelism

Network import does not support loading metadata in parallel

No parallelism on BasicFile LOBs

Convert *old* BasicLOBs to SecureFile LOBs

No parallelism for metadata export and import with Transportable Tablespaces

Parallel metadata export and import with Transportable Tablespace is added in Oracle Database 21c

Photo by Matt Chesin on Unsplash

Faster Patching

DPLoad Patch and Bundle Patch

Patching | Symptoms

`dpload.sql` is used when Data Pump gets patched

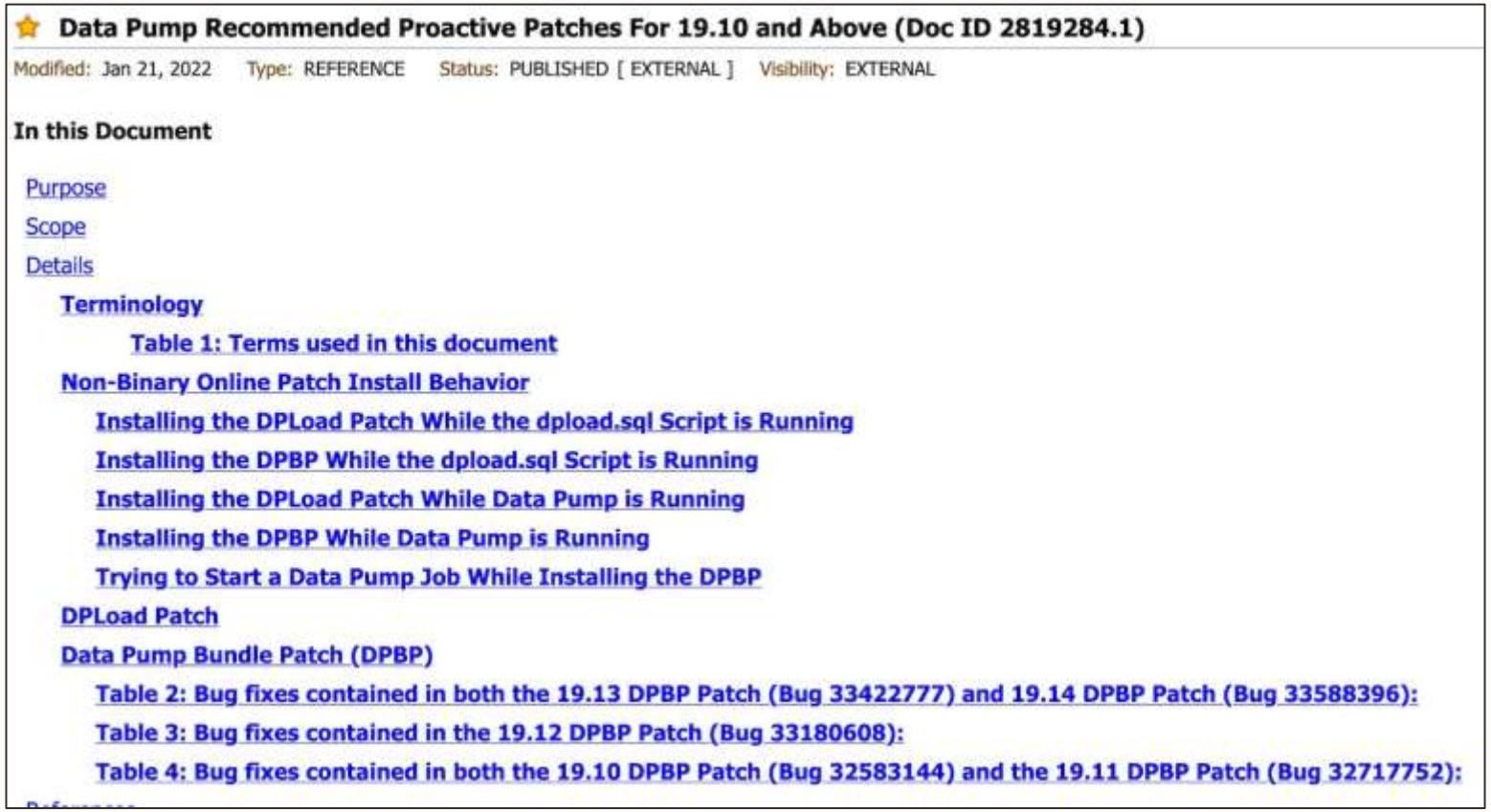
- Patching can be time consuming

```
Rem    NAME
Rem      dpload.sql - load entire Data Pump utility.
Rem
Rem    DESCRIPTION
Rem      load entire Data Pump utility (including the metadata layer).
Rem
Rem    NOTES
Rem      When there are changes to any of the following Data Pump files it
Rem      is necessary to unload and then reload the entire Data Pump utility.
Rem         src/server/datapump/services/prvtkupc.sql
Rem         src/server/datapump/ddl/prvtmetd.sql
Rem         admin/dbmsdp.sql
Rem         admin/catmeta.sql
Rem         admin/catmettypes.sql
Rem         admin/catmetviews.sql
Rem         admin/catmetinsert.sql
```

Patching | Solution

MOS Note: 2819284.1 - Data Pump Recommended Proactive Patches For 19.10 and Above

- Download



Data Pump Recommended Proactive Patches For 19.10 and Above (Doc ID 2819284.1)

Modified: Jan 21, 2022 Type: REFERENCE Status: PUBLISHED [EXTERNAL] Visibility: EXTERNAL

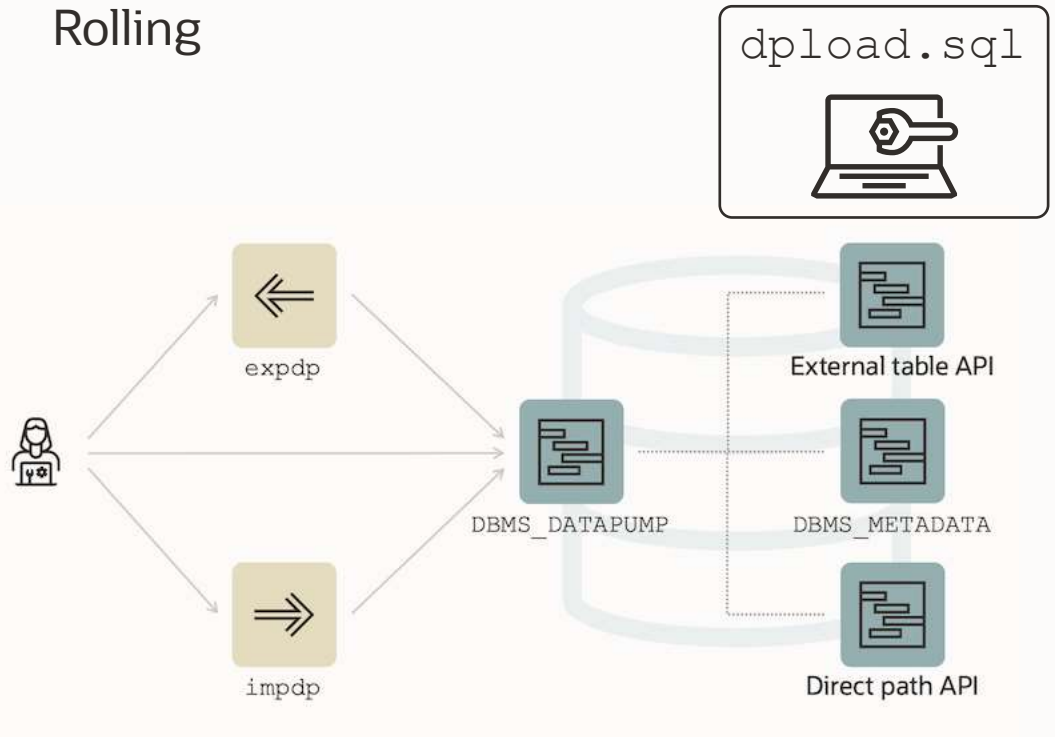
In this Document

Purpose
Scope
Details
Terminology
Table 1: Terms used in this document
Non-Binary Online Patch Install Behavior
Installing the DPLoad Patch While the dpload.sql Script is Running
Installing the DPBP While the dpload.sql Script is Running
Installing the DPLoad Patch While Data Pump is Running
Installing the DPBP While Data Pump is Running
Trying to Start a Data Pump Job While Installing the DPBP
DPLoad Patch
Data Pump Bundle Patch (DPBP)
Table 2: Bug fixes contained in both the 19.13 DPBP Patch (Bug 33422777) and 19.14 DPBP Patch (Bug 33588396):
Table 3: Bug fixes contained in the 19.12 DPBP Patch (Bug 33180608):
Table 4: Bug fixes contained in both the 19.10 DPBP Patch (Bug 32583144) and the 19.11 DPBP Patch (Bug 32717752):

Patching | DPload vs DP Bundle

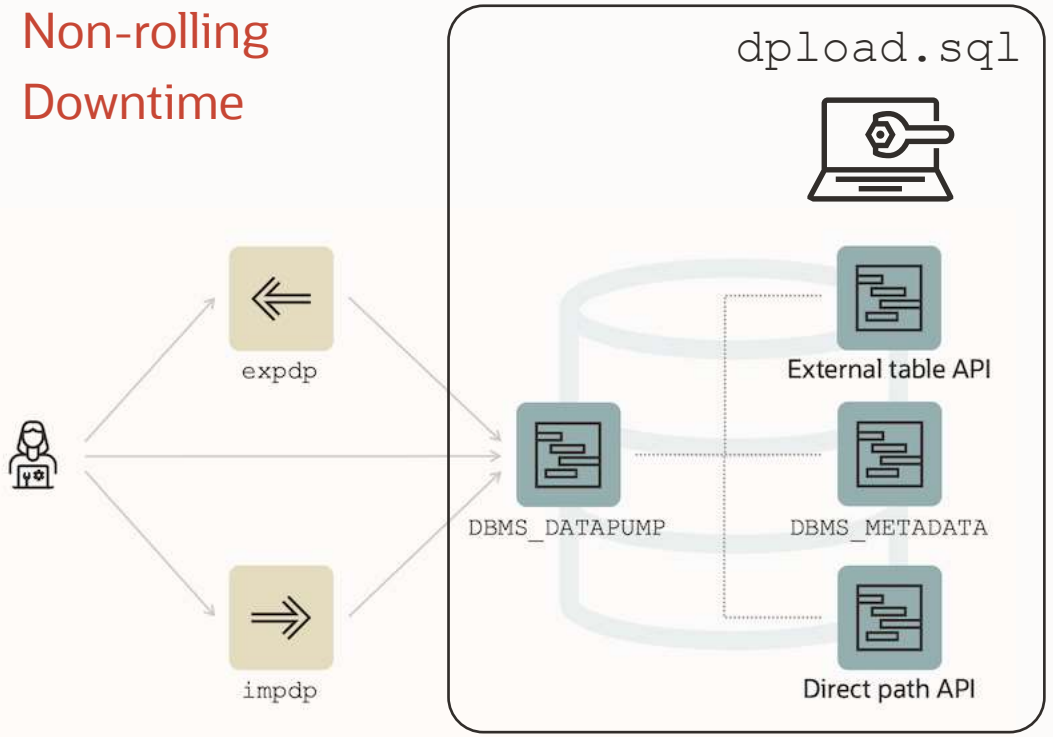
DPload Patch

- Speeds up Data Pump patching
- No downtime
- Rolling



Data Pump Bundle Patch

- Speeds up Data Pump patching
- Adds lots of important fixes
- Non-rolling
- Downtime



Patching | DPload vs DP Bundle

DPload Patch

- Patch 32919937

Data Pump Bundle Patch

- Patch 33588396 - 19.14.0
- Patch 33422777 - 19.13.0
- Patch 33180608 - 19.12.0
- Patch 32717752 - 19.11.0
- Patch 32583144 - 19.10.0

Data Pump Bundle Patch | Non-Binary Online Patching Safeguards

Installing the Data Pump Bundle Patch when Data Pump is in use:

- Built-in 3-minute timeout before signaling an error

```
BEGIN ku$_dpload.initial_phase; END;
*
ERROR at line 1:
ORA-20000: Retry dpload.sql script later when
Data Pump and Metadata API are not in use; current users are:
pid:11720, user:SYS, machine:<Machine>, sid:263,
module:sqlplus@<ConnectString> (TNS V1-
ORA-06512: at "SYS.KU$_DPLOAD", line 1042
ORA-06512: at line 1
```


Data Pump Bundle Patch | Non-Binary Online Patching Safeguards

Attempting to run Data Pump while patching is in progress:

```
Connected to: Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
ORA-31626: job does not exist
ORA-31637: cannot create job SYS_EXPORT_FULL_01 for user SYSTEM
ORA-06512: at "SYS.KUPV$FT", line 1142
ORA-06512: at "SYS.DBMS_SYS_ERROR", line 95
ORA-06512: at "SYS.KUPV$FT", line 1751
ORA-39062: error creating master process DM00
ORA-39107: Master process DM00 violated startup protocol. Master error:
ORA-06533: Subscript beyond count
ORA-06512: at "SYS.KUPP$PROC", line 59
ORA-06512: at "SYS.KUPP$PROC", line 310
ORA-06512: at "SYS.KUPV$FT", line 1691
ORA-06512: at "SYS.KUPV$FT", line 1103
```

Note: The 19.14 version will provide a much better error message:

- ORA-39442: Data Pump software update in progress

The DPload Patch speeds up future Data Pump patching by 50-80%

deep dive

DATA PUMP

with development

Photo by Maksym Kaharlytskyi on Unsplash

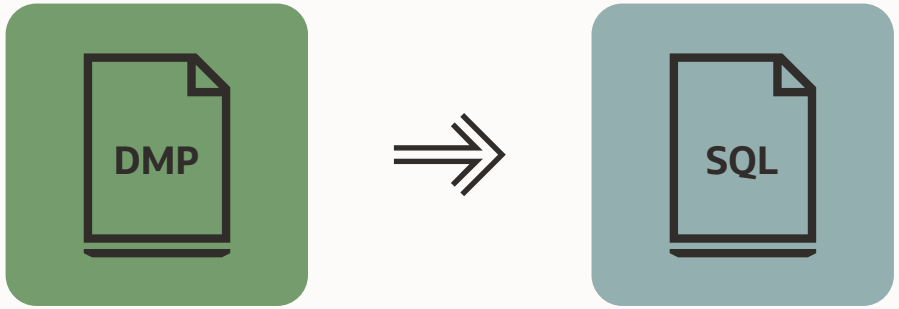
SQLFILES

You can extract metadata information from a dump file using parameter `SQLFILE`

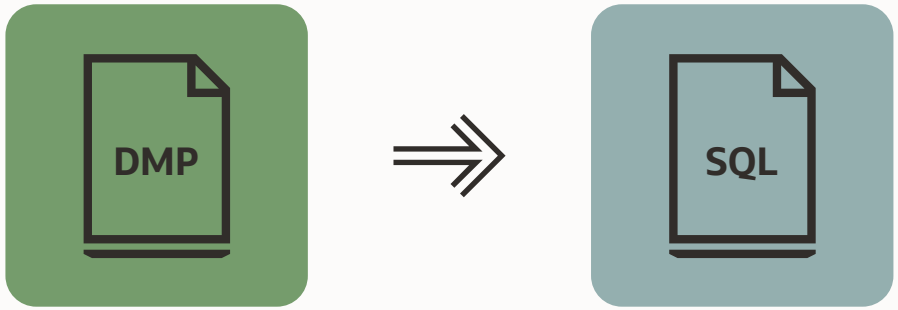
SQLFILE | Generate SQL Statements

Generate DDLs that `impdp` will execute

```
$ more import.par
...
sqlfile=all_statements.sql
...

$ impdp system parfile=import.par
```



SQLFILE | Generate SQL Statements



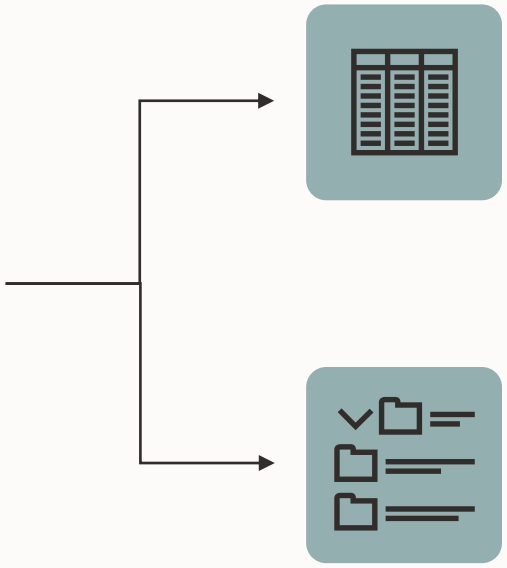

```
CREATE USER "TPCC" IDENTIFIED BY VALUES '...'
      DEFAULT TABLESPACE "TPCCTAB"
      TEMPORARY TABLESPACE "TEMP";
GRANT UNLIMITED TABLESPACE TO "TPCC";
GRANT "CONNECT" TO "TPCC";
GRANT "RESOURCE" TO "TPCC";
DECLARE
  TEMP_COUNT NUMBER;
  SQLSTR VARCHAR2(200);
BEGIN
  SQLSTR := 'ALTER USER "TPCC" QUOTA UNLIMITED ON "TPCCTAB"';
  EXECUTE IMMEDIATE SQLSTR;
EXCEPTION
  WHEN OTHERS THEN
    IF SQLCODE = -30041 THEN
      SQLSTR := 'SELECT COUNT(*) FROM USER_TABLESPACES
                 WHERE TABLESPACE_NAME = ''TPCCTAB'' AND CONTENTS =
''TEMPORARY''';
      EXECUTE IMMEDIATE SQLSTR INTO TEMP_COUNT;
      IF TEMP_COUNT = 1 THEN RETURN;
      ELSE RAISE;
      END IF;
    ELSE
      RAISE;
    END IF;
END;
/
```

Example

Creating big indexes

SQLFILE | Example

Imagine a schema



Tables

Small tables	10.000
Big tables	1

Indexes

Small indexes	10.000
Big indexes	1

SQLFILE | Example

Data Pump creates indexes with parallel degree 1

Many indexes are created simultaneously

Very efficient for many small indexes

Very inefficient for large indexes

SQLFILE | Example

Data Pump creates small indexes

You create big indexes with desired parallel degree

SQLFILE | Example

Find indexes of interest

```
SQL> select   segment_name, round(bytes/1024/1024/1024) as GB
     from     user_segments
     where    segment_type='INDEX'
     order by GB desc;
```

Exclude indexes from import

```
$ cat import.par
...
exclude=INDEX:"='BIG1','BIG2','BIG3'"
...

impdp ... parfile=import.par
```


SQLFILE | Example

Generate metadata for big indexes

```
$ cat import-sqlfile.par
...
include=INDEX:"='BIG1','BIG2','BIG3'"
sqlfile=index.sql
...

impdp ... parfile=import-sqlfile.par
```

Change parallel degree and create indexes

```
SQL> CREATE INDEX BIG1 .... PARALLEL n;
SQL> ALTER INDEX INDEX BIG1 .... PARALLEL 1;
...
```

SQLFILE | Example

Watch on YouTube

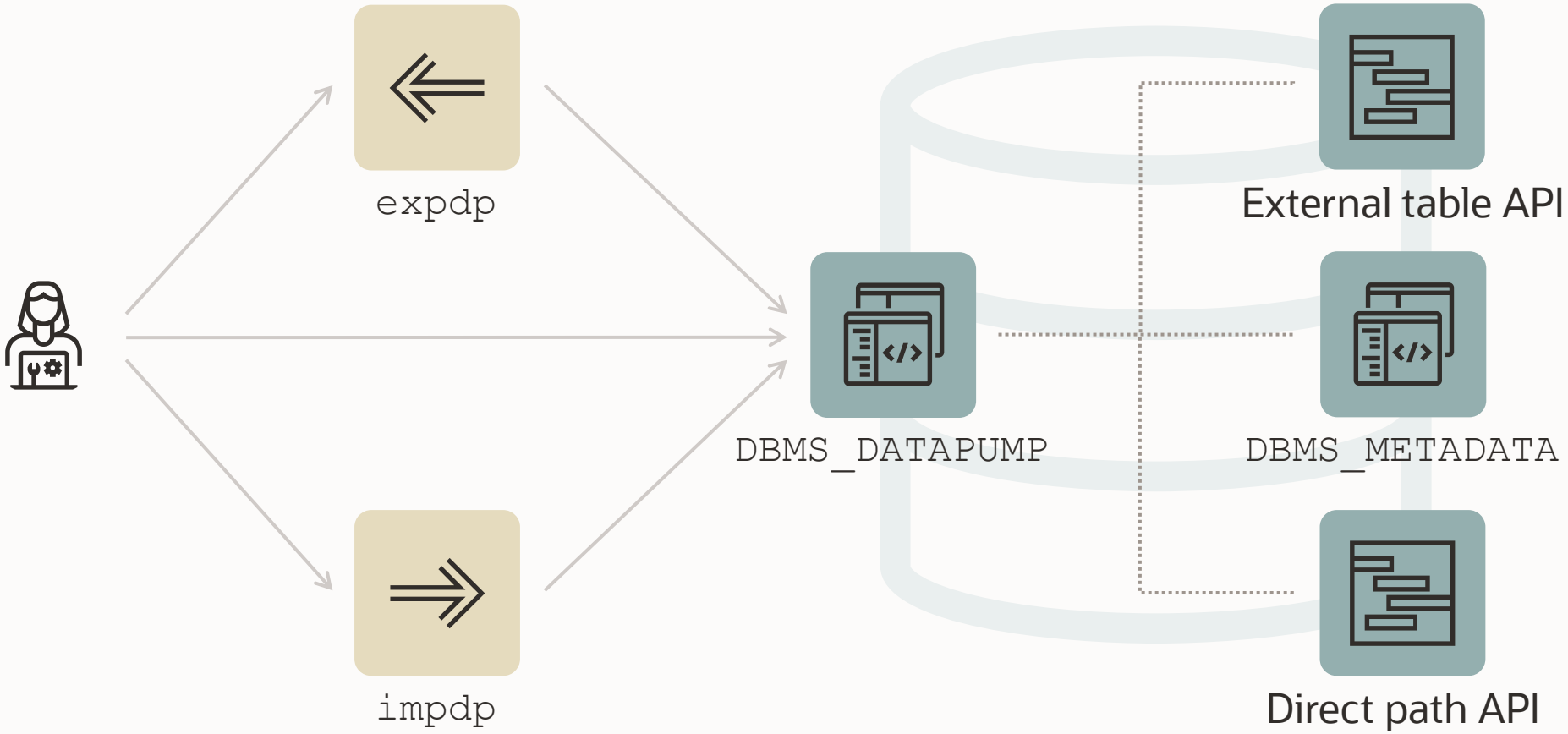
You can also get index definition from DBMS_METADATA.GET_DLL

Photo by Louis Hansel on Unsplash

DBMS_DATAPUMP

Usage

DBMS_DATAPUMP | Overview



DBMS_DATAPUMP | API

The Data Pump API (`DBMS_DATAPUMP`) is used many places:

- Zero Downtime Migration
- Enterprise Manager
- SQL Developer
- SQLcl

...

You can use it as well,
it is documented and supported

DBMS_DATAPUMP | API

Ideas:

- Use Data Pump functionality without installing a client
- Schedule export or imports using `DBMS_SCHEDULER`
- Dynamically build Data Pump jobs
- Integrate into automation tools (Ansible, Puppet)
- Accessible via ORDS / REST API as well
- Rename schema using a loopback database link
- Take a snapshot of a schema during application development

DBMS_DATAPUMP | Comparison

Client

```
expdp directory=mydir \
   logfile=exp.log \
   dumpfile=exp%u.dmp \
   schemas=app \
   parallel=4 \
   metrics=y \
   logtime=all
```

API

```
h1 := DBMS_DATAPUMP.OPEN(
        operation => 'EXPORT',
        job_mode => 'SCHEMA',
        remote_link => null,
        job_name => 'MY_JOB',
        version => null);

-- Create a Data Pump job to do a schema
-- export. Give it a meaningful name
```

DBMS_DATAPUMP | Comparison

Client

```
expdp directory=mydir \
   logfile=exp.log \
   dumpfile=exp%u.dmp \
   schemas=app \
   parallel=4 \
   metrics=y \
   logtime=all
```

API

```
DBMS_DATAPUMP.METADATA_FILTER(
   handle => h1,
   name => 'SCHEMA_EXPR',
   value => 'IN (''APP'')');

-- Specify the schema to be exported. We let
-- the object_path parameter default in this
-- call, so this applies to all objects in
-- the job
```

DBMS_DATAPUMP | Comparison

Client

```
expdp directory=mydir \
   logfile=exp.log \
   dumpfile=exp%u.dmp \
   schemas=app \
   parallel=4 \
   metrics=y \
   logtime=all
```

API

```
DBMS_DATAPUMP.ADD_FILE(
   handle => h1,
   filename => 'exp%u.dmp',
   directory => 'MYDIR',
   filetype=>DBMS_DATAPUMP.KU$_FILE_TYPE_DUMP_FILE);

-- Specify the dumpfile for the job using a
-- wildcard. The directory object must be
-- supplied for each file added to the job
-- FILETYPE defaults to dumpfile but we
-- specify it anyway to be clear
```

DBMS_DATAPUMP | Comparison

Client

```
expdp directory=mydir \
   logfile=exp.log \
   dumpfile=exp%u.dmp \
   schemas=app \
   parallel=4 \
   metrics=y \
   logtime=all
```

API

```
DBMS_DATAPUMP.ADD_FILE(
   handle => h1,
   filename => 'exp.log',
   directory => 'MYDIR',
   filetype=>DBMS_DATAPUMP.KU$_FILE_TYPE_LOG_FILE);

-- Specify the log file for the job. The directory
-- object must be supplied for each file added to
-- the job.
```

DBMS_DATAPUMP | Comparison

Client

```
expdp directory=mydir \
   logfile=exp.log \
   dumpfile=exp%u.dmp \
   schemas=app \
   parallel=4 \
   metrics=y \
   logtime=all
```

API

```
DBMS_DATAPUMP.SET_PARALLEL(
   handle => h1,
   degree => 4 );

-- Set the parallelism for the job
-- Or get a little creative

select value into parallel_degree
from   v$parameter
where  name='cpu_count';
DBMS_DATAPUMP.SET_PARALLEL(
   handle => h1,
   degree => parallel_degree);
```

DBMS_DATAPUMP | Comparison

Client

```
expdp directory=mydir \
   logfile=exp.log \
   dumpfile=exp%u.dmp \
   schemas=app \
   parallel=4 \
   metrics=y \
   logtime=all
```

API

```
DBMS_DATAPUMP.SET_PARAMETER(
   handle => h1,
   name => 'METRICS',
   value => 1);

DBMS_DATAPUMP.SET_PARAMETER(
   handle => h1,
   name => 'LOGTIME',
   value => 'ALL');

-- set other job parameters
```

DBMS_DATAPUMP | Comparison

Client

API

```
DBMS_DATAPUMP.START_JOB (
   handle => h1);

-- now start the job
-- wait for it to complete

DBMS_DATAPUMP.WAIT_FOR_JOB (
   handle => h1,
   job_state);
```

Use 10046 trace to generate DBMS_DATAPUMP calls

Data Pump | Generate PL/SQL

1. Enable SQL trace on a test database

```
SQL> alter system
     set event='10046 trace name context forever, level 4';
```

2. Execute your Data Pump command

```
$ impdp system ... parfile=import.par
```

3. Examine the trace file

```
$ vi ORCL_ora_12345.trc
```

Pro tip: Grep for *DBMS_DATAPUMP* to find the right trace file

The documentation has many good examples on using `DBMS_DATAPUMP`

Photo by SpaceX on Unsplash

Restartability

Export and Import

Data Pump export and import jobs can be stopped and restarted again

Restart | Export

Export can be restarted after the ESTIMATE phase has been completed

- Tracked in the Control Table
- Workers create/update records with COMPLETION_TIME
- Restart: Workers check for records with missing COMPLETION_TIME

OBJECT_TYPE	START_TIME	COMPLETION_TIME
TABLESPACE	12-SEP-2021:9:04.01	12-SEP-2021:9:05.23
USER	12-SEP-2021:9:05.27	

- Example
  - USER object is incomplete
  - Will be removed and restarted

Restart | Import

Import can be restarted using the Control Table

- Workers track import status via `STATE` and `STATUS`

OBJECT	OBJECT_SCHEMA	OBJECT_NAME	PROCESSING_STATE	PROCESSING_STATUS
TABLE	SCOTT	EMP	W	C
TABLE	SCOTT	DEPT	U	C
INDEX	SCOTT	IDX1_EMP	R	C
INDEX	SCOTT	IDX1_DEPT	R	C

  - R = objects were Retrieved (exported)
  - C = objects are Current (successfully imported
  - W = objects are Written (imported)
  - U = objects are Unknown (import started but did not finish)

Photo by Erik Mclean on Unsplash

Interactive Command Mode

Usage

Interactive Command Mode | Overview

Interact with a running job

- Changing parameters
- Changing attributes
- Monitoring
- Starting/stopping jobs and workers

```
Export> status

Job: SYS_EXPORT_SCHEMA_01
  Operation: EXPORT
  Mode: SCHEMA
  State: EXECUTING
  Bytes Processed: 13,454,650,144
  Percent Done: 85
  Current Parallelism: 1
  Job Error Count: 0
  Job heartbeat: 5
  Dump File: /tmp/dpdir/exp01.dmp
    bytes written: 13,454,696,448
  Dump File: /tmp/dpdir/exp%l.dmp

Worker 1 Status:
  Instance ID: 1
  Instance name: DB19
  Host name: hol.localdomain
  Object start time: Thursday, 17 February, 2022 12:16:04
  Object status at: Thursday, 17 February, 2022 12:18:24
  Process Name: DW00
  State: EXECUTING
  Object Schema: APP
```

Interactive Command Mode | Overview

Different commands are available for exports and imports.

Command	Mode	Description
`PARALLEL=n`	Both	Change the parallelism for current job. Increases almost immediately.
`STATUS`	Both	Get the job and worker status. Includes Operation, Mode, State, Percent Done, and Current Parallelism .
`STATUS=120`	Both	As above but refreshes every 120 second
`FILESIZE=n`	Export	Changes the file size (in bytes) of the dump files. Optionally specify denominator, e.g., FILESIZE=5G
`ADD_FILE=name`	Export	Adds an additional dump file. Or a dump file pattern, e.g., ADD_FILE=more_files%L.dmp
`TRACE=nnn`	Both	Adds tracing, see MOS ID 286496.1 for details

More commands are found in the documentation

Interactive Command Mode | Overview

Two ways to start Interactive Command Mode

- Break from Data Pump client (`expdp` and `impdp`)

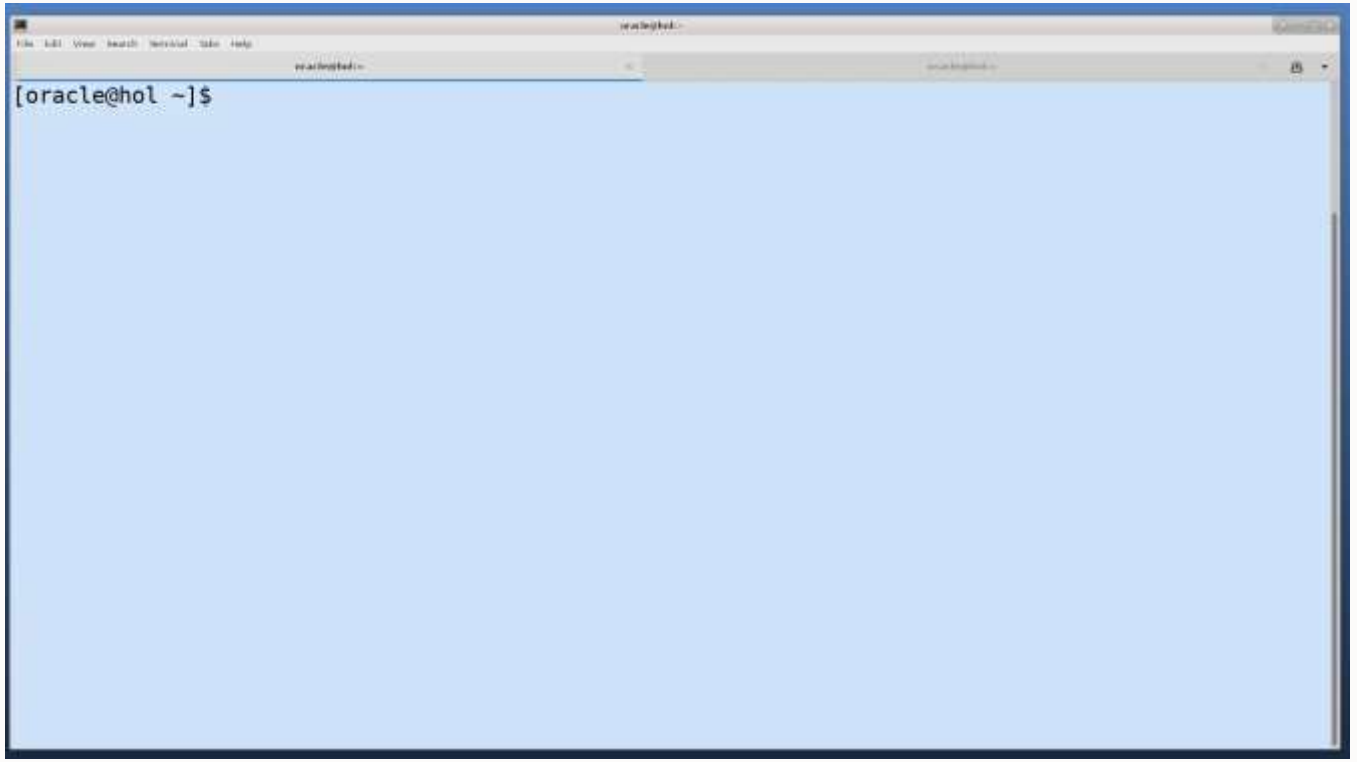
  Hit CTRL + C while Data Pump is logging to console

- Attach to a running job

```
expdp .... attach=<job name>

impdp .... attach=<job name>
```

Pro tip: Type `help` to see all commands in Interactive Command Mode

Interactive Command Mode | Demo



Watch on YouTube

Photo by Mitchell Luo on Unsplash

Troubleshooting

In case of an issue ...

Data Pump is not doing what you'd expect?

What should you check and collect?

Troubleshooting | Step-by-step

Log files

Troubleshooting | Log files

Log files are essential

- Use `METRICS=YES` and `LOGTIME=ALL`
  - Without log parameters:

```
Processing object type DATABASE_EXPORT/FINAL_POST_INSTANCE_IMPCALLOUT/MARKER
Processing object type DATABASE_EXPORT/AUDIT_UNIFIED/AUDIT_POLICY_ENABLE
Processing object type DATABASE_EXPORT/POST_SYSTEM_IMPCALLOUT/MARKER
. . exported "SYS"."KU$_USER_MAPPING_VIEW"                5.890 KB      25 rows
. . exported "SYSTEM"."REDO_DB"                           25.59 KB       1 rows
```

  - With log parameters:

```
DATABASE_EXPORT/FINAL_POST_INSTANCE_IMPCALLOUT/MARKER
02-NOV-21 19:43:56.064: W-1      Completed 1 MARKER objects in 0 seconds
02-NOV-21 19:43:59.171: W-1 Processing object type DATABASE_EXPORT/AUDIT_UNIFIED/AUDIT_POLICY_ENABLE
02-NOV-21 19:43:59.195: W-1      Completed 2 AUDIT_POLICY_ENABLE objects in 0 seconds
02-NOV-21 19:43:59.380: W-1 Processing object type DATABASE_EXPORT/POST_SYSTEM_IMPCALLOUT/MARKER
02-NOV-21 19:43:59.387: W-1      Completed 1 MARKER objects in 0 seconds
02-NOV-21 19:43:59.830: W-1 . . exported "SYS"."KU$_USER_MAPPING_VIEW"    5.890 KB  25 rows in 0 seconds using external_table
02-NOV-21 19:43:59.923: W-1 . . exported "SYSTEM"."REDO_DB"               25.59 KB   1 rows in 0 seconds using direct_path
```

Troubleshooting | Log files

Check `alert.log` and upload it with an SR

```
2022-02-21T11:31:23.315021+01:00
db_recovery_file_dest_size of 18432 MB is 1.23% used. This is a
user-specified limit on the amount of space that will be used by this
database for recovery-related files, and does not reflect the amount of
space available in the underlying filesystem or ASM diskgroup.
2022-02-21T11:31:25.810983+01:00
DM00 started with pid=80, OS id=17226, job DPUSER.SYS_EXPORT_SCHEMA_01
2022-02-21T11:31:56.980017+01:00
Thread 1 advanced to log sequence 20 (LGWR switch),  current SCN: 6660216
  Current log# 2 seq# 20 mem# 0: /u02/oradata/DB19/redo02.log
2022-02-21T11:31:57.197532+01:00
ARC1 (PID:16810): Archived Log entry 3 added for T-1.S-19 ID 0x31223092 LAD:1
2022-02-21T11:32:01.650969+01:00
TABLE SYS.WRP$_REPORTS: ADDED INTERVAL PARTITION SYS_P865 (4435) VALUES LESS THAN (TO_DATE(' 2022-02-22 01:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
TABLE SYS.WRP$_REPORTS_DETAILS: ADDED INTERVAL PARTITION SYS_P866 (4435) VALUES LESS THAN (TO_DATE(' 2022-02-22
01:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
TABLE SYS.WRP$_REPORTS_TIME_BANDS: ADDED INTERVAL PARTITION SYS_P869 (4434) VALUES LESS THAN (TO_DATE(' 2022-02-21
01:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
2022-02-21T11:32:12.822559+01:00
ALTER SYSTEM SET streams_pool_size=256M SCOPE=BOTH;
```

Troubleshooting | Log files

Check for Data Pump trace files in `$ORACLE_BASE/diag/rdbms/../../trace`

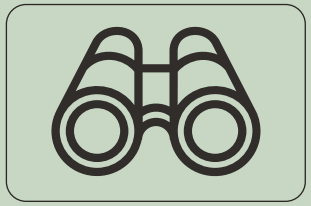
```
Trace file /u01/app/oracle/diag/rdbms/db19/DB19/trace/DB19_dm00_17468.trc
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.14.0.0.0
Build label:    RDBMS_19.14.0.0.0DBRU_LINUX.X64_211224.3
ORACLE_HOME:    /u01/app/oracle/product/19
System name:    Linux
Node name:      hol.localdomain
Release:        5.4.17-2136.302.7.2.1.el7u
Version:        #2 SMP Tue Jan 18 13:44:44
Machine:        x86_64
Instance name: DB19
Redo thread mounted by this instance: 1
Oracle process number: 58
Unix process pid: 17468, image: oracle@hol


*** 2022-02-21T11:33:25.374300+01:00
*** SESSION ID:(253.19643) 2022-02-21T11:3
*** CLIENT ID:() 2022-02-21T11:33:25.37431
*** SERVICE NAME:(SYS$USERS) 2022-02-21T11
*** MODULE NAME:(Data Pump Master) 2022-02
*** ACTION NAME:(SYS_EXPORT_SCHEMA_01) 202
*** CLIENT DRIVER:() 2022-02-21T11:33:25.374327+01:00
```

```
================== skgfqio Request Dump ======================
OSD Context: aiopend=0, aiodone=0, limfsiz=42949672951, sigwinchslot=0
Request flags: READ
- - - - skgfrrq request element 1 - - - -
BLOCKNO = 1
IOV: addr=0x0x6ef687d8, fib=0x0x6d0d2478, maxaio=0, seal=0x45726963,
fd=260
     fsync required?=TRUE, offset=18446744073709551615, aiopend=0
FIB: addr=0x0x6d0d2478, lblksiz=0, ora ftype=18, pblksiz=512, filsiz=1
     maxvec=16, fname=/home/oracle/dp/export.log, serr=0, seal=0x45726963
     fstype=0x58465342, unix ftype=0x81a4, last
block=18446744073709551615
IOSB: addr=0x0x7f0da829dc38, status=3, time=0, qstatus=8,AIO start
time=139696632618072
err=27072 errno=25 ose[0]=4 ose[1]=1 ose[2]=333
BUFFER: addr=0x0x7f0da76b2000, len=4096
```

Troubleshooting | Step-by-step

Log files

Database Views



Troubleshooting | Database Views

Monitor a Data Pump process in DBA_DATAPUMP_JOBS

- MOS Note: 1471766.1 - How To Monitor The Progress Of Data Pump Jobs
- Use parameter JOB_NAME with expdp and impdp

```
SQL> select * from DBA_DATAPUMP_JOBS;

OWNER_NAME  JOB_NAME  OPERATION  JOB_MODE  STATE  DEGREE  ATTACHED  DATAPUMP_SESSIONS
----------- --------- ---------- --------- ------ ------- --------- -----------------
SYS         MYEXPDP1  EXPORT     FULL      EXECUT       1         1                 3
```

Troubleshooting | Database Views

Monitor a Data Pump process in `DBA_DATAPUMP_SESSIONS`

- MOS Note: 1528301.1 - Finding Out The Current SQL Statement A Data Pump Process Is Executing
  - Use the script from MOS Note: 1528301.1 to:
    - Diagnose possible hangs
    - Slow Data Pump processes

Troubleshooting | Database Views

Monitor a Data Pump process in `V$SESSION_LONGOPS`

- MOS Note: 455720.1 - How can we monitor a DataPump Job's Progress?
- Use parameter `JOB_NAME` with expdp and impdp

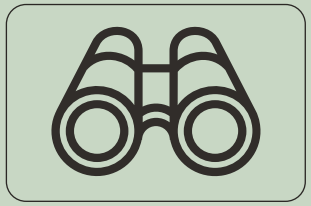
```
select sid, serial#, sofar, totalwork
from   V$SESSION_LONGOPS
where  opname = '<your export job name>' and
       sofar != totalwork;
```

  - `sofar`:
    Shows how much work in MB has been done so far in relation to `totalwork`
  - `totalwork`:
    Shows the total amount of work in MB

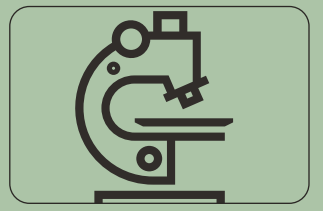
Troubleshooting | Step-by-step

Log files

Database Views



Tracing



Troubleshooting | TRACE

MOS Note: 286496.1 - DataPump Parameter TRACE - How to Diagnose Oracle Data Pump

Export/Import DataPump Parameter TRACE - How to Diagnose Oracle Data Pump (Doc ID 286496.1)

In this Document

Purpose
Scope
Details
1. Introduction.
2. How to create a Data Pump trace file ? Parameter: TRACE
3. How to start tracing the Data Pump job ?
4. How are Data Pump trace files named, and where to find them ?
5. How to get a detailed status report of a Data Pump job ? Parameter: STATUS
6. How to get timing details on processed objects ? Parameter: METRICS
7. How to get SQL trace files of the Data Pump processes ?
8. How to get header details of Export Data Pump dumpfiles ?
9. How to get Data Definition Language (DDL) statements ? Parameter: SQLFILE
10. How to get the DDL both as SQL statements and as XML data ?
Additional Resources
References

Troubleshooting | TRACE

Best practices

- Requires privileged user - DBA role or EXP[/IMP]_FULL_DATABASE Role
- Ensure `MAX_DUMP_FILE_SIZE` is large enough to capture the trace (default=unlimited)

Three options

- TRACE parameter
- TRACE in interactive mode
- TRACE event in SPFILE

Troubleshooting | TRACE

TRACE parameter

- Allows tracing of each individual component of Data Pump
- Bitmap format

```
trace=1FF0B00
```

- Most important TRACE bitmaps:
  - `1FF0300 – Recommended Tracing`
  - `1FFF0300 – Full Tracing`
  - For a comprehensive list and further explanation, see MOS Note: 286496.1

- Generates two trace files:
  - `<SID>_dm<number>_<process_id>.trc` - Control process trace
  - `<SID>_dw<number>_<process_id>.trc` - Worker trace file (one for each worker)

Troubleshooting | TRACE

TRACE option in interactive mode

- Type ^C while the job is running
- Add tracing
- Resume job

```
impdp user/password attach=user.imp_job_1 trace=400300


Import> start_job [=SKIP_CURRENT=YES]
```


Troubleshooting | TRACE

TRACE event 39089

- Add event to SPFILE/PFILE or set it via ALTER SYSTEM

```
EVENT="39089 trace name context forever, level 0x300"
```

```
ALTER SYSTEM SET EVENTS = '39089 trace name context forever, level 0x300' ;
ALTER SYSTEM SET EVENTS = '39089 trace name context off' ;
```

- For further explanation, see MOS Note: 286496.1

Troubleshooting | TRACE

TRACE event 10046

- Multi-purpose SQL trace event
- Usually set for the worker process(es)
- For further explanation, see:
  MOS Note: 376442.1 - How To Collect 10046 Trace (SQL_TRACE) Diagnostics for Performance Issues

Photo by Zbynek Burival on Unsplash

105 minutes – Feb 4, 2021

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

120 minutes – Mar 16, 2021

Episode 5

Migration Strategies – Insights, Tips and Secrets

120 minutes – Mar 25, 2021

Episode 6

Move to the Cloud – Not only for techies

115 minutes – Apr 8, 2021

Episode 7

Cool Features – Not only for DBAs

110 minutes – Jan 14, 2021

Episode 8

Database Upgrade Internals – and so much more

110 minutes – Feb 11, 2021

Episode 9

Performance Testing Using the Oracle Cloud for Upgrades and Migrations

90 minutes – May 19, 2021

NEW Episode 10

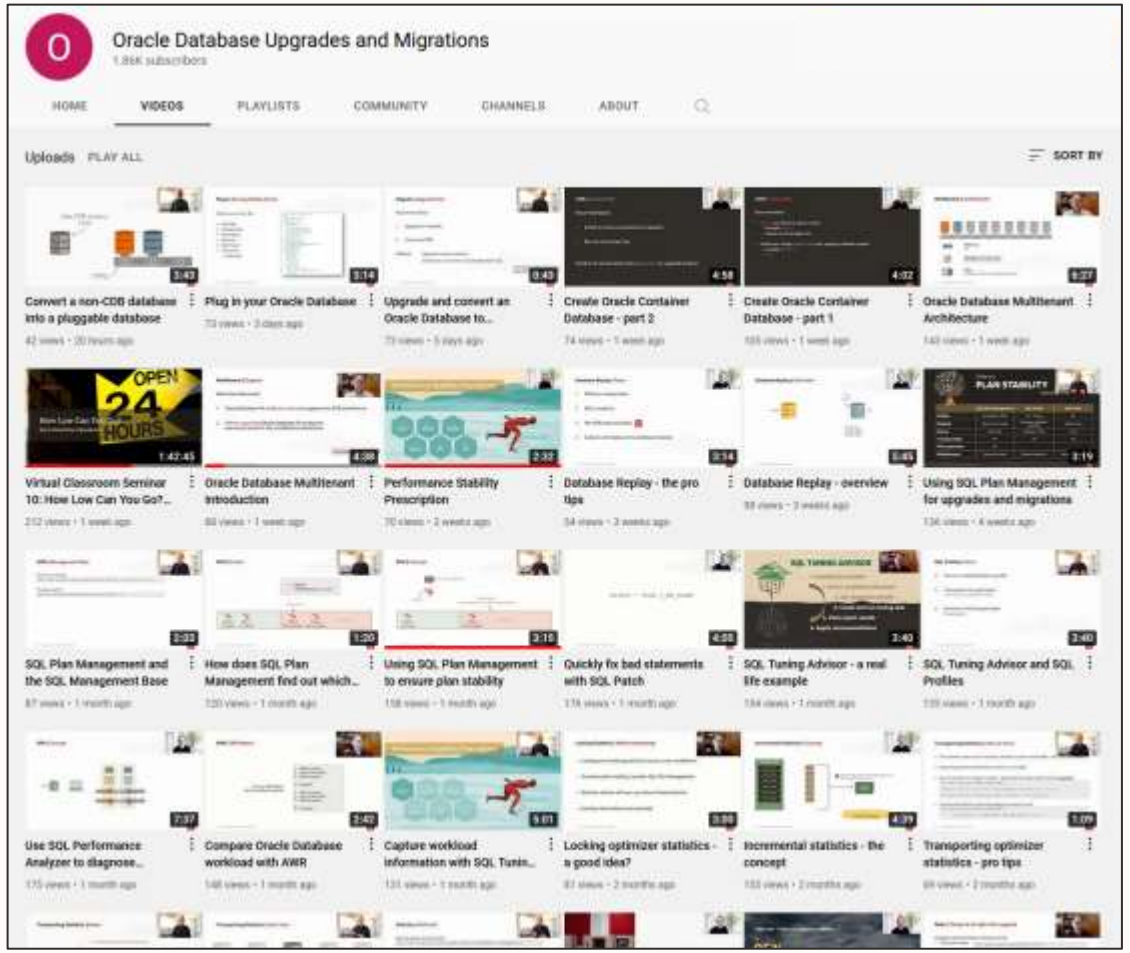
How Low Can You Go? Minimal Downtime Upgrade Strategies

100 minutes – Oct 26, 2021

Recorded Web Seminars

https://MikeDietrichDE.com/videos

YouTube | Oracle Database Upgrades and Migrations



Link

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

SCAN ME

THANK YOU

Visit our blogs:

https://MikeDietrichDE.com

https://DOHdatabase.com

https://www.dbarj.com.br/en

THANK YOU

Webinars:

https://MikeDietrichDE.com/videos

YouTube channel:

OracleDatabaseUpgradesandMigrations

THANK YOU

AUTOUPGRADE 2.0
New features and use cases

May 5, 2022 – 10:00h CET

THANK YOU