ORACLE

Oracle Data Pump

Best Practices

May, 2024, Version 2.0

Public

ORACLE

Table of contents

Introduction

There are multiple approaches for migrating on-premises Oracle databases. A common method is Oracle Data Pump, a feature of Oracle Database since Release 10g and successor to the Oracle Export and Import (`exp/imp`) utilities in Release 9i and earlier. Oracle Data Pump is useful for migrating data among schemas, databases of different versions and on different operating systems, and from on-premises to on-premises and to Oracle Cloud. Oracle Data Pump should be considered for migrating an on-premises Oracle database or another Oracle Cloud database to Oracle cloud, including under the following circumstances:

- The source Oracle Database is release 10g or higher. Older databases must perform two hops, first use the original Export and Import utilities that came with the database to migrate to 10g then use Data Pump.
- Migrating data cross-endian
- Migrating from a non-CDB Oracle database to Oracle Multitenant
- Migrating with changes to database structure, iuncluding encryption, compression, character sets, IOTs to heap tables, remapping tablespaces, or schemas, converting BasicFile LOBs to SecureFile LOBs
- Including and/or excluding objects in and/or from the migration
- combining a migration and a version upgrade

Migrating data using Oracle Data Pump is a three-step process. Assuming your cloud instance and PDB are created:

- export the on-premises database using expdp,
- copy the dump files to the target system or to the Oracle Object Store, if required, and
- import into the cloud PDB using impdb.
- Or combine these steps into a Network mode import using Database links and avoid the dumpfiles.

This Technical Brief describes some best practices for using Oracle Data Pump to help make the process go smoothly and successfully.

The following paragraphs describe best practices for using Oracle Data Pump to accomplish a migration.

Note: The term "migration" can also be used when discussing the move of data from a non-Oracle database into Oracle. This technical brief will cover migrations only when both the source and destination are Oracle databases.

Obtaining Support and the Data Pump Bundle Patch

Oracle recommends using the latest Data Pump Bundle patch to benefit from important fixes. It is not included in the Database Release Update. Oracle is investigating this for a future release. See this MOS note for details, "Data Pump Recommended Proactive Patches For 19.10 and Above - 2819284.1".

Helpful information to include in a request for support, includes:

1. Details about the source/target environment, including the database version & and Release Update and the Data Pump client version
2. Database and job information
   - STREAMS_POOL_SIZE setting
   - Whether statistics have been run recently relative to the amount
   - The error message(s) encountered if they aren't in the logfile
   - Table definition if the error is table related
   - Whether the LOB storage is `SECUREFILE` or `BASICFILE`
3. Relevant files
   - The parfile.par or command used to invoke the export or import job
   - The logfile.log created using with diagnostics parameters `METRICS=Y LOGTIME=ALL`
   - The alert_SID.log found in the ADR directory specified by the parameter DIAGNOSTIC_DEST in the initialization parameter file
   - An AWR report for a representative period, approximately 15 minutes

Do Not Invoke Export Using SYS as SYSDBA

`SYSDBA` is used internally and has specialized functions; it doesn't behave like generalized users. `AS SYSDBA` has some restrictions on the use of parallel DML. In addition, in some situations `AS SYSDBA` can circumvent some internal consistency checks. This is not desirable.

You should not need to start Export as SYSDBA except at the request of Oracle technical support or when importing a transportable tablespace set.

Use A Parameter File

A parameter file, also referred to as a "parfile", enables you to specify command-line parameters in a file for ease of reuse. It also helps avoid typographical errors from typing long ad hoc Data Pump commands on the command line, especially if you use parameters whose values require quotation marks.

Here is an example of a parfile named `my_data_pump_parfile.par` that you might use to migrate to Autonomous Database::

```
PARALLEL=16
DUMPFiLE=adbdump%L.dmp
FILESIZE=10GB
METRICS=Y
LOGTIME=ALL
JOB_NAME=ADBMIGRATION
DIRECTORY=DP_DIR
EXCLUDE=STATISTICS
COMPRESSION=ALL
COMPRESSION_ALGORITHM=MEDIUM
FLASHBACK_SCN=SYSTIMESTAMP
```

The command to execute the par file looks like this:

```
expdp parfile=my_data_pump_parfile.par
```

Notes:

Prior to Release 12.2 use `DUMPFULE=adbdump_1%U.dmp, adbdump_2%U.dmp...` to allow 99 dumpfiles for each prefix

Beginning with Release 12.1 use `LOGTIME=ALL`

For Autonomous Database the maximum dumpfile size is 10GB. It okay to specify a smaller size.

As an alternative to `flashback_scn`, If the system is static during maintenance and you want to make sure that no job starts during export, set `job_queue_process=0`. It may be necessary to increase the undo retention. By default it is set to 900 seconds. If you're worried about connections to the database during the export job, use a restricted session.

Make a Consistent Data Pump Export

By default, Oracle Data Pump preserves consistency within a single database table. For example, if you export a table that has 1000 partitions, the exported table will be consistent as of the specific System Change Number (SCN) at which you started the export. When exporting multiple tables, the next table exported would then be consistent as of a different SCN. For any export of more than one table, you will probably want your export dump file to represent all of the objects as of the same SCN.

This can be accomplished by using either `FLASHBACK_SCN=<scn>` or `FLASHBACK_TIME=<timestamp>` to enable the Flashback Query utility. A particularly convenient approach is to specify `FLASHBACK_TIME=SYSTIMESTAMP`.

Using `FLASHBACK_SCN`, the export operation is performed with data that is consistent up to the specified SCN. For example, this command assumes that an existing SCN value of 384632 exists. It exports the `hr` schema up to SCN 384632:

```
expdp hr DIRECTORY=dpump_dir1 DUMPFILE=hr_scn.dmp FLASHBACK_SCN=384632
```

Using `FLASHBACK_TIME=<timestamp>`, the export operation is performed with data that is consistent up to the SCN that most closely matches the specified time. For example, this export operation is performed with data that is consistent up to the SCN closest to the specified time.

```
FLASHBACK_TIME="TO_TIMESTAMP('27-10-2012 13:16:00', 'DD-MM-YYYY HH24:MI:SS')"
```

When specifying `FLASHBACK_TIME=SYSTIMESTAMP`, the timestamp will be that of the current system time. Finally, you can still use the release 11.2 legacy interface, `CONSISTENT=Y` which is translated to `FLASHBACK_TIME=SYSTIMESTAMP`.

You can change an Oracle Data Guard standby database to a temporary snapshot to do an export. However, Data Pump must create a control table to coordinate the export. This requires a read-write instance. This can't be done

with Active Data Guard and a DML Redirect operation because it supports only DML, DDL is not supported for the `CREATE TABLE` command.

GoldenGate instantiation filtering can also help. It no longer requires `FLASHBACK_<SCN|TIME>` on export. This means that GoldenGate will know the SCN of each table when it was exported, and will start replicating changes as of that SCN on a table-by-table basis. That doesn't solve the problem 100%, because you could still see cases where table A is exported as SCN X and table B is exported at SCN Y which is some time later than X. If there are dependencies such as constraints across those tables, and if table B has additional rows that depend on not-yet-replicated rows in table A, then those constraint validations might fail.

Exclude Statistics from Export and Import

Oracle recommends that you do not export statistics during export. This will provide better performance on both export and import, even accounting for the need to gather statistics after the import. You can exclude statistics from an export operation using the `EXCLUDE=STATISTICS` parameter or `EXCLUDE=TABLE_STATISTICS,INDEX_STATISTICS` for Transportable Tablespaces. Instead, follow the best practice of either creating fresh statistics on the target database or using a `DBMS_STATS` staging table for transporting statistics.

Use Diagnostics

Always use the Logfile Diagnostic Parameters

Timestamp the messages that are displayed during an export operation using `LOGTIME=ALL`. This parameter is available beginning with Oracle Database release 12.1. Having timestamps on every line in the logfile helps when assessing export and import performance.

Record the details about the objects being moved, the elapsed time and the worker assignments in the Oracle Data Pump log file using the parameter `METRICS=YES`.

Example:

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
```

Check the Alert log

Check the `alert.log` for activity during the Data Pump job.

Monitor Processes with Database Views

Monitor a Data Pump process using the following views, especially if a Data Pump process seems slow or unresponsive:

- `DBA_DATAPUMP_JOBS`. Refer to MOS Note: 1471766.1 - How To Monitor The Progress Of Data Pump Jobs.
- `DBA_DATAPUMP_SESSIONS`. Refer to MOS Note: 1528301.1 - Finding Out The Current SQL Statement A Data Pump Process Is Executing

- V$SESSION_LONGOPS. Refer to MOS Note: 455720.1 - How can we monitor a DataPump Job's Progress?

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

Use Data Pump Trace

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

Tracing requires a privileged user with either the DBA role or the EXP_FULL_DATABASE / IMP_FULL_DATABASE role. Ensure MAX_DUMP_FILE_SIZE is large enough to capture the trace (default=unlimited).

Use one of three ways to initiate a Data Pump trace. Refer to MOS Note: 286496.1

- Set the TRACE parameter with the recommended bitmap: trace=1FF0300.
- Set the TRACE in interactive mode:
  impdp user/password attach=user.imp_job_1 trace=1FF0300
- Set a TRACE event in the SPFILE/PFILE or set it via ALTER SYSTEM

Use SQL Trace

Gather a 10046 trace for SQL_TRACE information to diagnose query performance issues. Refer to MOS Note: 376442.1 - How To Collect 10046 Trace (SQL_TRACE) Diagnostics for Performance Issues

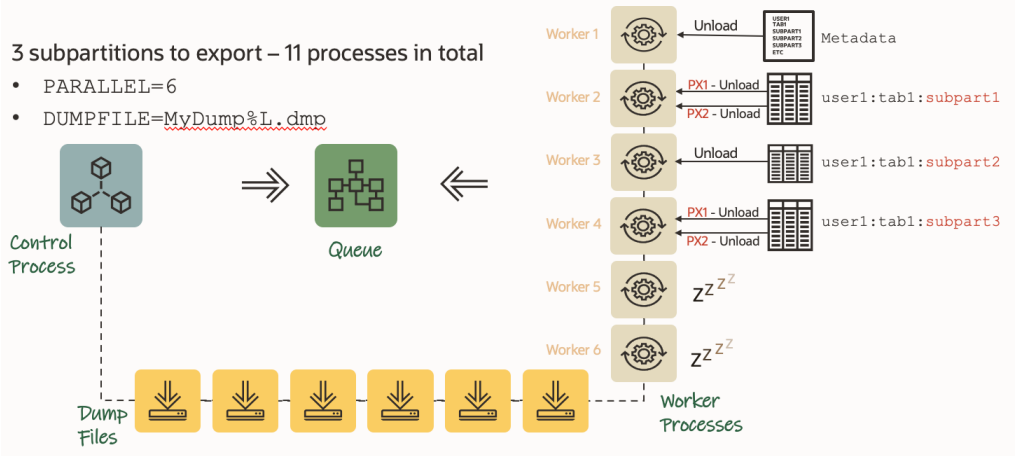
Improve Performance with Parallelism and Statistics

Use Parallelism

Accomplish more work in less time using parallelism. A new Data Pump Job consists of at least two background processes: a Control process and a Worker process, and 2 sessions. The Data Pump PARALLEL parameter creates additional worker processes for an export or import.

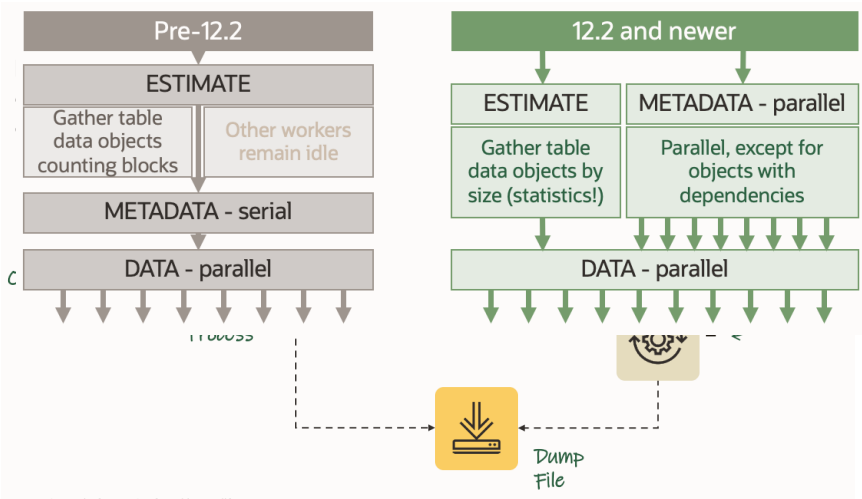
The PARALLEL=n parameter specifies the maximum number of worker processes and parallel query (PX) processes that can be active for the export or import job. This count does not include the Control Process. Typically, the value for n should be twice the number of CPU cores but adjust if need be.

Data Pump uses two kinds of parallelism.



One worker is assigned by the Control process to a small table or to a partition or sub-partition. This is called Inter-table parallelism. A worker can also coordinate one or more Parallel Execution (PX, formerly called a PQ slave) processes used for each large partition or a large unpartitioned table (Intra-table parallelism.) A worker that is coordinating PX processes does not count toward the maximum degree of parallelism for the job.

Beginning with Release 12.2, metadata export happens concurrently with estimate phase for table data.

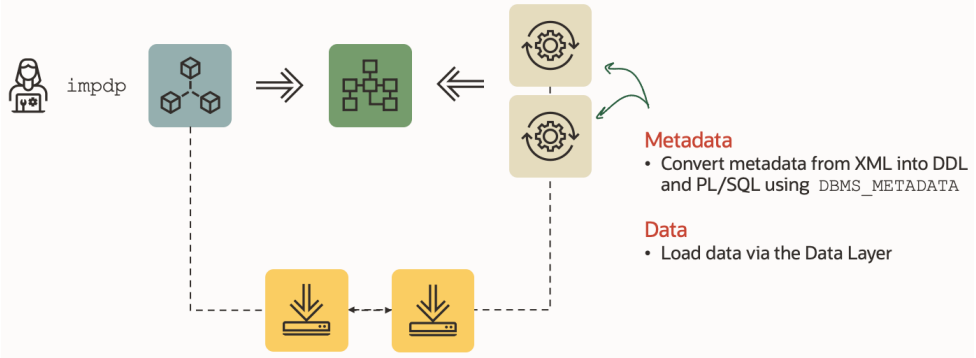


Most metadata & data objects are exported in parallel when PARALLEL=2 or greater. Oracle Data Pump has imported package bodies in parallel for several releases. Beginning with Release 12.2 and higher, for export and import by dumpfile, Oracle Data Pump imports most metadata and most database objects in parallel. Database metadata objects that have dependencies are still imported in a serial fashion, such as types (due to inheritance), schemas and procedural actions. The `PARALLEL` parameter also determines how many indexes are created in parallel, each index having one degree of parallelism.

If you are using Oracle Database release 11.2.0.4 or 12.1.0.2, you can apply the fix bug 22273229 to enable parallel import of constraints and indexes.

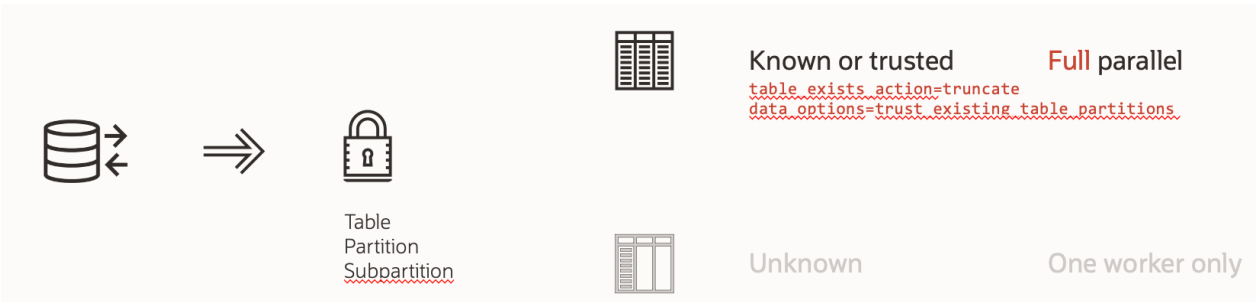
NOTE: *Parallel metadata export and import for Transportable Tablespace is available beginning with Oracle Database 21c.*

During import you may set `PARALLEL` to a different value than the setting used for export.



An import worker process uses the `DBMS_METADATA` API to convert metadata that was exported in XML documents to DDL and PL/SQL for import. One XML document is allocated to a worker at a time. It uses the Data Layer to load (import) or move (network mode import) the data. Network mode Import uses `DBMS_METADATA` API to convert source database metadata for execution and moves the data to the target.

Data Pump import must acquire a lock on a table, partition or subpartition being loaded.



The Data Pump `data_options=TRUST_EXISTING_TABLE_PARTITIONS` parameter confirms that the partitioning scheme on the existing target table is identical to that of the table being imported from the source, so full parallelism is used, otherwise one worker will access the table.

Specify wildcards to create multiple Dumpfiles

When using parallelism, use the `%L` or `%U` (legacy) substitution variable when specifying the dumpfile name. This enables simultaneous parallel writes to multiple logfiles. Also, multiple dump files of a smaller size can be easier to managc and copy.

Beginning with Release 12.2 the `%L wildcard` can be used to create more than the ninety-nine files created by `%U`. It expands the file names into a 3-digit to 10-digit, variable-width integer.

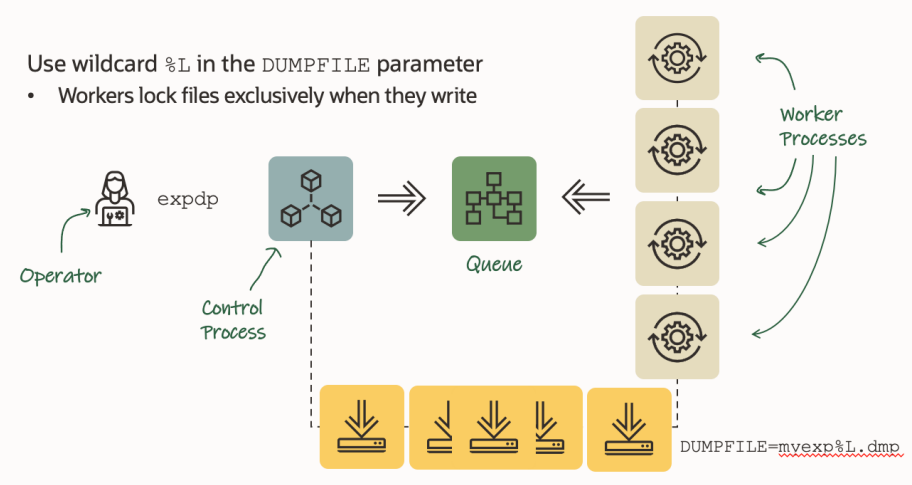
```
dumpfile=dumpfile%L.dmp
filesize=n
```

The legacy `%U` is the only wildcard available before Release 12.2. It can create one to ninety-nine files. It expands the file names into a 2-digit, fixed-width, incrementing integer.

```
dumpfile=dumpfile%U.dmp
filesize=n
```

If more than 99 dumpfiles are needed to migrate prior to Release 12.2, you can create multiple dumpfile filename prefixes and continue to use the %U wildcard, for example dump_01_%U.dmp, dump_02_%U.dmp, etc.

If you don't specify a wildcard in the dumpfile name, the dumpfile can restrict parallel workers. They will lock and write to the logfile sequentially.



Not specifying a wildcard can impact performance because each worker writing to the dumpfile will take an exclusive lock on the file and block the other worker processes.

Gather Statistics before and after Data Pump Operations

It is helpful having accurate statistics prior to an export operation for the best possible export performance. It is also helpful to gather statistics after import. The frequency of statistics updates that is needed to keep statistics current depends on how volatile the database is. Statistics gathering includes both dictionary statistics and object statistics. Dictionary statistics are used when data pump filters, orders, and collects metadata objects at various stages of the export. Object statistics are used to estimate table and index sizes, which helps produce optimal ordering and parallelism.

Concurrent with metadata export, table sizes are estimated and ranked from largest to smallest for parallel export. The table sizes are estimated using statistics. You can collect statistics using the `dbms_stats` package, with the `gather_table_stats, gather_schema_stats`, or `gather_database_stats` procedure.

Oracle recommends using `gather_schema_stats` because this procedure will gather stats on all objects regardless of staleness information.

Example:

```
SQL> BEGIN
        DBMS_STATS.GATHER_SCHEMA_STATS('SYS');
        DBMS_STATS.GATHER_SCHEMA_STATS('SYSTEM');
     END;
$ORACLE_HOME/perl/bin/perl $ORACLE_HOME/rdbms/admin/catcon.pl \
  -l /tmp \
  -b gatherstats -- \
  --x"begin dbms_stats.gather_schema_stats('SYS');
dbms_stats.gather_schema_stats('SYSTEM'); end;"
```

Note: Be aware, especially for production systems, that gathering statistics can be time consuming and resource intensive. There is the potential for cursor invalidation that leads to hard parses. New statistics can result in new optimizer plans. Automatic maintenance tasks may not gather statistics on indexes if the corresponding table stats are not stale.

Set Resources Utilization Appropriately

STREAMS_POOL_SIZE

`STREAMS_POOL_SIZE` initialization parameter should be set to a reasonable value in the range of 64MB to 256MB. Oracle Data Pump uses Advanced Queuing (AQ) functionality to communicate between processes. If the `SGA_TARGET` initialization parameter is set, then the `STREAMS_POOL_SIZE` initialization parameter should be set to

a reasonable minimum for database usage. See Oracle Database Reference for details on setting the STREAMS_POOL_SIZE parameter.

Restrict resource usage for Data Pump Jobs

A Multitenant Container Database (CDB) with many Plugable Databases (PDBs) may want to avoid having Data Pump operations in one PDB affect other PDBs.

Beginning with release 19c, you can set the maximum number of Data Pump jobs and the maximum parallelism for pluggable databases in a multitenant environment.

MAX_DATAPUMP_JOBS_PER_PDB parameter restricts the number of jobs created. By default, the parameter is set to 50 percent of SESSIONS. This value must be same for each RAC instance. It can be set and changed dynamically, on a per-PDB basis.

MAX_DATAPUMP_PARALLEL_PER_JOB parameter restricts the amount of parallelism in an individual Data Pump job. By default, the perimeter is set to 25 percent of SESSIONS. This value can be different for each RAC instance. It can be set and changed dynamically, on a per-PDB basis.

Note: If you encounter the error, "`ORA-00018: maximum number of sessions exceeded`*" or "*`ORA-00020: maximum number of processes (%s) exceeded`*" you may be allowing too many jobs or too much parallelism, respectively.*

Network Mode Import

You can start an import (impdp) from the target database over a database link. No dump file will be generated, which can eliminate the need for temporary storage during a migration. Beginning with Oracle Database release 12.2, there is support for Direct Path Load over a database link, including for LONG and LONG RAW data, using the parameter ACCESS_METHOD=DIRECT_PATH.

Use Securefile Lobs

Oracle recommends using SecureFile LOBs, especially with partitioning. SecureFile LOBs offer superior performance, functionality and scalability over BasicFile LOBs, including:

- parallel IO into and out of tables with LOB columns (No parallelism on BasicFile LOBs)
- compression
- encryption

You can use the impdp parameter LOB_STORAGE=SECUREFILE to automatically convert BasicFile LOBs to SecureFiles. Tables with SecureFile LOBs storage are automatically created in the target database.

Set Database Compatibility

The database compatibility level affects Data Pump export and import operations. The source database compatibility level determines the compatibility level of the export dumpfile set. If the target database has a lower compatibility level than the source database, use the expdp parameter VERSION to specify the target version. A network mode import can be performed if the compatibility level of the source database is the same as or differs by one major version from the target database.

Beginning with Oracle Database 18c, COMPATIBLE should be set using only the major release and three digits, for instance 18.0.0, with no Release Update number, for example not 19.3.0.

Use the Universal Client

Prior to Oracle Database 21c, exports and imports are performed using the expdp and impdp client versions that match the version of the source and target databases, respectively. However, impdp can always read earlier

version dumpfiles. Beginning with Oracle Database 21c, Data Pump has universal `expdp` and `impdp` clients that can be used with any version of the database that supports Data Pump.

Note: There is no interoperability between Data Pump and the Original Export and Import utilities. `impdp` *cannot read original* `exp` *dumpfiles and* `imp` *cannot read Data Pump dumpfiles.*

More details about Data Pump compatibility can be found in the My Oracle Support note: Export/Import DataPump Parameter VERSION - Compatibility of Data Pump Between Different Oracle Versions (Doc ID 553337.1)

Use Compression for Space Efficiency and Performance

Compression of metadata and/or data during an export can reduce dumpfiles sizes and number, and the size of the data stream during a Network Mode import. It may also improve performance in some cases, especially for Network Mode import. However, additional CPU resources are required to perform transformations on the raw data, so testing should be performed.

Compression can be applied to metadata, data, both or neither. The default is `COMPRESSION=METADATA_ONLY`.

The recommended compression algorithm in most cases is `COMPRESSION=ALL` and `COMPRESSION_ALGORITHM=MEDIUM`. The medium compression algorithm is the best compromise between size and performance, and has no significant difference in overhead compared to the default of BASIC.

Compression algorithms	Characteristics
`BASIC`	Good compression without impacting performance
`LOW`	Use when CPU utilization is more important than the compression ratio
`MEDIUM`	Recommended. Similar to BASIC with a different algorithm
`HIGH`	Maximum compression and CPU utilization

NOTE: Data Pump data compression requires a license for the Advanced Compression Option. No license is required for `COMPRESSION=METADATA_ONLY` *or importing a compressed dumpfile.*

Compression was not supported in Oracle Database 10g release 1 (10.1).

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

Check Database Parameters Before a Data Pump Job

AQ_TM_PROCESSES database parameter

Do not set AQ_TM_PROCESSES to zero. A value of zero can reduce the speed of Advanced Queue operations and of Data Pump operations that use Advanced Queueing. Either leave this parameter value as null or set it to a value greater than 0.

_OPTIMIZER_GATHER_STATS_ON_LOAD hidden parameter

Beginning with Oracle Database 12c, Online Statistics Gathering causes automatic statistics generation for data load operations with Create Table as Select (CTAS) or direct path inserts, such as an insert with an append hint. The recommended value for a Data Pump job is _OPTIMIZER_GATHER_STATS_ON_LOAD=false. However, the default setting is _OPTIMIZER_GATHER_STATS_ON_LOAD=TRUE. This can slow down import operations.

Remember to reset the parameter to TRUE after the Data Pump operation completes or manually gather database statistics using DBMS_STATS.GATHER_DATABASE_STATS.

_lm_share_lock_opt hidden parameter for RAC

Starting with Oracle Real Application Clusters (RAC) release 12.2, RAC allows the Library Cache to use the SHARE lock (S-lock) optimization.
To avoid 'Library Cache Lock' (Cycle) issues in 12.2 during an impdp operation with the parallel= parameter set to a value greater than one on RAC, consider setting "_lm_share_lock_opt"=FALSE on all RAC instances during the import or execute the metadata import with parallel=1.

More details can be found in the MyOracleSupport note: 'Library Cache Lock' (Cycle) Seen During DataPump Import in 12.2 RAC Environment (Doc ID 2407491.1)

Use Include and Exclude Parameters

A category of metadata is described by an object path, for example:

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

You can get a full list of object paths from these views corresponding to the mode of export:

```
DATABASE_EXPORT_OBJECTS
SCHEMA_EXPORT_OBJECTS
TABLE_EXPORT_OBJECTS
```

Some metadata have dependencies. For example, excluding a table will also exclude the associated indexes, constraints, grants, triggers and other table related objects. Excluding an index will also exclude statistics on that index.

Use the INCLUDE or EXCLUDE parameters to add or remove a specific category of metadata. Beginning with Oracle Database 21c these parameters can be combined. The INCLUDE parameter is processed first, including all objects identified by the parameter. Then the EXCLUDE parameter is processed. It removes any objects in the list of included objects.

In this example, the table object path is included and from that object path table statistics is excluded.

```
expdp hr DIRECTORY=dpump_dir1 DUMPFILE=exp%u.dmp SCHEMAS=hr,oe include=table
exclude=statistics
```

Exclude or Include Scheduler Jobs in the Export

SCHEDULER JOBS metadata is stored in the SYS schema. To exclude this metadata refer to the MOS note, "How To Exclude Scheduler Job From Data Pump Export (expdp) (Doc ID 1414478.1)". You can use the opposite approach to include this metadata in a full database export.

Exclude a Large index, Build it with Parallelism after Import

Each index is created by a single Data Pump worker. This enables fast processing of smaller indexes but it may increase the length of the import job when processing a large index. You can remedy this by creating a large index after the import process using the following steps:

4. Find the large 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;
```

5. Exclude the large index(es) from import

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

6. Generate metadata for big indexes using the SQLFILE parameter

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

7. After the import jobe completes, change the parallel degree and create the index(es)

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

Note: *An alternative to using* impdp *with the* sqlfile *parameter is* DBMS_METADATA*. The* DBMS_METADATA.GET_DLL *function returns the metadata of an index or any other object.* The function *can be run on the source database before the export operation.*

Additional Practices for Oracle Autonomous Database

Have you followed the best practices for Autonomous Database Services in the documentation Import Data Using Oracle Data Pump on Autonomous Database

Perform a Schema or table Level Export

Perform a SCHEMA or TABLES mode export when migrating Oracle Database to the Oracle Cloud because FULL mode export will try to get objects that the ADMIN user may not be allowed to access.

Export . Use the parameter schemas=schema_name,...for example

```
$ expdp system directory=dp_dir schemas=scott logfile=export_scott.log parallel=8 ...
```

This ensures only permissible user schemas are moved to the Oracle Cloud database. Be aware that In schema mode you manually handle references from one schema to the other, public synonyms, privileges, etc.. You can

create a script to accomplish this. n general, export from ADB should use SCHEMA or TABLE mode because FULL will try to get objects which the ADMIN user is maybe not allowed to get.

Some objects are exported only in full export:

- Audit trail and policies
- Database Vault
- Directories
- Profiles and password verify function Public synonyms
- Roles
- Public database links
- SQL Management Objects (plan histories, baselines, SQL profiles, etc.)
- Tablespaces
- Users (other than those specified in `SCHEMAS` parameter)
- Workspace manager (use `DBMS_WM.Export_Schemas`)

NOTE: Data Pump never exports grants on SYS objects or AWR

Use a Network Link mode Import to Autonomous Database

Beginning with Oracle Database 19c, you can start an import (`impdp`) from the target autonomous database over a database link. No dump file will be generated.

Consider the Access Method

The Data Pump Export `ACCESS_METHOD` parameter instructs Export to use a particular method to unload data. By default, it is set to `AUTOMATIC`. Beginning with release 12.2, `LONG` or `LONG RAW` types in the source database can be exported over database links using `ACCESS_METHOD=DIRECT_PATH` with `NETWORK_LINK=<dblink>`.

Use the DATAPUMP_CLOUD_IMP role

The `DATAPUMP_CLOUD_IMP` role is needed to perform a full import or to import objects that are owned by other users.

Verify the Dumpfile with Checksum

Beginning with Oracle Database 21c, Data Pump includes a checksum parameter that adds a checksum to the dumpfile. The checksum helps ensure the integrity of the contents of a dump file beyond the header block with a cryptographic hash and ensures that there are no unintentional errors in a dump file. This can help confirm the file is valid after a transfer over the network to or from the object store and ensure it has no malicious changes.

The Data Pump checksum is superior to a regular md5sum checksum. It is a more secure algorithm. The hashes are written in an encrypted format into the dump file header making it harder to tamper with the hash values.

Setting the value specifies whether Oracle Data Pump calculates checksums for the export dump file set, and which hash algorithm is used to calculate the checksum.

Before importing you can choose to perform a verification before starting the actual import. If you have already verified the dumpfile, you can choose to skip the verification during import.

Import into a non-partitioned table

If the source database has partitioned tables and you are migrating data into an Autonomous Data Warehouse database that does not use partitioning, use `DATA_OPTIONS=GROUP_PARTITION_TABLE_DATA` parameter. This allows Data Pump to use the parallel query engine more efficiently when loading data.

Use the AL32UTF8 database character set

Oracle recommends using `AL32UTF8` as the Oracle Database character set. It is a superset of all other character sets.

Creating, Transferring or Downloading an ADB Dumpfile Set

Use the `FILESIZE=10G` parameter and a file size of `10G` or less, that is the current limit for Autonomous Database. If you export directly to Object Store using Oracle Data Pump, the dump files in the Object Store bucket show a zero size. ADB divides each dump file part into smaller chunks for faster uploads. For example:

```
exp01.dmp
exp01.dmp_aaaaaa
exp02.dmp
exp02.dmp_aaaaaa
```

To download the full dump files from the Object Store, use a tool that supports Swift, such as Curl and provide your user login and Swift auth token.

For example:

```
curl -O -v -X GET -u 'user1@example.com:auth_token' \
   https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/namespace-
string/bucketname/exp01.dmp
```

The curl command does not support wildcards or substitution characters in its URL. You can use a script that supports substitution characters to download all the dump files from your Object Store in a single command. An example script can be found in this blog post.

CONCLUSION

Oracle Data Pump is a mature, full featured and flexible tool for Oracle Database logical migration. As discussed in this technical brief, each new release provides more options for optimizing performance and resource utilization. Following the best practices in this technical brief will help your Data Pump exports and imports run as smoothly and quickly as possible.

ORACLE

Connect with us

Call +**1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

blogs.oracle.com facebook.com/oracle twitter.com/oracle