19ᶜ ORACLE®
Database

Oracle Database Utilities Data Pump

Best Practices for Export and Import

WHITE PAPER / MARCH 6, 2019

ORACLE®

This document describes best practices for Oracle Data Pump Export and Import.

TABLE OF CONTENTS

There are multiple approaches for migrating on-premises Oracle databases to the Oracle cloud. A common method is Oracle Data Pump, a feature of Oracle Database since release 10*g* and successor to the Oracle Export and Import (`exp`/`imp`) utilities in release 9*i* 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 Oracle Cloud. It should be considered for migrating an on-premises Oracle database to Oracle cloud in the following circumstances:

• the source Oracle Database is release 10g or higher; (Older databases must use the original Export and Import utilities that came with the database.)

• migrating data cross-endian;

• migrating from a non-CDB Oracle database to an Oracle multitenant database;

• migrating with changes to database structure; or,

• 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-premise database using `expdp`,

- copy the dump files to the target system or to the Oracle Object Store, if required

- import into the cloud PDB using `impdb`.
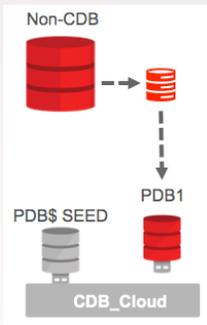
As can be the case, the devil is in the details. This white paper describes some best practices for using Oracle Data Pump to help make the process go smoothly and successfully.

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:

```
DIRECTORY=my_data_pump_dir
DUMPFILE=dumpfile.dmp
LOGFILE=logfile.log
SCHEMAS=HR
EXCLUDE=STATISTICS
LOGTIME=ALL
METRIC=YES
FLASHBACK_TIME=SYSTIMESTAMP
```

The command to execute the par file looks like this:

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

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 directly to FLASHBACK_TIME=SYSTIMESTAMP.

NOTE: If you use FLASHBACK_TIME or FLASHBACK_SCN, Data Pump Export must retain UNDO records for the duration of the export. If the database has insufficient UNDO retention, the result will be a "snapshot segment too old" error as Data Pump attempts to access UNDO records that are no longer available to the database.

AWAYS INCLUDE THESE PARAMETERS DURING EXPORT

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. Instead, follow the best practice of either creating fresh statistics on the target database or using a DBMS_STATS staging table for statistics.

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 number of objects and the elapsed time about the job in the Oracle Data Pump log file. Accomplish this using the parameter METRICS=YES. This gives an extra level of detail, such as the work performed by each process in a PARALLEL export or import.

ALWAYS INCLUDE THESE PARAMETERS DURING IMPORT

Timestamp the messages that are displayed during an import operation using LOGTIME=ALL. This parameter is available beginning with Oracle Database release 12.1.

Record the number of objects and the elapsed time about the job in the Oracle Data Pump log file. Accomplish this using the parameter METRICS=YES.

USE PARALLELISM AND COLLECT STATISTICS

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

The PARALLEL=n parameter specifies the maximum number of processes of active execution operating on behalf of the export or import job. Typically, the value for n should be twice the number of CPU cores but be prepared to adjust it if need be.

You should always use the %U or %L substitution variable when exporting with PARALLEL. Otherwise you end up with parallel writes to a single logfile, which can impact performance.

It is important to collect up-to-date statistics prior to an export operation. Beginning with release 12.2, for export and import by dumpfile only, metadata operations are performed in parallel. Concurrent with metadata export, table sizes are estimated and ranked from largest to smallest for parallel export. The table sizes are estimated using statistics. Collect statistics using the dbms_stats package, with the gather_table_stats, gather_schema_stats, or gather_database_stats procedure.

Oracle Data Pump has imported package bodies in parallel for several releases. Beginning with release 12.2, Oracle Data Pump imports most metadata and most database objects in parallel. The `PARALLEL` parameter also determines how many indexes get created 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.

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

SET RESOURCE UTILIZATION APPROPRIATELY

Set the initialization parameter `STREAMS_POOL_SIZE` 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.

Set the maximum number of Data Pump jobs and the maximum parallelism for pluggable databases in a multitenant environment.

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.

Beginning with release 19c, DBAs can restrict resource usage for Data Pump. You can set the `MAX_DATAPUMP_JOBS_PER_PDB` database parameter to restrict the number of jobs created. You can also allow the parameter to be set automatically to 50 percent of `SESSIONS`. This value must be same for each RAC instance. It can be set and changed dynamically, and it is modifiable per-PDB.

You can set the `MAX_DATAPUMP_PARALLEL_PER_JOB` database parameter to restrict the amount of parallelism in an individual Data Pump job. You can also allow the perimeter to be set automatically to 25 percent of `SESSIONS`. This value can be different for each RAC instance. It can be set and changed dynamically, and it is modifiable per-PDB.

USE A NETWOK LINK FOR SERVERS ON DIFFERENT OS AND STORAGE

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 dblink, including for LONG and LONG RAW data, using the parameter `ACCESS_METHOD=DIRECT_PATH`.

USE SECUREFILE LOBS

It is recommended that you use SecureFile LOBs, especially in conjunction with partitioning. SecureFile LOBs are more scalable then older BasicFile LOBs, and allow for parallel IO into and out of tables with LOB columns. You can use the `impdp` parameter `LOB_STORAGE=SECUREFILE` to convert old LOBs to SecureFiles without having to pre-create tables with the newer LOB storage in the target database.

ADDITIONAL RECOMMENDED PRACTICES FOR AUTONOMOUS DATABASE

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 dblinks using `ACCESS_METHOD=DIRECT_PATH` with `NETWORK_LINK=<dblink>`.

Importing 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 partitions then use `DATA_OPTIONS=GROUP_PARTITION_TABLE_DATA.` This allows Data Pump to use the parallel query engine to more efficiently load data into the data warehouse.

Use the AL32UTF8 database character set

Oracle recommends using AL32UTF8 as the database character set. It is the most common character set because it is a superset of all other character sets.

CONCLUSION

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

ORACLE CORPORATION

Worldwide Headquarters
500 Oracle Parkway, Redwood Shores, CA 94065 USA

Worldwide Inquiries
TELE   + 1.650.506.7000   + 1.800.ORACLE1
FAX   + 1.650.506.7200
oracle.com

CONNECT WITH US

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

B blogs.oracle.com/oracle   f facebook.com/oracle   twitter.com/oracle

Integrated Cloud Applications & Platform Services

Oracle is committed to developing practices and products that help protect the environment