Oracle Data Pump (expdp and impdp) in Oracle Database 10g

Oracle Data Pump is a newer, faster and more flexible alternative to the "exp" and "imp" utilities used in previous Oracle versions. In addition to basic import and export functionality data pump provides a PL/SQL API and support for external tables.

* [Getting Started](https://oracle-base.com/articles/10g/oracle-data-pump-10g#GettingStarted)
* [Table Exports/Imports](https://oracle-base.com/articles/10g/oracle-data-pump-10g#TableExpImp)
* [Schema Exports/Imports](https://oracle-base.com/articles/10g/oracle-data-pump-10g#SchemaExpImp)
* [Database Exports/Imports](https://oracle-base.com/articles/10g/oracle-data-pump-10g#DatabaseExpImp)
* [INCLUDE and EXCLUDE](https://oracle-base.com/articles/10g/oracle-data-pump-10g#IncludeExclude)
* [Network Exports/Imports (NETWORK\_LINK)](https://oracle-base.com/articles/10g/oracle-data-pump-10g#NetworkExportsImports)
* [Flashback Exports](https://oracle-base.com/articles/10g/oracle-data-pump-10g#Flashback)
* [Miscellaneous Information](https://oracle-base.com/articles/10g/oracle-data-pump-10g#MiscellaneousInformation)
* [Data Pump API](https://oracle-base.com/articles/10g/oracle-data-pump-10g#DataPumpAPI)
* [External Tables (Unloading/Loading Data Using External Tables)](https://oracle-base.com/articles/10g/oracle-data-pump-10g#ExternalTables)
* [Help](https://oracle-base.com/articles/10g/oracle-data-pump-10g#Help)  
  + [expdp](https://oracle-base.com/articles/10g/oracle-data-pump-10g#expdp)
  + [impdp](https://oracle-base.com/articles/10g/oracle-data-pump-10g#impdp)

Getting Started

For the examples to work we must first unlock the SCOTT account and create a directory object it can access. The directory object is only a pointer to a physical directory, creating it does not actually create the physical directory on the file system of the database server.

CONN / AS SYSDBA

ALTER USER scott IDENTIFIED BY tiger ACCOUNT UNLOCK;

CREATE OR REPLACE DIRECTORY test\_dir AS '/u01/app/oracle/oradata/';

GRANT READ, WRITE ON DIRECTORY test\_dir TO scott;

Existing directories can be queried using the ALL\_DIRECTORIES view.

 Data Pump is a server-based technology, so it typically deals with directory objects pointing to physical directories on the database server. It does not write to the local file system on your client PC.

Table Exports/Imports

The TABLES parameter is used to specify the tables that are to be exported. The following is an example of the table export and import syntax.

expdp scott/tiger@db10g tables=EMP,DEPT directory=TEST\_DIR dumpfile=EMP\_DEPT.dmp logfile=expdpEMP\_DEPT.log

impdp scott/tiger@db10g tables=EMP,DEPT directory=TEST\_DIR dumpfile=EMP\_DEPT.dmp logfile=impdpEMP\_DEPT.log

For example output files see [expdpEMP\_DEPT.log](https://oracle-base.com/articles/10g/expdpEMP_DEPT.log) and [impdpEMP\_DEPT.log](https://oracle-base.com/articles/10g/impdpEMP_DEPT.log).

The TABLE\_EXISTS\_ACTION=APPEND parameter allows data to be imported into existing tables.

Schema Exports/Imports

The OWNER parameter of exp has been replaced by the SCHEMAS parameter which is used to specify the schemas to be exported. The following is an example of the schema export and import syntax.

expdp scott/tiger@db10g schemas=SCOTT directory=TEST\_DIR dumpfile=SCOTT.dmp logfile=expdpSCOTT.log

impdp scott/tiger@db10g schemas=SCOTT directory=TEST\_DIR dumpfile=SCOTT.dmp logfile=impdpSCOTT.log

For example output files see [expdpSCOTT.log](https://oracle-base.com/articles/10g/expdpSCOTT.log) and [impdpSCOTT.log](https://oracle-base.com/articles/10g/impdpSCOTT.log).

Database Exports/Imports

The FULL parameter indicates that a complete database export is required. The following is an example of the full database export and import syntax.

expdp system/password@db10g full=Y directory=TEST\_DIR dumpfile=DB10G.dmp logfile=expdpDB10G.log

impdp system/password@db10g full=Y directory=TEST\_DIR dumpfile=DB10G.dmp logfile=impdpDB10G.log

For an example output file see [expdpDB10G.log](https://oracle-base.com/articles/10g/expdpDB10G.log).

INCLUDE and EXCLUDE

The INCLUDE and EXCLUDE parameters can be used to limit the export/import to specific objects. When the INCLUDE parameter is used, only those objects specified by it will be included in the export/import. When the EXCLUDE parameter is used, all objects except those specified by it will be included in the export/import. The two parameters are mutually exclusive, so use the parameter that requires the least entries to give you the result you require. The basic syntax for both parameters is the same.

INCLUDE=object\_type[:name\_clause] [, ...]

EXCLUDE=object\_type[:name\_clause] [, ...]

The following code shows how they can be used as command line parameters.

expdp scott/tiger@db10g schemas=SCOTT include=TABLE:"IN ('EMP', 'DEPT')" directory=TEST\_DIR dumpfile=SCOTT.dmp logfile=expdpSCOTT.log

expdp scott/tiger@db10g schemas=SCOTT exclude=TABLE:"= 'BONUS'" directory=TEST\_DIR dumpfile=SCOTT.dmp logfile=expdpSCOTT.log

If the parameter is used from the command line, depending on your OS, the special characters in the clause may need to be escaped, as follows. Because of this, it is easier to use a parameter file.

include=TABLE:\"IN (\'EMP\', \'DEPT\')\"

A single import/export can include multiple references to the parameters, so to export tables, views and some packages we could use either of the following approaches.

INCLUDE=TABLE,VIEW,PACKAGE:"LIKE '%API'"

or

INCLUDE=TABLE

INCLUDE=VIEW

INCLUDE=PACKAGE:"LIKE '%API'"

Multiple objects can be targeted in once statement using the LIKE and IN operators.

EXCLUDE=SCHEMA:"LIKE 'SYS%'"

EXCLUDE=SCHEMA:"IN ('OUTLN','SYSTEM','SYSMAN','FLOWS\_FILES','APEX\_030200','APEX\_PUBLIC\_USER','ANONYMOUS')"

The valid object type paths that can be included or excluded can be displayed using the DATABASE\_EXPORT\_OBJECTS, SCHEMA\_EXPORT\_OBJECTS, and TABLE\_EXPORT\_OBJECTS views.

Network Exports/Imports (NETWORK\_LINK)

The NETWORK\_LINK parameter identifies a database link to be used as the source for a network export/import. The following database link will be used to demonstrate its use.

CONN / AS SYSDBA

GRANT CREATE DATABASE LINK TO test;

CONN test/test

CREATE DATABASE LINK remote\_scott CONNECT TO scott IDENTIFIED BY tiger USING 'DEV';

In the case of exports, the NETWORK\_LINK parameter identifies the database link pointing to the source server. The objects are exported from the source server in the normal manner, but written to a directory object on the local server, rather than one on the source server. Both the local and remote users require the EXP\_FULL\_DATABASE role granted to them.

expdp test/test@db10g tables=SCOTT.EMP network\_link=REMOTE\_SCOTT directory=TEST\_DIR dumpfile=EMP.dmp logfile=expdpEMP.log

For imports, the NETWORK\_LINK parameter also identifies the database link pointing to the source server. The difference here is the objects are imported directly from the source into the local server without being written to a dump file. Although there is no need for a DUMPFILE parameter, a directory object is still required for the logs associated with the operation. Both the local and remote users require the IMP\_FULL\_DATABASE role granted to them.

impdp test/test@db10g tables=SCOTT.EMP network\_link=REMOTE\_SCOTT directory=TEST\_DIR logfile=impdpSCOTT.log remap\_schema=SCOTT:TEST

Flashback Exports

The exp utility used the CONSISTENT=Y parameter to indicate the export should be consistent to a point in time. By default the expdp utility exports are only consistent on a per table basis. If you want all tables in the export to be consistent to the same point in time, you need to use the FLASHBACK\_SCN or FLASHBACK\_TIME parameter.

The FLASHBACK\_TIME parameter value is converted to the approximate SCN for the specified time.

expdp ..... flashback\_time=systimestamp

# In parameter file.

flashback\_time="to\_timestamp('09-05-2011 09:00:00', 'DD-MM-YYYY HH24:MI:SS')"

# Escaped on command line.

expdp ..... flashback\_time=\"to\_timestamp\(\'09-05-2011 09:00:00\', \'DD-MM-YYYY HH24:MI:SS\'\)\"

Not surprisingly, you can make exports consistent to an earlier point in time by specifying an earlier time or SCN, provided you have enough UNDO space to keep a read consistent view of the data during the export operation.

If you prefer to use the SCN, you can retrieve the current SCN using one of the following queries.

SELECT current\_scn FROM v$database;

SELECT DBMS\_FLASHBACK.get\_system\_change\_number FROM dual;

SELECT TIMESTAMP\_TO\_SCN(SYSTIMESTAMP) FROM dual;

That SCN is then used with the FLASHBACK\_SCN parameter.

expdp ..... flashback\_scn=5474280

The following queries may prove useful for converting between timestamps and SCNs.

SELECT TIMESTAMP\_TO\_SCN(SYSTIMESTAMP) FROM dual;

SELECT SCN\_TO\_TIMESTAMP(5474751) FROM dual;

In 11.2, the introduction of legacy mode means that you can use the CONSISTENT=Y parameter with the expdp utility if you wish.

Miscellaneous Information

Unlike the original exp and imp utilities all data pump ".dmp" and ".log" files are created on the Oracle server, not the client machine.

All data pump actions are performed by multiple jobs (server processes not DBMS\_JOB jobs). These jobs are controlled by a master control process which uses Advanced Queuing. At runtime an advanced queue table, named after the job name, is created and used by the master control process. The table is dropped on completion of the data pump job. The job and the advanced queue can be named using the JOB\_NAME parameter. Cancelling the client process does not stop the associated data pump job. Issuing "ctrl+c" on the client during a job stops the client output and presents a command prompt. Typing "status" at this prompt allows you to monitor the current job.

Export> status

Job: SYS\_EXPORT\_FULL\_01

Operation: EXPORT

Mode: FULL

State: EXECUTING

Bytes Processed: 0

Current Parallelism: 1

Job Error Count: 0

Dump File: D:\TEMP\DB10G.DMP

bytes written: 4,096

Worker 1 Status:

State: EXECUTING

Object Schema: SYSMAN

Object Name: MGMT\_CONTAINER\_CRED\_ARRAY

Object Type: DATABASE\_EXPORT/SCHEMA/TYPE/TYPE\_SPEC

Completed Objects: 261

Total Objects: 261

Data pump performance can be improved by using the PARALLEL parameter. This should be used in conjunction with the "%U" wildcard in the DUMPFILE parameter to allow multiple dumpfiles to be created or read. The same wildcard can be used during the import to allow you to reference multiple files.

expdp scott/tiger@db10g schemas=SCOTT directory=TEST\_DIR parallel=4 dumpfile=SCOTT\_%U.dmp logfile=expdpSCOTT.log

impdp scott/tiger@db10g schemas=SCOTT directory=TEST\_DIR parallel=4 dumpfile=SCOTT\_%U.dmp logfile=impdpSCOTT.log

The DBA\_DATAPUMP\_JOBS view can be used to monitor the current jobs.

system@db10g> select \* from dba\_datapump\_jobs;

OWNER\_NAME JOB\_NAME OPERATION

------------------------------ ------------------------------ ------------------------------

JOB\_MODE STATE DEGREE ATTACHED\_SESSIONS

------------------------------ ------------------------------ ---------- -----------------

SYSTEM SYS\_EXPORT\_FULL\_01 EXPORT

FULL EXECUTING 1 1

Data Pump API

Along with the data pump utilities Oracle provide an PL/SQL API. The following is an example of how this API can be used to perform a schema export.

SET SERVEROUTPUT ON SIZE 1000000

DECLARE

l\_dp\_handle NUMBER;

l\_last\_job\_state VARCHAR2(30) := 'UNDEFINED';

l\_job\_state VARCHAR2(30) := 'UNDEFINED';

l\_sts KU$\_STATUS;

BEGIN

l\_dp\_handle := DBMS\_DATAPUMP.open(

operation => 'EXPORT',

job\_mode => 'SCHEMA',

remote\_link => NULL,

job\_name => 'EMP\_EXPORT',

version => 'LATEST');

DBMS\_DATAPUMP.add\_file(

handle => l\_dp\_handle,

filename => 'SCOTT.dmp',

directory => 'TEST\_DIR');

DBMS\_DATAPUMP.add\_file(

handle => l\_dp\_handle,

filename => 'SCOTT.log',

directory => 'TEST\_DIR',

filetype => DBMS\_DATAPUMP.KU$\_FILE\_TYPE\_LOG\_FILE);

DBMS\_DATAPUMP.metadata\_filter(

handle => l\_dp\_handle,

name => 'SCHEMA\_EXPR',

value => '= ''SCOTT''');

DBMS\_DATAPUMP.start\_job(l\_dp\_handle);

DBMS\_DATAPUMP.detach(l\_dp\_handle);

END;

/

Once the job has started the status can be checked using.

system@db10g> select \* from dba\_datapump\_jobs;

External Tables (Unloading/Loading Data Using External Tables)

Oracle have incorporated support for data pump technology into external tables. The ORACLE\_DATAPUMP access driver can be used to unload data to data pump export files and subsequently reload it. The unload of data occurs when the external table is created using the "AS" clause.

CREATE TABLE emp\_xt

ORGANIZATION EXTERNAL

(

TYPE ORACLE\_DATAPUMP

DEFAULT DIRECTORY test\_dir

LOCATION ('emp\_xt.dmp')

)

AS SELECT \* FROM emp;

The data can then be queried using the following.

SELECT \* FROM emp\_xt;

The syntax to create the external table pointing to an existing file is similar, but without the "AS" clause. In this case we will do it the same schema, but this could be in a different schema in the same instance, or in an entirely different instance.

DROP TABLE emp\_xt;

CREATE TABLE emp\_xt (

EMPNO NUMBER(4),

ENAME VARCHAR2(10),

JOB VARCHAR2(9),

MGR NUMBER(4),

HIREDATE DATE,

SAL NUMBER(7,2),

COMM NUMBER(7,2),

DEPTNO NUMBER(2))

ORGANIZATION EXTERNAL (

TYPE ORACLE\_DATAPUMP

DEFAULT DIRECTORY test\_dir

LOCATION ('emp\_xt.dmp')

);

SELECT \* FROM emp\_xt;

Creating an external table using the ORACLE\_DATAPUMP access driver is restricted to dump files created by the external table unload.

Help

The HELP=Y option displays the available parameters.

expdp

expdp help=y

Export: Release 10.1.0.2.0 - Production on Tuesday, 23 March, 2004 8:33

Copyright (c) 2003, Oracle. All rights reserved.

The Data Pump export utility provides a mechanism for transferring data objects

between Oracle databases. The utility is invoked with the following command:

Example: expdp scott/tiger DIRECTORY=dmpdir DUMPFILE=scott.dmp

You can control how Export runs by entering the 'expdp' command followed

by various parameters. To specify parameters, you use keywords:

Format: expdp KEYWORD=value or KEYWORD=(value1,value2,...,valueN)

Example: expdp scott/tiger DUMPFILE=scott.dmp DIRECTORY=dmpdir SCHEMAS=scott

or TABLES=(T1:P1,T1:P2), if T1 is partitioned table

USERID must be the first parameter on the command line.

Keyword Description (Default)

------------------------------------------------------------------------------

ATTACH Attach to existing job, e.g. ATTACH [=job name].

CONTENT Specifies data to unload where the valid keywords are:

(ALL), DATA\_ONLY, and METADATA\_ONLY.

DIRECTORY Directory object to be used for dumpfiles and logfiles.

DUMPFILE List of destination dump files (expdat.dmp),

e.g. DUMPFILE=scott1.dmp, scott2.dmp, dmpdir:scott3.dmp.

ESTIMATE Calculate job estimates where the valid keywords are:

(BLOCKS) and STATISTICS.

ESTIMATE\_ONLY Calculate job estimates without performing the export.

EXCLUDE Exclude specific object types, e.g. EXCLUDE=TABLE:EMP.

FILESIZE Specify the size of each dumpfile in units of bytes.

FLASHBACK\_SCN SCN used to set session snapshot back to.

FLASHBACK\_TIME Time used to get the SCN closest to the specified time.

FULL Export entire database (N).

HELP Display Help messages (N).

INCLUDE Include specific object types, e.g. INCLUDE=TABLE\_DATA.

JOB\_NAME Name of export job to create.

LOGFILE Log file name (export.log).

NETWORK\_LINK Name of remote database link to the source system.

NOLOGFILE Do not write logfile (N).

PARALLEL Change the number of active workers for current job.

PARFILE Specify parameter file.

QUERY Predicate clause used to export a subset of a table.

SCHEMAS List of schemas to export (login schema).

STATUS Frequency (secs) job status is to be monitored where

the default (0) will show new status when available.

TABLES Identifies a list of tables to export - one schema only.

TABLESPACES Identifies a list of tablespaces to export.

TRANSPORT\_FULL\_CHECK Verify storage segments of all tables (N).

TRANSPORT\_TABLESPACES List of tablespaces from which metadata will be unloaded.

VERSION Version of objects to export where valid keywords are:

(COMPATIBLE), LATEST, or any valid database version.

The following commands are valid while in interactive mode.

Note: abbreviations are allowed

Command Description

------------------------------------------------------------------------------

ADD\_FILE Add dumpfile to dumpfile set.

ADD\_FILE=dumpfile-name

CONTINUE\_CLIENT Return to logging mode. Job will be re-started if idle.

EXIT\_CLIENT Quit client session and leave job running.

HELP Summarize interactive commands.

KILL\_JOB Detach and delete job.

PARALLEL Change the number of active workers for current job.

PARALLEL=.

START\_JOB Start/resume current job.

STATUS Frequency (secs) job status is to be monitored where

the default (0) will show new status when available.

STATUS=[interval]

STOP\_JOB Orderly shutdown of job execution and exits the client.

STOP\_JOB=IMMEDIATE performs an immediate shutdown of the

Data Pump job.

Oracle 10g Release 2 (10.2) added the following parameters.

Keyword Description (Default)

------------------------------------------------------------------------------

COMPRESSION Reduce size of dumpfile contents where valid

keyword values are: (METADATA\_ONLY) and NONE.

ENCRYPTION\_PASSWORD Password key for creating encrypted column data.

SAMPLE Percentage of data to be exported;

The following commands are valid while in interactive mode.

Note: abbreviations are allowed

Command Description

------------------------------------------------------------------------------

FILESIZE Default filesize (bytes) for subsequent ADD\_FILE commands.

Oracle 11g Release 1 (11.1) added the following parameters.

Keyword Description (Default)

------------------------------------------------------------------------------

DATA\_OPTIONS Data layer flags where the only valid value is:

XML\_CLOBS-write XML datatype in CLOB format

ENCRYPTION Encrypt part or all of the dump file where valid keyword

values are: ALL, DATA\_ONLY, METADATA\_ONLY,

ENCRYPTED\_COLUMNS\_ONLY, or NONE.

ENCRYPTION\_ALGORITHM Specify how encryption should be done where valid

keyword values are: (AES128), AES192, and AES256.

ENCRYPTION\_MODE Method of generating encryption key where valid keyword

values are: DUAL, PASSWORD, and (TRANSPARENT).

REMAP\_DATA Specify a data conversion function,

e.g. REMAP\_DATA=EMP.EMPNO:REMAPPKG.EMPNO.

REUSE\_DUMPFILES Overwrite destination dump file if it exists (N).

TRANSPORTABLE Specify whether transportable method can be used where

valid keyword values are: ALWAYS, (NEVER).

The following commands are valid while in interactive mode.

Note: abbreviations are allowed

Command Description

------------------------------------------------------------------------------

REUSE\_DUMPFILES Overwrite destination dump file if it exists (N).

Oracle 11g Release 2 (11.2) altered the format of the help output as well as adding the following parameters.

CLUSTER

Utilize cluster resources and distribute workers across the Oracle RAC.

Valid keyword values are: [Y] and N.

SERVICE\_NAME

Name of an active Service and associated resource group to constrain Oracle RAC resources.

SOURCE\_EDITION

Edition to be used for extracting metadata.

Oracle 12c Release 1 (12.1) added the following parameters.

COMPRESSION\_ALGORITHM

Specify the compression algorithm that should be used.

Valid keyword values are: [BASIC], LOW, MEDIUM and HIGH.

ENCRYPTION\_PWD\_PROMPT

Specifies whether to prompt for the encryption password.

Terminal echo will be suppressed while standard input is read.

VIEWS\_AS\_TABLES

Identifies one or more views to be exported as tables.

For example, VIEWS\_AS\_TABLES=HR.EMP\_DETAILS\_VIEW.

impdp

impdp help=y

Import: Release 10.1.0.2.0 - Production on Saturday, 11 September, 2004 17:22

Copyright (c) 2003, Oracle. All rights reserved.

The Data Pump Import utility provides a mechanism for transferring data objects

between Oracle databases. The utility is invoked with the following command:

Example: impdp scott/tiger DIRECTORY=dmpdir DUMPFILE=scott.dmp

You can control how Import runs by entering the 'impdp' command followed

by various parameters. To specify parameters, you use keywords:

Format: impdp KEYWORD=value or KEYWORD=(value1,value2,...,valueN)

Example: impdp scott/tiger DIRECTORY=dmpdir DUMPFILE=scott.dmp

USERID must be the first parameter on the command line.

Keyword Description (Default)

------------------------------------------------------------------------------

ATTACH Attach to existing job, e.g. ATTACH [=job name].

CONTENT Specifies data to load where the valid keywords are:

(ALL), DATA\_ONLY, and METADATA\_ONLY.

DIRECTORY Directory object to be used for dump, log, and sql files.

DUMPFILE List of dumpfiles to import from (expdat.dmp),

e.g. DUMPFILE=scott1.dmp, scott2.dmp, dmpdir:scott3.dmp.

ESTIMATE Calculate job estimates where the valid keywords are:

(BLOCKS) and STATISTICS.

EXCLUDE Exclude specific object types, e.g. EXCLUDE=TABLE:EMP.

FLASHBACK\_SCN SCN used to set session snapshot back to.

FLASHBACK\_TIME Time used to get the SCN closest to the specified time.

FULL Import everything from source (Y).

HELP Display help messages (N).

INCLUDE Include specific object types, e.g. INCLUDE=TABLE\_DATA.

JOB\_NAME Name of import job to create.

LOGFILE Log file name (import.log).

NETWORK\_LINK Name of remote database link to the source system.

NOLOGFILE Do not write logfile.

PARALLEL Change the number of active workers for current job.

PARFILE Specify parameter file.

QUERY Predicate clause used to import a subset of a table.

REMAP\_DATAFILE Redefine datafile references in all DDL statements.

REMAP\_SCHEMA Objects from one schema are loaded into another schema.

REMAP\_TABLESPACE Tablespace object are remapped to another tablespace.

REUSE\_DATAFILES Tablespace will be initialized if it already exists (N).

SCHEMAS List of schemas to import.

SKIP\_UNUSABLE\_INDEXES Skip indexes that were set to the Index Unusable state.

SQLFILE Write all the SQL DDL to a specified file.

STATUS Frequency (secs) job status is to be monitored where

the default (0) will show new status when available.

STREAMS\_CONFIGURATION Enable the loading of Streams metadata

TABLE\_EXISTS\_ACTION Action to take if imported object already exists.

Valid keywords: (SKIP), APPEND, REPLACE and TRUNCATE.

TABLES Identifies a list of tables to import.

TABLESPACES Identifies a list of tablespaces to import.

TRANSFORM Metadata transform to apply (Y/N) to specific objects.

Valid transform keywords: SEGMENT\_ATTRIBUTES and STORAGE.

ex. TRANSFORM=SEGMENT\_ATTRIBUTES:N:TABLE.

TRANSPORT\_DATAFILES List of datafiles to be imported by transportable mode.

TRANSPORT\_FULL\_CHECK Verify storage segments of all tables (N).

TRANSPORT\_TABLESPACES List of tablespaces from which metadata will be loaded.

Only valid in NETWORK\_LINK mode import operations.

VERSION Version of objects to export where valid keywords are:

(COMPATIBLE), LATEST, or any valid database version.

Only valid for NETWORK\_LINK and SQLFILE.

The following commands are valid while in interactive mode.

Note: abbreviations are allowed

Command Description (Default)11g

------------------------------------------------------------------------------

CONTINUE\_CLIENT Return to logging mode. Job will be re-started if idle.

EXIT\_CLIENT Quit client session and leave job running.

HELP Summarize interactive commands.

KILL\_JOB Detach and delete job.

PARALLEL Change the number of active workers for current job.

PARALLEL=.

START\_JOB Start/resume current job.

START\_JOB=SKIP\_CURRENT will start the job after skipping

any action which was in progress when job was stopped.

STATUS Frequency (secs) job status is to be monitored where

the default (0) will show new status when available.

STATUS=[interval]

STOP\_JOB Orderly shutdown of job execution and exits the client.

STOP\_JOB=IMMEDIATE performs an immediate shutdown of the

Data Pump job.

Oracle 10g Release 2 (10.2) added the following parameter.

Keyword Description (Default)

------------------------------------------------------------------------------

ENCRYPTION\_PASSWORD Password key for accessing encrypted column data.

This parameter is not valid for network import jobs.

Oracle 11g Release 1 (11.1) added the following parameters.

Keyword Description (Default)

------------------------------------------------------------------------------

DATA\_OPTIONS Data layer flags where the only valid value is:

SKIP\_CONSTRAINT\_ERRORS-constraint errors are not fatal.

PARTITION\_OPTIONS Specify how partitions should be transformed where the

valid keywords are: DEPARTITION, MERGE and (NONE)

REMAP\_DATA Specify a data conversion function,

e.g. REMAP\_DATA=EMP.EMPNO:REMAPPKG.EMPNO

REMAP\_TABLE Table names are remapped to another table.

For example, REMAP\_TABLE=HR.EMPLOYEES:EMPS.

Oracle 11g Release 2 (11.2) altered the format of the help output as well as adding the following parameters.

CLUSTER

Utilize cluster resources and distribute workers across the Oracle RAC.

Valid keyword values are: [Y] and N.

SERVICE\_NAME

Name of an active Service and associated resource group to constrain Oracle RAC resources.

SOURCE\_EDITION

Edition to be used for extracting metadata.

TARGET\_EDITION

Edition to be used for loading metadata.

Oracle 12c Release 1 (12.1) added the following parameters.

ENCRYPTION\_PWD\_PROMPT

Specifies whether to prompt for the encryption password.

Terminal echo will be suppressed while standard input is read.

TRANSPORTABLE

Options for choosing transportable data movement.

Valid keywords are: ALWAYS and [NEVER].

Only valid in NETWORK\_LINK mode import operations.

VIEWS\_AS\_TABLES

Identifies one or more views to be imported as tables.

For example, VIEWS\_AS\_TABLES=HR.EMP\_DETAILS\_VIEW.

Note that in network import mode, a table name may be appended

to the view name.

Data Pump Enhancements in Oracle Database 11g Release 1 (expdp and impdp)

Oracle [Data Pump](https://oracle-base.com/articles/10g/oracle-data-pump-10g) was introduced in Oracle 10g. This article provides an overview of the main Data Pump enhancements in Oracle Database 11g Release 1, including the following.

* [COMPRESSION](https://oracle-base.com/articles/11g/data-pump-enhancements-11gr1#compression)
* [Encryption Parameters](https://oracle-base.com/articles/11g/data-pump-enhancements-11gr1#encryption_parameters)  
  + [ENCRYPTION and ENCRYPTION\_PASSWORD](https://oracle-base.com/articles/11g/data-pump-enhancements-11gr1#encryption_and_encryption_password)
  + [ENCRYPTION\_ALGORITHM](https://oracle-base.com/articles/11g/data-pump-enhancements-11gr1#encryption_algorithm)
  + [ENCRYPTION\_MODE](https://oracle-base.com/articles/11g/data-pump-enhancements-11gr1#encryption_mode)
* [TRANSPORTABLE](https://oracle-base.com/articles/11g/data-pump-enhancements-11gr1#transportable)
* [PARTITION\_OPTIONS](https://oracle-base.com/articles/11g/data-pump-enhancements-11gr1#partition_options)
* [REUSE\_DUMPFILES](https://oracle-base.com/articles/11g/data-pump-enhancements-11gr1#reuse_dumpfiles)
* [REMAP\_TABLE](https://oracle-base.com/articles/11g/data-pump-enhancements-11gr1#remap_table)
* [DATA\_OPTIONS](https://oracle-base.com/articles/11g/data-pump-enhancements-11gr1#data_options)  
  + [SKIP\_CONSTRAINT\_ERRORS](https://oracle-base.com/articles/11g/data-pump-enhancements-11gr1#skip_constraint_errors)
  + [XML\_CLOBS](https://oracle-base.com/articles/11g/data-pump-enhancements-11gr1#xml_clobs)
* [REMAP\_DATA](https://oracle-base.com/articles/11g/data-pump-enhancements-11gr1#remap_data)
* [Miscellaneous Enhancements](https://oracle-base.com/articles/11g/data-pump-enhancements-11gr1#miscellaneous)

COMPRESSION

The COMPRESSION parameter allows you to decide what, if anything, you wish to compress in your export. The syntax is shown below.

COMPRESSION={ALL | DATA\_ONLY | METADATA\_ONLY | NONE}

The available options are:

* ALL: Both metadata and data are compressed.
* DATA\_ONLY: Only data is compressed.
* METADATA\_ONLY: Only metadata is compressed. This is the default setting.
* NONE: Nothing is compressed.

Here is an example of the COMPRESSION parameter being used.

expdp test/test schemas=TEST directory=TEST\_DIR dumpfile=TEST.dmp logfile=expdpTEST.log

compression=all

The COMPATIBLE initialization parameter should be set to "11.0.0" or higher to use these options, except for the METADATA\_ONLY option, which is available with a COMPATIBLE setting of "10.2".

Data compression requires the Advanced Compression Option option of Enterprise Edition, as described [here](http://www.oracle.com/technetwork/database/storage/advanced-compression-whitepaper-130502.pdf).

Encryption Parameters

Data pump encryption is an Enterprise Edition feature, so the parameters described below are only relevant for Enterprise Edition installations. In addition, the COMPATIBLE initialisation parameter must be set to "11.0.0" or higher to use these features.

ENCRYPTION and ENCRYPTION\_PASSWORD

The use of encryption is controlled by a combination of the ENCRYPTION or ENCRYPTION\_PASSWORD parameters. The syntax for the ENCRYPTION parameter is shown below.

ENCRYPTION = {ALL | DATA\_ONLY | ENCRYPTED\_COLUMNS\_ONLY | METADATA\_ONLY | NONE}

The available options are:

* ALL: Both metadata and data are encrypted.
* DATA\_ONLY: Only data is encrypted.
* ENCRYPTED\_COLUMNS\_ONLY: Only encrypted columns are written to the dump file in an encrypted format.
* METADATA\_ONLY: Only metadata is encrypted.
* NONE: Nothing is encrypted.

If neither the ENCRYPTION or ENCRYPTION\_PASSWORD parameters are set, it is assumed the required level of encryption is NONE. If only the ENCRYPTION\_PASSWORD parameter is specified, it is assumed the required level of encryption is ALL. Here is an example of these parameters being used.

expdp test/test schemas=TEST directory=TEST\_DIR dumpfile=TEST.dmp logfile=expdpTEST.log

encryption=all encryption\_password=password

ENCRYPTION\_ALGORITHM

The ENCRYPTION\_ALGORITHM parameter specifies the encryption algorithm to be used during the export, with the default being "AES128". The syntax is shown below.

ENCRYPTION\_ALGORITHM = { AES128 | AES192 | AES256 }

The ENCRYPTION\_ALGORITHM parameter must be used in conjunction with the ENCRYPTION or ENCRYPTION\_PASSWORD parameters, as shown below.

expdp test/test schemas=TEST directory=TEST\_DIR dumpfile=TEST.dmp logfile=expdpTEST.log

encryption=all encryption\_password=password encryption\_algorithm=AES256

ENCRYPTION\_MODE

The ENCRYPTION\_MODE parameter specifies the type of security used during export and import operations. The syntax is shown below.

ENCRYPTION\_MODE = { DUAL | PASSWORD | TRANSPARENT }

The allowable values and their default settings are explained below:

* DUAL: This mode creates a dump file that can be imported using an Oracle Encryption Wallet, or the the ENCRYPTION\_PASSWORD specified during the export operation. This is the default setting if the ENCRYPTION\_PASSWORD parameter is set and there is an open wallet.
* PASSWORD: This mode creates a dump file that can only be imported using the ENCRYPTION\_PASSWORD specified during the export operation. This is the default setting if theENCRYPTION\_PASSWORD parameter is set and there isn't an open wallet.
* TRANSPARENT: This mode creates an encrypted dump file using and open Oracle Encryption Wallet. If the ENCRYPTION\_PASSWORD is specified while using this mode and error is produced. This is the default setting of only the ENCRYPTION parameter is set.

Wallet setup is described [here](https://oracle-base.com/articles/11g/tablespace-encryption-11gr1#wallet_creation).

The ENCRYPTION\_MODE requires either the ENCRYPTION or ENCRYPTION\_PASSWORD parameter to be specified.

expdp test/test schemas=TEST directory=TEST\_DIR dumpfile=TEST.dmp logfile=expdpTEST.log

encryption=all encryption\_password=password encryption\_mode=password

TRANSPORTABLE

The TRANSPORTABLE parameter is similar to the TRANSPORT\_TABLESPACES parameter available previously in that it only exports and imports metadata about a table, relying on you to manually transfer the relevent tablespace datafiles. The export operation lists the tablespaces that must be transfered. The syntax is shown below.

TRANSPORTABLE = {ALWAYS | NEVER}

The value ALWAYS turns on the transportable mode, while the default value of NEVER indicates this is a regular export/import.

The following restrictions apply during exports using the TRANSPORTABLE parameter:

* This parameter is only applicable during table-level exports.
* The user performing the operation must have the EXP\_FULL\_DATABASE privilege.
* Tablespaces containing the source objects must be read-only.
* The COMPATIBLE initialization parameter must be set to 11.0.0 or higher.
* The default tablespace of the user performing the export must not be the same as any of the tablespaces being transported.

Some extra restictions apply during import operations:

* The NETWORK\_LINK parameter must be specified during the import operation. This parameter is set to a valid database link to the source schema.
* The schema performing the import must have both EXP\_FULL\_DATABASE and IMP\_FULL\_DATABASE privileges.
* The TRANSPORT\_DATAFILES parameter is used to identify the datafiles holding the table data.

Examples of the export and import operations are shown below.

expdp system tables=TEST1.TAB1 directory=TEST\_DIR dumpfile=TEST.dmp logfile=expdpTEST.log

transportable=ALWAYS

impdp system tables=TEST1.TAB1 directory=TEST\_DIR dumpfile=TEST.dmp logfile=impdpTEST.log

transportable=ALWAYS network\_link=DB11G transport\_datafiles='/u01/oradata/DB11G/test01.dbf'

PARTITION\_OPTIONS

The PARTITION\_OPTIONS parameter determines how partitions will be handled during export and import operations. The syntax is shown below.

PARTITION\_OPTIONS={none | departition | merge}

The allowable values are:

* NONE: The partitions are created exactly as they were on the system the export was taken from.
* DEPARTITION: Each partition and sub-partition is created as a separate table, named using a combination of the table and (sub-)partition name.
* MERGE: Combines all partitions into a single table.

The NONE and MERGE options are not possible if the export was done using the TRANSPORTABLE parameter with a partition or subpartition filter. If there are any grants on objects being departitioned, an error message is generated and the objects are not loaded.

expdp test/test directory=TEST\_DIR dumpfile=TEST.dmp logfile=expdpTEST.log tables=test.tab1

partition\_options=merge

REUSE\_DUMPFILES

The REUSE\_DUMPFILES parameter can be used to prevent errors being issued if the export attempts to write to a dump file that already exists.

REUSE\_DUMPFILES={Y | N}

When set to "Y", any existing dumpfiles will be overwritten. When the default values of "N" is used, an error is issued if the dump file already exists.

expdp test/test schemas=TEST directory=TEST\_DIR dumpfile=TEST.dmp logfile=expdpTEST.log

reuse\_dumpfiles=y

REMAP\_TABLE

This parameter allows a table to be renamed during the import operations performed using the TRANSPORTABLE method. It can also be used to alter the base table name used duringPARTITION\_OPTIONS imports. The syntax is shown below.

REMAP\_TABLE=[schema.]old\_tablename[.partition]:new\_tablename

An example is shown below.

impdp test/test tables=TAB1 directory=TEST\_DIR dumpfile=TEST.dmp logfile=impdpTEST.log

remap\_table=TEST.TAB1:TAB2

Existing tables are not renamed, only tables created by the import.

DATA\_OPTIONS

SKIP\_CONSTRAINT\_ERRORS

During import operations using the external table acces method, setting the DATA\_OPTIONS parameter to SKIP\_CONSTRAINT\_ERRORS allows load operations to continue through non-deferred constraint violations, with any violations logged for future reference. Without this, the default action would be to roll back the whole operation. The syntax is shown below.

DATA\_OPTIONS=SKIP\_CONSTRAINT\_ERRORS

An example is shown below.

impdp test/test tables=TAB1 directory=TEST\_DIR dumpfile=TEST.dmp logfile=impdpTEST.log

data\_options=SKIP\_CONSTRAINT\_ERRORS

This parameter has no impact on deferred constraints, which still cause the operation to be rolled back once a violation is detected. If the object being loaded has existing unique indexes or constraints, the APPEND hint will not be used, which may adversely affect performance.

XML\_CLOBS

During an export, if XMLTYPE columns are currently stored as CLOBs, they will automatically be exported as uncompressed CLOBs. If on the other hand they are currently stored as any combination of object-relational, binary or CLOB formats, they will be exported in compressed format by default. Setting the DATA\_OPTIONS parameter to XML\_CLOBS specifies that all XMLTYPE columns should be exported as uncompressed CLOBs, regardless of the default action. The syntax is shown below.

DATA\_OPTIONS=XML\_CLOBS

An example is shown below.

expdp test/test tables=TAB1 directory=TEST\_DIR dumpfile=TEST.dmp logfile=expdpTEST.log

version=11.1 data\_options=XML\_CLOBS

Both the export and import must use the same XML schema and the job version must be set to 11.0.0 or higher.

REMAP\_DATA

During export and import operations, the REMAP\_DATA parameter allows you to associate a remap packaged function that will accept the column value as a parameter and return a modified version of the data. The syntax is shown below.

REMAP\_DATA=[schema.]tablename.column\_name:[schema.]pkg.function

This can be used to mask sensitive data during export and import operations by replacing the original data with random alternatives. The mapping is done on a column-by-column basis, as shown below.

expdp test/test tables=TAB1 directory=TEST\_DIR dumpfile=TEST.dmp logfile=expdpTEST.log

remap\_data:tab1.col1:remap\_pkg.remap\_col1

remap\_data:tab1.col2:remap\_pkg.remap\_col2

The remapping function must return the same datatype as the source column and it must not perform a commit or rollback.

Miscellaneous Enhancements

Worker processes that have stopped due to certain errors will now have a one-time automatic restart. If the process stops a second time, it must be restarted manually.

Data Pump Enhancements in Oracle Database 12c Release 1 (expdp, impdp)

Oracle [Data Pump](https://oracle-base.com/articles/10g/oracle-data-pump-10g) was introduced in Oracle 10g. This article provides an overview of the main Data Pump enhancements in Oracle Database 12c Release 1, including the following.

* [NOLOGGING Option (DISABLE\_ARCHIVE\_LOGGING)](https://oracle-base.com/articles/12c/data-pump-enhancements-12cr1#nologging-option)
* [LOGTIME Parameter](https://oracle-base.com/articles/12c/data-pump-enhancements-12cr1#logtime-parameter)
* [Export View as Table](https://oracle-base.com/articles/12c/data-pump-enhancements-12cr1#export-view-as-table)
* [Change Table Compression at Import](https://oracle-base.com/articles/12c/data-pump-enhancements-12cr1#change-table-compression-at-import)
* [Change Table LOB Storage at Import](https://oracle-base.com/articles/12c/data-pump-enhancements-12cr1#change-table-lob-storage-at-import)
* [Dumpfile Compression Options](https://oracle-base.com/articles/12c/data-pump-enhancements-12cr1#dumpfile-compression-options)
* [Multitenant Option Support (CDB and PDB)](https://oracle-base.com/articles/12c/data-pump-enhancements-12cr1#multitenant-option-support)
* [Audit Commands](https://oracle-base.com/articles/12c/data-pump-enhancements-12cr1#audit-commands)
* [Encryption Password Enhancements](https://oracle-base.com/articles/12c/data-pump-enhancements-12cr1#encryption-password-enhancements)
* [Transportable Database](https://oracle-base.com/articles/12c/data-pump-enhancements-12cr1#transportable-database)
* [Miscellaneous Enhancements](https://oracle-base.com/articles/12c/data-pump-enhancements-12cr1#miscellaneous-enhancements)

NOLOGGING Option (DISABLE\_ARCHIVE\_LOGGING)

The TRANSFORM parameter of impdp has been extended to include a DISABLE\_ARCHIVE\_LOGGING option. The default setting of "N" has no affect on logging behaviour. Using a value "Y" reduces the logging associated with tables and indexes during the import by setting their logging attribute to NOLOGGING before the data is imported and resetting it to LOGGING once the operation is complete.

TRANSFORM=DISABLE\_ARCHIVE\_LOGGING:Y

The effect can be limited to a specific type of object (TABLE or INDEX) by appending the object type.

TRANSFORM=DISABLE\_ARCHIVE\_LOGGING:Y:TABLE

TRANSFORM=DISABLE\_ARCHIVE\_LOGGING:Y:INDEX

An example of its use is shown below.

$ impdp system/Password1@pdb1 directory=test\_dir dumpfile=emp.dmp logfile=impdp\_emp.log \

remap\_schema=scott:test **transform=disable\_archive\_logging:y**

The DISABLE\_ARCHIVE\_LOGGING option has no effect if the database is running in FORCE LOGGING mode.

LOGTIME Parameter

The LOGTIME parameter determines if timestamps should be included in the output messages from the expdp and impdp utilities.

LOGTIME=[NONE | STATUS | LOGFILE | ALL]

The allowable values are explained below.

* NONE : The default value, which indicates that no timestamps should be included in the output, making the output look similar to that of previous versions.
* STATUS : Timestamps are included in output to the console, but not in the associated log file.
* LOGFILE : Timestamps are included in output to the log file, but not in the associated console messages.
* ALL : Timestamps are included in output to the log file and console.

An example of the output is shown below.

$ expdp scott/tiger@pdb1 tables=emp directory=test\_dir dumpfile=emp.dmp logfile=expdp\_emp.log **logtime=all**

Export: Release 12.1.0.1.0 - Production on Wed Nov 20 22:11:57 2013

Copyright (c) 1982, 2013, Oracle and/or its affiliates. All rights reserved.

Connected to: Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 - 64bit Production

With the Partitioning, Oracle Label Security, OLAP, Advanced Analytics

and Real Application Testing options

20-NOV-13 22:12:09.312: Starting "SCOTT"."SYS\_EXPORT\_TABLE\_01": scott/\*\*\*\*\*\*\*\*@pdb1 tables=emp directory=test\_dir dumpfile=emp.dmp logfile=expdp\_emp.log logtime=all

20-NOV-13 22:12:13.602: Estimate in progress using BLOCKS method...

20-NOV-13 22:12:17.797: Processing object type TABLE\_EXPORT/TABLE/TABLE\_DATA

20-NOV-13 22:12:18.145: Total estimation using BLOCKS method: 64 KB

20-NOV-13 22:12:30.583: Processing object type TABLE\_EXPORT/TABLE/TABLE

20-NOV-13 22:12:33.649: Processing object type TABLE\_EXPORT/TABLE/INDEX/INDEX

20-NOV-13 22:12:37.744: Processing object type TABLE\_EXPORT/TABLE/CONSTRAINT/CONSTRAINT

20-NOV-13 22:12:38.065: Processing object type TABLE\_EXPORT/TABLE/INDEX/STATISTICS/INDEX\_STATISTICS

20-NOV-13 22:12:38.723: Processing object type TABLE\_EXPORT/TABLE/CONSTRAINT/REF\_CONSTRAINT

20-NOV-13 22:12:41.052: Processing object type TABLE\_EXPORT/TABLE/STATISTICS/TABLE\_STATISTICS

20-NOV-13 22:12:41.337: Processing object type TABLE\_EXPORT/TABLE/STATISTICS/MARKER

20-NOV-13 22:13:38.255: . . exported "SCOTT"."EMP" 8.75 KB 14 rows

20-NOV-13 22:13:40.483: Master table "SCOTT"."SYS\_EXPORT\_TABLE\_01" successfully loaded/unloaded

20-NOV-13 22:13:40.507: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

20-NOV-13 22:13:40.518: Dump file set for SCOTT.SYS\_EXPORT\_TABLE\_01 is:

20-NOV-13 22:13:40.545: /home/oracle/emp.dmp

20-NOV-13 22:13:40.677: Job "SCOTT"."SYS\_EXPORT\_TABLE\_01" successfully completed at Wed Nov 20 22:13:40 2013 elapsed 0 00:01:36

$

Export View as Table

The VIEWS\_AS\_TABLES parameter allows Data Pump to export the specified views as if they were tables. The table structure matches the view columns, with the data being the rows returned by the query supporting the views.

VIEWS\_AS\_TABLES=[schema\_name.]view\_name[:table\_name], ...

To see it working, create a view.

CONN scott/tiger@pdb1

CREATE VIEW emp\_v AS

SELECT \* FROM emp;

Now export the view using the VIEWS\_AS\_TABLES parameter.

$ expdp scott/tiger **views\_as\_tables=scott.emp\_v** directory=test\_dir dumpfile=emp\_v.dmp logfile=expdp\_emp\_v.log

By default expdp creates a temporary table as a copy of the view, but with no data, to provide a source of the metadata for the export. Alternatively to can specify a table with the appropriate structure. This probably only makes sense if you are using this functionality in a read-only database.

The are a number of restrictions relating to this parameter, which you can read about [here](http://docs.oracle.com/cd/E16655_01/server.121/e17639/dp_export.htm#SUTIL3904).

Change Table Compression at Import

The TABLE\_COMPRESSION\_CLAUSE clause of the TRANSFORM parameter allows the table compression characteristics of the tables in an import to be altered on the fly.

TRANSFORM=TABLE\_COMPRESSION\_CLAUSE:[NONE | compression\_clause]

The allowable values for the TABLE\_COMPRESSION\_CLAUSE include the following.

* NONE : The table compression clause is omitted, so the table takes on the compression characteristics of the tablespace.
* NOCOMPRESS : Disables table compression.
* COMPRESS : Enables basic table compression.
* ROW STORE COMPRESS BASIC : Same as COMPRESS.
* ROW STORE COMPRESS BASIC : Same as COMPRESS.
* ROW STORE COMPRESS ADVANCED : Enables advanced compression, also known as OLTP compression.
* COLUMN STORE COMPRESS FOR QUERY : Hybrid Columnar Compression (HCC) available in Exadata and ZFS storage appliances.
* COLUMN STORE COMPRESS FOR ARCHIVE : Hybrid Columnar Compression (HCC) available in Exadata and ZFS storage appliances.

 Compression clauses that contain whitespace must be enclosed by single or double quotes.

An example of its use is shown below.

$ impdp system/Password1@pdb1 directory=test\_dir dumpfile=emp.dmp logfile=impdp\_emp.log \

remap\_schema=scott:test **transform=table\_compression\_clause:compress**

Change Table LOB Storage at Import

The LOB\_STORAGE clause of the TRANSFORM parameter allows the table compression characteristics of the tables in a non-transportable import to be altered on the fly.

TRANSFORM=LOB\_STORAGE:[SECUREFILE | BASICFILE | DEFAULT | NO\_CHANGE]

The allowable values for the LOB\_STORAGE clause include the following.

* SECUREFILE : The LOBS are stored as SecureFiles.
* BASICFILE : The LOBS are stored as BasicFiles.
* DEFAULT : The LOB storage is determined by the database default.
* NO\_CHANGE : The LOB storage matches that of the source object.

An example of its use is shown below.

$ impdp system/Password1@pdb1 directory=test\_dir dumpfile=lob\_table.dmp logfile=impdp\_lob\_table.log \

**transform=lob\_storage:securefile**

Dumpfile Compression Options

As part of the Advanced Compression option, you can specify the COMPRESSION\_ALGORITHM parameter to determine the level of compression of the export dumpfile. This is not related to table compression discussed previously.

COMPRESSION\_ALGORITHM=[BASIC | LOW | MEDIUM | HIGH]

The meanings of the available values are described below.

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

An example of its use is shown below.

$ expdp scott/tiger tables=emp directory=test\_dir dumpfile=emp.dmp logfile=expdp\_emp.log \

**compression=all compression\_algorithm=medium**

Multitenant Option Support (CDB and PDB)

Oracle Database 12c introduced the multitenant option, allowing multiple pluggable databases (PDBs) to reside in a single container database (CDB). For the most part, using data pump against a PDB is indistinguishable from using it against a non-CDB instance.

Exports using the FULL option from 11.2.0.2 or higher can be imported into a clean PDB in the same way you would expect for a regular full import.

There are some minor restrictions, which you can read about [here](http://docs.oracle.com/cd/E16655_01/server.121/e17639/dp_overview.htm#SUTIL4334).

Audit Commands

Oracle 12c allows data pump jobs to be audited by creating an audit policy.

CREATE AUDIT POLICY policy\_name

ACTIONS COMPONENT=DATAPUMP [EXPORT | IMPORT | ALL];

When this policy is applied to a user, their data pump jobs will appear in the audit trail. The following policy audits all data pump operations. The policy is applied to the SCOTT user.

CONN / AS SYSDBA

CREATE AUDIT POLICY audit\_dp\_all\_policy ACTIONS COMPONENT=DATAPUMP ALL;

AUDIT POLICY audit\_dp\_all\_policy BY scott;

Run the following data pump command.

$ expdp scott/tiger tables=emp directory=test\_dir dumpfile=emp.dmp logfile=expdp\_emp.log

Checking the audit trail shows the data pump job was audited.

-- Flush audit information to disk.

EXEC DBMS\_AUDIT\_MGMT.FLUSH\_UNIFIED\_AUDIT\_TRAIL;

SET LINESIZE 200

COLUMN event\_timestamp FORMAT A30

COLUMN dp\_text\_parameters1 FORMAT A30

COLUMN dp\_boolean\_parameters1 FORMAT A30

SELECT event\_timestamp,

dp\_text\_parameters1,

dp\_boolean\_parameters1

FROM unified\_audit\_trail

WHERE audit\_type = 'Datapump';

EVENT\_TIMESTAMP DP\_TEXT\_PARAMETERS1 DP\_BOOLEAN\_PARAMETERS1

------------------------------ ------------------------------ ------------------------------

14-DEC-13 09.47.40.098637 PM MASTER TABLE: "SCOTT"."SYS\_EX MASTER\_ONLY: FALSE, DATA\_ONLY:

PORT\_TABLE\_01" , JOB\_TYPE: EXP FALSE, METADATA\_ONLY: FALSE,

ORT, METADATA\_JOB\_MODE: TABLE\_ DUMPFILE\_PRESENT: TRUE, JOB\_RE

EXPORT, JOB VERSION: 12.1.0.0. STARTED: FALSE

0, ACCESS METHOD: AUTOMATIC, D

ATA OPTIONS: 0, DUMPER DIRECTO

RY: NULL REMOTE LINK: NULL, T

ABLE EXISTS: NULL, PARTITION O

PTIONS: NONE

SQL>

Encryption Password Enhancements

In previous versions, data pump encryption required the ENCRYPTION\_PASSWORD parameter to be entered on the command line, making password snooping relatively easy.

In Oracle 12c, the ENCRYPTION\_PWD\_PROMPT parameter enables encryption without requiring the password to be entered as a command line parameter. Instead, the user is prompted for the password at runtime, with their response not echoed to the screen.

ENCRYPTION\_PWD\_PROMPT=[YES | NO]

An example of its use is shown below.

$ expdp scott/tiger tables=emp directory=test\_dir dumpfile=emp.dmp logfile=expdp\_emp.log \

**encryption\_pwd\_prompt=yes**

Export: Release 12.1.0.1.0 - Production on Sat Dec 14 21:09:11 2013

Copyright (c) 1982, 2013, Oracle and/or its affiliates. All rights reserved.

Connected to: Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 - 64bit Production

With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

**Encryption Password:**

Starting "SCOTT"."SYS\_EXPORT\_TABLE\_01": scott/\*\*\*\*\*\*\*\* tables=emp directory=test\_dir

dumpfile=emp.dmp logfile=expdp\_emp.log encryption\_pwd\_prompt=yes

Estimate in progress using BLOCKS method...

Processing object type TABLE\_EXPORT/TABLE/TABLE\_DATA

Total estimation using BLOCKS method: 64 KB

Processing object type TABLE\_EXPORT/TABLE/TABLE

Processing object type TABLE\_EXPORT/TABLE/INDEX/INDEX

Processing object type TABLE\_EXPORT/TABLE/CONSTRAINT/CONSTRAINT

Processing object type TABLE\_EXPORT/TABLE/INDEX/STATISTICS/INDEX\_STATISTICS

Processing object type TABLE\_EXPORT/TABLE/CONSTRAINT/REF\_CONSTRAINT

Processing object type TABLE\_EXPORT/TABLE/STATISTICS/TABLE\_STATISTICS

Processing object type TABLE\_EXPORT/TABLE/STATISTICS/MARKER

Processing object type TABLE\_EXPORT/TABLE/POST\_TABLE\_ACTION

. . exported "SCOTT"."EMP" 8.765 KB 14 rows

Master table "SCOTT"."SYS\_EXPORT\_TABLE\_01" successfully loaded/unloaded

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dump file set for SCOTT.SYS\_EXPORT\_TABLE\_01 is:

/tmp/emp.dmp

Job "SCOTT"."SYS\_EXPORT\_TABLE\_01" successfully completed at Sat Dec 14 21:09:55 2013 elapsed 0 00:00:41

$

Transportable Database

The TRANSPORTABLE option can now be combined with the FULL option to transport a whole database.

$ expdp system/Password1 **full=Y transportable=always** version=12 directory=TEMP\_DIR \

dumpfile=orcl.dmp logfile=expdporcl.log

This method can also be used to upgrade the database as described [here](https://oracle-base.com/articles/12c/upgrading-to-12c#transport-database).

Miscellaneous Enhancements

* Data Pump supports [extended data types](https://oracle-base.com/articles/12c/extended-data-types-12cR1), provided the VERSION parameter is not set to a value prior to 12.1.
* LOB columns with a domain index can now take advantage of the direct path load method.