Oracle Data Pump - Expdp, Impdp

- Create Data Pump Directory
- Table Export and Import
- Schema Export and Import
- Rows Export and Import
- Database Export and Import
- Data Pump Import Over Network
- Data Pump Performance Tuning
- EXPDP PAR File Example
- Schedule Data Pump Export in crontab
- Data Pump Export Progress %

Create Data Pump Directory

The first step in Oracle Data Pump is to create an OS level directory which will be used by Oracle for performing exports and imports. Create directory at OS level

```
mkdir -p /u02/dp_exp_dir
```

Create directory inside the database

```
SQL> create directory datapump as '/u02/dp_exp_dir';
```

Grant permissions on directory

```
SQL> grant read,write on directory datapump to scott;
```

View directory information

```
SQL> select * from dba_directories;
```

Get help on expdp or impdp utility

```
expdp help=y
impdp help=y
```

Table Export and Import

Take table level export

```
expdp directory=datapump dumpfile=emp_bkp.dmp logfile=emp_bkp.log tables='SCOTT.EMP'
```

Import table where source and target schema are same

```
$ impdp directory=datapump dumpfile=emp_bkp.dmp logfile=imp_emp.log tables='EMP'
```

Import table to another schema

```
$ impdp directory=datapump dumpfile=emp_bkp.dmp logfile=imp_emp.log tables='EMP'
remap_schema='SCOTT:HR'
```

Import tables to another tablespace (only in datapump)

```
$ impdp directory=datapump dumpfile=emp_bkp.dmp logfile=imp_emp.log tables='EMP'
remap_schema='SCOTT:HR' remap_tablespace='USERS:MYTBS'
```

Import table to a different name or rename table or remap_table

```
$ impdp directory=datapump dumpfile=emp_bkp.dmp logfile=imp_emp.log tables='EMP'
remap_table='SCOTT.EMP:HR.EMPLOYEE'
```

Import only the rows from an exported table without loading table any table definitions

```
$ impdp directory=datapump dumpfile=emp_bkp.dmp logfile=imp_emp.log tables='EMP'
content=DATA_ONLY
```

Schema Export and Import

Take schema level export

```
$ expdp directory=datapump dumpfile=scott_bkp.dmp logfile=scott_bkp.log schemas='SCOTT'
```

Import source schema objects into same schema on target

```
$ impdp directory=datapump dumpfile=scott_bkp.dmp logfile=imp_schema.log
remap_schema='SCOTT:SCOTT'
```

Import source schema objects into a different schema on target

```
$ impdp directory=datapump dumpfile=scott_bkp.dmp logfile=imp_schema.log
remap_schema='SCOTT:HR'
```

Rows Export and Import

Take row level export

```
$ expdp directory=datapump dumpfile=emprows_bkp.dmp logfile=emprows_bkp.log
tables='SCOTT.EMP' query=\"where deptno=10\"
```

Import rows where source and target schema are same

```
$ impdp directory=datapump dumpfile=emprows_bkp.dmp logfile=imp_emprows.log
tables='SCOTT.EMP'
```

Database Export and Import

Take database level export

```
$ expdp directory=datapump dumpfile=fullprod.dmp logfile=fullprod.log full=y
```

Import full database

```
-- On source
SQL> select name from v$tablespace;

-- On target
SQL> select name from v$tablespace;

-- Create missing tablespaces on target
-- Make sure target tablespace has enough free space

$ impdp directory=datapump dumpfile=fullprod.dmp logfile=imp_fullprod.log full=y
```

Data Pump Import Over Network

When you try to move large tables or schema between two Oracle databases, datapump export might take lot of disk space. The exported dump files may take lot of space on the disk. The best workaround is to use dblink with Oracle datapump to move data from one oracle database to another.

Note: when you use datapump with dblink, there are no dumpfiles created on source. The data is transferred from one database to another over network

Add source database TNS entry into tnsnames.ora of the target database

```
SHADB =
 (DESCRIPTION =
   (ADDRESS_LIST =
     (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.11.98)(PORT = 1521))
```

```
  )
 (CONNECT_DATA =
   (SERVICE_NAME = SHADB)
 )
)
```

On target, we need to create a database link using the TNS entry created above

```
create database link SOURCE_DB connect to scott identified by tiger  using 'SHADB';

Database link created.
```

It's time to import source schema on target database via db link. Run below command on target database to start import

```
SQL> create or replace directory MY_DUMP_DIRas'/u01/dump_files';
```

```
impdp directory=MY_DUMP_DIR LOGFILE=dblink_transfer.log
network_link=SOURCE_DB remap_schema=scott:hr
```

To import multiple schemas, make sure to use a DBA user (sys) to perform this action

```
impdp sys directory=MY_DUMP_DIR LOGFILE=dblink_transfer.log
network_link=SOURCE_DB schemas=IJS,scott,hr
```

Data Pump Performance Tuning

You can always use DIRECT=y parameter to perform faster exports and imports. You can also use PARALLEL parameter to start multiple export and import process for faster performance.

Make sure to use %U with the dumpfile name so multiple dumpfiles can be read/write simultaneously

```
expdp directory=DUMP_DIR dumpfile=SCOTT_%U.dmp logfile=expdp_SCOTT.log schemas=SCOTT
parallel=4
```

```
impdp directory=DUMP_DIR dumpfile=SCOTT_%U.dmp logfile=impdp_SCOTT.log schemas=SCOTT
parallel=4
```

EXPDP PAR File Example

Data Pump jobs can be automated using PAR file. You basically create one par file which contains all the export or import parameters and just call the par file at expdp utility

```
vi exp.par

Username=scott/tiger
tables=scott.emp
directory=EXP_DIR
```

```
dumpfile=QUERY_EXP.dmp
logfile=QUERY_EXP.log
parallel=7
```

```
expdp parfile=exp.par
```

You can specify any extension but it is recommended to use .par

Schedule Data Pump Export in crontab

Create a file which contains expdp script

```
vi daily_export.sh
```

```
export DATE=$(date +%m_%d_%y_%H_%M)
export ORACLE_SID=orcl
export ORACLE_HOME=/u01/app/oracle/product/11.2.0/db_1

$ORACLE_HOME/bin/expdp username/password@sid directory=export_dir
dumpfile=backup_$DATE.dmp logfile=backup_$DATE.log full=y
```

Give permissions to execute on above file

```
chmod 755 daily_export.sh
```

Schedule export under crontab

```
crontab –e –u oracle
```

```
0020***/home/oracle/backup_script
```

Data Pump Export Progress %

When you run a data pump export in the background and want to know the progress status, use below query to get the percentage (%) completion of the export process

```
SELECT SID, SERIAL#, USERNAME, CONTEXT, SOFAR,
TOTALWORK,ROUND(SOFAR/TOTALWORK*100,2) "%_COMPLETE"FROM V$SESSION_LONGOPS
WHERE TOTALWORK != 0 AND SOFAR <> TOTALWORK;
```

DATAPUMP STATUS

```
set line 190
col OWNER_NAME for a30
col JOB_NAME for a30
col OPERATION for a20
col JOB_MODE for a20
select OWNER_NAME,JOB_NAME,OPERATION,JOB_MODE,STATE from dba_datapump_jobs;
```