An Oracle White Paper
February 2009

Oracle Data Pump Quick Start

ORACLE®

Introduction

Oracle Data Pump is the replacement for the original Export and Import utilities. Available starting in Oracle Database 10g, Oracle Data Pump enables very high-speed movement of data and metadata from one database to another.

The Data Pump Export and Import utilities have a similar look and feel to the original utilities, but they are much more efficient and give you greater control and management of your import and export jobs.

This paper gives users of the original Export and Import utilities a primer on how to move up to the faster, more powerful, and more flexible Data Pump Export and Import utilities.

See the Oracle Database Utilities Guide for more comprehensive information about Oracle Data Pump.

New Concepts in Oracle Data Pump

There are two new concepts in Oracle Data Pump that are different from original Export and Import.

Directory Objects

Data Pump differs from original Export and Import in that all jobs run primarily on the server using server processes. These server processes access files for the Data Pump jobs using directory objects that identify the location of the files. The directory objects enforce a security model that can be used by DBAs to control access to these files.

Interactive Command-Line Mode

Besides regular operating system command-line mode, there is now a very powerful interactive command-line mode which allows the user to monitor and control Data Pump Export and Import operations.

Changing from Original Export/Import to Oracle Data Pump

Creating Directory Objects

In order to use Data Pump, the database administrator must create a directory object and grant privileges to the user on that directory object. If a directory object is not specified, a default directory object called data_pump_dir is provided. The default data_pump_dir is available only to privileged users unless access is granted by the DBA.

In the following example, the following SQL statement creates a directory object named dpump_dir1 that is mapped to a directory located at /usr/apps/datafiles.

create a directory.

1. **SQL> CREATE DIRECTORY dpump_dir1 AS '/usr/apps/datafiles';**

After a directory is created, you need to grant READ and WRITE permission on the directory to other users. For example, to allow the Oracle database to read and to write to files on behalf of user scott in the directory named by dpump_dir1, you must execute the following command:

2. SQL> GRANT READ,WRITE ON DIRECTORY dpump_dir1 TO scott;

Note that READ or WRITE permission to a directory object means only that the Oracle database will read or write that file on your behalf. You are not given direct access to those files outside of the Oracle database unless you have the appropriate operating system privileges. Similarly, the Oracle database requires permission from the operating system to read and write files in the directories.

Once the directory access is granted, the user scott can export his database objects with command arguments:

3. >expdp username/password DIRECTORY=dpump_dir1 dumpfile=scott.dmp

Comparison of command-line parameters from Original Export and Import to Data Pump

Data Pump commands have a similar look and feel to the original Export and Import commands, but are different. Below are a few examples that demonstrate some of these differences.

1) Example import of tables from scott's account to jim's account

Original Import:

> imp username/password FILE=scott.dmp FROMUSER=scott TOUSER=jim TABLES=(*)

Data Pump Import:

> impdp username/password DIRECTORY=dpump_dir1 DUMPFILE=scott.dmp TABLES=scott.emp REMAP_SCHEMA=scott:jim

Note how the FROMUSER/TOUSER syntax is replaced by the REMAP_SCHEMA option.

2) Example export of an entire database to a dump file with all GRANTS, INDEXES, and data

> exp username/password FULL=y FILE=dba.dmp GRANTS=y INDEXES=y ROWS=y

> expdp username/password FULL=y INCLUDE=GRANT INCLUDE= INDEX
DIRECTORY=dpump_dir1 DUMPFILE=dba.dmp CONTENT=ALL

Data Pump offers much greater metadata filtering than original Export and Import. The INCLUDE parameter allows you to specify which object (and its dependent objects) you want to keep in the export job. The EXCLUDE parameter allows you to specify which object (and its dependent objects) you want to keep out of the export job. You cannot mix the two parameters in one job. Both parameters work with Data Pump Import as well, and you can use different INCLUDE and EXCLUDE options for different operations on the same dump file.

3) Tuning Parameters

Unlike original Export and Import, which used the BUFFER, COMMIT, COMPRESS, CONSISTENT, DIRECT, and RECORDLENGTH parameters, Data Pump needs no tuning to achieve maximum performance. Data Pump chooses the best method to ensure that data and metadata are exported and imported in the most efficient manner. Initialization parameters should be sufficient upon installation.

4) Moving data between versions

The Data Pump method for moving data between different database versions is different from the method used by original Export and Import. With original Export, you had to run an older version of Export to produce a dump file that was compatible with an older database version. With Data Pump, you use the current Export version and simply use the VERSION parameter to specify the target database version. You cannot specify versions earlier than Oracle Database 10g (since Data Pump did not exist before 10g).

Example:

> expdp username/password TABLES=hr.employees VERSION=10.1
DIRECTORY=dpump_dir1 DUMPFILE=emp.dmp

Data Pump Import can always read dump file sets created by older versions of Data Pump Export.

Note that Data Pump Import cannot read dump files produced by original Export.

Maximizing the Power of Oracle Data Pump

Data Pump works great with default parameters, but once you are comfortable with Data Pump, there are new capabilities that you will want to explore.

Parallelism

Data Pump Export and Import operations are processed in the database as a Data Pump job, which is much more efficient that the client-side execution of original Export and Import. Now Data Pump operations can take advantage of the server's parallel processes to read or write multiple data streams simultaneously (PARALLEL is only available in the Enterprise Edition of Oracle Database.)

The number of parallel processes can be changed on the fly using Data Pump's interactive command-line mode. You may have a certain number of processes running during the day and decide to change that number if more system resources become available at night (or vice versa).

For best performance, you should do the following:

- Make sure your system is well balanced across CPU, memory, and I/O.

- Have at least one dump file for each degree of parallelism. If there aren't enough dump files, performance will not be optimal because multiple threads of execution will be trying to access the same dump file.

- Put files that are members of a dump file set on separate disks so that they will be written and read in parallel.

- For export operations, use the %U variable in the DUMPFILE parameter so multiple dump files can be automatically generated.

Example:

> expdp username/password DIRECTORY=dpump_dir1 JOB_NAME=hr
DUMPFILE=par_exp%u.dmp PARALLEL=4

REMAP

- **REMAP_TABLESPACE** – This allows you to easily import a table into a different tablespace from which it was originally exported. The databases have to be 10.1 or later.

  Example:

  > impdp username/password REMAP_TABLESPACE=tbs_1:tbs_6
  DIRECTORY=dpumpdir1 DUMPFILE=employees.dmp

- **REMAP_DATAFILES** – This is a very useful feature when you move databases between platforms that have different file naming conventions. This parameter changes the source datafile name to the target datafile name in all SQL statements where the source datafile is referenced. Because the REMAP_DATAFILE value uses quotation marks, it's best to specify the parameter within a parameter file.

Example:

The parameter file, payroll.par, has the following content:

DIRECTORY=dpump_dir1

FULL=Y

DUMPFILE=db_full.dmp

REMAP_DATAFILE="'C:\DB1\HRDATA\PAYROLL\tbs6.dbf':'/db1/hrdata/payroll/tbs6.dbf '"

You can then issue the following command:

> impdp username/password PARFILE=payroll.par

Even More Advanced Features of Oracle Data Pump

Beyond the command-line and performance features of Oracle Data Pump are new capabilities that DBAs will find invaluable. A couple of prominent features are described here.

Interactive Command-Line Mode

You have much more control in monitoring and controlling Data Pump jobs with interactive command-line mode. Because Data Pump jobs run entirely on the server, you can start an export or import job, detach from it, and later reconnect to the job to monitor its progress. Here are some of the things you can do while in this mode:

- See the status of the job. All of the information needed to monitor the job's execution is available.

- Add more dump files if there is insufficient disk space for an export file.

- Change the default size of the dump files.

- Stop the job (perhaps it is consuming too many resources) and later restart it (when more resources become available).

- Restart the job. If a job was stopped for any reason (system failure, power outage), you can attach to the job and then restart it.

- Increase or decrease the number of active worker processes for the job. (Enterprise Edition only.)

- Attach to a job from a remote site (such as from home) to monitor status.

Network Mode

Data Pump gives you the ability to pass data between two databases over a network (via a database link), without creating a dump file on disk. This is very useful if you're moving data between databases, like data marts to data warehouses, and disk space is not readily available. Note that if you are moving large volumes of data, Network mode is probably going to be slower than file mode.

Network export creates the dump file set on the instance where the Data Pump job is running and extracts the metadata and data from the remote instance.

Network export gives you the ability to export read-only databases. (Data Pump Export cannot run locally on a read-only instance because the job requires write operations on the instance.) This is useful when there is a need to export data from a standby database.

Generating SQLFILES

In original Import, the INDEXFILE parameter generated a text file which contained the SQL commands necessary to recreate tables and indexes that you could then edit to get a workable DDL script.

With Data Pump, it's a lot easier to get a workable DDL script. When you run Data Pump Import and specify the SQLFILE parameter, a text file is generated that has the necessary DDL (Data Definition Language) in it to recreate all object types, not just tables and indexes. Although this output file is ready for execution, the DDL statements are not actually executed, so the target system will not be changed.

SQLFILEs can be particularly useful when pre-creating tables and objects in a new database. Note that the INCLUDE and EXCLUDE parameters can be used for tailoring sqlfile output. For example, if you want to create a database that contains all the tables and indexes of the source database, but that does not include the same constraints, grants, and other metadata, you would issue a command as follows:

>impdp username/password DIRECTORY=dpumpdir1 DUMPFILE=expfull.dmp
SQLFILE=dpump_dir2:expfull.sql INCLUDE=TABLE,INDEX

The SQL file named expfull.sql is written to dpump_dir2 and would include SQL DDL that could be executed in another database to create the tables and indexes as desired.

Frequently Asked Questions

Why are directory objects needed?

They are needed to ensure data security and integrity. Otherwise, users would be able to read data that they should not have access to and perform unwarranted operations on the server.

What makes Data Pump faster than original Export and Import?

There are three main reasons that Data Pump is faster than original Export and Import. First, the Direct Path data access method (which permits the server to bypass SQL and go right to the data blocks on disk) has been rewritten to be much more efficient and now supports Data Pump Import and Export.   Second, because Data Pump does its processing on the server rather than in the client, much less data has to be moved between client and server. Finally, Data Pump was designed from the ground up to take advantage of modern hardware and operating system architectures in ways that original Export/ and Import cannot. These factors combine to produce significant performance improvements for Data Pump over original Export and Import

How much faster is Data Pump than the original Export and Import utilities?

For a single stream, Data Pump Export is approximately 2 times faster than original Export and Data Pump Import is approximately 15 to 40 times faster than original Import. Speed can be dramatically improved using the PARALLEL parameter.

Why is Data Pump slower on small jobs?

Data Pump was designed for big jobs with lots of data. Each Data Pump job has a master table that has all the information about the job and is needed for restartability. The overhead of creating this master table makes small jobs take longer, but the speed in processing large amounts of data gives Data Pump a significant advantage in medium and larger jobs.

Are original Export and Import going away?

Original Export is being deprecated with the Oracle Database 11g release. Original Import will always be supported so that dump files from earlier releases (release 5.0 and later) will be able to be imported. Original and Data Pump dump file formats are not compatible.

Are Data Pump dump files and original Export and Import dump files compatible?

No, the dump files are not compatible or interchangeable. If you have original Export dump files, you must use original Import to load them.

How can I monitor my Data Pump jobs to see what is going on?

In interactive mode, you can get a lot of detail through the STATUS command. In SQL, you can query the following views:

• DBA_DATAPUMP_JOBS - all active Data Pump jobs and the state of each job

• USER_DATAPUMP_JOBS – summary of the user's active Data Pump jobs

• DBA_DATAPUMP_SESSIONS – all active user sessions that are attached to a Data Pump job

• V$SESSION_LONGOPS – shows all progress on each active Data Pump job

Can I use Oracle Enterprise Manager with Data Pump?

Yes, OEM supports a fully functional interface to Data Pump.

Can I use gzip with Data Pump?

Because Data Pump uses parallel operations to achieve its high performance, you cannot pipe the output of Data Pump export through gzip. Starting in Oracle Database 11g, the COMPRESSION parameter can be used to compress a Data Pump dump file as it is being created. The COMPRESSION parameter is available as part of the Advanced Compression Option for Oracle Database 11g.

Conclusion

Data Pump is fast and flexible. It replaces original Export and Import starting in Oracle Database 10g. Moving to Data Pump is easy, and opens up a world of new options and features.

ORACLE®

Oracle Data Pump Quick Start
February 2009
Author: Carol Palmer

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

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