C

Migrating Data by Using Oracle Data Pump

ORACLE

Objectives

After completing this lesson, you should be able to migrate data by using Oracle Data Pump.

Oracle Data Pump: Overview

As a server-based facility for high-speed data and metadata movement, Oracle Data Pump:

- Is callable via `DBMS_DATAPUMP`
- Provides the following tools:
  - `expdp`
  - `impdp`
  - GUI interface in Enterprise Manager Cloud Control
- Provides four data movement methods:
  - Data file copying
  - Direct path
  - External tables
  - Network link support
- Detaches from and re-attaches to long-running jobs
- Restarts Data Pump jobs

Oracle Data Pump enables high-speed data and metadata loading and unloading of Oracle databases. The Data Pump infrastructure is callable via the `DBMS_DATAPUMP` PL/SQL package. Thus, custom data movement utilities can be built by using Data Pump.

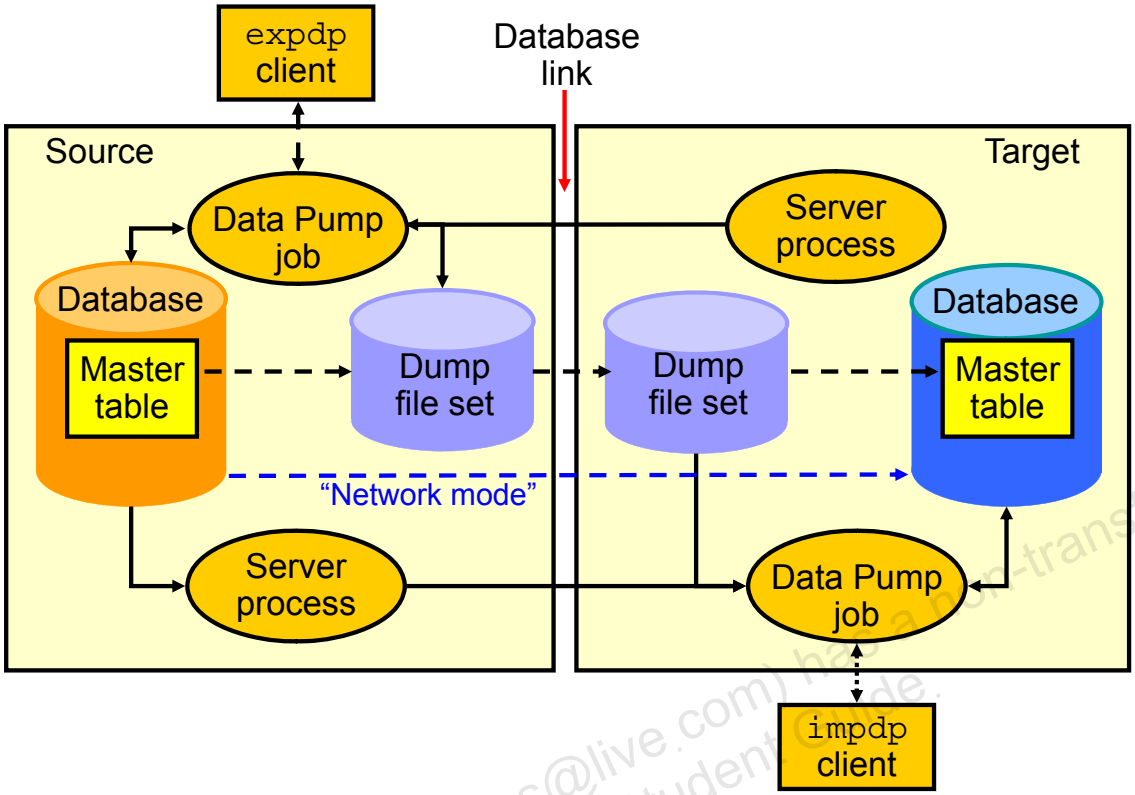
Oracle Database provides the following tools:

- Command-line export and import clients called `expdp` and `impdp`, respectively
- Export and import interface in Enterprise Manager Cloud Control

Data Pump automatically decides the data access methods to use; these can be either direct path or external tables. Data Pump uses direct path load and unload when a table's structure allows it and when maximum single-stream performance is desired. However, if there are clustered tables, referential integrity constraints, encrypted columns, or several other items, Data Pump uses external tables rather than direct path to move the data.

The ability to detach from and re-attach to long-running jobs without affecting the job itself enables you to monitor jobs from multiple locations while they are running. All stopped Data Pump jobs can be restarted without loss of data as long as the metainformation remains undisturbed. It does not matter whether the job is stopped voluntarily or involuntarily due to a crash.

Data Pump Export and Import Clients: Overview



Data Pump Export is a utility for unloading data and metadata into a set of operating system files called *dump file sets*. Data Pump Import is used to load metadata and data stored in an export dump file set into a target system.

The Data Pump API accesses its files on the server rather than on the client.

These utilities can also be used to export from a remote database directly to a dump file set, or to load the target database directly from a source database with no intervening files. This is known as *network mode*. This mode is particularly useful to export data from a read-only source database.

At the center of every Data Pump operation is the master table (MT), which is a table created in the schema of the user running the Data Pump job. The MT maintains all aspects of the job. The MT is built during a file-based export job and is written to the dump file set as the last step. Conversely, loading the MT into the current user's schema is the first step of a file-based import operation and is used to sequence the creation of all objects imported.

Note: The MT is the key to Data Pump's restart capability in the event of a planned or unplanned stopping of the job. The MT is dropped when the Data Pump job finishes normally.

Data Pump Utility: Interfaces and Modes

- Data Pump Export and Import interfaces:
  - Command line
  - Parameter file
  - Interactive command line
  - Enterprise Manager Cloud Control
- Data Pump Export and Import modes:
  - Full
  - Schema
  - Table
  - Tablespace
  - Transportable tablespace
  - Transportable database

You can interact with Data Pump Export and Import by using one of the following interfaces:

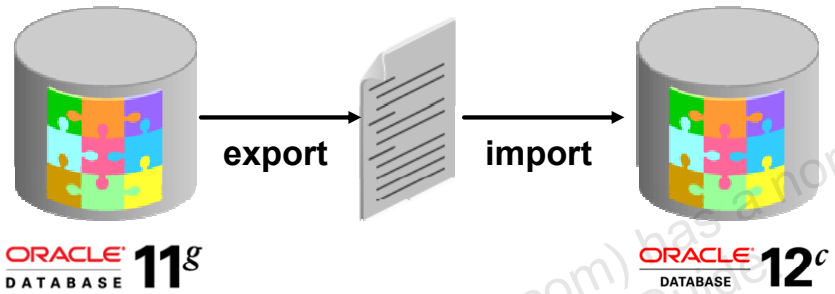
- **Command-line interface:** Enables you to specify most of the export parameters directly on the command line
- **Parameter file interface:** Enables you to specify all command-line parameters in a parameter file. The only exception is the `PARFILE` parameter.
- **Interactive-command interface:** Stops logging to the terminal and displays the export or import prompts, where you can enter various commands. This mode is enabled by pressing Ctrl + C during an export operation that is started with the command-line interface or the parameter file interface. Interactive-command mode is also enabled when you attach to an executing or stopped job.
- **GUI interface:** Select Schema > Database Export/Import. In the menu select the export or import operation you want to execute.

Data Pump Export and Import provide different modes for unloading and loading different portions of the database. The mode is specified on the command line by using the appropriate parameter. The available modes are listed in the slide and are the same as in the original export and import utilities.

Migrating by Using Oracle Data Pump

To migrate to Oracle Database 12*c* by using Oracle Data Pump:

1. Export data from the source database
2. Install Oracle Database 12*c* and create a database
3. Import data into the new Oracle Database 12*c* database

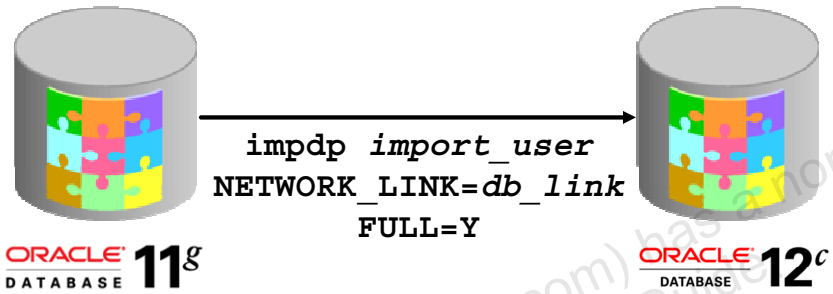


You can use Oracle Dump Export and Import to migrate your database to Oracle Database 12*c*. This method provides the following advantages:

- The database can remain available for user access during the upgrade. If a consistent snapshot of the database is required, the database must operate in restricted mode or must otherwise be protected from changes when the export executes.
- The upgraded database can be created on a different operating system or hardware platform.
- The database can be restructured as part of the import. You can create new tablespaces or modify existing tables, tablespaces, or partitions to be populated by imported data.

Importing by Using a Network Link

- Use Data Pump Import with a database link for a full database import
- User performing the Export must have the `DATAPUMP_EXP_FULL_DATABASE` role
- User performing the Import must have the `DATAPUMP_IMP_FULL_DATABASE` role

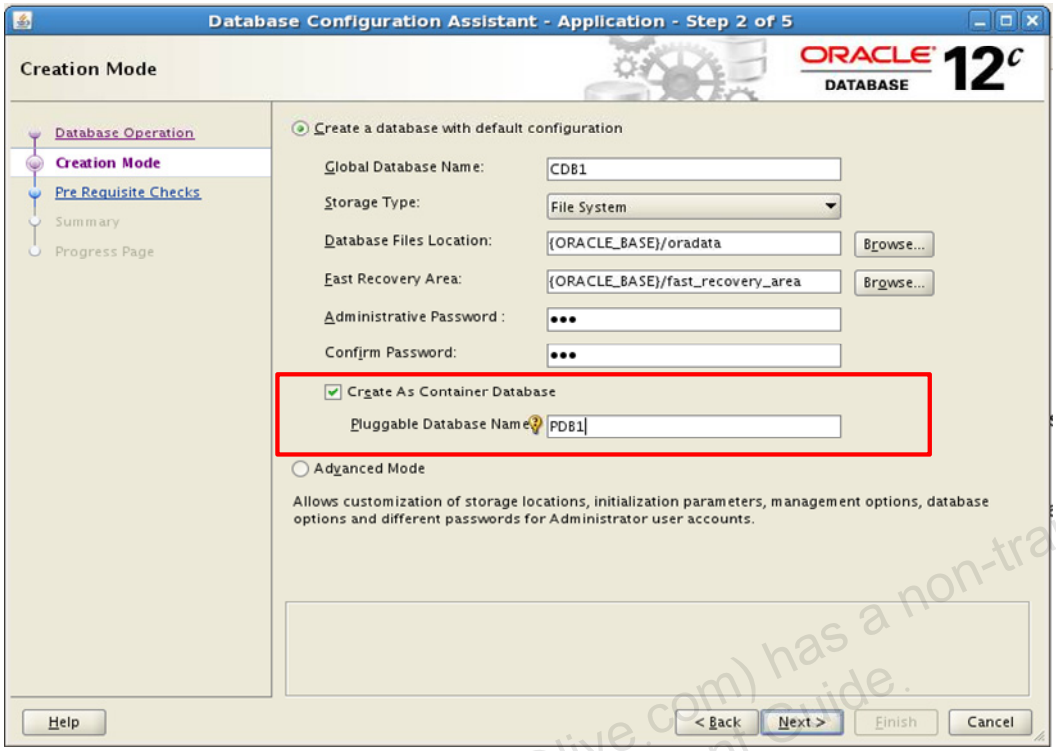


Data Pump Import can be used with a database link to perform a full database import. When you use this method, no intermediate dump files are created.

The user performing the export must be granted the `DATAPUMP_EXP_FULL_DATABASE` role. Specify this user when the database link is created.

The user performing the import must be granted the `DATAPUMP_IMP_FULL_DATABASE` role.

Creating a Container Database by Using DBCA

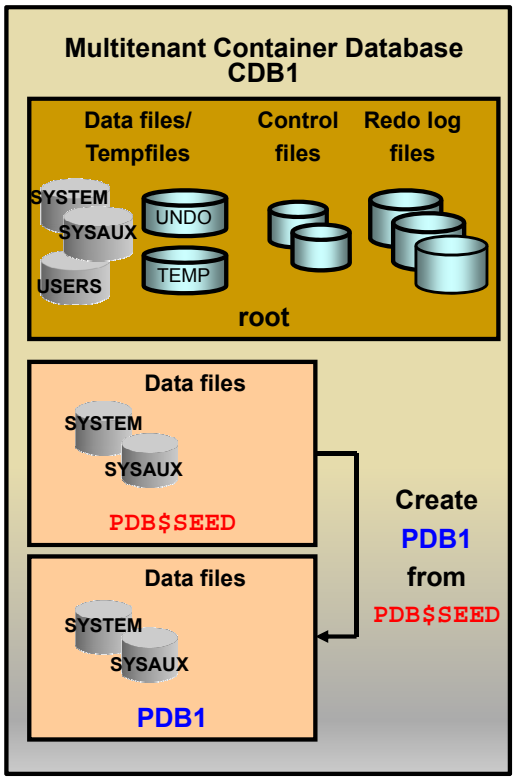


Another way to migrate a non-CDB database to a CDB is to create a multitenant container database and then import the database. To create a CDB by using DBCA, select “Create As Container Database.”

You also must provide a pluggable database name in the “Pluggable Database Name” field. If you select “Advanced Mode” you can create an empty CDB with only the root and seed containers.

In Advanced Mode, you can register the CDB with Enterprise Manager Cloud Control, configure the CDB for Enterprise Manager Database Express, and set passwords for the `SYS` and `SYSTEM` users.

Creating a New PDB from PDB$SEED



- Copies the data files from PDB$SEED data files
- Creates SYSTEM, SYSAUX tablespaces
- Creates a full catalog including metadata pointing to Oracle supplied objects
- Creates common users:
  - Superuser SYS
  - SYSTEM
- Creates a local user (PDBA) granted local PDB_DBA role
- Creates a new default service

ORACLE

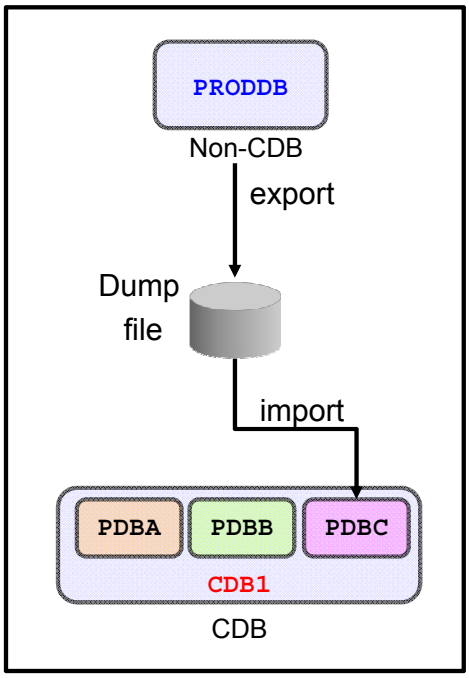
The creation of a new PDB from the seed is nearly instantaneous. The operation copies the data files from the READ ONLY seed PDB to the target directory defined in the CREATE PLUGGABLE DATABASE statement.

It creates tablespaces such as SYSTEM to store a full catalog including metadata pointing to Oracle-supplied objects, SYSAUX for local auxiliary data.

It creates default schemas and common users that exist in seed PDB, SYS who continues to have all superuser privileges, and SYSTEM who can administer the PDB.
It creates a local user (the PDBA) granted a local PDB_DBA role. Until the PDB SYS user grants privileges to the local PDB_DBA role, the new PDBA cannot perform any other operation than connecting to the PDB.

A new default service is also created for the PDB.

Exporting from a Non-CDB and Importing into a PDB



1. Export **PRODDB** with FULL clause:

```
$ expdp system@PRODDB FULL=Y
    DUMPFILE=proddb.dmp
```

2. Create **PDBC** if it does not exist in **CDB1**:

```
SQL> CONNECT sys@CDB1
SQL> CREATE PLUGGABLE DATABASE
  2  PDBC USING pdb$seed;
```

3. Open **PDBC**
4. Create a Data Pump directory in **PDBC**
5. Copy the dumpfile to the directory
6. Create same **PRODDB** tablespaces in **PDBC** for new local users objects
7. Import into **PDBC** with FULL and REMAP clauses:

```
$ impdp system@PDBC FULL=Y
    DUMPFILE=proddb.dmp
```

ORACLE

To export data from a non-CDB and import it into a PDB of a CDB, perform the steps as described in the slide.

The example in the slide describes how to perform a conventional full database export from the non-CDB and a conventional full database import into the PDB. You can also perform a full transportable, a tablespace, schema, or table-level export and import.

The tablespace export and import can be of either type: conventional or transportable.

The users exported from the non-CDB database are re-created as local users in the PDB.

The tablespaces for the new local users and objects must be created in the PDB before the import.

Summary

In this lesson, you should have learned how to:

- Migrate data by using Oracle Data Pump

Practice C

- C-1: Performing a Transportable Tablespace Import
- C-2: Importing a Non-CDB Application into a CDB