**TRANSPORTABLE TABLESPACES IN ORACLE**

A **Transportable Tablespace** (TTS) is a feature in Oracle that allows you to move large amounts of data between Oracle databases efficiently.

This is particularly useful when migrating or replicating data across databases with minimal downtime.

It involves transporting the **tablespaces** (including their data files) from one database to another while maintaining the integrity of the data.

TTS can be performed with both **local** and **cross-platform** (different operating systems or hardware) transportable tablespaces.

**Key Concepts of Transportable Tablespaces:**

1. **Tablespace**: A logical storage container for database objects such as tables and indexes.
2. **Data Files**: Physical files on disk that store the data of the tablespaces.
3. **Metadata**: The dictionary information about the tablespace that is stored in the database.

**Types of Transportable Tablespaces:**

* **Local Transportable Tablespaces**: Involves the movement of tablespace files from one Oracle database to another within the same platform and operating system.
* **Cross-Platform Transportable Tablespaces**: Enables the transport of tablespace data between different platforms, such as from UNIX to Windows, or between different endianness (e.g., Big Endian to Little Endian).

**Advantages of Using Transportable Tablespaces:**

* **Efficient Data Movement**: The tablespace data is moved without requiring a full export/import process, which is much slower.
* **Minimal Downtime**: TTS allows you to move the data with minimal downtime, especially in cases where the source and target database are both running during the process.
* **Large Data Handling**: It is effective for moving large amounts of data, especially when the data set spans several tablespaces.

**Steps to Perform Transportable Tablespace:**

There are several steps involved in transporting a tablespace:

1. **Pre-requisites**:
   * Ensure that the target database is the same version (or higher) than the source database.
   * Both databases must be in the same endianness format for local transport. For cross-platform TTS, you need to use Oracle’s **Data Pump** to convert the data.
2. **Preparing the Source Database**:
   * Ensure that the tablespace is in read-only mode.
   * You need to verify that no active DML operations are happening in the source database.
3. **Steps to Export the Tablespace**:
   * Use the **expdp** (Data Pump Export) utility to export the tablespace metadata.
   * Copy the data files of the tablespace.
4. **Steps to Import the Tablespace**:
   * Use the **impdp** (Data Pump Import) utility to import the metadata into the target database.

**Detailed Steps for Transportable Tablespace (TTS)**

**Step 1: Preparing the Source Database**

1. **Set the Source Tablespace to Read-Only:**

Before exporting the tablespace, set the tablespace to **READ-ONLY** mode to ensure no changes are made while the data is being moved.

ALTER TABLESPACE <tablespace\_name> READ ONLY;

1. **Check for Dependent Objects**:

Ensure there are no dependent objects (such as foreign keys, materialized views, etc.) before proceeding. This can be done by checking the database objects associated with the tablespace:

SELECT \* FROM DBA\_DEPENDENCIES WHERE REFERENCED\_OBJECT\_NAME = '<tablespace\_name>';

1. **Verify the Tablespace**:

Verify that the tablespace is in a consistent state for transport:

SELECT tablespace\_name, status FROM dba\_tablespaces WHERE tablespace\_name = '<tablespace\_name>';

**Step 2: Exporting the Tablespace Metadata Using Data Pump (expdp)**

The **expdp** utility is used to export the metadata of the tablespace to a dump file. This includes the dictionary information (tables, indexes, etc.).

1. **Create a Directory for Export**:

Create a directory on the source database file system to store the export dump file:

CREATE OR REPLACE DIRECTORY data\_pump\_dir AS '/path/to/export/directory';

GRANT READ, WRITE ON DIRECTORY data\_pump\_dir TO <username>;

1. **Export the Tablespace**:

Run the **expdp** command to export the tablespace metadata:

expdp <username>/<password> DIRECTORY=data\_pump\_dir DUMPFILE=<tablespace\_name>\_metadata.dmp LOGFILE=expdp\_<tablespace\_name>.log TRANSPORTABLE=ALWAYS TABLESPACES=<tablespace\_name>;

* + **TRANSPORTABLE=ALWAYS**: This ensures that the metadata is exported for transportable tablespaces.
  + **TABLESPACES=<tablespace\_name>**: Specifies the tablespace to be exported.

After the export completes, you should have the metadata dump file ready for transport.

**Step 3: Copy the Data Files**

1. **Identify Data Files**:

Identify the data files associated with the tablespace you want to transport. You can find them in the **dba\_data\_files** view:

SELECT file\_name FROM dba\_data\_files WHERE tablespace\_name = '<tablespace\_name>';

1. **Copy the Data Files to the Target Database**:

Copy the identified data files to the target database's filesystem. Ensure that the directories are accessible from the target database.

**Step 4: Import the Tablespace into the Target Database**

1. **Prepare the Target Database**:

Before importing, create the target tablespace if it doesn't already exist, and make sure the target database has enough storage space for the new data files.

CREATE TABLESPACE <tablespace\_name> DATAFILE '/path/to/datafile' SIZE <size> AUTOEXTEND ON;

1. **Import the Tablespace Metadata**:

Use the **impdp** utility to import the tablespace metadata into the target database:

impdp <username>/<password> DIRECTORY=data\_pump\_dir DUMPFILE=<tablespace\_name>\_metadata.dmp LOGFILE=impdp\_<tablespace\_name>.log TRANSPORTABLE=ALWAYS;

This will import the metadata dump into the target database, creating the necessary dictionary information and database objects.

**Step 5: Make the Tablespace Read/Write in the Target Database**

After the import is complete, you need to set the tablespace to **read/write** mode on the target database:

ALTER TABLESPACE <tablespace\_name> READ WRITE;

**Step 6: Verifying the Transportable Tablespace**

To verify the success of the transportable tablespace operation, you can check the dba\_tablespaces view in the target database:

SELECT tablespace\_name, status FROM dba\_tablespaces WHERE tablespace\_name = '<tablespace\_name>';

You should now see the imported tablespace in the target database, and the data should be available for use.

**Example Scenario**

Let’s assume we have a tablespace named **users\_tbs** in the source database and want to transport it to a target database.

1. **Source Database:**
2. ALTER TABLESPACE users\_tbs READ ONLY;
3. **Exporting Metadata:**
4. expdp admin/password DIRECTORY=data\_pump\_dir DUMPFILE=users\_tbs\_metadata.dmp LOGFILE=expdp\_users\_tbs.log TRANSPORTABLE=ALWAYS TABLESPACES=users\_tbs;
5. **Copy Data Files**: Identify the data files for users\_tbs and copy them to the target system.
6. **Target Database**:
   * Create a tablespace (if necessary):
   * CREATE TABLESPACE users\_tbs DATAFILE '/path/to/new/datafile/users\_tbs.dbf' SIZE 100M AUTOEXTEND ON;
   * Import metadata:
   * impdp admin/password DIRECTORY=data\_pump\_dir DUMPFILE=users\_tbs\_metadata.dmp LOGFILE=impdp\_users\_tbs.log TRANSPORTABLE=ALWAYS;
7. **Make the Tablespace Read/Write**:
8. ALTER TABLESPACE users\_tbs READ WRITE;

**Important Considerations**

* **Compatibility**: Ensure that the source and target database versions are compatible for transport. In some cases, using Oracle Data Pump for export/import is necessary.
* **Cross-Platform**: For cross-platform TTS, you need to convert data files from the source platform's format to the target platform’s format. Oracle Data Pump handles this conversion.
* **Foreign Keys & Constraints**: If the tablespace contains foreign keys or constraints that reference objects in other tablespaces, you need to address these dependencies before transport.
* **Downtime**: While the transport process minimizes downtime, you should still plan for some downtime, especially when switching the tablespace from read-only to read/write mode.

**Conclusion**

Transportable Tablespaces (TTS) provide an efficient and fast way to move large amounts of data between Oracle databases. By following the steps outlined above, you can successfully perform TTS operations with minimal downtime, ensuring that your data is moved quickly and accurately between databases.