Migrating an Oracle 19c database from Windows to Linux using **Transportable Tablespaces (TTS)** is an efficient method, especially for large databases, as it avoids the need to move the entire database and focuses on transporting only the tablespaces. Here’s a step-by-step guide with commands and expected outputs:

**Prerequisites**

1. Ensure both source (Windows) and target (Linux) databases are **Oracle 19c** and have compatible character sets and endian formats.
   * Check the endian format:
   * SELECT \* FROM v$transportable\_platform;
   * Verify the source platform:
   * SELECT PLATFORM\_NAME FROM v$database;
2. The target database must have the same tablespaces created or remapped before the import.

**Step-by-Step Process**

**1. Identify Tablespaces to Transport**

On the source (Windows) database, list the tablespaces to transport:

SELECT tablespace\_name FROM dba\_tablespaces WHERE tablespace\_name NOT IN ('SYSTEM', 'SYSAUX', 'TEMP', 'UNDO');

Example output:

TABLESPACE\_NAME

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

USERS

DATA\_TBS

**2. Put Tablespaces in Read-Only Mode**

Set the tablespaces to read-only mode to ensure no changes occur during the migration:

ALTER TABLESPACE USERS READ ONLY;

ALTER TABLESPACE DATA\_TBS READ ONLY;

**3. Export Tablespace Metadata**

Use **Data Pump Export (expdp)** to export the metadata of the tablespaces:

expdp "sys/oracle as sysdba" directory=DATAPUMP\_DIR dumpfile=tts\_metadata.dmp logfile=tts\_export.log transport\_tablespaces=USERS,DATA\_TBS

**Expected Output (Log File tts\_export.log):**

Export: Release 19.0.0.0.0 - Production

Starting "SYS"."SYS\_EXPORT\_TRANSPORTABLE\_01": ...

Processing object type TRANSPORTABLE\_EXPORT/PLUGTS\_BLK

Processing object type TRANSPORTABLE\_EXPORT/TABLE

Processing object type TRANSPORTABLE\_EXPORT/INDEX

. . exported "HR"."EMPLOYEES" ...

Exported 2 tablespaces: USERS, DATA\_TBS

**4. Copy Datafiles and Metadata to Linux**

* Copy the datafiles of the tablespaces from the Windows server to the Linux server.

List datafiles for the tablespaces:

SELECT file\_name FROM dba\_data\_files WHERE tablespace\_name IN ('USERS', 'DATA\_TBS');

Example output:

FILE\_NAME

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

C:\ORACLE\ORADATA\PRODDB\USERS01.DBF

C:\ORACLE\ORADATA\PRODDB\DATA\_TBS01.DBF

Transfer files using scp or a similar tool:

scp C:\ORACLE\ORADATA\PRODDB\USERS01.DBF oracle@linux-server:/u01/oradata/PRODDB/

scp C:\ORACLE\ORADATA\PRODDB\DATA\_TBS01.DBF oracle@linux-server:/u01/oradata/PRODDB/

scp C:\ORACLE\DATAPUMP\tts\_metadata.dmp oracle@linux-server:/u01/datapump/

**5. Import Tablespace Metadata on Linux**

On the target (Linux) database, ensure the tablespaces' datafiles are accessible in the correct path and use **Data Pump Import (impdp)** to import the metadata:

impdp "sys/oracle as sysdba" directory=DATAPUMP\_DIR dumpfile=tts\_metadata.dmp logfile=tts\_import.log transport\_datafiles='/u01/oradata/PRODDB/USERS01.DBF','/u01/oradata/PRODDB/DATA\_TBS01.DBF'

**Expected Output (Log File tts\_import.log):**

Import: Release 19.0.0.0.0 - Production

Processing object type TRANSPORTABLE\_EXPORT/PLUGTS\_BLK

Processing object type TRANSPORTABLE\_EXPORT/TABLE

Processing object type TRANSPORTABLE\_EXPORT/INDEX

Successfully imported tablespace USERS

Successfully imported tablespace DATA\_TBS

**6. Set Tablespaces to Read/Write Mode**

After verifying the import, change the tablespaces back to read/write mode:

ALTER TABLESPACE USERS READ WRITE;

ALTER TABLESPACE DATA\_TBS READ WRITE;

**Verification and Post-Migration Steps**

**1. Verify Tablespaces**

Check if the tablespaces are online and accessible:

SELECT tablespace\_name, status FROM dba\_tablespaces;

Expected output:

TABLESPACE\_NAME STATUS

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

USERS ONLINE

DATA\_TBS ONLINE

**2. Verify Data**

Ensure the data in the tablespaces matches the source database by querying tables:

SELECT \* FROM hr.employees WHERE rownum <= 5;

**3. Compile Invalid Objects**

Compile any invalid objects:

EXEC DBMS\_UTILITY.compile\_schema(schema => 'HR');

**Rollback Plan (If Needed)**

If the migration fails or inconsistencies are found:

1. Revert the tablespaces to read/write mode on the source:
2. ALTER TABLESPACE USERS READ WRITE;
3. ALTER TABLESPACE DATA\_TBS READ WRITE;
4. Fix the issue and repeat the migration process.

**Advantages of TTS**

* Faster than full export/import for large databases.
* Minimizes downtime as you can transport only the required tablespaces.

Let me know if you face any specific issues or want more details on any step! 😊