

Cross Platform Transportable Tablespaces Migration in Oracle 11g



Prepared by
ViSolve Migration Team

June 2012

Contact

ViSolve, Inc.
4010, Moorpark Avenue,
#205 San Jose, California 95117
(602) 842 2738
migration_engg@visolve.com
www.visolve.com

Table of Contents

Introduction	3
Assumptions made	3
<i>Oracle version on source and target.....</i>	<i>3</i>
<i>Source and target platforms.....</i>	<i>3</i>
<i>Compatibility criteria</i>	<i>3</i>
<i>Service status of the platform</i>	<i>3</i>
<i>Transportable Technologies</i>	<i>4</i>
<i>About Cross platform transportable tablespaces</i>	<i>4</i>
<i>Benefits of Transportable Tablespace</i>	<i>4</i>
<i>Limitations on Transportable Tablespace</i>	<i>5</i>
How Transportable tablespace Works	5
Cross-Platform Transportable Tablespace Migration procedure.....	6
Cross-platform Tranportable Tablespace Migration Steps.....	6
<i>Determine Endianness in source and target platform.....</i>	<i>6</i>
<i>Select a self-contained set of tablespaces</i>	<i>7</i>
<i>Steps needed before Generate a transportable tablespace set.....</i>	<i>9</i>
<i>Generate a transportable tablespace set.....</i>	<i>9</i>
<i>Transport the tablespace set</i>	<i>11</i>
<i>Import the Tablespace Set</i>	<i>12</i>
<i>Verify the imported tablespace and data in target database.....</i>	<i>12</i>
Conclusion:.....	13
For more information	13

Introduction

Data migration is the process of making an exact copy of an organization's current data from one device to another device—preferably without disrupting or disabling active applications—and then redirecting all input/output (I/O) activity to the new device.

There are a variety of circumstances that might cause an organization to undertake a data migration, including:

- Server or storage technology replacement or upgrade
- Server or storage consolidation
- Relocation of the data center
- Server or storage equipment maintenance, including workload balancing or other performance-related maintenance.

This document explains how to migrate an Oracle database from a big-endian platform like HPUX on PA-RISC to a little-endian platform like Linux on X86_64. This migration is based on Cross-Platform Transportable Tablespaces. Transportable tablespaces allow you to copy an entire tablespace between Oracle databases.

Assumptions made

Oracle version on source and target

- Oracle database used in both the platforms is Oracle 11g.
- Oracle database is installed in both the source and the target platforms.

Source and target platforms

1. Source machine is big-endian platform (HPUX on PA-RISC)
2. Target machine is little-endian platform (Linux on X86-64)

Compatibility criteria

A tablespace can always be transported to a database with the same or higher compatibility setting, whether the target database is on the same or a different platform. The database signals an error if the compatibility level of the transportable set is higher than the compatibility level of the target database.

Service status of the platform

The status of the set of source tablespaces to be transported in Read-only mode..

Transportable Technologies

Transportable technologies provide transportable database and transportable tablespace:

1. **Transportable database** is used to transport an entire database (user data and the Oracle dictionary) to a new platform with the same endian format. Transportable database permits a minimal downtime migration to a new platform by avoiding the time-consuming method of unloading all user data from the source database and loading it into the target database.
2. **Transportable tablespaces** is a feature designed to move a subset of one database into another, even among platforms that differ in endian format. The cross-platform capability of transportable tablespaces can be used to migrate all user data within a database to a new platform with a different endian format. Leveraging transportable tablespaces in this manner permits a minimal downtime migration to a new platform by avoiding the time-consuming method of unloading all user data from the source database and loading it into the target database.

About Cross platform transportable tablespaces

Starting with Oracle Database 11g, you can transport tablespaces across platforms. This functionality can be used to:

1. Allow a database to be migrated from one platform to another
2. Provide an easier and more efficient means for content providers to publish structured data and distribute it to customers running Oracle Database on different platforms
3. Simplify the distribution of data from a data warehouse environment to data marts, which are often running on smaller platforms
4. Enable the sharing of read-only tablespaces between Oracle Database installations on different operating systems or platforms, assuming that your storage system is accessible from those platforms and the platforms all have the same endianness, as described in the sections that follow

Benefits of Transportable Tablespace

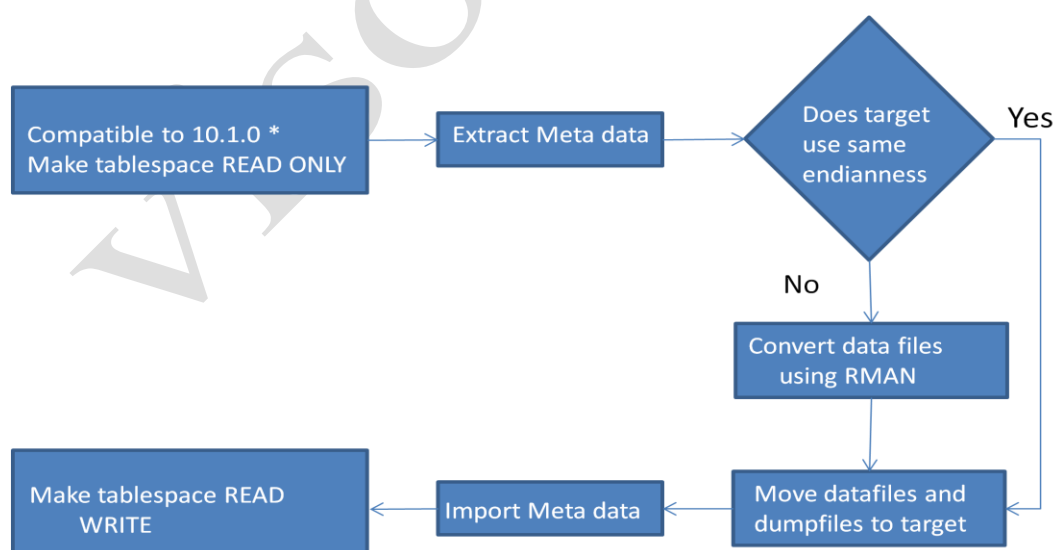
1. Move entire tablespace data
2. Supports media recovery
3. Reduces Server Burden - When transportable tablespaces replace large export/import or other loading, a significant processing burden disappears.
4. Higher Reliability - Fewer processing steps and quicker processing reduce the opportunities for human Errors and system problems

Limitations on Transportable Tablespace

The following are the limitations of using Transportable tablespace.

1. The source and target database must use the same character set and national character set.
2. We cannot transport a tablespace to a target database in which a tablespace with the same name already exists. However, you can rename either the tablespace to be transported or the destination tablespace before the transport operation.
3. Objects with underlying objects (such as materialized views) or contained objects (such as partitioned tables) are not transportable unless all of the underlying or contained objects are in the tablespace set.
4. Tablespaces that do not use block encryption but that contain tables with encrypted columns cannot be transported. You must use Oracle Data Pump to export and import the tablespace's schema objects. You can take advantage of Oracle Data Pump features that enable you to maintain encryption for the data while it is being exported and imported.
5. Beginning with Oracle Database 10g Release 2, we can transport tablespaces that contain XMLTypes. Beginning with Oracle Database 11g Release 1, you must use only Data Pump to export and import the tablespace metadata for tablespaces that contain XMLTypes.
6. We cannot transport the SYSTEM tablespace or objects owned by the user SYS. Some examples of such objects are PL/SQL, Java classes, callouts, views, synonyms, users, privileges, dimensions, directories, and sequences.
7. Transportable tablespaces do not support 8.0-compatible advanced queues with multiple recipients

How Transportable tablespace Works



Cross-Platform Transportable Tablespace Migration procedure

The following steps summarize the process of transporting a tablespace. Details for each step are provided in the subsequent example.

1. For cross-platform transport, check the endian format of both platforms.
2. Select a self-contained set of tablespaces.
3. Generate a transportable tablespace set.
4. A transportable tablespace set (or transportable set) consists of datafiles for the set of tablespaces being transported and an export file containing structural information (metadata) for the set of tablespaces. use Data Pump to perform the export.
5. If you are transporting the tablespace set to a platform with different endianness from the source platform, you must convert the tablespace set to the endianness of the target platform. You can perform a source-side conversion at this step in the procedure, or you can perform a target-side conversion as part of step 4.
6. convert the tablespace set to the endianness of the target platform
7. We are transporting the tablespace set to a platform with different endianness from the source platform, so we must convert the tablespace set to the endianness of the target platform
8. Transport the tablespace set.
9. Copy the datafiles and the export file to a place that is accessible to the target database.
10. Import the tablespace set.
11. Invoke the Data Pump utility to import the metadata for the set of tablespaces into the target database.

Cross-platform Transportable Tablespace Migration Steps

Determine Endianness in source and target platform

In Source Machine

1. Login as oracle user on the source machine
2. Export the environment variables like ORACLE_HOME, ORACLE_SID, PATH appropriately.

```
$ export ORACLE_HOME=/u02/product/11.1.0/db_1  
$ export ORACLE_SID=sample  
$ export PATH=$PATH:$ORACLE_HOME/bin
```
3. Login to oracle as dba user

```
$ sqlplus / as sysdba
```
4. Start the database

```
SQL> startup
```

5. Find the Endian format
 SQL> SELECT d.PLATFORM_NAME, ENDIAN_FORMAT FROM
 V\$TRANSPORTABLE_PLATFORM tp, V\$DATABASE d WHERE tp.PLATFORM_NAME =
 d.PLATFORM_NAME;
 Result:
 PLATFORM_NAME ENDIAN_FORMAT
 ----- -----
 HP-UX (64-bit) Big

In Target Machine

1. Login as oracle user on the target machine.
2. Export the environment variables like ORACLE_HOME, ORACLE_SID, PATH appropriately.

```
$ ORACLE_HOME=/u02/product/11.1.0/db_1
$ ORACLE_SID=SAMPLE
$ PATH=$PATH:$ORACLE_HOME/bin
$ export ORACLE_HOME ORACLE_SID PATH
```
3. Login to oracle as dba user

```
$ sqlplus / as sysdba
```
4. Start the database

```
SQL> startup
```
5. Find the Endian format

```
SQL> SELECT d.PLATFORM_NAME, ENDIAN_FORMAT FROM
V$TRANSPORTABLE_PLATFORM tp, V$DATABASE d WHERE tp.PLATFORM_NAME =
d.PLATFORM_NAME;
Result:
PLATFORM_NAME            ENDIAN_FORMAT
-----            -----
Linux IA (32-bit)            Little
```

Select a self-contained set of tablespaces

There may be logical or physical dependencies between objects in the transportable set and those outside of the set. We can only transport a set of tablespaces that is self-contained.

Here self-contained tablespace means that there are no references from inside the set of tablespaces pointing outside of the tablespaces.

Self contained tablespace violations

Some examples of self contained tablespace violations are listed here,

1. An index inside the set of tablespaces is for a table outside of the set of tablespaces.
2. A partitioned table is partially contained in the set of tablespaces.
3. A referential integrity constraint points to a table across a set boundary
4. A table inside the set of tablespaces contains a LOB column that points to LOBs outside the set of tablespaces.

In source machine

1. Find the list of table spaces in source database

```
SQL> select name from v$tablespace;
```

```
NAME
-----
SYSTEM
SYSaux
UNDOTBS1
USERS
TEMP
SAMPLE
```

Here we cannot transport system, sysaux, undotbs1, and temp, because these tablespace are system tablespace or contain objects owned by the user SYS. So we select USERS and SAMPLE tablespace to transport from source database to target database.

2. Check any self contained violations present in the tablespaces

```
SQL> EXECUTE DBMS_TTS.TRANSPORT_SET_CHECK('USERS,SAMPLE',TRUE);
```

PL/SQL procedure successfully completed.

```
SQL> SELECT * FROM TRANSPORT_SET_VIOLATIONS;
```

no rows selected

Here the set of tablespaces is self-contained, this view is empty. If these query returns any violations then this view return the violations. Then those violations must be resolved before set of tablespaces are transportable.

Steps needed before generating a transportable tablespace set

Sys user or Any EXP_FULL_DATABASE privileged user can perform the following steps.

1. Find the required space needed for storing the dump files and converted data files in the source database.
2. Login to oracle as dba user

```
$ cd $ORACLE_HOME/bin
$ sqlplus / as sysdba
SQL> SELECT TABLESPACE_NAME, FILE_NAME, BYTES FROM DBA_DATA_FILES
where TABLESPACE_NAME in('USERS','SAMPLE');
```

3. Exit from the database

```
SQL> exit
```

4. Based on the size of the data files, check for the availability of free space on the disks using the below command.

```
$ bdf
```

5. Now create a folder 'backup' in the disk that have sufficient space and store the backup file.
6. Note: Here we have the directory '/u02' with sufficient space to store the exported data. Hence we have created a folder 'backup' in /u02.

```
$ mkdir /u02/backup
```

```
$ chmod 700 /u02/backup
```

Generate a transportable tablespace set

1. Login to oracle as dba user

```
$ cd $ORACLE_HOME/bin
$ sqlplus / as sysdba
```

2. Create a directory in the source database

```
SQL> create directory expdp_dir as '/u02/backup';
```

3. Grant permission to that directory

```
SQL> grant read, write on directory expdp_dir to system;
```

4. Make all tablespaces in the set that we are copying read-only.

```
SQL> alter tablespace users read only;
```

```
Tablespace altered.
```

```
SQL> alter tablespace sample read only;
```

Tablespace altered.

5. Invoke the Data Pump export utility on the host system and specify which tablespaces are in the transportable set.

```
SQL> host
```

```
$ ./expdp system/sys DUMPFILE=expdata.dmp DIRECTORY=expdp_dir  
TRANSPORT_TABLESPACES =USERS,SAMPLE LOGFILE=expdata.log
```

Result:

```
Job "SYSTEM"."SYS_EXPORT_TRANSPORTABLE_01" successfully completed
```

6. Collect the required users,roles,grants,profile to create again in the target database

```
$ ./expdp system/sys DUMPFILE=exp_meta_dmp.dmp DIRECTORY=expdp_dir  
full=y INCLUDE=USER,ROLE,ROLE_GRANT,PROFILE content=METADATA_ONLY
```

Result:

```
Job "SYSTEM"."SYS_EXPORT_FULL_01" successfully completed
```

7. exit back to SQL*Plus

```
$ exit
```

```
SQL> exit
```

Here users and sample tablespace are being transported to a different platform, and the endianness of the platforms is different. So we want to convert before transporting the tablespace set, and then convert the datafiles composing the users and sample tablespaces.

8. The RMAN CONVERT command is used to do the conversion. Start RMAN and connect to the target database.

```
$ RMAN TARGET /
```

9. Convert the datafiles into a backup location on the source platform.

```
RMAN> CONVERT TABLESPACE 'USERS','SAMPLE' TO PLATFORM 'Linux IA (32-bit)'  
FORMAT='/u02/backup/%U';
```

Result:

```
channel ORA_DISK_1: datafile conversion complete, elapsed time: 00:00:02
```

```
Finished conversion at source at 02-FEB-11
```

10. Exit from Rman

```
RMAN> exit
```

```
Recovery Manager complete.
```

Transport the tablespace set

Transport both the datafiles and the export file of the tablespaces to a place that is accessible to the target database

Here dump files and converted data files are present in the backup directory (/u02/backup).

1. Login as oracle user on the target machine
2. In order to restore the backup of the database taken, check for the availability of free space on the disks using the below command

```
$ df
```

3. Now create a folder 'backup' in the disk that have sufficient space and store the backup file.
4. Note: Here we have the directory '/u02' with sufficient space to store the exported data. Hence we have created a folder 'backup' in /u02. In order to avoid confusion in restoring the database, use the same directory names as in the source machine.

```
$ mkdir /u02/backup
```

```
$ chmod 700 /u02/backup
```

5. Copy the source backup directory files (~ /u02/backup) to the target directory /u02/backup

Convert the data files in Target machine

1. The RMAN CONVERT command is used to do the conversion. Start RMAN and connect to the target database.

```
$ cd $ORACLE_HOME/bin
```

```
$ RMAN TARGET /
```

```
RMAN> CONVERT DATAFILE '/u02/backup/data_D-RAN_I-2884613691_TS-SAMPLE_FNO-5_0em3mj05' format '/*existing database data file path*/sample01.dbf";
```

Result:

```
channel ORA_DISK_1: datafile conversion complete, elapsed time: 00:00:07
```

```
Finished conversion at target at 04-FEB-11
```

```
RMAN> CONVERT DATAFILE '/u02/ran/data_D-RAN_I-2884613691_TS-USERS_FNO-
```

```
4_ofm3mj0r' format '/* existing database data file path* /users01.dbf';
```

Result:

```
channel ORA_DISK_1: datafile conversion complete, elapsed time: 00:00:03
```

```
Finished conversion at target at 04-FEB-11
```

```
RMAN> exit
```

Recovery Manager complete.

Import the Tablespace Set

1. Import the required users, roles, grants, profiles metadata using the Data Pump Import utility, impdp.

```
$ cd $ORACLEHOME/bin
```

```
$ ./impdp system/sys DIRECTORY=expdp_dir DUMPFILE=exp_meta_dmp.dmp  
full=y
```

Result:

```
Job "SYSTEM"."SYS_IMPORT_FULL_02" successfully completed
```

2. Now Import the tablespace metadata using Data Pump Import utility, impdp.

```
$ ./impdp system/sys DIRECTORY=expdp_dir DUMPFILE=expdata.dmp  
TRANSPORT_DATAFILES='/ existing database data file path/users01.dbf',/  
existing database data file path /sample01.dbf';
```

Result:

```
Job "SYSTEM"."SYS_IMPORT_TRANSPORTABLE_01" successfully completed
```

3. Import the completed tablespaces and check the imported tablespace and data in the target machine.

Verify the imported tablespace and data in target database

1. Login to the database as a dba user

```
$ cd $ORACLE_HOME/bin
```

```
$ sqlplus / as sysdba
```

2. List the tablespaces for verify the imported tablespace

```
SQL> select name from v$tablespace;
NAME
-----
SYSTEM
SYSAUX
UNDOTBS1
TEMP
SAMPLE
```

3. USERS

Here USERS and SAMPLE tablespaces are imported from source machine and issue the queries to check the availability of data in the database.

4. Enable read/write mode for the specific imported tablespaces.

```
SQL> alter tablespace USERS read write;
Tablespace altered.
SQL> alter tablespace SAMPLE read write;
Tablespace altered.
SQL> exit
```

Now imported Tablespaces in target machine is ready to use. At this point the tablespace with data file from the source platform is migrated to the target platform.

Conclusion:

With Oracle 10g and 11g, the cross-platform transportable tablespace feature provides a powerful method to easily and efficiently move data between heterogeneous systems.

This methodology can be used to:

1. Provide an easy and different method for content providers to publish structures data and distribute it to customers running Oracle on different platform.
2. Simplify the distribution of data from a data warehouse environment to data marts, which are often running on lower-end storage and host platforms.
3. Simplify the migration of a database from one host platform to another (through the use of Data Pump or import/export)

For more information

For More information about this document and support, Visit:

<http://www.visolve.com/migration>