White Paper

Cross-Platform Oracle Database Migration Using Transportable Tablespaces and EMC Open Replicator for Symmetrix

Best Practices Planning

Abstract

Starting with Oracle 10g, cross-platform transportable tablespaces have made data movement between heterogeneous systems possible. This white paper discusses best practices and considerations when moving Oracle data between different platforms and illustrates a practical example of moving data from a Solaris host to a Linux host leveraging EMC® Open Replicator for Symmetrix®.

July 2008

Part Number H2510.1

Table of Contents

Executive summary

The advantage of using Oracle transportable tablespaces in migration planning is that instead of extracting the full Oracle data from the database, a process that can take a significant amount of time, only the metadata is extracted while the data itself can be replicated by storage means such as EMC® Open Replicator for Symmetrix®, TimeFinder®, or SRDF®.

The major difference between this solution and a typical EMC business continuance solution is that by using Open Replicator the secondary storage array does not need to be a Symmetrix and can be any qualified third-party storage array on the SAN. In addition, by adding the cross-platform ability to the migration process, the host platform used at the target is different from that of the source. Examples for such use are the ability to migrate from expensive SMP systems to lower-cost server hardware, the ability to move data between OLTP and DSS systems that operate on different platforms, the ability to create and refresh database clones, and more.

Introduction

Oracle 8i first provided a mechanism for migrating tablespace data between database instances called Oracle transportable tablespaces. This functionality provided DBAs with the ability to migrate one or more tablespaces between homogeneous host environments. With release 10g of the database, Oracle introduced cross-platform transportable tablespaces that make data movement between heterogeneous systems possible. This functionality continues to allow tablespace data to be associated with other database instances; however, it now also provides a new mechanism to migrate a tablespace or an entire database from one host platform to another.

Users need to understand that moving Oracle datafiles across heterogeneous operating systems might require some OS-specific details to effectively use this methodology. The section “Special considerations when moving data between server platforms” provides more information.

When using the transportable tablespaces feature, Oracle only requires the tablespaces’ metadata in order to plug them into the destination database. The physical data movement has to be performed by other means outside of Oracle’s scope. Though a simple file transfer could be used with a few very small databases, more efficient and robust replication technology can play an important role. EMC has provided remote replication software in the form of Symmetrix Remote Data Facility (SRDF) since the mid-1990s, which requires both source and target to be Symmetrix arrays. Recently, EMC introduced software called Open Replicator that allows remote replications between a Symmetrix DMX™ array and a variety of disparate storage arrays, including CLARiiON®, IBM Shark, HP/Compaq StorageWorks, or other Symmetrix arrays. EMC Open Replicator provides a fast and reliable mechanism to replicate data across heterogeneous storage arrays that complements EMC SRDF.

This white paper describes the steps needed to configure and implement Oracle’s cross-platform transportable tablespaces and EMC Open Replicator. Transportable tablespaces utilize either the Oracle export utility or the newer Oracle Data Pump to generate a file containing the tablespace metadata. This metadata, in conjunction with a copy of the datafiles, is used to “plug in” the tablespace into a separate database instance.

In addition to reliably replicating data and presenting it to the host operating system, another important consideration for migrating tablespace data between platforms is the “endianness,” or byte ordering, of the data blocks. Different operating systems use different endian types (big endian or little endian). The Oracle RMAN utility is used to convert datafiles with one type of endianness to the other for presentation to a different operating system type. These steps are documented in this paper.

This paper also describes the steps needed to configure and manage the Open Replicator remote replication software running on a Symmetrix DMX. This software is included in EMC Solutions Enabler version 6.0 and later and requires Enginuity™ 5670+ running on the source Symmetrix array. This software is used to manage the remote replication to the target storage array. For this paper, a Symmetrix DMX array was used as the target environment. However, any compatible storage array (for example, CLARiiON, IBM

Shark) could be used. Additionally, other EMC replication software such as SRDF (for remote replication) or TimeFinder (for local replication) could be used instead of Open Replicator.

Audience

The primary target audience of this paper is database administrators, system administrators, storage administrators, and architects who are responsible for architecting, implementing, integrating, and maintaining robust database and storage systems. Readers should be familiar with Oracle 10g and 11g, basic Red Hat Enterprise Linux system administration, and EMC products.

Oracle 10g and 11g cross-platform transportable tablespaces

Oracle's transportable tablespace feature allows DBAs to quickly move tablespace data across Oracle database instances. It is a very efficient method as the data need not be physically imported to the new database. Rather, the metadata alone of the tablespaces is disassociated from one database instance and associated with another, pointing to the appropriate datafiles (that should be copied over by other means outside of Oracle).

Transportable tablespaces can also be used in conjunction with replicated tablespaces to present an exact copy of a tablespace's data to a secondary database instance. The datafiles of the tablespace are replicated using operating system, volume manager, or storage-based replication techniques. These files are then presented to the same or another host and "plugged into" a second instance.

One important limitation in database versions prior to Oracle 10g was the inability to utilize transportable tablespaces in heterogeneous host environments. Transportable tablespaces required the source and target database instances to run on the same host platforms. Although the datafiles were structured by Oracle similarly, the byte ordering, or endianness of the blocks in each file system, prevented transportable tablespaces to be presented to different types of hosts.

Some operating systems, including Windows, Tru64, and Linux, store multibyte binary data with the least significant byte in the lowest memory address; these systems are called *little endian*. Conversely, other operating systems, including Solaris, HP-UX, and AIX, store the most significant byte in the highest memory address, hence the term *big endian*. Figure 1 shows platforms divided by big or little endianness. When a big endian system tries to read data from a little endian one, a conversion process is required—otherwise, the byte order will lead to an incorrect interpretation of the data read. When transporting tablespaces between platforms of the same "endianness," however, no conversion is required.

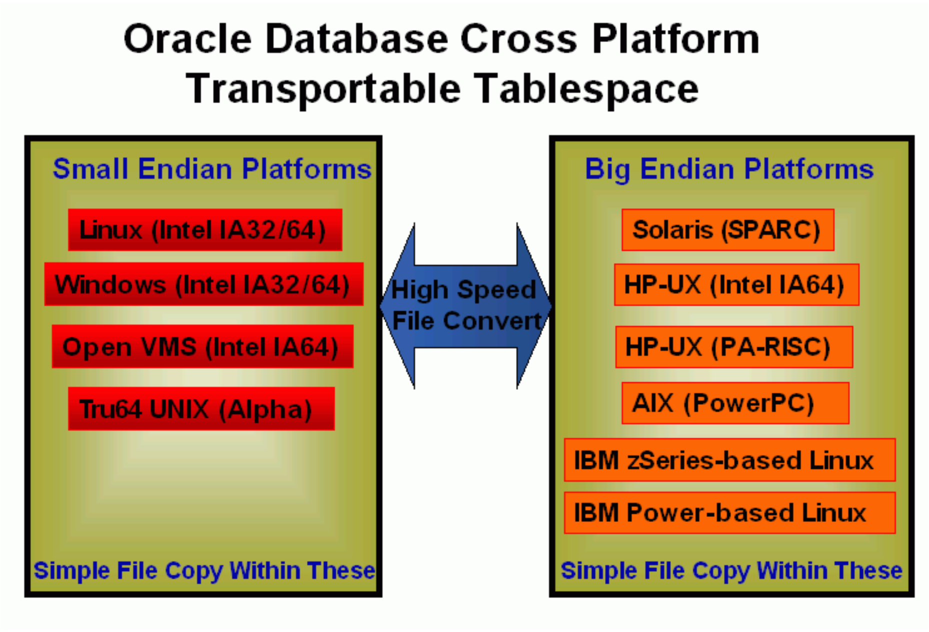

Figure 1. Small and big Endian platforms[1]

This restriction was lifted in Oracle 10g, when Oracle enhanced the RMAN utility to convert Oracle datafiles from one endianness to the other. As a result, Oracle datafiles utilized by a database running on one operating system, for example Solaris, could be transformed and copied over to a target host, say a Linux host, and plugged into an instance running there. This greatly increased the opportunities for using the transportable tablespace functionality.

The steps needed to implement cross-platform transportable tablespaces are relatively simple. The steps include:

1. The tablespace, or set of tablespaces, must be verified to be self-contained. This means that objects in the tablespace set must not have associated objects (for example, indexes, materialized views, partitioned tables) outside of the specified tablespace set. Oracle provides the procedure `TRANSPORT_SET_CHECK` as part of the Oracle provided package `DBMS_TTS`. For example:

   ```
   EXECUTE DBMS_TTS.TRANSPORT_SET_CHECK(‘TBS1’, TRUE);
   ```

   Any violations of the tablespace being self-contained are written to the `TRANSPORT_SET_VIOLATIONS` view, and can be queried using:

   ```
   SELECT *
   FROM TRANSPORT_SET_VIOLATIONS;
   ```

   If no rows are selected, the tablespace is self-contained.

2. Confirm the endian type of database that is used on both the source and target databases. If the source and target platforms are of different endianness, then an additional conversion step must be performed to convert the tablespace being transported to the target format. If they are of the same endianness, then no conversion is necessary and tablespaces can be transported as if they were on the same

[1] Image from the Oracle Technology Network website – “Oracle Database 11g Cross Platform Transportable Tablespace”

platform. The following SQL command can be run on both the source and target databases to determine the platform endianness:

```
SELECT a.platform_name, endian_format
FROM v$transportable_platform b, v$database a
WHERE b.platform_name = a.platform_name;
```

An alternative to this step is to query the V$TRANSPORTABLE_PLATFORM view and identify the type of host used by the source and target databases.

```
SQL> select * from v$transportable_platform order by platform_id;

PLATFORM_ID PLATFORM_NAME                       ENDIAN_FORMAT
----------- ----------------------------------- --------------
          1 Solaris[tm] OE (32-bit)             Big
          2 Solaris[tm] OE (64-bit)             Big
          3 HP-UX (64-bit)                      Big
          4 HP-UX IA (64-bit)                   Big
          5 HP Tru64 UNIX                       Little
          6 AIX-Based Systems (64-bit)          Big
          7 Microsoft Windows IA (32-bit)       Little
          8 Microsoft Windows IA (64-bit)       Little
          9 IBM zSeries Based Linux             Big
         10 Linux IA (32-bit)                   Little
         11 Linux IA (64-bit)                   Little
         12 Microsoft Windows 64-bit for AMD    Little
         13 Linux 64-bit for AMD                Little
         15 HP Open VMS                         Little
         16 Apple Mac OS                        Big
         17 Solaris Operating System (x86)      Little
         18 IBM Power Based Linux               Big
         19 HP IA Open VMS                      Little
         20 Solaris Operating System (AMD64)    Little
```

3. Put the tablespace(s) in the transportable tablespace set into read-only mode. The tablespaces need to be in READ ONLY mode in order to make a clean and valid copy of the datafiles that comprise the transportable tablespace set, as well as to successfully run a transport tablespace export. A tablespace (in this example, TBS1) can be put into read-only mode with the following SQL command:

   ```
   SQL> ALTER TABLESPACE TBS1 READ ONLY;
   ```

4. Metadata of the transportable tablespace set needs to be created and copied to the target environment. Either the Oracle export utility or the new Data Pump facility may be used to create this file. The following shows the commands needed to create the tablespace metadata information using Oracle Data Pump:

   ```
   expdp system/manager dumpfile=or_ufs.dmp directory=dpump_dir
   transport_tablespaces=tbs1 transport_full_check=Y
   ```

   Note**:** If transport tablespace contains XMLTypes, you must use the IMP and EXP utilities, not Data Pump.

5. After putting the tablespace in read-only mode, the datafiles of the transportable tablespace set can be copied and presented to the target host. In the example shown later in this paper, the Symmetrix-based Open Replicator is used to migrate the data from one storage array to another. These new target volumes are then presented to the target host.

6. The endianness of the data may be converted either on the source or the target host. In this example, the conversion process is performed after migrating the data to the target. The Oracle RMAN utility is used to convert the datafile. The following shows an example of the RMAN conversion process:

```
RMAN> CONVERT DATAFILE “/ufs/ora_db1.dbf”
   2> TO PLATFORM=”Linux IA (64-bit)”
   3> FROM PLATFORM=”Solaris[tm] OE (64-bit)”
   4> DB_FILE_NAME_CONVERT=”/ufs”,”/ext3”
   5> PARALLELISM=2;
```

   The datafile is converted to little endian and is written to the new directory location `/ext3` from the `/ufs` directory using the same filename.

7. After converting the file, the tablespace may now be plugged into the target database. The Data Pump utility is used to facilitate the process.

   ```
   impdp system/manager dumpfile=ora_db1.dmp directory=dpump_dir
   transport_datafiles=/ext3/ora_db1.dbf
   ```

Figure 2 summarizes the procedure to transport a tablespace from one platform to another.

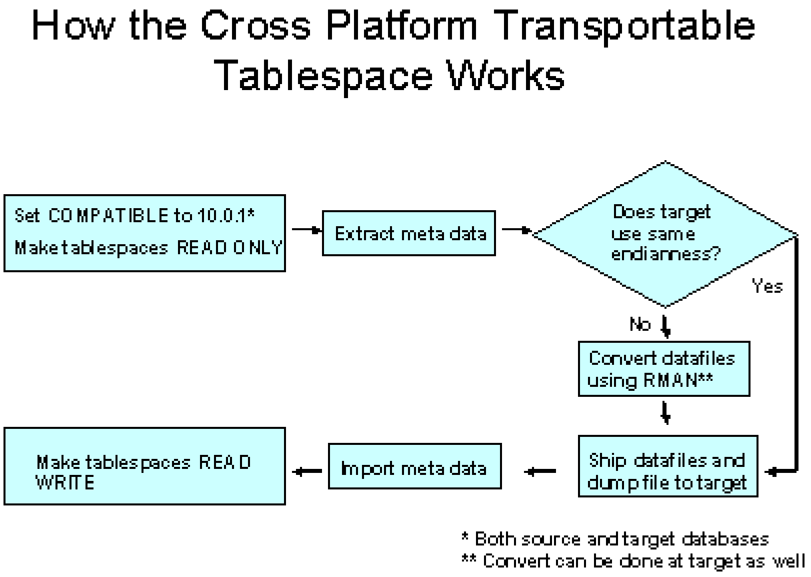
How the Cross Platform Transportable Tablespace Works
Set COMPATIBLE to 10.0.1*
Make tablespaces READ ONLY
Extract meta data
Does target use same endianness?
Yes
No
Convert datafiles using RMAN**
Ship datafiles and dump file to target
Import meta data
Make tablespaces READ WRITE
* Both source and target databases
** Convert can be done at target as well

Figure 2. Cross-platform procedures[2]

Using RMAN to convert endianness

Oracle utilizes RMAN to convert datafiles from big endian to little endian, and vice versa. First, the endianness of the source and target operating systems needs to be identified. If different, then the datafiles must be converted by RMAN. Upon completion, the endianness of the datafiles is transformed to the format needed in the target environment.

The process of converting the datafiles can take place either on the source database host before copying to the new environment, or once it has been transmitted to the target host. Different commands are used to convert the tablespace or datafiles depending on whether conversion is performed on the source side, where

[2] Image from the Oracle Technology Network website – “Oracle Database 11g Cross Platform Transportable Tablespace

the source database understands the underlying datafile structures associated with each tablespace, or the target side, where knowledge of the tablespace is unknown.

The following shows an example of the syntax required to convert two tablespaces, CUST and ORDERS, on the source side to a Microsoft Windows environment (little endian):

```
RMAN> CONVERT TABLESPACE CUST,ORDERS
2> TO PLATFORM ‘Microsoft Windows IA (32-bit)’
3> FROM PLATFORM ‘Solaris[tm] OE (64-bit)’
4> PARALLELISM=4;
```

When conversion of the datafiles is going to take place on the target host, the CONVERT TABLESPACE command does not work, since the target is unaware of the tablespace that will be plugged in. In these situations, the CONVERT DATAFILE command is used to transform the endianness of the datafiles. This syntax is the same as that shown in step 6 on page 8.

Other than utilizing this conversion process using RMAN, the steps for cross-platform transportable tablespaces are the same as those for normal transportable tablespaces.

Special considerations when moving data between server platforms

Users must pay attention to the format of the data on disk and the technology used to move data from one operating system to another. SRDF and Open Replicator use storage technology to replicate data. As such they will replicate the full content of the storage device(s) as an exact image of the source. So, if the source Oracle tablespace is residing within a volume manager disk group or a file system, you are really replicating the image of the storage device together with the volume manager and file system, not just the Oracle DB files. This means that the target host will need to be able to mount that disk group or file system, even if in a read-only mode, for RMAN to be able to do the endian conversion. The requirement for a common denominator if a volume manager or file system is used will most likely determine whether the endian conversion will be performed on the source or target host. For example, this paper shows how to migrate datafiles from a Solaris UFS file system to a Linux EXT3 file system. Since Linux can mount UFS in read-only mode, the RMAN endian conversion was performed on the target Linux host.

If, however, the Oracle tablespace or datafile is stored on the device as raw, then there are no such strong compatibility requirements as there is no disk group, volume group, or target host file system to mount.

If a volume manager is used, for example an ASM disk group, then the Oracle tablespace or database file is scattered as pieces inside the ASM disk group. All the disks that participate in that disk group must be replicated together in order to create an ASM remountable disk group at the target host. In addition, if ASM is used with either the source or destination database, you must use RMAN to transport/convert the tablespace from and/or to ASM.

To summarize, it is key that once replicated, the Oracle datafiles be accessible by the target host. If the replicated storage device is a UFS file system, it must be target host mountable as a UFS file system. If it is a raw partition, the raw partition must be picked up using the proper raw partition semantics on the target host. If it is a logical volume disk group, such as ASM or VxVM, the logical volume manager must be able to pick up the replicated storage object(s) coherently and allow the logical volume to be remounted on the target host.

In the case where there is no common denominator between source and target logical volume manager or file system, EMC offers other tools for migration such as EMC Open Migrator/LM. The EMC Open Migrator/LM Product Guides provide more information.

This paper shows how to migrate datafiles from a Solaris UFS file system to a Linux EXT3 file system to perform cross-platform Oracle database migration using transportable tablespace and EMC Open Replicator. However, the actual Oracle database to be migrated and/or host system of your real environment may vary. You may have to make necessary adjustments, based on the principle of this white paper, to best suit the migration needs of your environment.

EMC Open Replicator for Symmetrix

EMC Open Replicator for Symmetrix is a software product that runs entirely within the Symmetrix DMX array, making use of an existing SAN infrastructure to provide data migration, backup, and ILM solutions. Open Replicator enables movement of data between a DMX and other types of storage arrays on the SAN, including CLARiiON, IBM Shark, HP/Compaq StorageWorks, and other Symmetrix arrays.

Open Replicator has the ability to both “push” data (replicate data from the DMX to a remote array) and to “pull” data (replicate data from the remote array back to the DMX). In both cases, the DMX array acts as the controlling entity for replication operations. Connectivity between the arrays is established through the SAN. Both full and incremental copies of the data may be pushed from the DMX, while pull operations from another array to the DMX are full.

Once Open Replicator has started to replicate data, it can be accessed immediately as if the copy was completed. If data that was not yet copied is requested by the host, it will be immediately read from the other storage array to satisfy the request. For that reason, downtime during migration is minimal.

Control of remote copy operations is handled from a host locally attached to the DMX array. Devices on the DMX control array are known as *control devices.* Devices on the remote array are called *remote devices.* Device pairings between the DMX control array and the remote array are established through a file. An example of a file containing device pairings between the storage arrays is shown below:

```
symdev=000187900754:0061   wwn=6006048000018790067153594D343144
symdev=000187900754:0062   wwn=6006048000018790067153594D343145
```

The devices 0061 and 0062 on Symmetrix DMX array 187900754 are paired up with devices on another array (in this case, another Symmetrix DMX array 187900671) specified by their WWNs on the SAN. Copying between the arrays can either be immediate or deferred:

- To immediately start the copy of a control device, either to or from remote device, the –copy option is used. When data is being pushed, the `–differential` option may be used, which means that any subsequent remote copy operation for the same session will copy only the changed data.
- With a deferred copy of those tracks on a full device that contain the data of interest, copying occurs only when there are writes to or from the DMX control device. The device pair state remains in `CopyOnWrite` (with push) or `CopyOnAccess` (with pull) until the copy session is terminated or all of the tracks have been read/written.

Online operations allow a DMX host application such as a database or file server to be up and running while the DMX is copying data to remote devices during push operations. While the local DMX control array is pushing data and the control devices are online, a consistent point-in-time copy can be sent to the remote array by using ECA (Enginuity Consistent Assist) to temporarily restrict host I/O to the control devices while the remote copy process starts the session. With online operations, only one copy of the data may be replicated to the remote site. Offline operations allow up to 16 copies of the data to be replicated to the remote site.

The following shows the steps need to create and activate an online push operation between a Symmetrix DMX and another array. The device file (devpair.txt) is used for these operations. The first command creates the device pairings between the arrays, specifies an immediate copy operation, defines the control DMX array as the pushing source of the data, identifies the session named as `10g`, and specifies that the data is active on the control devices. The second command initiates the copy process between the arrays using the –consistent flag (that is, using ECA to ensure consistency).

```
symrcopy create –copy –name 10g –push –hot –file devpair.txt
symrcopy activate –file devpair.txt -consistent
```

Mounting cross-platform file systems under Linux

As discussed earlier, an important consideration with migrating transportable tablespaces between disparate operating systems is the ability to read file systems from one host on another. In the example below datafiles on a Solaris UFS file system are migrated to Linux EXT3 file system. The endian conversion is performed on the Linux server as Linux can mount many types of file systems including UFS in read-only mode. Linux's ability to mount a UFS file system requires the appropriate kernel module to be available. It is a prerequisite for the principles in this white paper that the target Linux server has already been built with proper kernel modules to mount UFS file systems. The following command mounts the Solaris UFS file system on the Linux host:

```
mount -t –r ufs /dev/sda2 /mnt/ext3
```

Example: Pushing data online to a remote array using Open Replicator for Oracle transportable tablespace operation

This example shows how a tablespace can be moved from an Oracle database running on a Solaris host to an Oracle database running on a Linux host. The Oracle version on both the source host (Solaris system) and target host (Linux system) is 11.1.0.6.0. To move the Oracle datafile, EMC Open Replicator will be used; to move the tablespace metadata, Oracle Data Pump will be used. The migration was performed using EMC Solution Enabler version 6.5.0. The hardware setup consists of the DMX control array with an array ID of 000187900754 connected to a Sun Solaris host. The remote array on the SAN is another DMX and 10 remote devices are each identified by their LUN WWN. The Oracle database transportable tablespace set was laid out on the corresponding 10 devices on the control array.

Pushing Oracle transportable tablespace data online using Open Replicator

The transportable tablespaces need to be in READ ONLY mode in order to make a clean and valid copy of the datafiles that comprise the transportable tablespace set, as well as to successfully run a transport tablespace export.

```
SQL> alter tablespace TBS1 read only;

Tablespace altered.

SQL> select tablespace_name,status from dba_tablespaces;

TABLESPACE_NAME                STATUS
------------------------------ ---------
SYSTEM                         ONLINE
SYSAUX                         ONLINE
UNDO_1                         ONLINE
WARE_0                         ONLINE
CUST_0                         ONLINE
DIST_0                         ONLINE
HIST_0                         ONLINE
STOK_0                         ONLINE
ITEM_0                         ONLINE
ORDR_0                         ONLINE
NORD_0                         ONLINE
IWARE_0                        ONLINE
ICUST1_0                       ONLINE
```

```
ICUST2_0                        ONLINE
IDIST_0                         ONLINE
ISTOK_0                         ONLINE
IITEM_0                         ONLINE
IORDR2_0                        ONLINE
TEMP_0                          ONLINE
SP_0                            ONLINE
TESTRAW                         ONLINE
OR_RAW                          ONLINE
TBS1                            READ ONLY
OR_VXFS                         ONLINE

24 rows selected.

SQL> exit
```

To set up an Open Replicator session so that datafiles can be moved across platforms quickly, control/remote device pairings in a text file need to be set up. Control or remote devices are either specified by their unique LUN WWNs or by a combination of the array ID and the device names (array ID: device name). Enter the DMX control devices in the left-hand column, and the remote devices in the right-hand column. In this example, a text file named device.txt has been edited using the vi text editor. The first pair entered is device 061 on DMX control array 0754 paired with the remote device whose LUN WWN is `60060480000187900067153594D343144`.

```
<root>licoc025:/download/scripts>>vi devfile.txt

symdev=000187900754:0061 wwn=60060480000187900067153594D343144
symdev=000187900754:0062 wwn=60060480000187900067153594D343145
symdev=000187900754:0063 wwn=60060480000187900067153594D343146
symdev=000187900754:0503 wwn=60060480000187900067153594D323030
symdev=000187900754:0065 wwn=60060480000187900067153594D343231
symdev=000187900754:0066 wwn=60060480000187900067153594D343232
symdev=000187900754:0067 wwn=60060480000187900067153594D343233
symdev=000187900754:0068 wwn=60060480000187900067153594D343234
symdev=000187900754:0069 wwn=60060480000187900067153594D343235
symdev=000187900754:006A wwn=60060480000187900067153594D343236
```

The `symrcopy create` command creates 10 online copy sessions (one for each device in the file) so that data on the control devices can be copied to the remote devices specified in file devfile.txt. The *–push* parameter specifies that the DMX control array is pushing the data. The *-hot* parameter indicates that the source DMX devices remain online during the operation. Remote copy operations allow the use of the *-differential* option the first time a full copy is performed. By doing this, all subsequent copies of data during that copy session are automatically incremental copies, with only new writes to the control device being passed to the remote device. The *–copy* option specifies a full and immediate copy. The *-name* parameter gives these sessions the label name 10g.

```
# symrcopy -file devfile.txt -nop create -v -push -hot -name 10g
-copy -differential
'Create' operation execution is in progress for the device list in
device file 'devfile.txt'. Please wait...


SELECTING Control device - Remote devices:

  (Ctl)Sym: 000187900754 Device: 00061 -   LUN WWN: 60060480000187900067153594D343144
  (Ctl)Sym: 000187900754 Device: 00062 -   LUN WWN: 6006048000018790067153594D343145
  (Ctl)Sym: 000187900754 Device: 00063 -   LUN WWN: 6006048000018790067153594D343146
  (Ctl)Sym: 000187900754 Device: 00503 -   LUN WWN: 6006048000018790067153594D323030
  (Ctl)Sym: 000187900754 Device: 00065 -   LUN WWN: 6006048000018790067153594D343231
  (Ctl)Sym: 000187900754 Device: 00066 -   LUN WWN: 6006048000018790067153594D343232
  (Ctl)Sym: 000187900754 Device: 00067 -   LUN WWN: 6006048000018790067153594D343233
  (Ctl)Sym: 000187900754 Device: 00068 -   LUN WWN: 6006048000018790067153594D343234
  (Ctl)Sym: 000187900754 Device: 00069 -   LUN WWN: 6006048000018790067153594D343235
  (Ctl)Sym: 000187900754 Device: 0006A -   LUN WWN: 6006048000018790067153594D343236

STARTING a REMOTE Copy CREATE (PUSH) (HOT) (DIFFERENTIAL)
STARTING a RCOPY 'CREATE' operation.

...

The Rcopy 'CREATE' operation SUCCEEDED.

'Create' operation successfully executed for the device list
in device file 'devfile.txt'.
```

The `symrcopy activate` command activates the copy session for the pairings in the file devfile.txt. Replication from the control devices to the remote ones is initiated.

```
<root>licoc025:/download/scripts >>./activate.sh
+ symrcopy -file devfile.txt -nop activate -consistent -v

'Activate' operation execution is in progress for the device list in
device file 'devfile.txt'. Please wait...

  (Ctl)Sym: 000187900754 Device: 00061 -   LUN WWN: 6006048000018790067153594D343144 -
[SELECTED]
  (Ctl)Sym: 000187900754 Device: 00062 -   LUN WWN: 6006048000018790067153594D343145 -
[SELECTED]
  (Ctl)Sym: 000187900754 Device: 00063 -   LUN WWN: 6006048000018790067153594D343146 -
[SELECTED]
  (Ctl)Sym: 000187900754 Device: 00503 -   LUN WWN: 6006048000018790067153594D323030 -
[SELECTED]
  (Ctl)Sym: 000187900754 Device: 00065 -   LUN WWN: 6006048000018790067153594D343231 -
[SELECTED]
  (Ctl)Sym: 000187900754 Device: 00066 -   LUN WWN: 6006048000018790067153594D343232 -
[SELECTED]
  (Ctl)Sym: 000187900754 Device: 00067 -   LUN WWN: 6006048000018790067153594D343233 -
[SELECTED]
  (Ctl)Sym: 000187900754 Device: 00068 -   LUN WWN: 6006048000018790067153594D343234 -
[SELECTED]
  (Ctl)Sym: 000187900754 Device: 00069 -   LUN WWN: 6006048000018790067153594D343235 -
[SELECTED]
  (Ctl)Sym: 000187900754 Device: 0006A -   LUN WWN: 6006048000018790067153594D343236 -
[SELECTED]

SELECTING Control device - Remote devices:

...
STARTING a REMOTE Copy ACTIVATE CONSISTENT
STARTING a RCOPY 'ACTIVATE' operation.


The Rcopy 'ACTIVATE' operation SUCCEEDED.
'Activate' operation successfully executed for the device list
in device file 'devfile.txt'.
```

The symrcopy query command indicates that the device pairs defined in the devfile.txt file are in the CopyInProg state and that copying is 1 percent complete.

```
<root>licoc025:/download/scripts >>./query_30.sh
+ symrcopy -file devfile.txt query -i 30

Device File Name      : devfile.txt

       Control Device                 Remote Device             Flags     Status      Done
---------------------------- ----------------------------------- ----- -------------- ----
                   Protected
SID:symdev         Tracks    Identification                   RI CDSH   CTL <=> REM     (%)
------------------ --------- -------------------------------- -- ----- -------------- ----
000187900754:0061     135518 000187900671:041D                SD XXXX  CopyInProg        1
000187900754:0062     135461 000187900671:041E                SD XXXX  CopyInProg        1
000187900754:0063     135501 000187900671:041F                SD XXXX  CopyInProg        1
000187900754:0503     542303 000187900671:0200                SD XXXX  CopyInProg        1
000187900754:0065     135494 000187900671:0421                SD XXXX  CopyInProg        1
000187900754:0066     135466 000187900671:0422                SD XXXX  CopyInProg        1
000187900754:0067     135577 000187900671:0423                SD XXXX  CopyInProg        1
000187900754:0068     135696 000187900671:0424                SD XXXX  CopyInProg        1
000187900754:0069     135723 000187900671:0425                SD XXXX  CopyInProg        1
000187900754:006A     135616 000187900671:0426                SD XXXX  CopyInProg        1

Total              ---------
  Track(s)           1762355
  MB(s)              55073.6

Legend:
R:  (Remote Device Vendor Identification)
  S = Symmetrix, C = Clariion, . = Unknown.

I:  (Remote Device Specification Identifier)
  D = Device Name, W = LUN WWN, World Wide Name.

Flags:
(C): X = The background copy setting is active for this pair.
     . = The background copy setting is not active for this pair.
(D): X = The session is a differential copy session.
     . = The session is not a differential copy session.
(S): X = The session is pushing data to the remote device(s).
     . = The session is pulling data from the remote device(s).
(H): X = The session is a hot copy session.
     . = The session is a cold copy session.
```

As soon as Open Replicator started the copy, the next step of exporting the tablespace metadata can take place. Since only the tablespace definitions are exported, the export job runs very quickly.

Export of the tablespace metadata using Data Pump

First, create the Oracle directory used by Oracle Data Pump to export tablespace metadata (this step can happen before the migration starts).

```
SQL> CREATE OR REPLACE DIRECTORY dpump_dir AS '/tmp/dump' ;
SQL> GRANT READ,WRITE ON DIRECTORY dpump_dir TO system;
```

Export transportable tablespace metadata using Data Pump.

```
<oracle>licoc025:/download/scripts >>more export.sh
set -x
DumpFile=$1
Tablespace=$2
expdp system/manager dumpfile=$DumpFile directory=dpump_dir
transport_tablespaces=$Tablespace

<oracle>licoc025:/download/scripts >>./export.sh or_ufs TBS1
DumpFile=or_ufs
```

```
Tablespace=TBS1
+ expdp system/manager dumpfile=or_ufs directory=dpump_dir
transport_tablespaces=TBS1

Export: Release 11.1.0.6.0 - 64bit Production on Tuesday, 15 April,
2008 11:07:37

Copyright (c) 2003, 2007, Oracle.  All rights reserved.

Connected to: Oracle Database 11g Enterprise Edition Release 11.1.0.6.0
- 64bit Production
With the Partitioning, Real Application Clusters, Data Mining and Real
Application Testing options
Starting "SYSTEM"."SYS_EXPORT_TRANSPORTABLE_01":  system/********
dumpfile=or_ufs directory=dpump_dir transport_tablespaces=TBS1
Processing object type TRANSPORTABLE_EXPORT/PLUGTS_BLK
Processing object type TRANSPORTABLE_EXPORT/TABLE
Processing object type TRANSPORTABLE_EXPORT/INDEX
Processing object type TRANSPORTABLE_EXPORT/CONSTRAINT/CONSTRAINT
Processing object type TRANSPORTABLE_EXPORT/INDEX_STATISTICS
Processing object type TRANSPORTABLE_EXPORT/TABLE_STATISTICS
Processing object type TRANSPORTABLE_EXPORT/TTE_POSTINST/PLUGTS_BLK
Master table "SYSTEM"."SYS_EXPORT_TRANSPORTABLE_01" successfully
loaded/unloaded
***********************************************************************
*******
Dump file set for SYSTEM.SYS_EXPORT_TRANSPORTABLE_01 is:
  /tmp/dump/or_ufs.dmp
Job "SYSTEM"."SYS_EXPORT_TRANSPORTABLE_01" successfully completed at
11:10
```

The metadata can now be transferred to the target system via FTP, for example.

Mounting cross-platform file systems

As stated in the “Introduction” section, it is a prerequisite for cross-platform transportable tablespace that the target host, which is the Linux server in this example, has already been built with proper kernel modules to mount UFS file systems. To mount the Solaris UFS file system on the Linux host once the Open Replicator session has finished copying the data, do the following:

```
<root>licoc045:/download/scripts >> mount -t –r ufs /dev/sda2 /mnt/ext3
```

For illustration purposes only, the Oracle utility `dbverify` is used to check the datafile. In the following example the check will fail as Solaris and Linux use different endianness (big versus little). A file system conversion from UFS to Linux EXT3 is thus required.

```
[ora11@licoc045 oracle]$ dbv file=/ufs/db1.dbf blocksize=8192 start=1
end=1000

DBVERIFY: Release 11.1.0.6.0 - Production on Tue Apr 15 13:47:31 2008

Copyright (c) 1982, 2007, Oracle.  All rights reserved.

DBVERIFY - Verification starting : FILE = /ufs/db1.dbf
Page 1 is marked corrupt
Corrupt block relative dba: 0x00000001 (file 0, block 1)
Bad header found during dbv:
```

```
Data in bad block:
 type: 11 format: 2 rdba: 0x01008005
 last change scn: 0x0000.00000000 seq: 0x1 flg: 0x04
 spare1: 0x0 spare2: 0x0 spare3: 0x0
 consistency value in tail: 0x010b0000
 check value in block header: 0xbf4b
 computed block checksum: 0x0

………………………..


DBVERIFY - Verification complete

Total Pages Examined         : 1000
Total Pages Processed (Data) : 0
Total Pages Failing   (Data) : 0
Total Pages Processed (Index): 0
Total Pages Failing   (Index): 0
Total Pages Processed (Other): 0
Total Pages Processed (Seg)  : 0
Total Pages Failing   (Seg)  : 0
Total Pages Empty            : 0
Total Pages Marked Corrupt   : 1000
Total Pages Influx           : 0
Highest block SCN            : 0 (0.0)
```

RMAN conversion of cross-platform datafiles

RMAN is used to complete the conversion of datafiles from UFS to EXT3.

```
[ora11@licoc045 scripts]$ ./convert.sh convUFS.crv ufs.log
++ export ORACLE_SID=ilm2
++ ORACLE_SID=ilm2
++ export ORACLE_HOME=/oracle
++ ORACLE_HOME=/oracle
++ export ConvFile=convUFS.crv
++ ConvFile=convUFS.crv
++ export LogFile=ufs.log
++ LogFile=ufs.log
++ rman target / cmdfile convUFS.crv log ufs.log
RMAN> 2> 3> 4> 5> 6>
[ora11@licoc045 scripts]$
[ora11@licoc045 scripts]$ more ufs.log

Recovery Manager: Release 11.1.0.6.0 - on Tue Apr 15 14:42:57 2008

Copyright (c) 1982, 2007, Oracle.  All rights reserved.

connected to target database: ware (DBID=3836358240)

RMAN> convert datafile '/ufs/db1.dbf'
2> to platform="Linux IA (64-bit)"
3> from platform="Solaris[tm] OE (64-bit)"
4> db_file_name_convert="/ufs","/ext3"
5> parallelism=2;
```

```
Starting backup at 15-APR-08
using target database controlfile instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=29 devtype=DISK
allocated channel: ORA_DISK_2
channel ORA_DISK_2: sid=28 devtype=DISK
channel ORA_DISK_1: starting datafile conversion
input filename=/ufs/db1.dbf
converted datafile=/ext3/db1.dbf
channel ORA_DISK_1: datafile conversion complete, elapsed time:
00:06:17
Finished backup at 15-APR-08

Recovery Manager complete.
```

Once RMAN conversion is completed, for illustration purposes a subsequent *dbverify* will run smoothly.

```
[ora11@licoc045 scripts]$ dbv file=/ext3/db1.dbf blocksize=8192 start=1
end=1000

DBVERIFY: Release 11.1.0.6.0 - Production on Fri May 09 11:47:31 2008

Copyright (c) 1982, 2007, Oracle.  All rights reserved.

DBVERIFY - Verification starting : FILE = /ext3/db1.dbf

DBVERIFY - Verification complete

Total Pages Examined         : 1000
Total Pages Processed (Data) : 991
Total Pages Failing   (Data) : 0
Total Pages Processed (Index): 0
Total Pages Failing   (Index): 0
Total Pages Processed (Other): 9
Total Pages Processed (Seg)  : 0
Total Pages Failing   (Seg)  : 0
Total Pages Empty            : 0
Total Pages Marked Corrupt   : 0
Total Pages Influx           : 0
Highest block SCN            : 3731238 (0.3731238)
```

The next step is to plug the tablespace into the target instance by importing back the tablespace metadata definition. Again Oracle Data Pump is used.

```
SQL> select tablespace_name from dba_tablespaces;

TABLESPACE_NAME
------------------------------
SYSTEM
SYS_UNDOTS
SYSAUX
RMAN
TEMP

5 rows selected.
```

Import of the tablespace metadata using Data Pump

For importing transportable tablespace metadata using Data Pump, you need to create the Oracle directory the same way as for the export process (this step can take place before the migration operations start).

SQL> CREATE OR REPLACE DIRECTORY dpump_dir AS '/tmp/dump' ;
SQL> GRANT READ,WRITE ON DIRECTORY dpump_dir TO system;

Import transportable tablespace metadata using Data Pump.

```
[ora11@licoc045 scripts]$ more ./import.sh
set -x
export DumpFile=$1
export SourceFile=$2
impdp system/manager dumpfile=$DumpFile directory=dpump_dir
transport_datafiles='$SourceFile' remap_schema=(tpcc:rman)

[oracle@l82aq057 scripts]$ ./import.sh or_ufs.dmp /ext3/db1.dbf
Import: Release 11.1.0.6.0 - 64bit Production on Friday, 09 May, 2008
13:29:23

Copyright (c) 2003, 2007, Oracle.  All rights reserved.
Connected to: Oracle Database 11g Enterprise Edition Release 11.1.0.6.0
- 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing
options
Master table "SYSTEM"."SYS_IMPORT_TRANSPORTABLE_01" successfully
loaded/unloaded
 remap_schema=(tpcc:rman)sport_datafiles=/ext3/db1.dbf
dumpfile=or_ufs.dmp
Processing object type TRANSPORTABLE_EXPORT/PLUGTS_BLK
Processing object type TRANSPORTABLE_EXPORT/TABLE
Processing object type TRANSPORTABLE_EXPORT/INDEX
Processing object type TRANSPORTABLE_EXPORT/CONSTRAINT/CONSTRAINT
Processing object type TRANSPORTABLE_EXPORT/INDEX_STATISTICS
Processing object type TRANSPORTABLE_EXPORT/TABLE_STATISTICS
Processing object type TRANSPORTABLE_EXPORT/TTE_POSTINST/PLUGTS_BLK
Job "SYSTEM"."SYS_IMPORT_TRANSPORTABLE_01" successfully completed at
13:51:09
```

To confirm the tablespace set has been transported into the target database:

```
[ora11@licoc045 scripts]$ sqlplus system/manager

SQL*Plus: Release 11.1.0.6.0 - on Fri May 09 13:51:18 2008

Copyright (c) 1982, 2007, Oracle.  All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0
With the Partitioning, Data Mining and Real Application Testing options

SQL> select tablespace_name from dba_tablespaces;

TABLESPACE_NAME
------------------------------
SYSTEM
```

```
SYS_UNDOTS
SYSAUX
RMAN
TEMP
TBS1

6 rows selected.
```

Finally, put the transported tablespace set in read/write mode in the target database for data access.

```
SQL> ALTER TABLESPACE TBS1 READ WRITE;

SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition Release
11.1.0.6.0
With the Partitioning, Data Mining and Real Application Testing options
```

Conclusion

With Oracle 10g and 11g, the cross-platform transportable tablespace feature combined with EMC's proven storage technologies provide a powerful method to easily and efficiently move data between heterogeneous systems.

This functionality can be used to:

- Provide an easy and efficient method for content providers to publish structured data and distribute it to customers running Oracle on different platforms.
- Simplify the distribution of data from a data warehouse environment to data marts, which are often running on lower-end storage and host platforms.
- Simplify the migration of a database from one host platform to another (through the use of Data Pump or import/export).
- Implement Information Lifecycle Management (ILM) solutions, where data is made available on tiered levels of storage and host resources as the business value of the data changes in time.

References

The following EMC and Oracle sources provide information related to the concepts discussed in this white paper:

- *EMC Solutions Enabler Symmetrix Open Replicator CLI Product Guide*
- Oracle documentation library at http://www.oracle.com