Technical Report

Cross-Platform Database Migration Using Oracle Transportable Tablespaces and NetApp FlexClone

Kannan Mani and Matt DeMarco, NetApp

September 2009 | TR-3791

ABSTRACT

This document explains the process of cross-platform database migration using Oracle® transportable tablespaces, and it also provides the steps to clone Oracle Databases using NetApp® FlexClone®.

TABLE OF CONTENTS

1 INTRODUCTION

The movement of data from OLTP databases to data warehouses has been one of the most time- and space-consuming processes for Oracle Database administrators. Oracle offers many tools to address these issues such as import/export, data pump, streams, change data capture, and transportable tablespaces.

With transportable tablespaces, one requirement prior to Oracle Database 10*g* was that the database hosts share the same platform due to endianness restrictions. The endianness is the byte ordering of blocks in a file system, which differs between host operating systems. For example, platforms such as Linux®, Windows®, OpenVMS, and Tru64 are small endian platforms, and Solaris™, HP-UX (Intel® IA64 and PA-RISC), AIX, IBM zSeries-based Linux, and IBM Power-based Linux are big endian platforms.

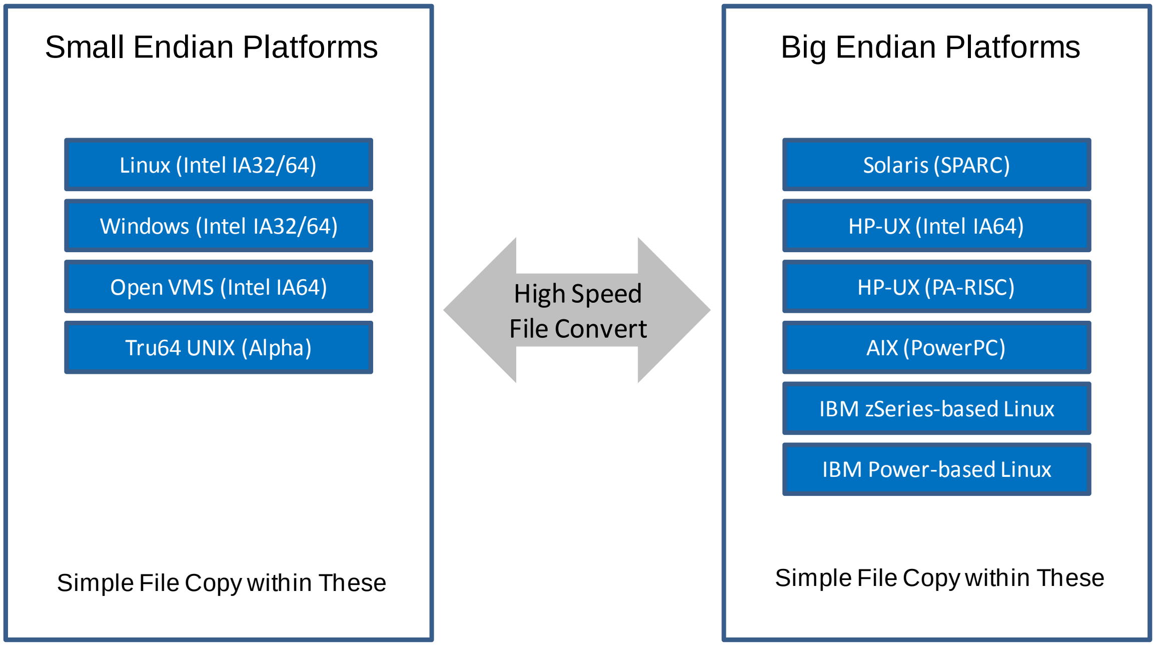


Figure 1) Supported platforms for Oracle Database cross-platform transportable tablespaces.

This restriction is now lifted with the introduction of the new Oracle Database 11*g* feature cross-platform transportable tablespaces. With this new feature, it is now possible to move Oracle data files between platforms with different endianness.

With the removal of this restriction, Oracle DBAs can now move data between platforms in a rapid manner. The movement of the tablespaces still relies on classic data movement techniques such as FTP, rsync, or copy. Customers using NetApp® FAS or V-Series can leverage Data ONTAP® FlexClone technology to quickly clone files that make up a tablespace without consuming additional space.

NetApp FlexClone software enables true cloning: instant replication of data files, LUNs, and volumes without requiring additional storage space at the time of creation. Each cloned file, LUN, or volume is a transparent, virtual copy of the data that can be used for essential enterprise operations such as testing and bug fixing, platform and upgrade checks, multiple simulations against large data sets, remote office testing and staging, and provisioning of server and desktop images. NetApp FlexClone provides substantial storage space savings, which means you can manage many more data set variations in less time and with less risk at a lower cost to your organization.

NetApp FlexClone allows you to create data set replicas of entire Data ONTAP volumes as well as replicas at the individual file or LUN level. For volume-level cloning, a FlexClone replica is a writable Snapshot™ copy with all of the capabilities of a NetApp FlexVol® volume. It is a truly virtual data container that can be provisioned, sized, and resized dynamically to simplify operations and increase the responsiveness of your organization. For those applications in which more granularities is needed, file- and LUN-level cloning leverages NetApp block sharing technology, resulting in maximum storage utilization since incremental capacity is needed only for clone-specific metadata and non-redundant data blocks.

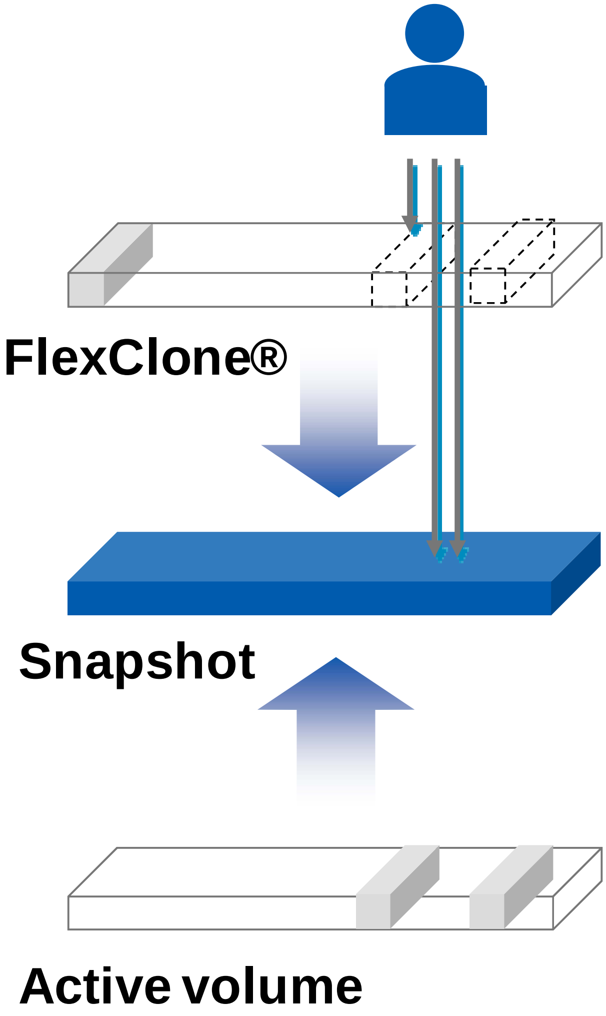
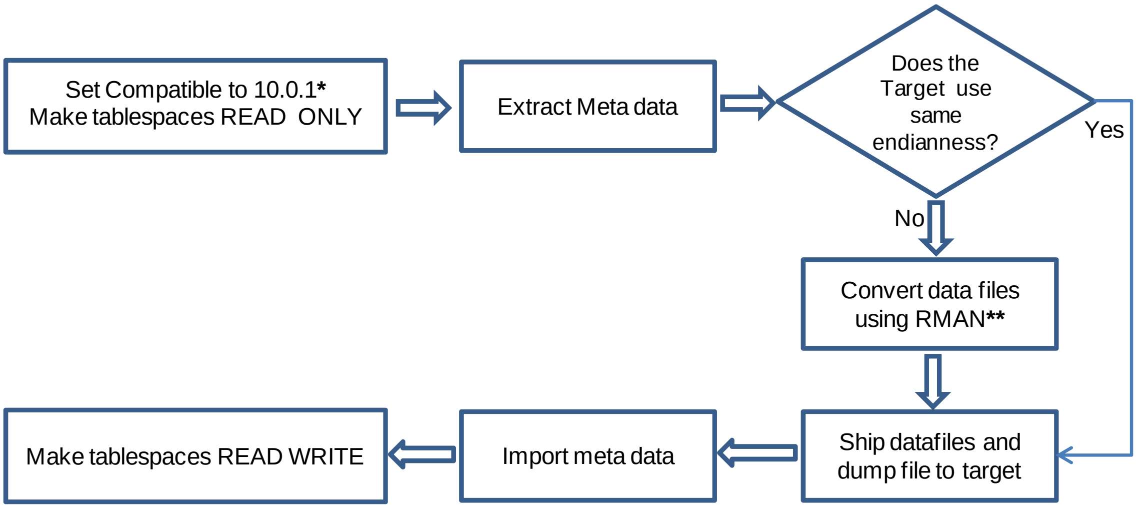


Figure 2) NetApp FlexClone.

By using the technologies available to NetApp customers, we can transport Oracle tablespaces and eliminate the “Ship data files to target” step in the workflow shown in Figure 3.


Set Compatible to 10.0.1*
Make tablespaces READ ONLY
Extract Meta data
Does the Target use same endianness?
Yes
No
Convert data files 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 3) Oracle cross-platform transportable tablespaces workflow.

2 CROSS-PLATFORM TRANSPORTABLE TABLESPACES BASED ON FLEXCLONE

The steps involved with performing this solution based on FlexClone are very similar to standard transportable tablespace movement with the exception that you will be creating a FlexClone copy rather than moving data from one storage system to the other.

2.1 CHECK THE ENDIANNESS OF THE SOURCE AND TARGET PLATFORMS

For cross-platform transport, check the endian format of both platforms by querying the V$TRANSPORTABLE_PLATFORM view.

The following SQL statement can be used to check the endianness of both the source and target:

```
SELECT d.PLATFORM_NAME, ENDIAN_FORMAT FROM
V$TRANSPORTABLE_PLATFORM tp, V$DATABASE d WHERE tp.PLATFORM_NAME
= d.PLATFORM_NAME;
```

The output of the query on the source platform should look like the following:

```
PLATFORM_NAME                                                      ENDIAN_FORMAT

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

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

Linux IA (32-bit)                                                       Little
```

The output of the query on the target platform should look like the following:

```
PLATFORM_NAME                                                      ENDIAN_FORMAT

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

Solaris[tm] OE (32-bit)   64-bit)                                       Big
```

2.2 PICK A SELF-CONTAINED SET OF TABLESPACES

There might be logical or physical dependencies between objects in the transportable set and those outside of the set. You can only transport a set of tablespaces that is self-contained. In this context "self-contained" means that there are no references from inside the set of tablespaces pointing outside of the tablespaces.

In order to make sure that the tablespaces are self-contained, we have to execute the following PL/SQL package on our source tablespace:

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

After invoking this PL/SQL package, you can see all violations by selecting from the TRANSPORT_SET_VIOLATIONS view. If the set of tablespaces is self-contained, this view is empty.

You can check for violations by querying the view as follows:

```
SELECT * FROM TRANSPORT_SET_VIOLATIONS;

no rows selected.
```

2.3 GENERATE TRANSPORTABLE TABLESPACE METADATA

Any privileged user can perform this step. However, you must have been assigned the EXP_FULL_DATABASE role to perform a transportable tablespace export operation.

After making sure that you have a self-contained set of tablespaces that you want to transport, generate a transportable tablespace set by performing the following actions:

```
SYS@thor>alter tablespace sales_data read only;

Tablespace altered.
```

Once the source tablespace is read-only, invoke data pump or export to create the metadata for the tablespace.

The data pump utility is used to export only data dictionary structural information (metadata) for the tablespaces. No actual data is unloaded, so this operation goes relatively quickly even for large tablespace sets.

Enter the following command to export the metadata:

```
EXDP system/oracle DUMPFILE=endian.dmp DIRECTORY=etl_stage
TRANSPORT_TABLESPACES = sales_data

Export: Release 11.1.0.6.0 - Production on Thursday, 2 July, 2009 11:15:53

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

Connected to: Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - Production
With the Partitioning, Oracle Label Security, OLAP, Data Mining
and Real Application Testing options
Starting "SYSTEM"."SYS_EXPORT_TRANSPORTABLE_01":  system/********
dumpfile=endian.dmp directory=etl_stage transport_tablespaces = sales_data
Processing object type TRANSPORTABLE_EXPORT/PLUGTS_BLK
Processing object type TRANSPORTABLE_EXPORT/TABLE
Processing object type TRANSPORTABLE_EXPORT/POST_INSTANCE/PLUGTS_BLK
Master table "SYSTEM"."SYS_EXPORT_TRANSPORTABLE_01" successfully loaded/unloaded
******************************************************************************
Dump file set for SYSTEM.SYS_EXPORT_TRANSPORTABLE_01 is:
  /oracle/etl_stage/endian.dmp
******************************************************************************
Datafiles required for transportable tablespace SALES_DATA:
  /oracle/etl_stage/sales_data.dbf
Job "SYSTEM"."SYS_EXPORT_TRANSPORTABLE_01" successfully completed at 11:16:56
```

2.4 CREATE A FILE-LEVEL FLEXCLONE COPY OF THE TABLESPACES' DATA FILE

Once the transportable tablespace set is created, we can leverage the unique Data ONTAP feature of creating a FlexClone copy of the data file for the tablespace. This allows us to create a clone of the data file in seconds.

In order to leverage this feature, issue the following command for the clone on the NetApp controller:

```
CLONE START /vol/etl_volume/sales_data.dbf /vol/etl_volume/CLONE_sales_data.dbf
```

Once the operation completes, you can see the newly cloned data file on the source and target by issuing the following command:

On the source:

```
[oracle@orcl_demo1 etl_stage]$ ls -hl

total 201M

-rw-r-----  1 oracle oinstall 101M Jul  2  2009 CLONE_sales_data.dbf

-rwxr-xr-x  1    600      600  265 Jun 30 19:30 convert.sh

-rw-r-----  1 oracle oinstall  84K Jul  2  2009 endian.dmp

-rw-r--r--  1 oracle oinstall 1.2K Jul  2  2009 export.log

-rw-r-----  1 oracle oinstall 101M Jul  2  2009 sales_data.dbf

drwxr-xr-x  2    600      600 4.0K Jul  2  2009 solaris
```

On the target:

```
[oracle@atl4452][REPO][/oracle/etl_stage]$ ls -hl

total 410664

-rw-r-----   1 500      503         100M Jul  2  2009 CLONE_sales_data.dbf

-rwxr-xr-x   1 oracle   dba          265 Jun 30 19:30 convert.sh

-rw-r-----   1 500      503          84K Jul  2  2009 endian.dmp

-rw-r--r--   1 500      503         1.1K Jul  2  2009 export.log

-rw-r-----   1 500      503         100M Jul  2 10:26 sales_data.dbf

drwxr-xr-x   2 oracle   dba         4.0K Jul  2 10:22 solaris
```

Once the clone has been created successfully, we can now put the tablespace back into read/write access as we will be using the FlexClone copy for the transportable tablespace conversion.

Enter the following command to enable read/write access on the tablespace:

```
SYS@thor>alter tablespace sales_data read write;

Tablespace altered.
```

2.5 CONVERT THE DATA FILES WITH RMAN

We will leverage both RMAN and the NetApp FlexClone copy for the endianness conversion.

RMAN runs the CONVERT procedure to change the endianness of the data file for the target platform. This step will consume space as RMAN must output the new data file.

Run the following command on the target:

```
RMAN> connect target

connected to target database: REPO (DBID=2814474854)

RMAN> convert datafile
'/oracle/etl_stage/CLONE_sales_data.dbf'
db_file_name_convert
'/oracle/etl_stage', '/oracle/etl_stage/solaris'
from platform 'Linux IA (32-bit)'
 parallelism = 10 ;

Starting conversion at target at 02-JUL-09
using channel ORA_DISK_1
allocated channel: ORA_DISK_2
channel ORA_DISK_2: SID=134 device type=DISK
allocated channel: ORA_DISK_3
channel ORA_DISK_3: SID=138 device type=DISK
allocated channel: ORA_DISK_4
channel ORA_DISK_4: SID=140 device type=DISK
allocated channel: ORA_DISK_5
channel ORA_DISK_5: SID=135 device type=DISK
allocated channel: ORA_DISK_6
channel ORA_DISK_6: SID=133 device type=DISK
allocated channel: ORA_DISK_7
channel ORA_DISK_7: SID=132 device type=DISK
allocated channel: ORA_DISK_8
channel ORA_DISK_8: SID=139 device type=DISK
allocated channel: ORA_DISK_9
channel ORA_DISK_9: SID=131 device type=DISK
allocated channel: ORA_DISK_10
channel ORA_DISK_10: SID=130 device type=DISK
channel ORA_DISK_1: starting datafile conversion
input file name=/oracle/etl_stage/CLONE_sales_data.dbf
converted datafile=/oracle/etl_stage/solaris/CLONE_sales_data.dbf
channel ORA_DISK_1: datafile conversion complete, elapsed time: 00:01:46
Finished conversion at target 02-JUL-09
```

2.6 IMPORT THE CONVERTED DATA FILE

Once the conversion process is complete, we are now free to import the data file and associated tablespace into the target database.

At the target, it is a best practice to create a directory object that can read the data pump dump file and data file.

```
SQL> create directory "SOLARIS_STAGE" as '/oracle/etl_stage/solaris';
SQL> grant read, write on directory "SOLARIS_STAGE" to public;
```

Move or copy the data pump dump file into the /oracle/etl_stage/solaris directory. Enter the following command to copy the file:

```
mv endian.dmp /oracle/etl_stage/solaris/
```

Once the setup on the target is complete, issue the DataPump command to import into the target. On our target database, we have the same user who owned the objects on the source database, so remapping schema objects is unnecessary. Should you have different source and target schemas, use data pump’s remapping feature.

Enter the following command to import the files into the target database:

```
bash-3.00$ impdp system/oracle dumpfile=endian.dmp directory=SOLARIS_STAGE
transport_datafiles=/oracle/etl_stage/solaris/CLONE_sales_data.dbf

Import: Release 11.1.0.6.0 - 64bit Production on Thursday, 02 July, 2009 11:20:07

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
Starting "SYSTEM"."SYS_IMPORT_TRANSPORTABLE_01":  system/********
dumpfile=endian.dmp directory=SOLARIS_STAGE
transport_datafiles=/oracle/etl_stage/solaris/CLONE_sales_data.dbf
Processing object type TRANSPORTABLE_EXPORT/PLUGTS_BLK
Processing object type TRANSPORTABLE_EXPORT/TABLE
Processing object type TRANSPORTABLE_EXPORT/POST_INSTANCE/PLUGTS_BLK
Job "SYSTEM"."SYS_IMPORT_TRANSPORTABLE_01" successfully completed at 11:21:13
```

With the successful import of the data files, we must now put the tablespace back into read/write mode.

Enter the following commands select all of the tablespaces and grant them all read/write access:

```
bash-3.00$ sqlplus /  as sysdba

SQL*Plus: Release 11.1.0.6.0 - Production on Thu Jul 2 11:24:54 2009

Copyright (c) 1982, 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

SQL> select tablespace_name, status from dba_tablespaces;

TABLESPACE_NAME              STATUS
------------------------------ ---------
SYSTEM                       ONLINE
SYSAUX                       ONLINE
UNDOTBS1                     ONLINE
TEMP                         ONLINE
```

```
USERS                          ONLINE
SALES_DATA                     READ ONLY

6 rows selected.

SQL> alter tablespace sales_data read write;

Tablespace altered.

SQL> select tablespace_name, status from dba_tablespaces;

TABLESPACE_NAME                STATUS
------------------------------ ---------
SYSTEM                         ONLINE
SYSAUX                         ONLINE
UNDOTBS1                       ONLINE
TEMP                           ONLINE
USERS                          ONLINE
SALES_DATA                     ONLINE

6 rows selected.

SQL> select count(*) from sales;

  COUNT(*)
----------
    918843
```

3 CONCLUSION

With the introduction of cross-platform transportable tablespaces, Oracle has made moving data between heterogeneous platforms extremely easy. With NetApp technology this process becomes even simpler and faster by leveraging FlexClone copies.

By creating a FlexClone copy, we reduce the amount of time required for the source database's tablespaces to be in read-only mode. This process can also be efficiently repeated across numerous heterogeneous target systems by leveraging the single NetApp FlexClone copy as the source for the RMAN conversion.

4 ACKNOWLEDGEMENTS

Tom C. DeCristofaro, Regional Systems Engineer Manager, NetApp

Michael Doherty, CSE ll, Oracle, NetApp

Bill Heffelfinger, Database and Business Apps Global Field Technology Lead, NetApp

Antonio Jose Rodrigues Neto, Consulting Systems Engineer, NetApp

Steven Schuettinger, Technical Alliance Manager, NetApp

Craig Sullivan, SAP Alliance Engineer, NetApp

Lynne Thieme, Senior Manager, Oracle Alliances Engineering, NetApp

APPENDIX A: SUPPORTED PLATFORMS

Table 1) Supported Platforms

PLATFORM_ID	PLATFORM_NAME	ENDIAN_FORMAT
1	Solaris OE (32-bit)	Big
2	Solaris 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 NT®	Little
8	Linux IA (32-bit)	Little
9	Linux IA (64-bit)	Little

APPENDIX B: ENVIRONMENT LAYOUT

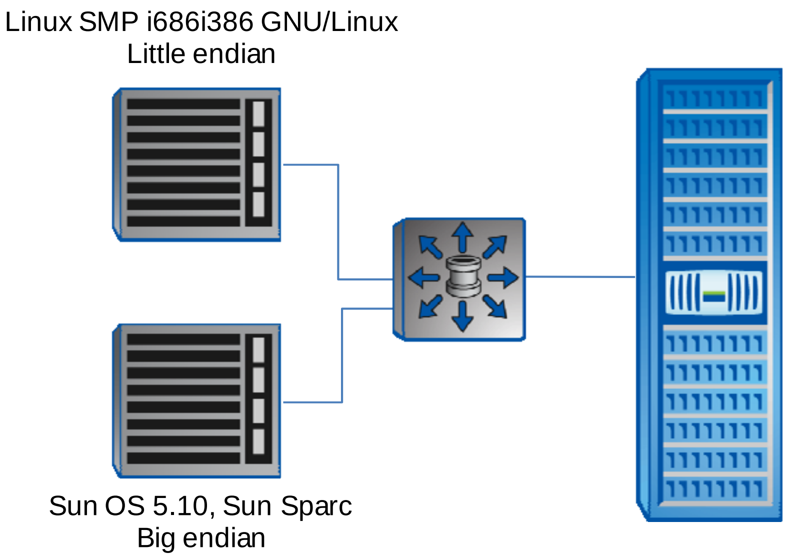


Figure 4) Environment Layout

SOURCE

```
CREATE DIRECTORY "ETL_STAGE" as '/oracle/etl_stage';
grant read,write on directory "ETL_STAGE" to public

expdp system/oracle dumpfile=endian.dmp directory=etl_stage transport_tablespaces =
sales_data
```

TARGET

```
CREATE DIRECTORY "SOLARIS_STAGE" as '/oracle/etl_stage/solaris';
grant read,write on directory "SOLARIS_STAGE" to public;
```

```
impdp system/oracle dumpfile=endian.dmp directory=SOLARIS_STAGE
transport_datafiles=/oracle/etl_stage/solaris/CLONE_sales_data.dbf
```

NetApp File Level FlexClone Command

```
clone start /vol/etl_volume/sales_data.dbf /vol/etl_volume/CLONE_sales_data.dbf
```

APPENDIX C: SAMPLE SHELL SCRIPT

```
#!/bin/sh

####################################################################################################

# Purpose:

# Use the Data ONTAP feature to clone a data file and

# Use RMAN to convert data file from one platform to another

# This script will connect to the database via RMAN

# put the target tablespace in read only

# and convert to target platform

# For a complete list of platforms issue this sql statement

# select * from v$transportable_platform order by platform_id;

# Some best practices state that this SHOULD be run on the TARGET server as not to burden the SOURCE

# Assumptions:

#               1) This script assumes that os user oracle has $ORACLE_HOME defined

#               2)  Assumes $ORACLE_HOME/bin in path

#  Modifications:

#               Please make me better, and send a copy to Matt D.

#               Thanks and enjoy

####################################################################################################

SCRIPT_HOME=`pwd`;export SCRIPT_HOME

# Get a list of the data files

sqlplus "sys/netapp as sysdba" <<EOF

set heading off

set timing off

SELECT  FILE_NAME, TABLESPACE_NAME FROM DBA_DATA_FILES;

EOF

#Ask for the tablespace to convert

printf "What tablespace would you like to convert? >"
```

```
read TS
# Check to see if the Tablespace is self-contained and is a candidate for Transportable Tablespace movement
# Future versions should reture a success or failure code and end the script on any violations
sqlplus "sys/netapp as sysdba" <<EOF
execute DBMS_TTS.TRANSPORT_SET_CHECK('$TS', TRUE);
/
SELECT * FROM TRANSPORT_SET_VIOLATIONS;
EOF
# Ask for the data file name
printf "What is the name of the DATAFILE for the tablespace? >"
echo "For example: users01.dbf"
read DF
#Ask for the data file path
printf "Where is the datafile you would like to convert? >"
echo "For example: /oracle/data01/demodb "
read DFPATH
# Ask for where to put the converted data files; this should be an NFS mount
printf "Where would you like the converted datafile to go? >"
read CLONEPATH
 # Ask for the degree of parallelism
 printf "What degree of parallelism would you like for the conversion? >"
 read PQ
#more $SCRIPT_HOME/platforms_list
echo "PLATFORM_NAME                                                   "
echo "----------------------------------------------------------------------------------------------------- "
echo "Solaris[tm] OE (32-bit)                                                      "
echo "Solaris[tm] OE (64-bit)                                                      "
echo "HP-UX (64-bit)                                                                            "
echo "HP-UX IA (64-bit)                                                         "
echo "HP Tru64 UNIX                                                             "
echo "AIX-Based Systems (64-bit)                                                "
echo "Microsoft Windows IA (32-bit)                                             "
echo "Microsoft Windows IA (64-bit)                                                   "
echo "IBM zSeries Based Linux                                                         "
echo "Linux IA (32-bit)                                                                         "
echo "Linux IA (64-bit)                                                                         "
echo "Microsoft Windows 64-bit for AMD                                                "
echo "Linux 64-bit for AMD                                                                      "
echo "HP Open VMS                                                                     "
echo "Apple Mac OS                                                                    "
echo "Solaris Operating System (x86)                                                  "
echo "IBM Power Based Linux                                                           "
echo "HP IA Open VMS                                                                  "
echo "Solaris Operating System (AMD64)                                                "
# Ask for platform target...copy and paste for now...
echo ""
echo ""
echo "Copy and paste in selection from above list....for now....sorry..."
printf "What platform is the target for conversion? >"
read OS
# Get the NetApp Controller name or IP
printf "Please enter the NetApp Controller or IP > "
read NTAP
# Need to get the full path of the data file as it exists on the Controller
printf "Please enter the full path of the datafile as it exists on the Controller >"
echo "For example: /vol/sunlab_ora_data/demodb"
read NTAPMASTER
# Need to get the clone path on the Controller
printf "Please enter the datafile clone name on the Controller >"
echo "For example: users_clone.dbf"
read NTAPCLONE
# Do the clone operation on the Controller
rsh $NTAP clone start ${NTAPMASTER}/${DF} ${NTAPMASTER}/${CLONE}
# Do the RMAN data file conversion
```

```
rman <<EOF
connect target
sql 'alter tablespace ${TS} read only';
CONVERT DATAFILE  '${DFPATH}/${NTAPCLONE}'  to platform '${OS}' parallelism = ${PQ}
   DB_FILE_NAME_CONVERT '${DFPATH}', '${CLONEPATH}';
exit
EOF
# Show the tablespace status
sqlplus "sys/netapp as sysdba" <<EOF
select tablespace_name, status from dba_tablespaces;
exit;
EOF
# Create the transportable tablespace metadata file
echo tablespaces=$TS > $SCRIPT_HOME/convert.par
echo transport_tablespace=y >> $SCRIPT_HOME/convert.par
echo file=$DIR/TTS_$TS.dmp >> $SCRIPT_HOME/convert.par
exp "'sys/netapp as sysdba'"  parfile=$SCRIPT_HOME/convert.par
# Put the tablespace back to read write
rman <<EOF
connect target
sql 'alter tablespace ${TS} read write';
exit
EOF
# Show the tablespace status
sqlplus "sys/netapp as sysdba" <<EOF
select tablespace_name, status from dba_tablespaces;
exit;
EOF
```

DISCLAIMER

NetApp provides no representations or warranties regarding the accuracy, reliability, or serviceability of any information or recommendations provided in this publication or with respect to any results that might be obtained by the use of the information or observance of any recommendations provided herein. The information in this document is distributed AS IS, and the use of this information or the implementation of any recommendations or techniques herein is a customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. This document and the information contained herein must be used solely in connection with the NetApp products discussed in this document.

www.netapp.com