



Technical White Paper  
August 2010

# Migrating to Oracle 11g Using Data Replicator Software with Transportable Tablespaces

## Contents

Contents.....	2
Introduction .....	3
Benefits of Using This Method.....	4
Setting Up the System Environment .....	4
Technical Specifications .....	5
Reference Architecture .....	6
Setting Up the Licensable Premium Features.....	8
Enabling a Premium Feature .....	8
Setting Up Data Replicator Software .....	9
Detailed Example Using This Method.....	9
Preparing the Source Database to Be Cloned .....	10
Preparing the Servers.....	10
Preparing the Target Database .....	12
Conclusion .....	15
Appendix A: References.....	17

## Introduction

Migrating a database system to a new release of Oracle has traditionally required an outage of the database. This requirement can be problematic with a production database. By combining the Data Replicator Software (DRS) premium feature of the Sun storage array with Oracle's Transportable Tablespace technology, no outage is required. For example, you can migrate from Oracle Database 10g to Oracle Database 11g without taking the production database offline.

This document describes a new solution for data migration that does not require you to put production tablespaces in read-only mode. This solution is especially desirable because it can be executed without disrupting the production database.

This solution supersedes several other methods:

- Import/Export
- SQL\*Plus Copy
- Create Table as Select (CTAS)
- Transportable Tablespace option

Using a test case as a detailed example, this document provides the information needed to successfully migrate data from an Oracle Database 10g to an Oracle Database 11g. This solution consists of a combination of the Oracle's Transportable Tablespace feature and Sun storage arrays with the Sun Storage Common Array Manager licensable premium feature, Data Replicator Software (DRS). This combination simplifies the migration by accomplishing it with no interruption to the source database.

**NOTE** To find out more about the restrictions and capabilities of Oracle's Transportable Tablespace method of moving data from one database to another, refer to Oracle's documentation on Transportable Tablespaces.

The intended reader for this document is an Oracle Database administrator with experience in the following areas:

- Oracle Database and its related components
- Cloning an Oracle Database

- Storage, including an understanding of data services and the premium features that support those services

## Benefits of Using This Method

When you use DRS, you can create a copy of the database, and then clone it onto a different server. From the target server, you can perform certain critical functions on the clone rather than on the production database. Using a copy for critical functions has several advantages:

- No disruption of the source production database
- Fewer staff resources
- Anytime duplication of the database
- Data replication for recovery of the production database

There are many benefits of using the DRS premium feature. The DRS premium feature is an option for real-time replication of data that creates copies of the source volumes on a second target storage array. The second storage array might reside within the same computing environment, or, for disaster recovery purposes, might reside many miles away or even in a different state or country.

A fully-synchronized DRS setup provides many ways to use the remote mirrors without disrupting the primary source volumes. DRS provides rapid recovery from a disaster or a catastrophic failure of the primary site by using role reversals to promote the secondary volumes to a primary role. Hosts can then read and write to the newly promoted volumes, allowing business operations to continue.

Implementing the DRS premium feature eliminates the issue of placing the production database tablespaces in read-only mode before running the Transportable Tablespace process. DRS mirrors the data to another set of drives that you can use to bring up a copy of the production database. Then you can copy the data from this database to the target database of choice with no impact to the source database.

## Setting Up the System Environment

Use the reference architectures as guidelines for setting up your own system environment. The reference architectures serve as examples only. You can modify the system environment as necessary to suit your needs. There are two reference architectures:

- The 6540 reference architecture consists of one primary server and one secondary server attached to two 6540 controllers. See [Figure 1](#) on page 5.
- The Sun Storage 6580/6780 reference architecture consists of one primary server and one secondary server attached to one Sun Storage 6580/6780 controller and one 6540 controller. See [Figure 2](#) on page 6.

The reference architectures were developed for testing seven common scenarios that an Oracle Database administrator would regularly encounter. This document describes only one of those scenarios. The other six scenarios are described in separate documents. Those documents are listed in [“Appendix A: References.”](#)

The reference architectures use three premium features—Volume Copy, Snapshot, and Data Replicator Software (DRS)—to test various backup and recovery methods and to test multiple development and upgrade scenarios. The specific Oracle scenario described in this document might not use all three premium features.

## Technical Specifications

Here are the specifications for components of the reference architectures.

### Operating System Information

Red Hat Enterprise Linux 5.1 Version 2.6.18-53.el5

The RDAC driver is MPP Driver Version 99.03.C000.0005

### Oracle Information

Oracle Database EE 10g Release 2 Version 10.2.0.3

Oracle Database EE 11g Release 1 Version 11.1.0.6

### Storage Information

6540

Firmware Version 07.60.36.13

NVSRAM Version N6091-760843-004

Sun Storage 6580/6780

Firmware version 07.60.36.13

NVSRAM version N7091-760843-004

Sun Storage Common Array Manager (CAM) 6.7.0

### Volume Configuration

For each reference architecture, the primary storage array has the following volumes:

- Oracle distribution volume /u01
- Primary Oracle Database volumes
- Snapshot volumes
- Oracle Volume Copy volumes

For each reference architecture, the secondary storage array has the following volumes:

- Snapshot volumes
- Secondary DRS volumes for the Oracle distribution and database volumes

## Reference Architecture

Figure 1 shows that the 6540 reference architecture consists of one primary server and one secondary server attached to two 6540 controllers.

Figure 2 on page 5 shows that the Sun Storage 6580/6780 reference architecture consists of one primary server and one secondary server attached to one Sun Storage 6580/6780 controller and one 6540 controller.

Figure 1 6540 Reference Architecture

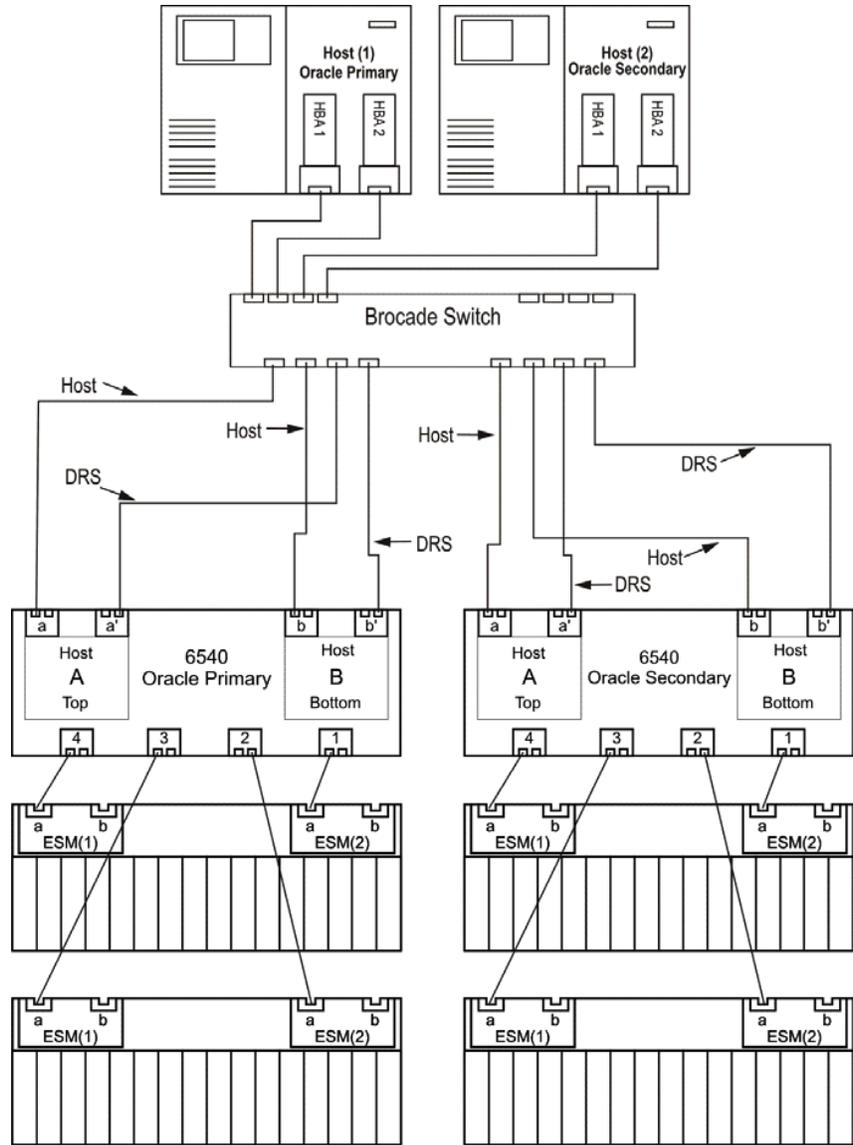
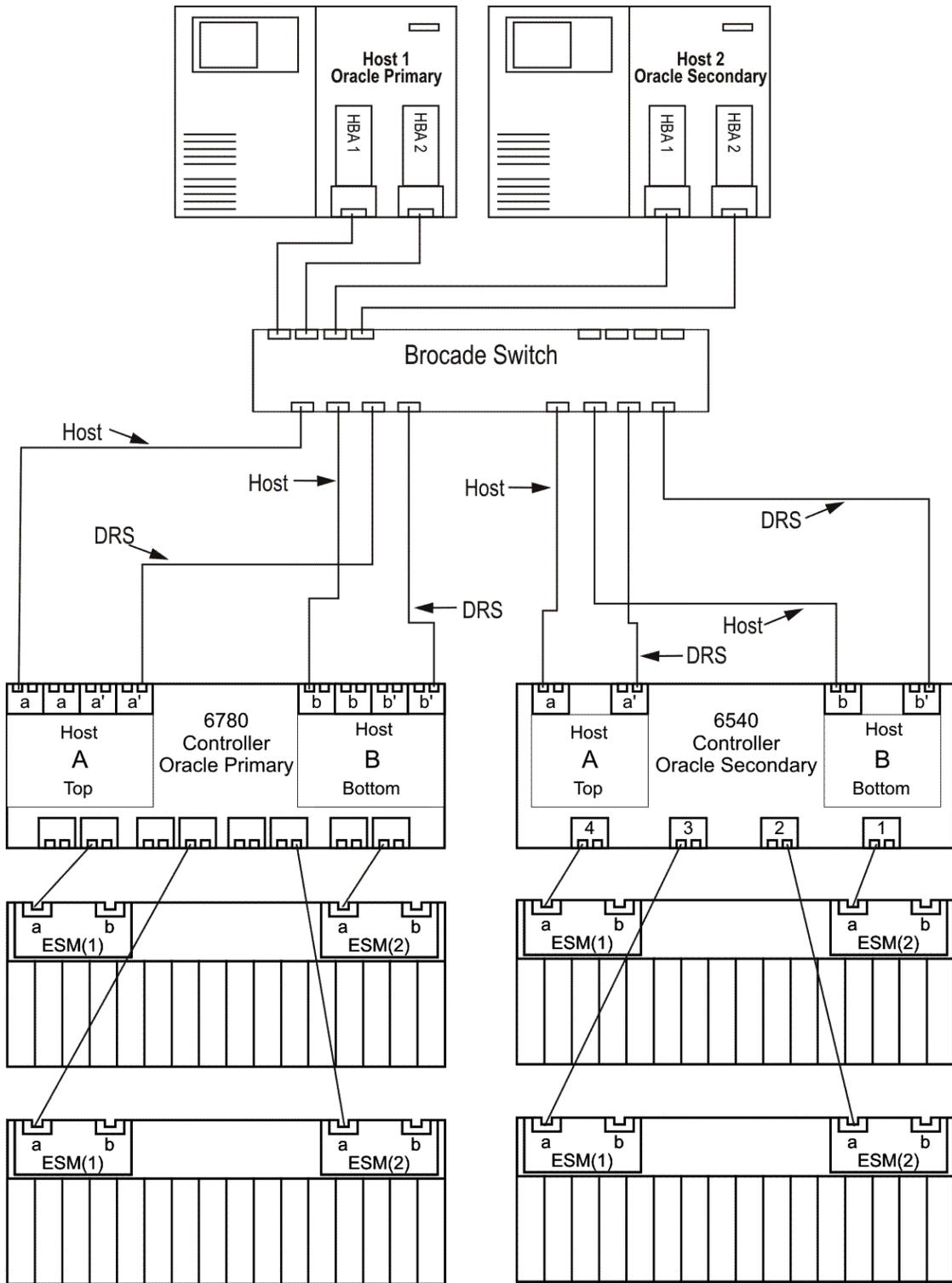


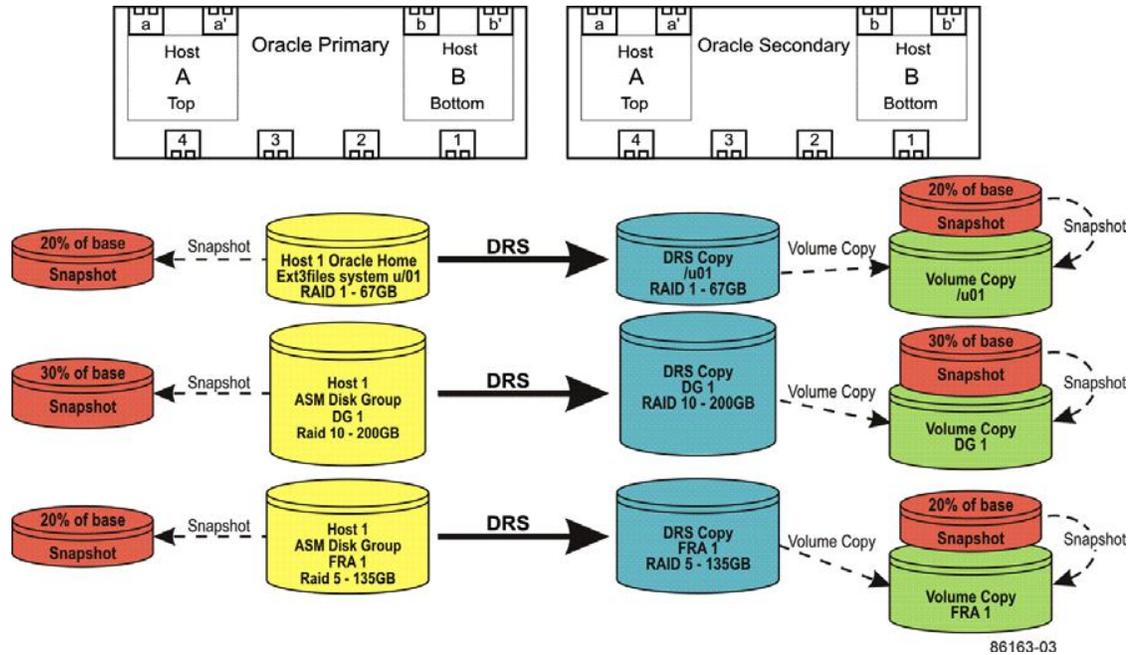
Figure 2 Sun Storage 6580/6780 Reference Architecture



## Volume Configuration

Figure 3 shows the volume configuration in the tests of Sun Storage Common Array Manager licensable premium features. Both reference architectures use the same volume configuration.

Figure 3 Volume Configuration



## Setting Up the Licensable Premium Features

After your system environment is set up, your next step is to enable the DRS licensable feature. Licensable features must be enabled on all arrays separately. You must enable a licensable feature only once for the feature to become fully enabled throughout each storage array. To enable the licensable feature, obtain the Feature Key file from Oracle for each array.

### Enabling a Premium Feature

1. Using the Sun Storage Common Array Manager browser interface, open the **Licensable Feature Summary** page by expanding the **Administration** folder under each storage array and click **Licensing**.
2. In the **Available Features** table, click the **Add License...** button.
3. In the **Add License** page, click the **Browse...** button and navigate to the \*.KEY file that contains the activation key for the licensable feature. When Oracle sent the \*.KEY file, you specified its file name and location.
4. Select the file, and then click **Open**.
5. Back in the **Add License** page, click the **Enable** button.

6. Repeat this procedure to activate an additional premium feature key.

## Setting Up Data Replicator Software

DRS lets you create an identical copy of the source volumes on a second storage array or on a remote storage array. DRS requires a physical hardware connection between two storage arrays. Connect the upper host ports to the last host ports on the storage array using a SAN switch infrastructure. Both controllers require the following connections:

- Connect the last host port of controller A on the target storage array to the last host port of controller A on the source storage array.
- Connect the last host port of controller B on the target storage array to the last host port of controller B on the source storage array.

DRS disables any host port connections to the last host port when DRS is initialized.

## Detailed Example Using This Method

This section describes in detail the server setup, the premium feature configuration, and the Oracle commands that were executed to execute the transportable tablespaces process. The test case serves as a detailed example. You can modify the instructions to suit your environment.

There is a need to copy all database objects from the TEST\_TTS tablespace from the production instance of Oracle Database 10g to the production instance of Oracle Database 11g. In the test case, both databases use filesystems for the tablespaces. The objects in the TEST\_TTS tablespace are owned by TTS\_USER.

**NOTE** If you are using Oracle Automatic Storage Management (Oracle ASM), the steps differ slightly but the concept is the same.

In the test case, the production instance of Oracle Database 10g, named pfd2, is running on HOST1 and the production instance of Oracle Database 11g, named pfd1, is running on HOST2. The Oracle 10g binary files on HOST1 are located on the /u01 filesystem. On HOST2, the instance of Oracle Database 11g uses the /db filesystem for its binary files. DRS is configured to mirror the /u01 filesystem from HOST1 which contains both the Oracle binary files and datafiles.

After mirroring the data, stop the DRS process to create a point-in-time image of the Oracle Database 10g on another set of drives. Then mount these drives on HOST2 and start the Oracle Database 10g. Because HOST1 uses /u01, the DRS drives were mounted as /u01. From this point, the database is set up to perform the Transportable Tablespaces process. After copying the data, you can remove the Oracle Database 10g, along with the DRS drives, from HOST2.

## Preparing the Source Database to Be Cloned

The source database, named pfdb2, resides on HOST1.

1. Create two tables. As a test, populate the tables with hundreds of rows of data. In the next steps, you can drop the unnecessary rows of data. Enter the following command.

```
SQL> create table scott.tb1 (name varchar2 (80), value varchar2(4000));
```

```
Table created.
```

```
SQL> insert into scott.tb1 select name, value from v$parameter;
```

```
258 rows created.
```

```
SQL> create table scott.tb2 (name varchar2(257));
```

```
Table created.
```

```
SQL> insert into scott.tb2 select name from v$datafile;
```

```
4 rows created.
```

```
SQL> Commit;
```

2. From SQL\*Plus, select the row count from the objects t1 and t2 in tablespace test\_tts. You will use these values to verify that the same number of rows have been migrated to the new database when finished.

```
SQL> select count(1) from tts_user.t1;
```

```
COUNT(1)
```

```
-----
```

```
258
```

```
SQL> select count(1) from tts_user.t2;
```

```
COUNT(1)
```

```
-----
```

```
4
```

## Preparing the Servers

1. Set up the primary server and the storage array. After the primary server and the storage array are running in an optimal state, validate the following components:
  - Databases are optimal and running.
  - Primary storage is optimal.
  - Secondary storage is optimal.
  - DRS feature is enabled and the DRS links are connected.

2. Set up the secondary server. Let the secondary server remain idle until it is needed to perform the flashback process. Do not map any volumes to the secondary server yet.
3. Using the Sun Storage Common Array Manager browser interface, initiate an DRS copy for source array volume mounted at /u01 (Oracle home, Oracle binary files, and filesystem database) to the target storage array by creating a new replication set. This is done using the **Create New Replication Set** wizard. The wizard is launched by clicking the **New..** button in the **Replication Sets** table found in the **Replication Set Summary** page. The **Replication Set Summary** page is shown by clicking on **Replication Sets** under the primary array. The target volume must already exist on the secondary array and be greater than or equal in size to the source volume on the primary array.

The test case used the following **Create New Replication Set** wizard properties:

- Mode: **Asynchronous**
- *Add to write consistency group* checkbox enabled.
- Replication priority: **Highest**
- Resynchronization method: **Automatic**

**NOTE: The source and target volumes must be owned by the same controller on their respective arrays in order to create a replication set relationship between them. For example, if controller A on the primary array owns the source volume, controller A must also own the target volume on the secondary array.**

4. Before preparing the source database, wait for the DRS volume copy to complete. Using the Common Array Manager browser interface, notice the **Synchronization Progress** field of the **Replication Sets** table in the **Replication Set Summary** page. The DRS volume copy has completed when the value of this field for the new replication set created in step 3 changes from “Synchronization In Progress” to “Replicating” which may take some time depending on the size of the volume.
5. After synchronization is complete, you must quiesce the database to allow for a clean break in the DRS mirrors:

```
SQL> alter system switch logfile;
SQL> alter database begin backup;
SQL> alter system quiesce restricted;
SQL> select active_state from v$instance;
ACTIVE_STATE
-----
```

QUIESCED

6. Suspend the replication set. Using Sun Storage Common Array Manager browser interface, click the replication set name in the **Replication Sets** table on the **Replication Set Summary** page for either the primary or secondary array. This will display the **Replication Set Details** page. Click the **Suspend** button. Back in the **Replication Set Summary** page, confirm that the **Synchronization Progress** field now displays the value “Suspended” for the replication set.

7. Bring the source database back online.

```
SQL> alter system unquiesce;
```

```
SQL> alter database end backup;
```

Database altered.

### Preparing the Target Database

Complete the following steps on the target, HOST2. The target HOST2 already has one database, an Oracle Database 11g named pfdb1. Add the Oracle Database 10g, named pfdb2.

1. Using the Sun Storage Common Array Manager browser interface, map the target DRS volume to HOST2 and make the volume visible to the operating system.
  - a. After expanding the secondary array's folder, click **Mappings** to display the **Mapping Summary** page.
  - b. Click **New...** to launch the **Create New Mappings** wizard.
  - c. Choose a specific LUN and make note of the number chosen.
  - d. Run the `/usr/sbin/mppBusRescan` utility on HOST2 to have the operating system look for the newly mapped target.
  - e. Run the `/opt/mpp/lsvdev` command on HOST2 to confirm that the LUN chosen for the target DRS volume in step c above is shown next to the secondary array's name.
  - f. Make a mount point with `mkdir /u01`.
  - e. Mount the file system as `/u01` using the device shown next to the LUN in step e.
2. Set the Oracle environment variables to point to the `/u01` mount point, and modify the `ORACLE_SID`.

This setting lets the Oracle user start the Oracle Database 10g using the replicated Oracle10g binary files and datafiles.

3. From SQL\*Plus, start the instance of Oracle Database 10g on HOST2.
4. Because the test case uses the Data Pump Export utility to extract the metadata of the test\_tts tablespace, create a dictionary in the Oracle Database 10g.

This dictionary contains the output file of the export process.

```
SQL> CREATE OR REPLACE DIRECTORY tts_dir AS '/u01/oradata/pfdb2/testtts';
```

5. Grant rights on the tts\_dir to public.

```
SQL> GRANT READ, WRITE ON DIRECTORY tts_dir TO public;
```

6. Check for tablespace self-consistency. Before transporting the test\_tts tablespace, make sure that all objects in the tablespace are self-contained. This means that the tablespace has no external dependent objects or objects owned by user SYS. As sysdba, enter the following commands.

```
SQL> EXECUTE sys.DBMS_TTS.TRANSPORT_SET_CHECK('test_tts',TRUE);
```

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

no rows selected

Finding no rows (“no rows selected”) means that the tablespace has no external dependent objects.

7. Alter the tablespace test\_tts to read-only mode.

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

8. To run expdp, open an operating system command prompt or go into host mode from the SQL\*Plus prompt to perform the metadata export:

```
$ expdp system/oracle DUMPFILE=test_tts.dmp DIRECTORY=tts_dir
```

```
TRANSPORT_TABLESPACES=test_tts
```

9. After exporting the metadata, and while the tablespace is still in read-only mode, either copy or FTP the datafiles to the target database:

```
cp /u01/oradata/pfdb2/test_tts.dbf /db/oradata/pfdb1/.
```

10. Shut down the instance of Oracle Database 10g named pfdb2.

11. Because the test case uses the Data Pump Import utility to import the metadata of the test\_tts tablespace, create a dictionary in the instance of Oracle Database 11g.

This dictionary contains the output file from the export process performed previously in step 8.

```
SQL> CREATE OR REPLACE DIRECTORY dest_tts_dir AS '/db/oradata/pfdb1/testtts';
```

12. Copy or FTP the metadata dump file from the export process above to the dest\_tts\_dir directory, which will be used by the impdp process.

```
$ cp /u01/oradata/pfdb2/testtts/test_tts.dmp /db/oradata/pfdb1/testtts/.
```

13. Because the objects of the tablespaces that are being migrated into the Oracle Database 11g were owned by user tts\_user, create a user named tts\_user in the Oracle Database 11g.

```
SQL> create user tts_user identified by tts_user  
default tablespace users temporary tablespace temp;
```

14. After the metadata and datafiles have been moved to the location where they can be accessed by the impdp utility, run the import process:

```
$ impdp system/oracle DUMPFILE=test_tts.dmp  
DIRECTORY=dest_tts_dir TRANSPORT_DATAFILES='/db/oradata/pfdb1/test_tts.dbf'
```

The following output is from the import data pump process.

```
oracle@host2$ impdp system/oracle DUMPFILE=test_tts.dmp  
DIRECTORY=dest_tts_dir TRANSPORT_DATAFILES='/db/oradata/pfdb1/  
test_tts.dbf'  
Import: Release 11.1.0.6.0 - 64bit Production on Thursday, 20  
March, 2008 13:36:44  
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=test_tts.dmp DIRECTORY=dest_tts_dir  
TRANSPORT_DATAFILES=/db/oradata/pfdb1/test_tts.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
13:36:47
```

15. After running the import, modify the owner of the objects to set quota and the default tablespace for user tts\_user.

```
SQL> alter user tts_user default tablespace test_tts quota unlimited on test_tts;
```

16. Because the export process was completed while the tablespace was in read-only mode, you must change the tablespace back to read-write mode to make the tablespace usable:

```
SQL> ALTER TABLESPACE test_tts READ WRITE;
```

17. Verify that the transported objects have the same row count as in the source database. Take the results of the following queries and compare them to the results of the same query from the source database in the beginning of this document, in step 2 in [“Preparing the Source Database to Be Cloned.”](#)

```
SQL> select count(1) from tts_user.t1;
COUNT(1)
-----
258
SQL> select count(1) from tts_user.t2;
COUNT(1)
-----
4
```

In this test case, the objects are validated.

18. If the replication set suspended using the steps above is not intended to be re-used, the stated best practice is to delete the replication set. Deleting a replication set will not alter the contents of either the source or target volumes. In fact, deletion of the replication set instead of the suspending it in the steps above will achieve the same intended results. Using Sun Storage Common Array Manager, replication sets can be deleted from the **Replication Set Summary** page.

## Conclusion

Using the procedures in this document, you can migrate an Oracle Database 10g to Oracle Database 11g without having to take it out of production. You used DRS to avoid having to put the production database in read-only mode. Then you were able to use Oracle’s Transportable Tablespaces feature to complete the migration quickly.

Practice this procedure on another database before implementing it on the production database. As you practice the migration, make note of modifications that were necessary due to your specific environment.

## Appendix A: References

This document is one of seven detailed examples that explain how to complete common but important tasks often required from an Oracle database administrator. Refer to the other documents as needed.

- Cloning an Oracle Database to the Same Server Using Snapshot and Volume Copy
- Cloning an Oracle Database Using Data Replicator Software
- Forward Recovery of an Oracle Database Using Data Replicator Software
- Migrating to Oracle 11g Using Data Replicator Software with Transportable Tablespaces
- Recovering from Catastrophic Failures Using Data Replicator Software for Data Replication
- Safely Upgrading an Oracle Database Using Data Replicator Software
- Selective Restores Using Data Replicator Software with Oracle Flashback Database



Migrating to Oracle 11g Using Data  
Replicator Software with Transportable  
Tablespaces  
August 2010

Oracle Corporation  
World Headquarters  
500 Oracle Parkway  
Redwood Shores, CA 94065  
U.S.A.

Worldwide Inquiries:  
Phone: +1.650.506.7000  
Fax: +1.650.506.7200  
oracle.com



| Oracle is committed to developing practices and products that help protect the environment

Copyright © 2010, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. UNIX is a registered trademark licensed through X/Open Company, Ltd. 0810

**SOFTWARE. HARDWARE. COMPLETE.**