



Technical White Paper
August 2010

Cloning an Oracle Database to the Same Server Using Snapshot and Volume Copy

Contents

Contents.....	2
Introduction	3
Benefits of Using This Method.....	4
Benefits of Using Volume Copy	4
Setting Up the System Environment.....	4
Technical Specifications	5
Reference Architecture.....	6
Setting Up the Licensable Premium Features.....	9
Setting Up Snapshot.....	10
Setting Up Volume Copy	10
Detailed Example Using This Method.....	10
Creating a Snapshot and a Volume Copy.....	11
Cloning the Database.....	12
Preparing the Source Database to Be Cloned	13
Preparing the Target Database on the Same Server	16
Conclusion	19
Appendix A: References.....	20

Introduction

Maximizing hardware return on investment is an important business concern. To conserve hardware resources, many IT environments run multiple databases on a single server. With this proliferation of databases also comes the proliferation of database clones. Database clones are very useful for testing, developing, or keeping a point-in-time image for evaluation purposes. However, too many database clones take up a lot of server space.

When there is a shortage of server hardware, and when the source server has enough resources to run another database, the clone can run on the same server as the source database. For example, if needed, you could clone a development database back onto the development server.

Cloning a database onto the same server is more complicated than cloning the database onto a different server. To isolate the clone from its source database on the same server, you have to know which files to edit and how to edit them. Otherwise, any mistake that you might make when you create the clone might overwrite the source database and corrupt it. Not surprisingly, Oracle Database administrators are often reluctant to try to create clones on the same server. Even though doubling up would conserve hardware resources, sometimes the risk is not worth taking.

This document explains how to isolate a cloned database from its source. This document explains how to set up the cloned database using different storage locations from the source database. Initially, all of the pointers in the cloned database point to the original locations for the source database. You have to change the locations before you can use the cloned database. This document describes how to edit all of the references to the file systems so that it is safe to use the cloned database.

This document also describes a solution that simplifies cloning. Specifically, this document focuses on expediting the important task of cloning an Oracle Database to the same server. The solution combines Oracle Database products with Sun™ storage arrays and the licensable premium features, Snapshot and Volume Copy, available Sun Storage Common Array Manager.

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

You can use Snapshot and Volume Copy with Oracle to create a copy of the database and clone it onto the same server as the source database. Then you can perform certain critical functions on the copy rather than on the production database. Using a copy for critical functions has many advantages:

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

Benefits of Using Snapshot

The Snapshot premium feature lets you create a point-in-time copy of data on a volume. This copy is the logical equivalent of data on the complete base volume, but the point-in-time copy is created much faster than a complete physical copy, typically, within a few seconds. Due to the architecture of Snapshot, creating the copy typically requires much less drive space. Snapshot volumes become a read/write volume that can be assigned to the same host or to a secondary host depending on the content of the volume. You can use the Snapshot volumes for many purposes, such as application backups, while leaving the base volume online and accessible to the user.

Benefits of Using Volume Copy

The Volume Copy premium feature is an in-box mechanism for replicating volume data within the same storage array. Volume Copy creates a complete physical copy of a source volume to another target volume within the same storage array. When you use Volume Copy with Snapshot, the replication occurs while the source volumes remain online and accessible to the user.

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 sever 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 sever 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

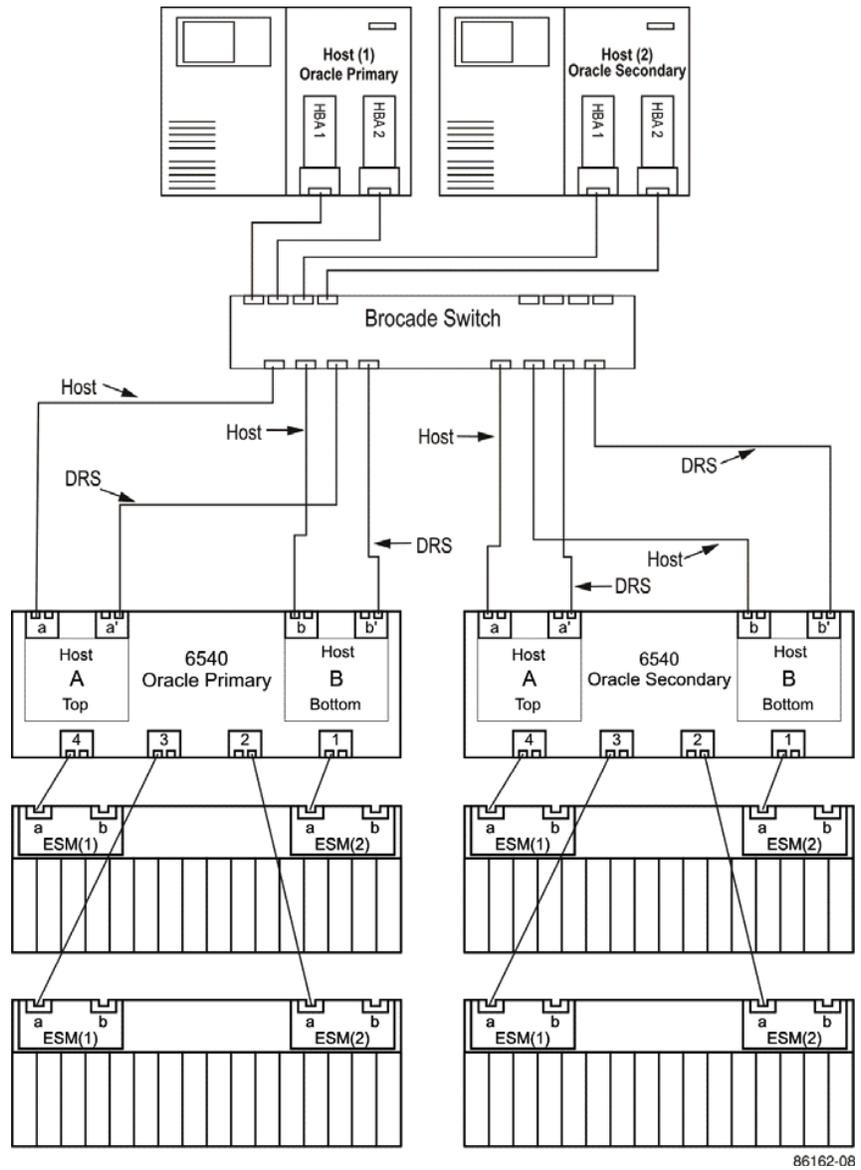
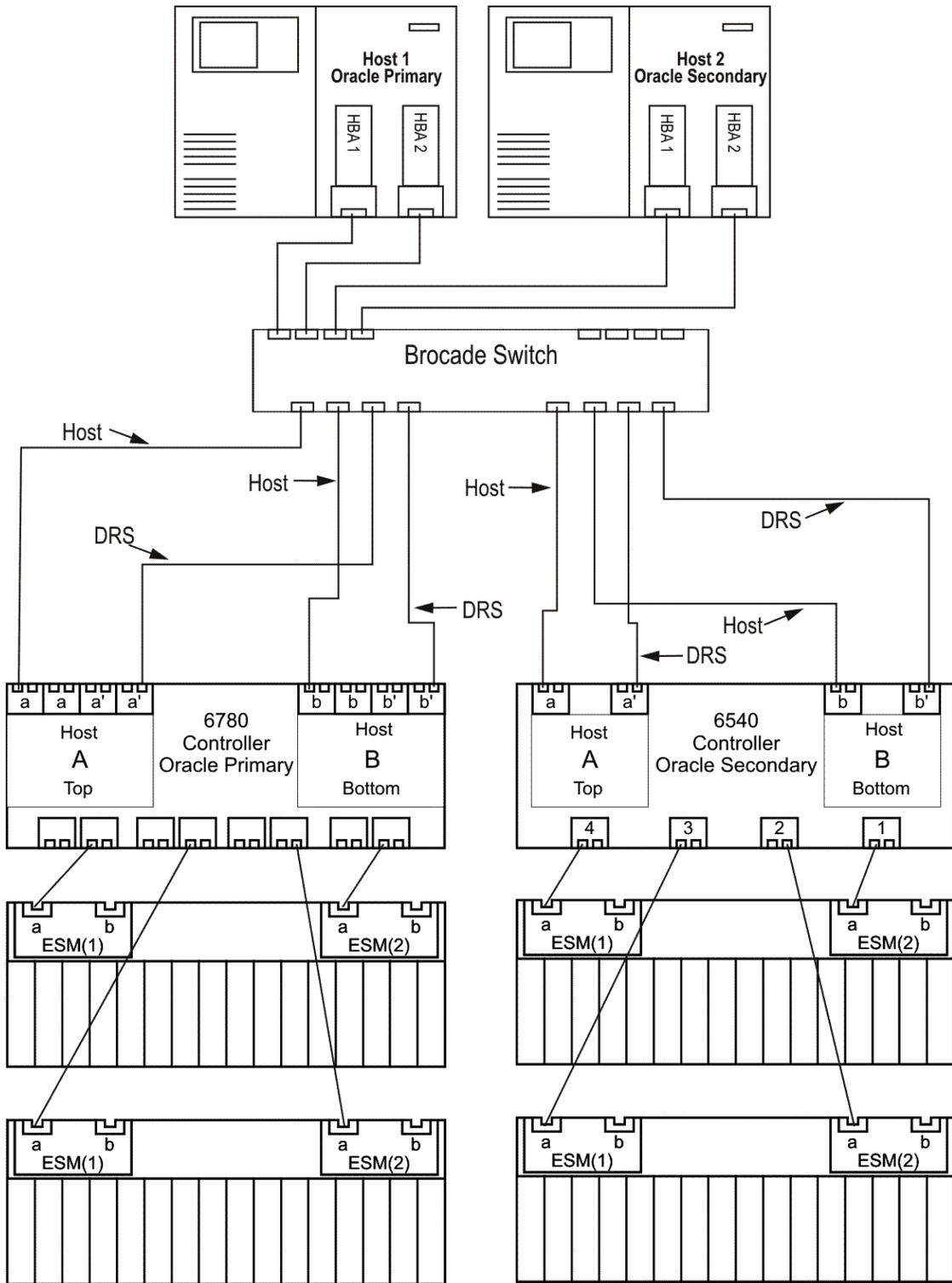


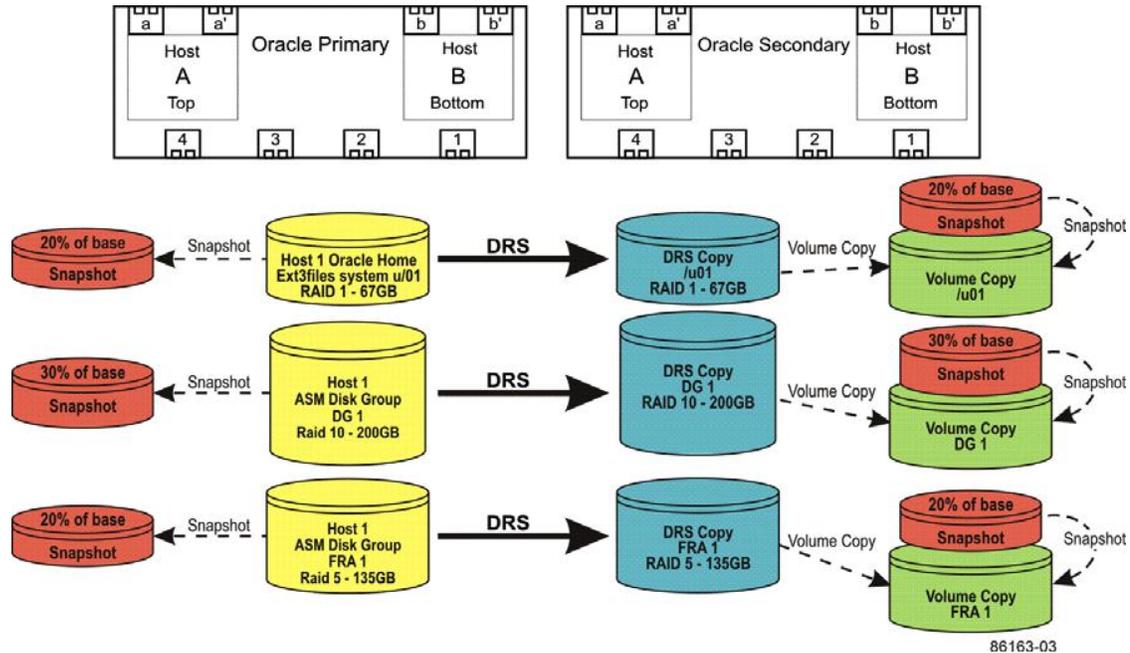
Figure 2 Sun Storage 6580/6780 Reference Architecture



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 Licensable 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 Snapshot

Snapshot requires a repository volume to store data that changes from the original data on the base volume. Snapshot is a point-in-time copy, not a complete replication. The Snapshot process reads data from the base volumes. When data is rewritten on the base volume, the original data is copied to the Snapshot repository to retain the point-in-time copy. The repository also contains any changes that the Snapshot replica incurs after being mapped to a host, because that host can perform writes to the Snapshot volume.

When you use a Snapshot replica against the primary application volumes, you must consider both the proper sizing of the Snapshot volume and the potential impact on performance. The default size for a volume is 20 percent. However, the optimal size of the repository volume depends on the application and on how much data changes during the time that the Snapshot remains active.

- When the application changes very little data, and when the Snapshot operation is performed for backup, the repository can be relatively small.
- When the application routinely changes the existing data, and when the Snapshot process is used for testing or upgrading purposes, the Snapshot repository must be relatively large—up to 100 percent of the base volume.

Setting Up Volume Copy

Volume Copy creates an identical copy of the source volumes within the same storage array. Volume Copy requires target volumes to be the same size as the source volume. Volume Copy turns the source volume into a read-only volume. Therefore, you must use Volume Copy with Snapshot on production volumes. Typically, you take a Snapshot of a volume, and then start a volume copy of the Snapshot replica.

Detailed Example Using This Method

After you set up the premium features, log on to Oracle to complete the procedures. This section gives a high-level overview of the work that you must do in Oracle. The following sections provide more detailed information.

In the test case, the source database resides on HOST1, named pfdb2. The Volume Copy procedure copies all of the pfdb2 data files and the Oracle binary files. The new database is cloned to the same host and is named pfdb3. The source database pfdb2 uses Oracle binary files and file systems on mount point /u01, and the new database uses mount point /u02 for all the data files and Oracle binary files.

NOTE When you want to clone a database that uses Oracle Automatic Storage Management (Oracle ASM), you cannot mount a copy of the ASM disks on the same server as the source database. You can use Volume Copy to clone an ASM database, but you must mount the ASM disks on another server that is not using Oracle ASM.

Perform a Snapshot and a Volume Copy to create a clone of mount point /u01. The mount point /u01 contains the binary files and all of the data files of the pfd2 database. When your database has more mount points, make sure that you include all of the mount points in the Snapshot process and the Volume Copy process. Mount this Volume Copy as /u02 on HOST1.

Creating a Snapshot and a Volume Copy

Volume Copy marks the source as read-only. Therefore in production environments, you must use Snapshot with Volume Copy. When you use Snapshot with Volume Copy, the source data remains read/write. Create the initial Snapshot replica before you quiesce the database to expedite quick Snapshot because the initial creation causes the LUN to be initialized. Subsequent use of the disable / re-create Snapshot is extremely quick.

1. Quiesce the database in preparation for a Snapshot and Volume Copy. Enter the following commands.

```
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
```

2. Create a new snapshot using the Sun Storage Common Array Manager browser interface. Open the **Snapshot Summary** page by clicking on **Snapshots** under the primary array. Click the **New...** button and follow the wizard. Use a repository volume size of 20 percent and use the **Do Not Map this Snapshot** option when prompted. Make note of the name chosen for the snapshot volume.
3. Bring the source database back online. Enter the following commands.

```
SQL> alter system unquiesce
SQL> alter database end backup;
SQL> alter system switch logfile;
```

Database altered.

NOTE You might need this online log file to complete the recovery of the database.

4. Using the Sun Storage Common Array Manager browser interface, create a new volume the same size as the source volume to be used as the copy target. In the **Volume Summary** page, click **New...** and follow the wizard. Make note of the name chosen for the new volume.
5. Create a copy of the snapshot created in step 2. In the **Snapshot Summary** page, click the name of the snapshot created in step 2 above from the **Snapshots** table. This will open the **Snapshot Details** page. Click the **Copy...** button. Follow the wizard and choose the volume created in step 4 as the copy target. After the copy completes, you can remove the snapshot, as it is no longer needed.
6. Map the copy target volume created in step 4 and make it visible to the operating system.
 - a. After expanding the array's folder, click **Volumes** to display the **Volume Summary** page. Click the target volume's name created in step 4 above in the **Volumes** table to show the **Volume Details** page.
 - b. Click **Map...** to launch the **Map Volume** wizard.
 - c. Choose a specific LUN and make note of the number chosen.
 - d. Run the `/usr/sbin/mppBusRescan` utility on the host to have the operating system look for the newly mapped target.
 - e. Run the `/opt/mpp/lsvdev` command to confirm that the LUN chosen for the target volume in step c above is shown next to the array's name.
 - f. Make a mount point with `mkdir /u02`.
 - g. Mount the file system as `/u02` using the device shown next to the LUN in step e.

Cloning the Database

To clone the database, you must prepare both the source database and the target location. You must prepare the source database first.

CAUTION Possible database corruption – You are likely to corrupt another database on the server if you try to start the cloned database on the same server as the source database without first changing all of the database references:

- Data files
- Log files
- Binary Locations

Preparing the Source Database to Be Cloned

In the test case, the source database was pfdb2.

1. Clone the source database and change its name.

Rebuild the controlfile using the new data file locations and the new mount points. First back up the controlfile in the source database to a text file, and then edit the controlfile.

```
SQL> alter database backup controlfile to trace as '/home/oracle/pfdb2ctrlfile.ora' reuse;
Database altered.
```

2. Find out where the data objects are located. Enter the following commands.

```
SQL> spool '/home/oracle/gatherpfdb2_info.txt'
SQL> col name format a30
SQL> col value format a60
SQL> select name, value from v$parameter where value like '%/u01%';
```

NAME	VALUE
spfile	/u01/app/oracle/product/10g/dbs/spfilepfdb2.ora
control_files	/u01/oradata/pfdb2/control01.ctl, /u01/oradata/pfdb2/control02.ctl, /u01/oradata/pfdb2/control03.ctl
db_recovery_file_dest	/u01/oradata/pfdb2/fra
background_dump_dest	/u01/app/oracle/admin/pfdb2/bdump
user_dump_dest	/u01/app/oracle/admin/pfdb2/udump
core_dump_dest	/u01/app/oracle/admin/pfdb2/cdump
audit_file_dest	/u01/app/oracle/admin/pfdb2/adump

```

dg_broker_config_file1    /u01/app/oracle/product/10g/dbs/dr1pfdb2.dat
dg_broker_config_file2    /u01/app/oracle/product/10g/dbs/dr2pfdb2.dat
9 rows selected.

```

SQL> show parameter dispatchers

NAME	TYPE	VALUE
dispatchers	string	(PROTOCOL=TCP) (SERVICE=pfdb2XDB)

SQL> col name format a60

SQL> select name from v\$datafile;

```

NAME
-----
/u01/oradata/pfdb2/system01.dbf
/u01/oradata/pfdb2/undotbs01.dbf
/u01/oradata/pfdb2/sysaux01.dbf
/u01/oradata/pfdb2/users01.dbf
/u01/oradata/pfdb2/example01.dbf
/u01/oradata/pfdb2/test_tts.dbf
6 rows selected.

```

SQL> col member format a80

SQL> select member from v\$logfile;

```

MEMBER
-----
/u01/oradata/pfdb2/pfdb2/redo03.log
/u01/oradata/pfdb2/pfdb2/redo02.log
/u01/oradata/pfdb2/pfdb2/redo01.log

```

3. Create the pfile for the new database.

- a. Create a new pfile to use initially when you start the database to create the controlfile.

```
SQL> create pfile='/home/oracle/initpfdb3Temp.ora' from spfile;
```

```
SQL> spool off
```

- b. Exit sqlplus.

4. Delete all of the control files on the newly cloned database.

Check the gatherpfdb2_info.txt file that you created in step 1 for the locations of the control files, and then delete them.

5. Gather the information that you must edit at the operating system level to make the files and directories unique. On the new mount point, the directories point to pfdb2, which is the source database. Enter these commands in the following order.

a. `cd /u02`

b. `find . -name *pfdb2*`

c. `find . -name *PFDB2*`

6. You must edit the information that you gathered from the two find commands in the previous step.

- Edit the directory names to reference the new database name pfdb3.
- Edit the content of the files.
- Ignore the trc files, log files, and so forth.

7. Change the name of a directory from the source name to the target name. In the test case, the name was changed from pfdb2 to pfdb3 using the following command.

```
mv /u02/oradata/pfdb2 /u02/oradata/pfdb3
```

8. Edit all of the files that you found.

- a. Rename the files to the new name.
- b. Change the content of the file to reference the new mount point and database name.

IMPORTANT Make sure that there is no reference to pfdb2 either in any files listed under the new mount point or in the contents of any files on the new mount point.

9. Change all of the Oracle environment variables. Point the environment variables to the new SID pfdb3 along with the new locations for ORACLE_HOME, ORACLE_BASE, PATH, and any other variable that points back to the old mount point or the source database pfdb2.

IMPORTANT Make sure that all of the Oracle environment variables are pointing to the new locations and new database by issuing env on the command prompt.



10. Edit the pfile.
 - a. Copy the pfile that you created from the source database in step 3 to the new `$ORACLE_HOME/dbs` directory. In the test case, the pfile was named `initpfdb3Temp.ora`.
 - b. Edit this pfile to change all references of the old mount point to the new mount point along with changing all references of `pfdb2` to `pfdb3`.
 - c. Save your changes.

11. Edit the controlfile that you created earlier (in step 1) from the source database with the following changes.
 - a. Change all of the references of the old mount point to the new mount point.
 - b. Change any references of `pfdb2` to `pfdb3`.
 - c. Edit the pfile entry on the `STARTUP NOMOUNT` command to reference the new pfile in the `$ORACLE_HOME/dbs` directory.
 - d. Change the create controlfile statement to say `set database` instead of `use database`.
 - e. Make sure that the Create Controlfile statement specifies `RESETLOGS`.
 - f. Delete all of the lines from the beginning to the controlfile file down to the Startup Nomount statement in the section that specify `RESETLOGS`.

Preparing the Target Database on the Same Server

Change the environment settings, and then complete the following steps. In the test case, the target database was `pfdb3`.

1. Create a control file for the newly-cloned database.

Open one window for editing the controlfile and one window for editing `sqlplus`. Then copy the commands from the controlfile window and paste them into the `sqlplus` window. Having both windows open lets you view each step of the cloning process and catch any errors that might arise. Alternatively, you can script this procedure and execute it from `sqlplus`. To execute the script, enter the following commands.

```
oracle@HOST1$ sqlplus
SQL> conn / as sysdba
```

```

SQL> STARTUP NOMOUNT pfile='/u02/app/oracle/admin/pfdb3/
scripts/initpfdb3Temp.ora'

SQL> CREATE CONTROLFILE set DATABASE "pfdb3" RESETLOGS
ARCHIVELOG

        MAXLOGFILES 16

        MAXLOGMEMBERS 3

        MAXDATAFILES 100

        MAXINSTANCES 8

        MAXLOGHISTORY 292

LOGFILE

        GROUP 1 '/u02/oradata/pfdb3/pfdb3/redo01.log' SIZE 50M,
        GROUP 2 '/u02/oradata/pfdb3/pfdb3/redo02.log' SIZE 50M,
        GROUP 3 '/u02/oradata/pfdb3/pfdb3/redo03.log' SIZE 50M

-- STANDBY LOGFILE

DATAFILE

        '/u02/oradata/pfdb3/system01.dbf',
        '/u02/oradata/pfdb3/undotbs01.dbf',
        '/u02/oradata/pfdb3/sysaux01.dbf',
        '/u02/oradata/pfdb3/users01.dbf',
        '/u02/oradata/pfdb3/example01.dbf',
        '/u02/oradata/pfdb3/test_tts.dbf'

CHARACTER SET WE8ISO8859P1

```

2. Query the mounted database to make sure that all destinations now point to the correct mount point and directories.

```

SQL> col name format a30SQL> col value format a60
SQL> select name, value from v$parameter where value like '%/u01%';

```

3. Create a new spfile for the new database.

```

SQL> create spfile from pfile='/u02/app/oracle/admin/pfdb3/scripts/initpfdb3Temp.ora';

```

4. Shut down the database to make sure that you can bring it up successfully.

```
SQL> Shutdown immediate;
```

```
SQL> STARTUP MOUNT
```

5. Edit the service_names parameters and the dispatchers parameters to the new values.

```
SQL> Alter system set service_names ='pfd3.domainname' scope=both;
```

```
SQL> Alter system set dispatchers=(PROTOCOL=TCP
```

```
(SERVICE=pfd3XDB)' scope=both;
```

6. Open the database.

```
SQL> ALTER DATABASE OPEN resetlogs
```

If the database needs more recovery before it can be opened, take the following steps.

- a. Find the location of the logfile that was created by the switch logfile command after taking the snapshot.
- b. Issue a recover database command using backup controlfile.
- c. Enter the path and name of the logfile in step a.
- d. Wait to receive the media recovery complete message.
- e. Enter the following command.

```
alter database open resetlogs
```

7. Add the temporary tablespace definitions. The following code is from the test case.

```
SQL> ALTER TABLESPACE TEMP ADD TEMPFILE '/u02/oradata/pfd3/
```

```
temp01.dbf' SIZE 20971520 REUSE AUTOEXTEND ON NEXT 655360
```

```
MAXSIZE 32767M;
```

Your TEMPFILE specifications will be different. Refer to the controlfile text file that was created in step 1 for the exact syntax.

8. Verify that the database is up and open for use. Run the following query.

```
SQL> select instance_name, host_name, database_status from v$instance;
```

INSTANCE_NAME	HOST_NAME	DATABASE_STATUS
pfdb3	HOST1	ACTIVE

9. (*Optional*) If you intend to access this database from outside of the server, you must edit the existing Listener.ora file to include the information about this new database.

NOTE To automatically start and stop this instance when invoking the Oracle dbstart and dbshut scripts, add an entry in the ORATAB file that specifies the following:

- The new instance
- The Oracle home
- Whether you want to start up and shut down the instance automatically

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



Cloning an Oracle Database to the Same Server
Using Snapshot and Volume Copy
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.