Technical Report

# NetApp Snapshot Technology Combined with Oracle dNFS CloneDB for Efficient Database Cloning

Brandon Hoang, NetApp
February 2012 | TR-4012

## TABLE OF CONTENTS

## LIST OF FIGURES

# 1 EXECUTIVE SUMMARY

IT organizations face the challenge of on-demand provisioning of database development and test environments. Database cloning and provisioning can be a demanding task that consumes considerable time and effort and a considerable amount of storage resources. In some cases, the cloning and provisioning stages occupy the majority of time spent in a development and test cycle, potentially affecting its overall efficiency and success. The frequency of database refreshes and the number of database images to be managed for development and test purposes are the other challenges in this scenario. Often, development and test databases must be refreshed regularly and many copies must be created and maintained with different datasets and versions in order to satisfy development and testing needs.

For these reasons, it is clear that the ability to quickly clone production databases and provision these cloned databases for development and test purposes has value to IT organizations. Customers who deploy Oracle® 11*g*™ Release 2 databases on NFS storage can leverage NetApp® Snapshot™ technology and the Oracle Direct NFS (dNFS) CloneDB feature to provide a simple, streamlined, and efficient database cloning solution.

# 2 INTRODUCTION

This document demonstrates the process of using NetApp Snapshot technology with Oracle dNFS CloneDB to enable the cloning of production databases for development and test use. dNFS CloneDB is a database solution that allows a database to be thinly cloned and provisioned by using a backup of the source database. The immediately resulting database files of the clone database consume very little space because only new blocks or updated blocks are written to the files of the clone database.

Oracle dNFS CloneDB relies on a backup of the source database to create the clone, and NetApp Snapshot technology is ideal for providing the needed backup. A NetApp Snapshot copy of a volume can be created in a second, regardless of the volume size, which makes it the perfect complementary solution to be deployed with Oracle dNFS CloneDB. The two technologies combine to deliver an extremely simple and space-efficient database cloning solution.

NetApp also offers FlexClone® technology, which is often used by NetApp customers to provision database clones by creating space-efficient clones of database volumes. With NetApp FlexClone technology, database volumes can be cloned in minutes if not seconds. From a database cloning technology perspective, it is necessary to understand the key difference between Oracle dNFS CloneDB and NetApp FlexClone. In simple terms, Oracle dNFS CloneDB is a host-based database cloning solution that leverages a backup image of the source database in which the backup image can be an RMAN backup or a non-RMAN backup, such as backups created using NetApp Snapshot copies. In contrast, NetApp FlexClone is a storage-based volume cloning technology that can be used to create space-efficient clones of volumes residing on NetApp storage, including database and application volumes.

The following topics are addressed in this report:

- Creating instant backup images of the production database using NetApp Snapshot copies
- Using Snapshot copies with Oracle Direct NFS CloneDB to thinly clone and provision databases required for development and test environments

## 2.1 AUDIENCE

This document is intended for NetApp customers and employees who want to leverage NetApp Snapshot copies with the Oracle Direct NFS CloneDB feature to rapidly clone and provision databases for development and testing. The target audience includes:

- Database administrators

- Data center managers
- Sales engineers
- Consulting sales engineers
- Professional services engineers
- Professional services consultants
- Contracted delivery partners
- Channel partner engineers

We assume that the reader has at least some knowledge of Oracle databases, the Oracle Direct NFS CloneDB feature, and NetApp storage and Snapshot technologies.

### 2.2 SCOPE

This document illustrates the use of NetApp Snapshot copies as a backup image of the source database to facilitate database cloning for development and testing by using the Oracle Direct NFS CloneDB feature. The emphasis is on the CloneDB feature of Oracle Direct NFS; therefore, descriptions of the architecture and configuration of the general Oracle Direct NFS client are outside the scope of this report.

# 3 SOLUTION OVERVIEW

The database cloning solution being discussed consists of two key components:

- Oracle Direct NFS CloneDB functionality
- NetApp Snapshot copies

### 3.1 ORACLE DIRECT NFS CLONEDB

Oracle Direct NFS CloneDB is a feature of the overall Oracle Direct NFS Client implementation, which is available in the Oracle Database 11.2.0.2 release. dNFS CloneDB is a database thin-cloning technology that leverages the backup of a source database to create a thinly provisioned clone database. dNFS CloneDB requires only an existing backup of the production database to perform the cloning, so it does not affect the performance and operation of the production environment.

dNFS CloneDB is based on copy-on-write technology, providing instant cloning with storage savings through the method of thin provisioning and sparse file implementation. A single backup image can be used to create multiple clones. dNFS CloneDB is available on all database server platforms.

The dNFS CloneDB feature is ideal for creating temporary or test environments to facilitate general development and testing needs, hardware and software upgrades and patching, and near-real-time query reporting.

For more information on dNFS CloneDB, see My Oracle Support (MOS) Note ID 1210656.1 - Clone Your DNFS Production Database for Testing.

### 3.2 NETAPP SNAPSHOT TECHNOLOGY

NetApp Snapshot technology is a point-in-time snapshot capability that offers unique advantages and is the foundation of many NetApp data protection solutions. NetApp Snapshot is a low-overhead, scalable, high-performance snapshot technology.

A Snapshot copy is a point-in-time file system image. A Snapshot copy operation involves only pointer manipulation and does not require any data block copy, so it is extremely time and space efficient. A NetApp Snapshot copy takes at most a few seconds to create—typically less than one second—regardless of the size of the volume or the level of activity on the NetApp storage system. This makes

NetApp Snapshot technology a very fast and efficient method for backing up Oracle databases, regardless of database size, with negligible or no impact on database performance.

For more information on NetApp Snapshot and storage technologies, go to www.netapp.com.

## 3.3 SOLUTION DESCRIPTION

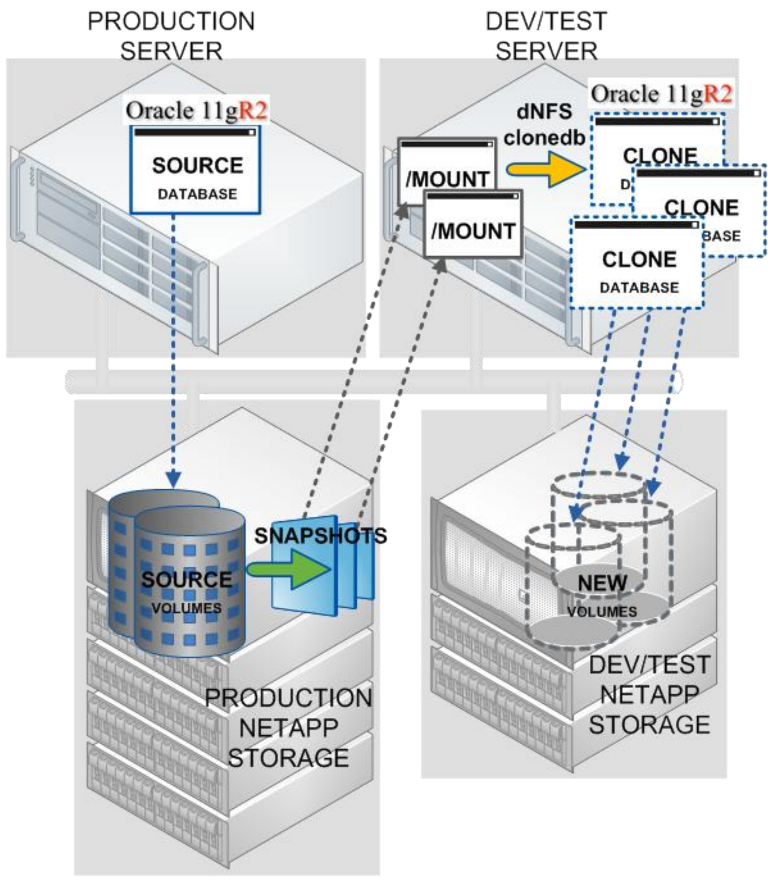
One primary requirement needed to use Oracle dNFS CloneDB for cloning is an existing backup of the source or production database. This backup can be in the form of an RMAN backup, an OS file copy, or a storage snapshot of the Oracle datafiles.

Because NetApp Snapshot technology can create a point-in-time copy in a few seconds or less, using it as the backup framework optimizes the solution, significantly reducing backup time and storage requirements. This benefit is especially apparent when taking a backup of an Oracle database in hot backup mode, which allows transactions to continue while the backup is running. Hot backup mode does, however, have some effect on application performance because additional redo data is recorded to provide data integrity during recovery. Therefore, it is important that the database hot backup process be completed as quickly as possible.

This solution uses NetApp Snapshot technology to create an instant backup of the database (Figure 1). The Snapshot copies of the source database volumes are taken while the database is placed into hot backup mode. The Snapshot copies are then mounted as read-only volumes on the development database server (by design, NetApp Snapshot copies are nonwritable). dNFS CloneDB is executed against the mounted backup volumes. The result of running the dNFS CloneDB program is two SQL scripts, which are used to create the clone database and new datafiles on designated volumes. Multiple clone databases can be created from the same backup volumes using the same procedures.

When the clone is first created, the datafiles of the clone database carry very few blocks. They are sparse files and are generally small compared to that of the backup image. As blocks are updated or new blocks are written in the clone database, the datafiles of the clone database grow. The size of the clone database is therefore determined by its own rate of change and not by the size of the source.

**Figure 1) Oracle Direct NFS CloneDB with NetApp Snapshot copies.**



## 4 SOLUTION IMPLEMENTATION

Cloning Oracle databases using Oracle Direct NFS CloneDB and NetApp Snapshot copies involves a set of simple procedures. The complete procedures are documented in individual steps for clarity. In summary, this section covers the following steps:

- Backing up a database using NetApp Snapshot copies
- Making the backup Snapshot copy available to the database server
- Enabling Oracle Direct NFS on the clone environment
- Preparing the initialization parameter file of the clone database
- Provisioning a volume for the clone database
- Working with Oracle Direct NFS CloneDB to generate required SQL scripts
- Executing SQL scripts to create the clone database
- Starting the clone database

### 4.1 BACKING UP A DATABASE USING NETAPP SNAPSHOT COPIES

A non-RMAN database backup is usually classified as either a cold backup or a hot backup. A cold backup, also known as an offline backup, refers to a backup that is taken when the database is down (or has been shut down). A hot backup, or online backup, can be performed while the database is open and available for use. To use hot backup mode, the database must be in ARCHIVELOG mode.

Hot backup mode is required for production environments because such databases cannot be shut down unless the backup is performed during planned downtime.

To take a hot backup of an Oracle database using NetApp Snapshot copies:

1. Assuming that the database is running in ARCHIVELOG mode, place the database in hot backup mode.

```
[oracle] $ sqlplus / as sysdba
SQL> alter database begin backup;
```

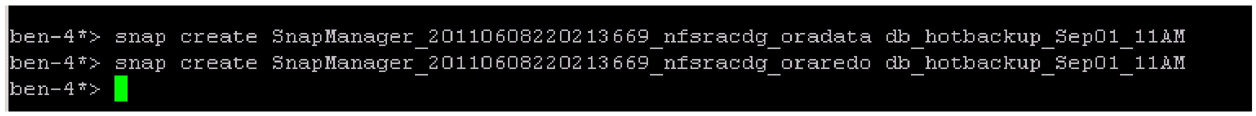
2. Create a Snapshot copy of the database's volumes, using either the CLI command on the storage system's console or the Web-based FilerView® tool interface. For example:

```
filerX> snap create <volume name> <snap name>
```

   Figure 2 is a screenshot showing the snap commands used to create the Snapshot copy of the database volumes.

**Note:** When hot backup mode is used to clone a database, a backup of redo logs and archived logs is required along with a backup of the datafiles, because the clone database needs access to the redo logs and archived logs for recovery during startup. However, a backup of temporary tablespaces is not required for cloning.

**Figure 2) Creating Snapshot copies on a NetApp storage system.**



```
ben-4*> snap create SnapManager_20110608220213669_nfsracdg_oradata db_hotbackup_Sep01_11AM
ben-4*> snap create SnapManager_20110608220213669_nfsracdg_oraredo db_hotbackup_Sep01_11AM
ben-4*>
```

3. End database hot backup mode after the volume Snapshot copies are created.

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

4. Archive all current logs.

```
SQL> alter system archive log current
```

## 4.2 MAKING THE BACKUP SNAPSHOT COPY AVAILABLE TO THE DATABASE SERVER

The backup Snapshot copies of the source database volumes can easily be made available by mounting them on the database server, as shown in Figure 1.

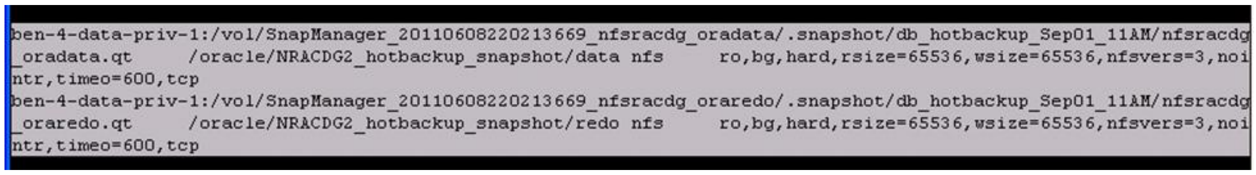
The Snapshot copies are nonwritable, point-in-time images of the volumes, so a Snapshot copy that is mounted is a read-only volume. Snapshot copies of a volume are maintained within the `.snapshot` directory of the volume itself. A Snapshot copy can be referenced as: `/vol/<volume name>/.snapshot/<snap name>`.

A Snapshot copy of a volume can be mounted as a read-only volume by specifying an entry in the `/etc/fstab` and executing the `mount` command. For example, a Snapshot copy can be mounted using the following mount options under Linux®.

```
ro,bg,hard,rsize=65536,wsize=65536,nfsvers=3,nointr,timeo=600,tcp
```

Figure 3 shows the `/etc/fstab` entries for two backup Snapshot copies that are being mounted as read-only volumes. In this particular example, the source volumes were created as qtrees; therefore, the mount device reference is in the format of `/vol/<volume name>/.snapshot/<snap name>/<qtree name>`.

Figure 3) Mount entries in the `/etc/fstab` file.



```
ben-4-data-priv-1:/vol/SnapManager_20110608220213669_nfsracdg_oradata/.snapshot/db_hotbackup_Sep01_11AM/nfsracdg
_oradata.qt     /oracle/NRACDG2_hotbackup_snapshot/data nfs     ro,bg,hard,rsize=65536,wsize=65536,nfsvers=3,noi
ntr,timeo=600,tcp
ben-4-data-priv-1:/vol/SnapManager_20110608220213669_nfsracdg_oraredo/.snapshot/db_hotbackup_Sep01_11AM/nfsracdg
_oraredo.qt     /oracle/NRACDG2_hotbackup_snapshot/redo nfs     ro,bg,hard,rsize=65536,wsize=65536,nfsvers=3,noi
ntr,timeo=600,tcp
```

## 4.3 ENABLING ORACLE DIRECT NFS ON THE CLONE ENVIRONMENT

Oracle Direct NFS must be enabled on the clone environment. This requirement applies strictly to the clone environment. The source environment can be a regular NFS client, OCFS, ASM, or Exadata. There is no restriction on the source environment as long as the backup is taken on an NFS volume.

To enable Oracle Direct NFS on the clone database environment, use the following commands:

```
$ cd $ORACLE_HOME/rdbms/lib
$ make -f ins_rdbms.mk dnfs_on
```

## 4.4 PREPARING THE INITIALIZATION PARAMETER FILE OF THE CLONE DATABASE

A new initialization parameter file is required for the clone database. It can be created from a copy of the source database's initialization parameter file or be created by the clonedb script. If the source database's initialization file is specified when executing the clonedb script, the clonedb script automatically generates an initialization file for the clone database that the user can edit further to satisfy the requirements of the clone environment.

If a copy of the source database's initialization parameter file is used, changes to some settings are necessary to make the initialization parameter file applicable for the clone database.

Here is an example of an initialization parameter file that has been modified to suit the clone environment.

```
*.audit_trail='DB'
*.cluster_database=FALSE
*.compatible='11.2.0.0.0'
control_files=/oracle/dnfs_clonedb/CLONEDB/CLONEDB_ctl.dbf
*.db_block_size=8192
db_name=CLONEDB
*.db_recovery_file_dest_size=4259315712
*.log_archive_format='%t_%s_%r.dbf'
*.log_archive_max_processes=30
*.memory_target=1073741824
*.open_cursors=100
*.processes=150
*.remote_login_passwordfile='EXCLUSIVE'
*.sessions=100
*.undo_management='AUTO'
*.undo_tablespace='UNDOTBS1'
db_create_file_dest=/oracle/dnfs_clonedb/CLONEDB
log_archive_dest=/oracle/dnfs_clonedb/CLONEDB
```

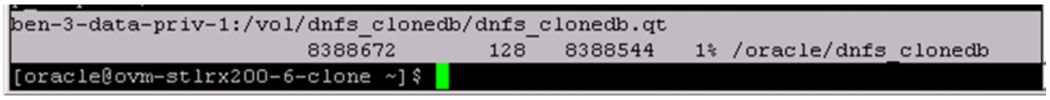
## 4.5 PROVISIONING A VOLUME FOR THE CLONE DATABASE

A new volume is needed for the clone database. To determine the size of the volume to host the datafiles of the clone database, you must consider the rate of change anticipated on the clone database.

Follow standard guidelines to create and provision a volume on NetApp storage. Once provisioned, the volume can be mounted on the clone database server. The mount options specified in mounting the volume are not critical, because during database startup Oracle Direct NFS client automates the mounting of the volume with the correct options.

Figure 4 shows an example of the provisioned and mounted volume for the clone database.

**Figure 4) Volume provisioned and mounted for a clone database.**



```
ben-3-data-priv-1:/vol/dnfs_clonedb/dnfs_clonedb.qt
                        8388672        128    8388544    1% /oracle/dnfs_clonedb
[oracle@ovm-stlrx200-6-clone ~]$
```

## 4.6 WORKING WITH ORACLE DIRECT NFS CLONEDB TO GENERATE REQUIRED SQL SCRIPTS

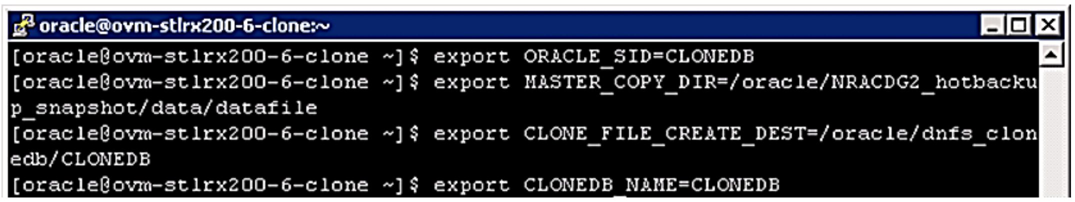
Oracle has made available a Perl script, clone.pl, which automates the generation of SQL scripts required to assist with the cloning process. You can download clone.pl by following the instructions provided in My Oracle Support (MOS) Note ID 1210656.1.

Before executing the clone.pl script, set the three required environment variables.

```
export MASTER_COPY_DIR=<directory containing the datafiles from the backup snapshot>
export CLONE_FILE_CREATE_DEST=<directory where the datafiles of clone database will be created>
export CLONEDB_NAME=<SID or name of clone database>
```

At this point, `ORACLE_SID` should also be set to the clone database SID, as shown in the example in Figure 5.

**Figure 5) Setting required environment variables for clone.pl script.**



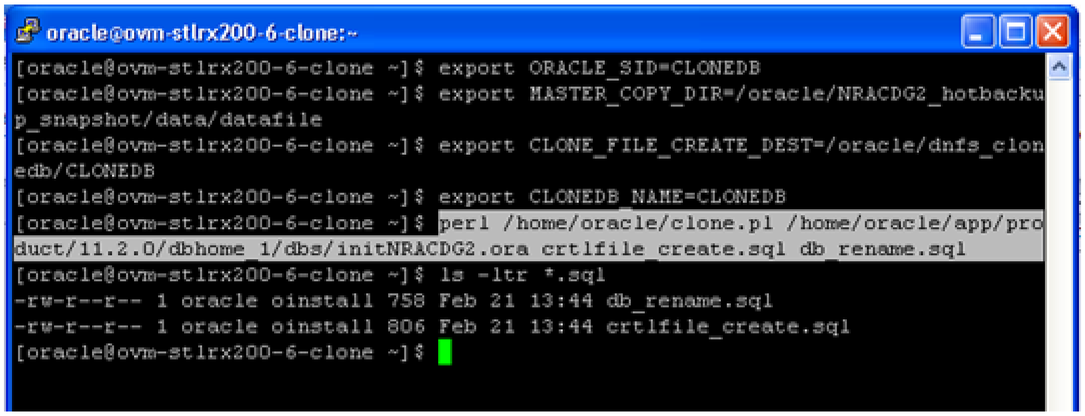
```
[oracle@ovm-stlrx200-6-clone ~]$ export ORACLE_SID=CLONEDB
[oracle@ovm-stlrx200-6-clone ~]$ export MASTER_COPY_DIR=/oracle/NRACDG2_hotbacku
p_snapshot/data/datafile
[oracle@ovm-stlrx200-6-clone ~]$ export CLONE_FILE_CREATE_DEST=/oracle/dnfs_clon
edb/CLONEDB
[oracle@ovm-stlrx200-6-clone ~]$ export CLONEDB_NAME=CLONEDB
```

Using Perl, execute the `clone.pl` script as follows (make sure that the Perl executable is in the path).

```
[oracle]$ perl <location of clone.pl> <location of source database init.ora> <filename
of control file create SQL script> <filename of db rename SQL script>
```

For example:

Figure 6) Running the clone.pl script.



```
[oracle@ovm-stlrx200-6-clone ~]$ export ORACLE_SID=CLONEDB
[oracle@ovm-stlrx200-6-clone ~]$ export MASTER_COPY_DIR=/oracle/NRACDG2_hotbacku
p_snapshot/data/datafile
[oracle@ovm-stlrx200-6-clone ~]$ export CLONE_FILE_CREATE_DEST=/oracle/dnfs_clon
edb/CLONEDB
[oracle@ovm-stlrx200-6-clone ~]$ export CLONEDB_NAME=CLONEDB
[oracle@ovm-stlrx200-6-clone ~]$ perl /home/oracle/clone.pl /home/oracle/app/pro
duct/11.2.0/dbhome_1/dbs/initNRACDG2.ora crtlfile_create.sql db_rename.sql
[oracle@ovm-stlrx200-6-clone ~]$ ls -ltr *.sql
-rw-r--r-- 1 oracle oinstall 758 Feb 21 13:44 db_rename.sql
-rw-r--r-- 1 oracle oinstall 806 Feb 21 13:44 crtlfile_create.sql
[oracle@ovm-stlrx200-6-clone ~]$
```

The execution of the clone.pl script generates two SQL scripts, a “control file create” SQL script and a “db file rename” SQL script.

## 4.7 EXECUTING SQL SCRIPTS TO CREATE A CLONE DATABASE

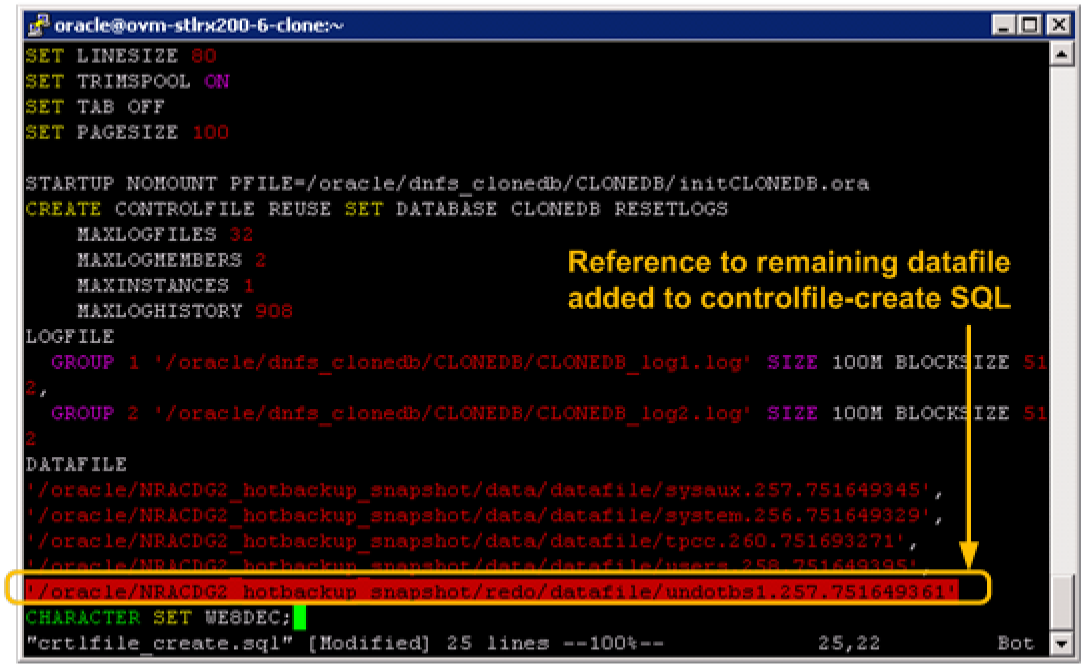
The two SQL scripts generated by running the clone.pl script can now be used to create the clone database.

Because the clone.pl script is designed to work with a single backup directory, the `MASTER_COPY_DIR` variable can reference only one directory. References to additional backup files that are part of the backup but reside in directories other than `MASTER_COPY_DIR` must be added to the SQL scripts. For instance, the source database might have datafiles residing on multiple volumes that require multiple NetApp Snapshot copies to be captured, one for each volume. The multiple Snapshot copies are mounted as volumes to provide the backup files. This results in a scenario in which multiple volumes provide the backup files.

**Note:** The alternative to adding backup volumes to the SQL scripts when you have multiple backup volumes is to first create a backup directory and then within the backup directory create soft links that reference the multiple backup volumes.

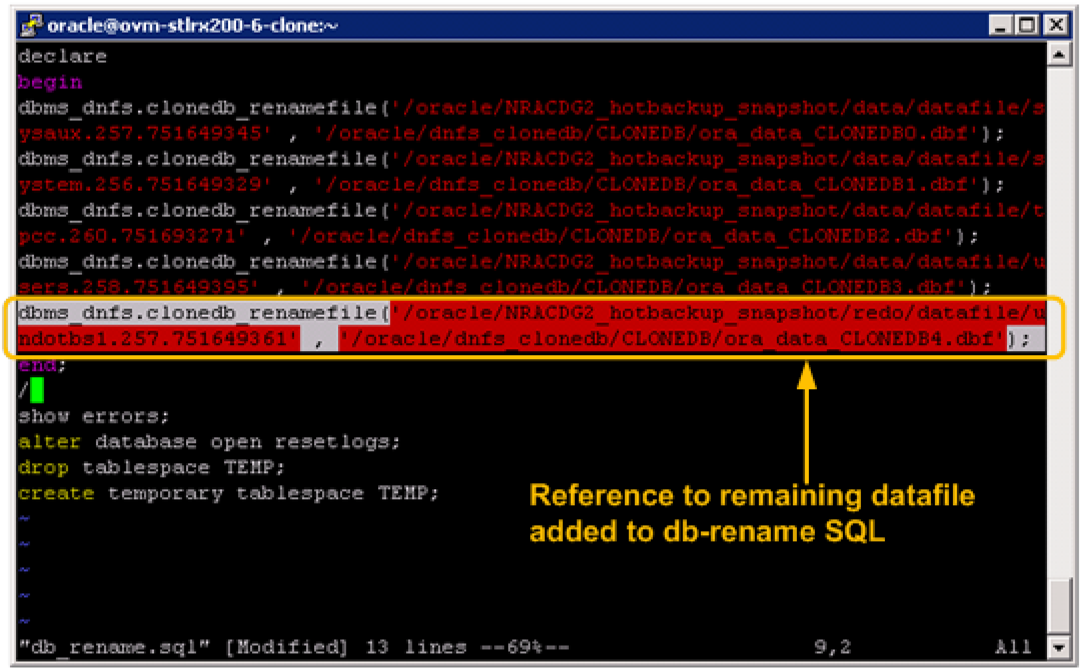
Figure 7 and Figure 8 demonstrate an example in which two backup volumes are involved. The `MASTER_COPY_DIR` variable is set to reference the location of the datafiles within the first backup volume. The remaining datafile, `undotbs1`, which is on the second backup volume, is not accounted for by the `clone.pl` script; thus, it is absent from both the `crtlfile_create.sql` and `db_rename.sql` SQL scripts. It must be added to both SQL scripts.

Figure 7) Datafile added to controlfile-create SQL.


oracle@ovm-stlrx200-6-clone:~
Reference to remaining datafile added to controlfile-create SQL

```
SET LINESIZE 80
SET TRIMSPOOL ON
SET TAB OFF
SET PAGESIZE 100

STARTUP NOMOUNT PFILE=/oracle/dnfs_clonedb/CLONEDB/initCLONEDB.ora
CREATE CONTROLFILE REUSE SET DATABASE CLONEDB RESETLOGS
    MAXLOGFILES 32
    MAXLOGMEMBERS 2
    MAXINSTANCES 1
    MAXLOGHISTORY 908
LOGFILE
  GROUP 1 '/oracle/dnfs_clonedb/CLONEDB/CLONEDB_log1.log' SIZE 100M BLOCKSIZE 51
2,
  GROUP 2 '/oracle/dnfs_clonedb/CLONEDB/CLONEDB_log2.log' SIZE 100M BLOCKSIZE 51
2
DATAFILE
'/oracle/NRACDG2_hotbackup_snapshot/data/datafile/sysaux.257.751649345',
'/oracle/NRACDG2_hotbackup_snapshot/data/datafile/system.256.751649329',
'/oracle/NRACDG2_hotbackup_snapshot/data/datafile/tpcc.260.751693271',
'/oracle/NRACDG2_hotbackup_snapshot/data/datafile/users.258.751649395',
'/oracle/NRACDG2_hotbackup_snapshot/redo/datafile/undotbs1.257.751649361'
CHARACTER SET WE8DEC;
"crtlfile_create.sql" [Modified] 25 lines --100%--              25,22          Bot
```

Figure 8) Datafile added to db-rename SQL.


oracle@ovm-stlrx200-6-clone:~
Reference to remaining datafile added to db-rename SQL

```
declare
begin
dbms_dnfs.clonedb_renamefile('/oracle/NRACDG2_hotbackup_snapshot/data/datafile/s
ysaux.257.751649345' , '/oracle/dnfs_clonedb/CLONEDB/ora_data_CLONEDB0.dbf');
dbms_dnfs.clonedb_renamefile('/oracle/NRACDG2_hotbackup_snapshot/data/datafile/s
ystem.256.751649329' , '/oracle/dnfs_clonedb/CLONEDB/ora_data_CLONEDB1.dbf');
dbms_dnfs.clonedb_renamefile('/oracle/NRACDG2_hotbackup_snapshot/data/datafile/t
pcc.260.751693271' , '/oracle/dnfs_clonedb/CLONEDB/ora_data_CLONEDB2.dbf');
dbms_dnfs.clonedb_renamefile('/oracle/NRACDG2_hotbackup_snapshot/data/datafile/u
sers.258.751649395' , '/oracle/dnfs_clonedb/CLONEDB/ora_data_CLONEDB3.dbf');
dbms_dnfs.clonedb_renamefile('/oracle/NRACDG2_hotbackup_snapshot/redo/datafile/u
ndotbs1.257.751649361' , '/oracle/dnfs_clonedb/CLONEDB/ora_data_CLONEDB4.dbf');
end;
/
show errors;
alter database open resetlogs;
drop tablespace TEMP;
create temporary tablespace TEMP;
~
~
~
~
~
"db_rename.sql" [Modified] 13 lines --69%--                    9,2           All
```

Start sqlplus as sysdba and execute the controlfile-create SQL script followed by the db-rename SQL script.

```
[oracle]$ sqlplus / as sysdba
SQL> @controlfile-create.sql
SQL> @db-rename.sql
```

The execution of the db-rename SQL script might fail at the `alter database open resetlogs` step. This is expected because the database backup was performed using hot backup mode, so media recovery is required.

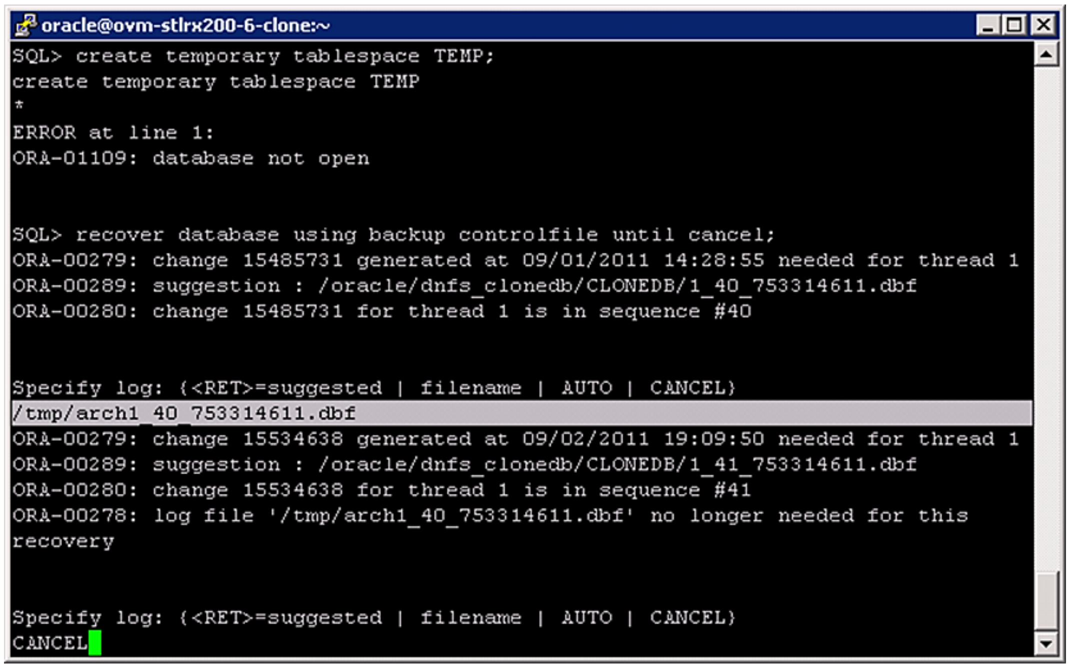
To perform media recovery, issue the following command.

```
SQL> recover database using backup controlfile until cancel;
```

Copy the appropriate archived log file from the source database to the database server clone and specify its location (Figure 9) to proceed with the recovery. As the recovery completes, specify `CANCEL` to exit.

For example:

**Figure 9) Specifying archived log for media recovery.**



```
oracle@ovm-stlrx200-6-clone:~
SQL> create temporary tablespace TEMP;
create temporary tablespace TEMP
*
ERROR at line 1:
ORA-01109: database not open


SQL> recover database using backup controlfile until cancel;
ORA-00279: change 15485731 generated at 09/01/2011 14:28:55 needed for thread 1
ORA-00289: suggestion : /oracle/dnfs_clonedb/CLONEDB/1_40_753314611.dbf
ORA-00280: change 15485731 for thread 1 is in sequence #40


Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
/tmp/arch1_40_753314611.dbf
ORA-00279: change 15534638 generated at 09/02/2011 19:09:50 needed for thread 1
ORA-00289: suggestion : /oracle/dnfs_clonedb/CLONEDB/1_41_753314611.dbf
ORA-00280: change 15534638 for thread 1 is in sequence #41
ORA-00278: log file '/tmp/arch1_40_753314611.dbf' no longer needed for this
recovery


Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
CANCEL
```

## 4.8 STARTING THE CLONE DATABASE

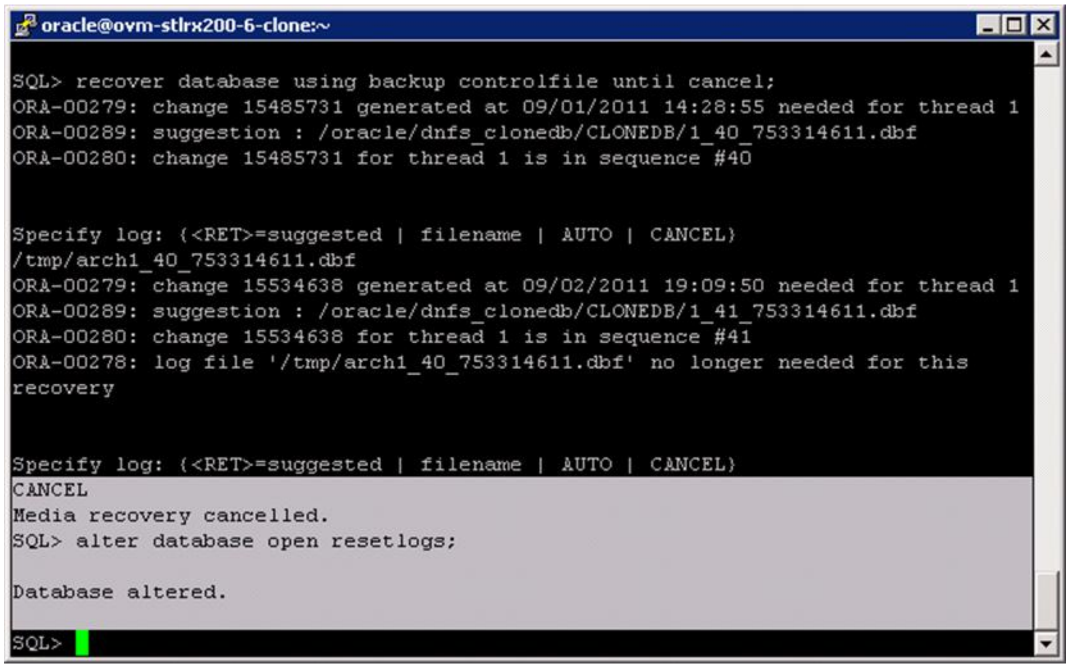
Upon the completion of media recovery, you can open the database.

Issue the following command to open the database.

```
SQL> alter database open resetlogs;
```

At this point, the clone database is open and is available for use.

Figure 10) Successfully opening the database after media recovery.

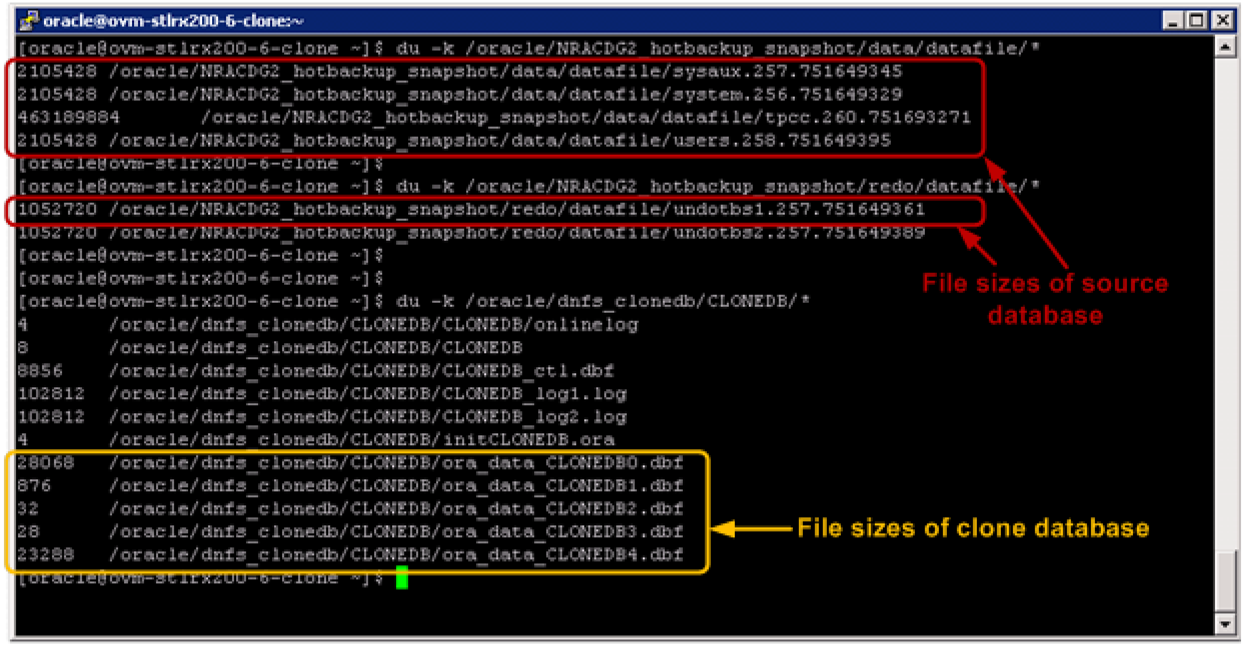


```
SQL> recover database using backup controlfile until cancel;
ORA-00279: change 15485731 generated at 09/01/2011 14:28:55 needed for thread 1
ORA-00289: suggestion : /oracle/dnfs_clonedb/CLONEDB/1_40_753314611.dbf
ORA-00280: change 15485731 for thread 1 is in sequence #40


Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
/tmp/arch1_40_753314611.dbf
ORA-00279: change 15534638 generated at 09/02/2011 19:09:50 needed for thread 1
ORA-00289: suggestion : /oracle/dnfs_clonedb/CLONEDB/1_41_753314611.dbf
ORA-00280: change 15534638 for thread 1 is in sequence #41
ORA-00278: log file '/tmp/arch1_40_753314611.dbf' no longer needed for this
recovery


Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
CANCEL
Media recovery cancelled.
SQL> alter database open resetlogs;

Database altered.

SQL>
```

If the clone database does not contain a temporary tablespace, you must create a default temporary tablespace.

The file sizes of the clone database are sparse, occupying minimal space compared to files of the source database.

Figure 11) File sizes of clone database.


oracle@ovm-stlrx200-6-clone:~
File sizes of source database
File sizes of clone database

```
[oracle@ovm-stlrx200-6-clone ~]$ du -k /oracle/NRACDG2_hotbackup_snapshot/data/datafile/*
2105428 /oracle/NRACDG2_hotbackup_snapshot/data/datafile/sysaux.257.751649345
2105428 /oracle/NRACDG2_hotbackup_snapshot/data/datafile/system.256.751649329
463189884       /oracle/NRACDG2_hotbackup_snapshot/data/datafile/tpcc.260.751693271
2105428 /oracle/NRACDG2_hotbackup_snapshot/data/datafile/users.258.751649395
[oracle@ovm-stlrx200-6-clone ~]$
[oracle@ovm-stlrx200-6-clone ~]$ du -k /oracle/NRACDG2_hotbackup_snapshot/redo/datafile/*
1052720 /oracle/NRACDG2_hotbackup_snapshot/redo/datafile/undotbs1.257.751649361
1052720 /oracle/NRACDG2_hotbackup_snapshot/redo/datafile/undotbs2.257.751649389
[oracle@ovm-stlrx200-6-clone ~]$
[oracle@ovm-stlrx200-6-clone ~]$
[oracle@ovm-stlrx200-6-clone ~]$ du -k /oracle/dnfs_clonedb/CLONEDB/*
4       /oracle/dnfs_clonedb/CLONEDB/CLONEDB/onlinelog
8       /oracle/dnfs_clonedb/CLONEDB/CLONEDB
8856    /oracle/dnfs_clonedb/CLONEDB/CLONEDB_ctl.dbf
102812  /oracle/dnfs_clonedb/CLONEDB/CLONEDB_log1.log
102812  /oracle/dnfs_clonedb/CLONEDB/CLONEDB_log2.log
4       /oracle/dnfs_clonedb/CLONEDB/initCLONEDB.ora
28068   /oracle/dnfs_clonedb/CLONEDB/ora_data_CLONEDB0.dbf
876     /oracle/dnfs_clonedb/CLONEDB/ora_data_CLONEDB1.dbf
32      /oracle/dnfs_clonedb/CLONEDB/ora_data_CLONEDB2.dbf
28      /oracle/dnfs_clonedb/CLONEDB/ora_data_CLONEDB3.dbf
23288   /oracle/dnfs_clonedb/CLONEDB/ora_data_CLONEDB4.dbf
[oracle@ovm-stlrx200-6-clone ~]$
```

## 5 CONCLUSION

The Oracle Direct NFS CloneDB feature offers customers a fast and efficient way of cloning production databases for development and test environments. Thin provisioning, instantaneous cloning, platform independence, and the ability to leverage a single backup image to create multiple clones are the primary benefits of Oracle Direct NFS CloneDB.

Combining Oracle Direct NFS CloneDB with NetApp Snapshot technology provides customers with an even better solution. Instantaneous, highly efficient Snapshot copies allow database backups to complete in seconds, regardless of database size. Reducing backup windows to seconds helps to greatly minimize any effect on production databases and increases the flexibility and frequency of backups, resulting in a more efficient cloning process.

## 6 REFERENCES

- My Oracle Support (MOS) Note ID 1210656.1—Clone Your DNFS Production Database for Testing
  support.oracle.com
- NetApp Snapshot Technology
  www.netapp.com/us/products/platform-os/snapshot.html

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 may 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 may be used solely in connection with the NetApp products discussed in this document.

Go further, faster®

NetApp®
www.netapp.com

© 2012 NetApp, Inc. All rights reserved. No portions of this document may be reproduced without prior written consent of NetApp, Inc. Specifications are subject to change without notice. NetApp, the NetApp logo, Go further, faster, FilerView, FlexClone, and Snapshot are trademarks or registered trademarks of NetApp, Inc. in the United States and/or other countries. Linux is a registered trademark of Linus Torvalds. Oracle is a registered trademark and Oracle 11*g* is a trademark of Oracle Corporation. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such.TR-4012-0212