

How to Copy an Oracle Database 12c Using Oracle Flash Storage LUN Copies

ORACLE WHITE PAPER | MARCH 2016





Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.



ORACLE®



Table of Contents

Disclaimer	1
Introduction	1
The Solution	2
Intended Audience	2
Software Used	3
Best Practices	3
LUN Layout	3
Database Layout	4
Disk Group Names	4
Database Copy Names	4
Prepare the Server for the Database Copies	4
Install Oracle Database Binaries	7
Task Workflow	8
Create Oracle Database Directories	9
Copy the Database	9
Rename the Oracle ASM Disks	11
Rename the Oracle ASM Disk Groups	13
Transfer the <code>pfile</code> From the Source Database	14
Rename the DATA, REDO, and TEMP Files	15
Rename the Database Copy	17
Conclusion	19
References	19





Introduction

This white paper presents a method of quickly creating copies of a running instance of Oracle Database 12c that you can use for test or development environments. This method shows you how to make consistent copies of your database while it is still running. You first copy the underlying LUNs and mount the copies to another server. Then you rename the database objects and the database copy itself to create another instance. LUN copies made with a Flash Storage (FS) system from Oracle allow you to create database copies in a few minutes with far less impact on your source database instance than copies made using Oracle Recovery Manager (Oracle RMAN).

Note: This white paper applies to both the Oracle FS1-2 System and the Oracle All Flash FS System. Throughout this document, the term “Oracle FS System” is used to apply to both types of systems.

When planning to create a copy of an Oracle Database 12c instance, consider the following points:

- » A fresh copy of data is the key to a successful data restoration. The key to fresh data is the ability to quickly create a copy of your database.
- » When copying your database, you must not impact current users.
- » If you plan to have multiple copies of a database on a server (for example, a test database and a development database, as shown in Figure 1), you will need to rename the diskgroups and databases so they can coexist.
- » The database copy might not need the same level of performance as the source.

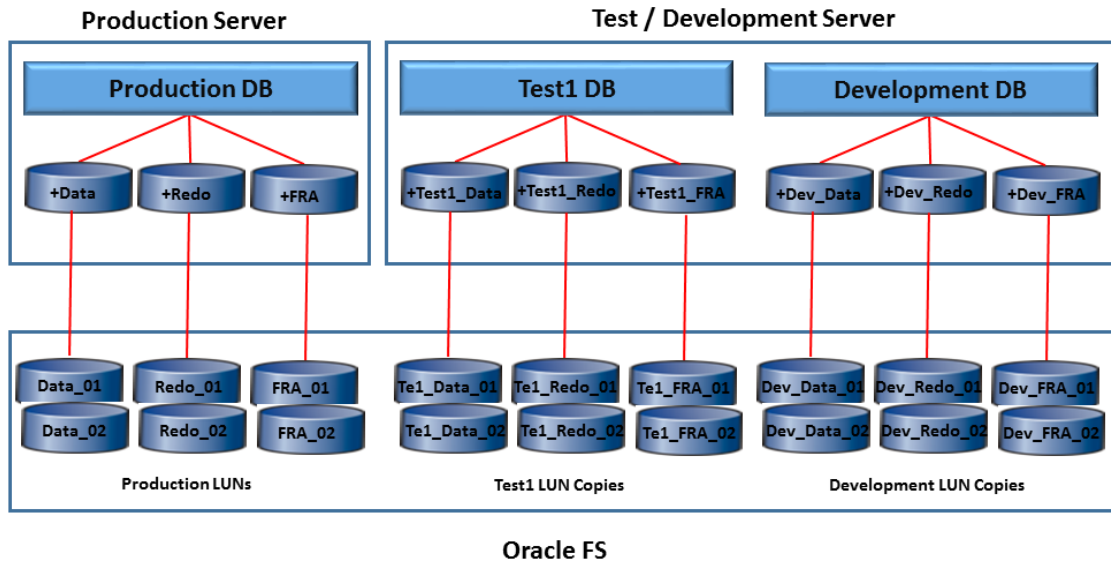


Figure 1. Multiple database copies on a single test/development server

The Solution

To copy an Oracle Database 12c instance, use an Oracle FS System to create LUN copies. By creating LUN copies, you reap the following benefits:

- » With database block-based tools such as Oracle RMAN, a database copy takes hours. Using an Oracle FS System to create LUN copies allows you to make a copy of your database, regardless of the size of the data, in minutes.
- » A copy of the source database is created while the database continues to run.
- » Most of the copy activity occurs on the Oracle FS System; therefore, the database servers are minimally affected.
- » Database copies are created with far less impact on the source database instance.
- » You can create a copy of the database with different quality of service (QoS) settings and types of disks. For example, the production database running on solid-state drives (SSDs) can be copied to a test database running on less expensive hard disk drives (HDDs). Optimizing QoS settings can result in significant cost savings and improved performance.

This white paper describes how to quickly create copies of a running Oracle Database 12c instance. Read this white paper to learn how to perform the following tasks:

- » Make consistent copies of your database while the database runs by copying the underlying LUNs.
- » Mount the LUN copies to another server.
- » Rename the database objects and the database copy itself to create another instance.

Intended Audience

This white paper is intended for Oracle DBAs and Oracle FS System storage administrators. You should be familiar with the following concepts:

- » Oracle database management system (DBMS) concepts
 - » Disk groups
 - » Database copies
 - » The Oracle Automatic Storage Management (Oracle ASM) feature of Oracle Database 12c
 - » Oracle Clusterware
- » Oracle FS System concepts
 - » LUN copies
 - » Clones
 - » Oracle FS Command Line Interface

Software Used

The examples in this white paper use the following versions of Oracle Linux 6, Oracle Database 12c, and Oracle FS System software:

- » Oracle Linux 6 with the ASMLib support library for Oracle ASM
- » Oracle Database 12c with Oracle ASM--note that Oracle Database 11g is supported, and most database-related steps are identical to those used with Oracle Database 12c
- » Oracle FS System software version 6.1 or higher
- » Oracle FS Command Line Interface

Best Practices

LUN Layout

For the purposes of this white paper, the following LUN layout and disk group layout are used. The design of the LUNs follows best practices for an Oracle Database instance running on an Oracle FS System. However, the Oracle Cluster Registry files must each be separated into their own disk groups (named +OCR in the following table). Each Oracle ASM instance to which databases are copied also has its own +OCR disk group, with the Oracle Cluster Registry files for that instance.

TABLE 1. LUN LAYOUT AND DISK GROUP LAYOUT

LUN Name	Controller	Disk Group Name	Redundancy	Purpose
DATA_01	1	+DATA	External	Data, Index, Temp, and Undo tablespaces
DATA_02	2			
REDO_01	1	+REDO	External	Online redo logs
REDO_02	2			
FRA_01	1	+FRA	External	Database fast recovery area (FRA): Archived redo logs, backups
FRA_02	2			
OCR_01	1	+OCR	External	Oracle Cluster Registry; the registry is not copied.

Database Layout

You can create copies of a running Oracle Database instance using the Oracle FS System LUN copies. You are not required to change the layout of your redo or data files, disk groups, or LUNs. Oracle ASM is not required, but it is used for all the examples in this white paper.

Best practices for laying out a database with multiple disk groups (+DATA, +REDO, +FRA) are used in this white paper. The best practices are described in [Best Practices for Optimizing Storage for Oracle Automatic Storage Management with Oracle FS1 Series Storage](#).

Important: Although any disk group layout works, all disk groups (except OCR and its underlying LUNs) must be copied at the exact same time.

Disk Group Names

If you plan to have multiple copies of a database on the same server, rename the Oracle ASM disks and disk groups. If you plan to have only one copy of your database on a server, this step is optional.

Database Copy Names

Oracle recommends that you change the name of your database copy. By changing the name, you can prevent mistakes, such as accidentally running a script against a production database when you thought it was running on a test database. You can use Oracle's DBNEWID utility (`nid`) to change the name of your database copy. Steps to change the name of the database copy are covered later in this white paper.

Prepare the Server for the Database Copies

The following configuration tasks prepare the Test/Development Server for the database copies.

1. Label the Oracle Cluster Registry LUN as an Oracle ASM disk using ASMLib.
 - a. Install ASMLib according to the instructions for your operating system.
 - b. Allocate a new Oracle Cluster Registry LUN and map it to your target server. The new Oracle Cluster Registry LUN is used to store the Oracle ASM configuration files for the Oracle ASM instance where you mount your database copies.
 - c. Ensure that the operating system recognizes the LUN. Consult the instructions for recognizing LUNs on your operating system. Typically, rebooting each node causes the operating system to recognize the new LUNs. Depending on your operating system, there could be less disruptive ways to recognize LUNs.
 - d. Label your Oracle Cluster Registry LUN as an Oracle ASM disk as follows:

```
[root@test-server]# oracleasm createdisk OCR /dev/dm-6
Writing disk header: done
Instantiating disk: done
```

2. Install Oracle Clusterware and Oracle ASM.

Perform an installation of Oracle Clusterware (if needed) and Oracle ASM on the target server.

Note: To create the first disk group, use the Oracle ASM disk labeled "OCR" in the previous step. Creating the first disk group ensures that the Oracle ASM and cluster configuration information is stored there.

3. Create the first disk group using the Oracle ASM Configuration Agent (Oracle ASMCA). To create the first disk group using Oracle ASMCA, complete the following steps:

- a. Set the disk discovery path appropriately for your environment.

Note: The default file path for ASMLib is `/dev/oracleasm/disks/*`.

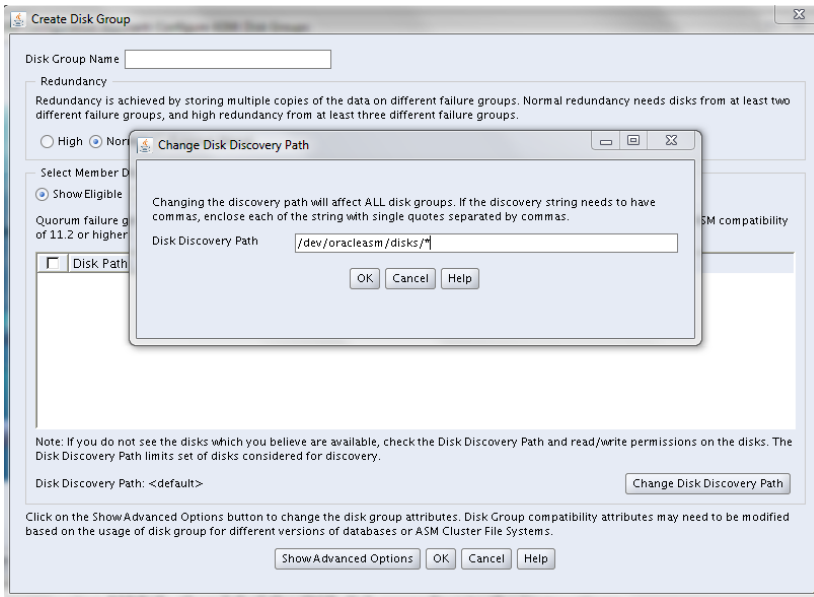


Figure 2. Setting the disk discovery path

- b. Select your Oracle ASM disk to be used for the OCR disk group.
- c. Enter a name in the **Disk Group Name** field.
- d. Because the Oracle FS System protects data using RAID, set the **Redundancy** value to **External (None)**.
- e. Click **OK**.

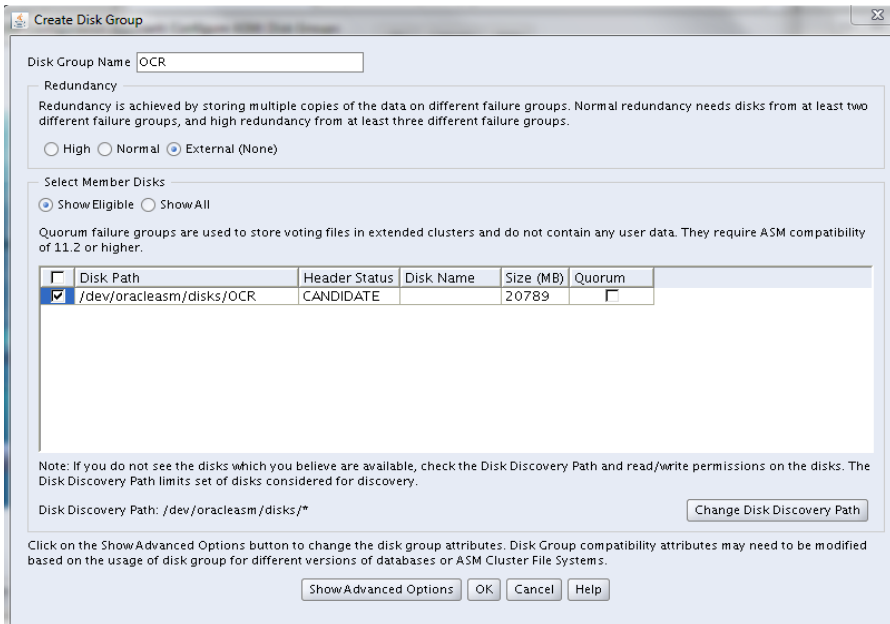


Figure 3. Setting the Redundancy value

Result: You have successfully created the Oracle ASM disk group named OCR.

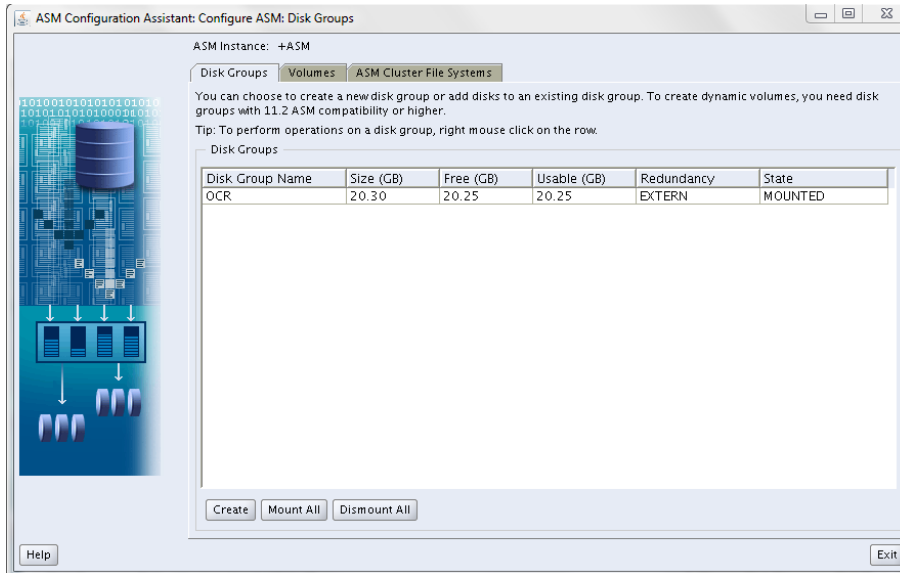


Figure 4. The result

Install Oracle Database Binaries

After installing Oracle ASM (and Oracle Clusterware, if you are using Oracle Real Application Clusters (Oracle RAC), select the **Install database software only** option. This option is found in Step 2 of the Oracle Database 12c Release 1 Installer (shown in Figure 5).

Note: Do not create a database now. The Oracle Database binaries are used to support the Oracle Database copies.

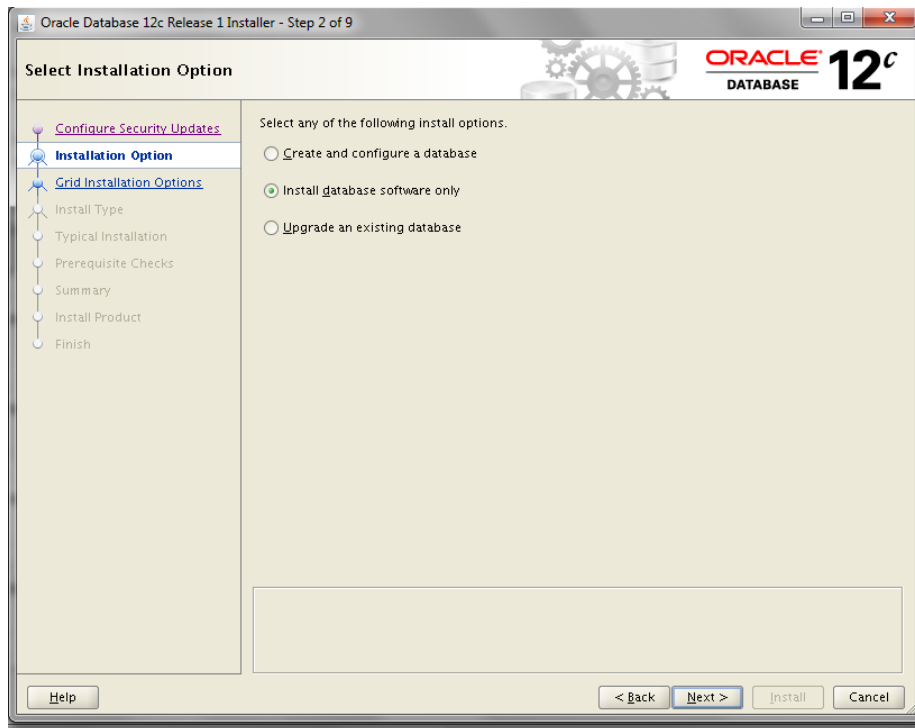


Figure 5. Installation Option — Install database software only

Task Workflow

After completing the preparation steps, perform the following tasks each time you create or refresh a copy of an Oracle Database 12c instance using LUN copies.

TABLE 2. TASK WORKFLOW TO COPY AN ORACLE DATABASE 12C INSTANCE USING LUN COPIES

Task	Tool Used to Perform Task
Create an Oracle Database directory for each database copy	Operating system commands
Copy the database: <ol style="list-style-type: none">1. Put the source database into hot-backup mode2. Create clones of the source LUNs3. Take the source database out of hot-backup mode4. Create LUN copies from the clones5. Delete the clones6. Ensure that the OS of each target server recognizes the LUNs	SQL *PLUS Oracle FS Command Line Interface SQL *PLUS Oracle FS Command Line Interface Oracle FS Command Line Interface Depends on your operating system
Configure the Oracle ASM disks: <ol style="list-style-type: none">1. Scan for new Oracle ASM disks2. List the Oracle ASM disks3. Rename the Oracle ASM disks	ASMLib on the test server, logged in as root
Configure the disk groups: <ol style="list-style-type: none">1. Rename the copied disk groups2. Mount the disk groups	renamedg utility as the Oracle ASM owner
Transfer the <code>pfile</code> from the source database and use it to mount the copy	SQL *PLUS
Change the paths of the redo, data, and temp files	SQL *PLUS
Rename the redo, data, and temp files	SQL *PLUS
Rename the database copy	SQL *PLUS

Create Oracle Database Directories

Create an Oracle directory for each instance of a database copy that you create. This task is required only the first time that you create each copy.

Note the following:

- » You must create the directories that the Oracle Database expects for each database copy, or the database copy will not start.
- » The directories that the Oracle Database expects have the `$ORACLE_SID` of your database copy. (Sometimes the directories use the SID in upper case or lower case.) In this example, the lower case SID is denoted as `$LC_NEW_ORACLE_SID`. For example, if your `$NEW_ORACLE_SID` is `TEST1`, then `$LC_NEW_ORACLE_SID` is `test1`.

After you create the Oracle Database directory, you are ready to refresh the database copy.

1. Set `$ORACLE_BASE` according to your installation.
2. Sign in to the Oracle Database 12c as the owner of the Oracle Database binaries (typically the `Oracle` user).
3. Create Oracle Database directories using the following commands.

```
export ORACLE_BASE=/u01/app/oracle
export NEW_ORACLE_SID=TEST1

export LC_NEW_ORACLE_SID=test1
mkdir -p $ORACLE_BASE/diag/rdbms/$LC_NEW_ORACLE_SID/$NEW_ORACLE_SID
mkdir -p $ORACLE_BASE/diag/rdbms/$LC_NEW_ORACLE_SID/$NEW_ORACLE_SID/alert
mkdir -p $ORACLE_BASE/diag/rdbms/$LC_NEW_ORACLE_SID/$NEW_ORACLE_SID/cdump
mkdir -p $ORACLE_BASE/diag/rdbms/$LC_NEW_ORACLE_SID/$NEW_ORACLE_SID/lck
mkdir -p $ORACLE_BASE/diag/rdbms/$LC_NEW_ORACLE_SID/$NEW_ORACLE_SID/stage
mkdir -p $ORACLE_BASE/admin/$NEW_ORACLE_SID/adump
```

Copy the Database

Complete the following tasks each time you create a copy or refresh a copy.

1. Put the source database into hot-backup mode.

The following example prepares the database for backup by putting the source database into hot-backup mode.

To perform this task, use the SQL `*PLUS` tool.

```
SQL> alter database begin backup;
Database altered.
```

2. Create clones of the source LUNs.

Before you perform this step, ensure that the clones have sufficient repository space.

All LUNs that are a part of the database must be copied at the same point in time. You can create a consistent collection of clones using the commands in the following example. The following commands prepare each clone, and then commit all clones at the same time.

To perform this task, use the Oracle FS Command Line Interface.

```
# fscli clone_lun -prepare -name CLONE_DATA_01 -source /MyHostGroup/DATA_01
```

```

# fscli clone_lun -prepare -name CLONE_DATA_02 -source /MyHostGroup/DATA_02
# fscli clone_lun -prepare -name CLONE_FRA_01 -source /MyHostGroup/FRA_01
# fscli clone_lun -prepare -name CLONE_FRA_02 -source /MyHostGroup/FRA_02
# fscli clone_lun -prepare -name CLONE_REDO_01 -source /MyHostGroup/REDO_01
# fscli clone_lun -prepare -name CLONE_REDO_02 -source /MyHostGroup/REDO_02

# fscli clone_lun -commit -cloneLun /CLONE_DATA_01,/CLONE_DATA_02,/CLONE_FRA_01,~/
/CLONE_FRA_02,/CLONE_REDO_01,/CLONE_REDO_02
Command Succeeded

```

3. Take the source database out of hot-backup mode.

To perform this task, use the SQL *PLUS tool.

```

SQL> alter database end backup;
Database altered

```

4. Create LUN copies from the clones.

The `clone_lun -copy` command creates a LUN by copying the contents and the settings of an existing clone.

To copy the LUN clones, use the Oracle FS CLI and enter the following commands:

```

# fscli clone_lun -copy -source /CLONE_DATA_01 -name COPY_DATA_01
# fscli clone_lun -copy -source /CLONE_DATA_02 -name COPY_DATA_02
# fscli clone_lun -copy -source /CLONE_FRA_01 -name COPY_FRA_01
# fscli clone_lun -copy -source /CLONE_FRA_02 -name COPY_FRA_02
# fscli clone_lun -copy -source /CLONE_REDO_01 -name COPY_REDO_01
# fscli clone_lun -copy -source /CLONE_REDO_02 -name COPY_REDO_02

```

These commands return in a few seconds and you can mount the LUN copies. The LUN copy process continues in the background. You can continue with all the following steps for copying the database. There might be a potential performance impact on the target database until the copy process is completed.

5. List the source clones.

After the full copies of the clones are created, you can delete the source clones to free up space in the Oracle FS System. Make sure that all LUN copies were created successfully before you delete the source clones.

To check the LUN copy status, use the Oracle FS Command Line Interface and enter the `task -list` command. The `task -list` command displays a list of the tasks that are currently running on the Oracle FS System.

The following example shows the copy tasks for the LUN clones that have not completed.

```

# fscli task -list

/CmBackgroundTypeVolumeCopyCmTask/151225/SYSTEM
Status: IN_PROCESS
/CmBackgroundTypeVolumeCopyCmTask/151109/SYSTEM
Status: IN_PROCESS
/CmBackgroundTypeVolumeCopyCmTask/151226/SYSTEM
Status: IN_PROCESS
/CmBackgroundTypeVolumeCopyCmTask/151228/SYSTEM
Status: IN_PROCESS
/CmBackgroundTypeVolumeCopyCmTask/151227/SYSTEM
Status: IN_PROCESS
/CmBackgroundTypeVolumeCopyCmTask/151107/SYSTEM
Status: IN_PROCESS
/CmBackgroundTypeVolumeCopyCmTask/151110/SYSTEM

```

```
Status: IN_PROCESS
/CmBackgroundTypeVolumeCopyCmTask/151111/SYSTEM
```

```
Status: IN_PROCESS
/CmBackgroundTypeVolumeCopyCmTask/151108/SYSTEM
Status: IN_PROCESS
```

When the copy operations have finished, the command returns `COMPLETED`, as shown in the following example.

```
# fscli task -list
/CmBackgroundTypeVolumeCopyCmTask/151225/SYSTEM
Status: COMPLETED
...
```

Caution: If you delete the clones before the LUN copies have finished, the clones are also deleted.

6. When you know the LUN copies completed successfully, delete the clones using the following command.

```
# fscli clone_lun -delete -cloneLUN ~
/CLONE_DATA_01,/CLONE_DATA_02,/CLONE_FRA_01,/CLONE_FRA_02,/CLONE_REDO_01,~
/CLONE_REDO_02
Command Succeeded
```

7. Ensure that the operating system of each target server recognizes the LUN.

Map the LUN copies to the target server. Consult the instructions for recognizing LUNs on your operating system. Typically, rebooting each node causes the operating system to recognize the new LUNs. Depending on your operating system, there could be less disruptive ways to recognize LUNs.

Rename the Oracle ASM Disks

Storage is added and removed from disk groups in units of Oracle ASM disks. To obtain optimal I/O performance, Oracle ASM disks should be independent of each other. With an Oracle FS System, you specify a LUN that is created from one or more physical disks. This LUN is presented to Oracle ASM as a single Oracle ASM disk.

1. Scan for new Oracle ASM disks.

The following example shows how to scan for new Oracle ASM disks using ASMLib.

```
[root@test-server]# oracleasm scandisks
Reloading disk partitions: done
Cleaning any stale ASM disks...
Scanning system for ASM disks...
```

2. List the Oracle ASM disks.

The LUN copies are presented to the target server and recognized as Oracle ASM disks. They contain the same names as the Oracle ASM disks on the source.

```
[root@test-server]# oracleasm listdisks
OCR
DATA_01
DATA_02
FRA_01
FRA_02
REDO_01
REDO_02
```

3. Rename the Oracle ASM disks.

To allow for multiple copies of databases on this server, rename the Oracle ASM disks as follows:

```
[root@test-server]# oracleasm renamedisk -f DATA_01 TEST1_DATA_01
Instantiating disk "TEST1_DATA_01": done
Writing disk header: done
Removing old ASM disk "DATA_01": done
[root@test-server]# oracleasm renamedisk -f DATA_02 TEST1_DATA_02
Instantiating disk "TEST1_DATA_02": done
Writing disk header: done
Removing old ASM disk "DATA_02": done
[root@test-server]# oracleasm renamedisk -f FRA_01 TEST1_FRA_01
Instantiating disk "TEST1_FRA_01": done
Writing disk header: done
Removing old ASM disk "FRA_01": done
[root@test-server]# oracleasm renamedisk -f FRA_02 TEST1_FRA_02
Instantiating disk "TEST1_FRA_02": done
Writing disk header: done
Removing old ASM disk "FRA_02": done
[root@test-server]# oracleasm renamedisk -f REDO_01 TEST1_REDO_01
Instantiating disk "TEST1_REDO_01": done
Writing disk header: done
Removing old ASM disk "REDO_01": done
[root@test-server]# oracleasm renamedisk -f REDO_02 TEST1_REDO_02
Instantiating disk "TEST1_REDO_02": done
Writing disk header: done
Removing old ASM disk "REDO_02": done
```

4. To verify the changes, list the Oracle ASM disks using ASMLib.

```
[root@test-server]# oracleasm listdisks
OCR
TEST1_DATA_01
TEST1_DATA_02
TEST1_FRA_01
TEST1_FRA_02
TEST1_REDO_01
TEST1_REDO_02
```


Rename the Oracle ASM Disk Groups

A disk group is one or more Oracle ASM disks that are managed as a logical unit. Oracle ASM rebalances the data to ensure an even I/O load to all disks in the disk group even as the disk group configuration changes.

1. Rename the copied disk groups.

If you plan to have multiple database copies on a server (for example, TEST1, TEST2, and DEV), you must rename the disk groups before mounting them to the server.

Note: This step is optional if you plan to have only one database copy.

The following example shows how to use the `renamedg` utility as the Oracle ASM owner (`grid`) to rename the `+FRA` disk group to `TEST1_FRA`.

The `renamedg` command takes the following parameters:

- » `dgname`: The existing disk group to rename (FRA).
- » `newdgname`: The new name to give the disk group (TEST1_FRA).
- » `asm_diskstring`: The discovery string that leads to your disks. `/dev/oracleasm/disks/*` is standard for ASMLib.
- » `confirm`: Automatically answers "Yes" to the "Are you sure?" prompt.
- » `verbose`: Specifies to give verbose output.

```
[grid@test-server]$ renamedg dgname=FRA newdgname=TEST1_FRA~
asm_diskstring=/dev/oracleasm/disks/* confirm=true verbose=true
Parsing parameters ...
```

Parameters in effect:

```
Old DG name      : FRA
New DG name      : TEST1_FRA
Phases           :
                  Phase 1
                  Phase 2
Discovery str    : /dev/oracleasm/disks/*
Confirm         : TRUE
Clean           : TRUE
Raw only        : TRUE
```

```
renamedg operation: dgname=FRA newdgname=TEST1_FRA
asm_diskstring=/dev/oracleasm/disks/* confirm=true verbose=true
Executing phase 1
Discovering the group
Performing discovery with string:/dev/oracleasm/disks/*
Identified disk UFS:/dev/oracleasm/disks/TEST1_FRA_01 with disk number:0 and
timestamp (33019274 2126130176)
Identified disk UFS:/dev/oracleasm/disks/TEST1_FRA_02 with disk number:1 and
timestamp (33019274 2126130176)
Checking for heartbeat...
Rediscovering the group
Performing discovery with string:/dev/oracleasm/disks/*
Identified disk UFS:/dev/oracleasm/disks/TEST1_FRA_01 with disk number:0 and
timestamp (33019274 2126130176)
Identified disk UFS:/dev/oracleasm/disks/TEST1_FRA_02 with disk number:1 and
timestamp (33019274 2126130176)
Checking if the diskgroup is mounted or used by CSS
Checking disk number:0
```

```

Checking disk number:1
Generating configuration file..
Completed phase 1
Executing phase 2
Looking for /dev/oracleasm/disks/TEST1_FRA_01
Modifying the header
Looking for /dev/oracleasm/disks/TEST1_FRA_02
Modifying the header
Completed phase 2
Terminating kgfd context 0x7f3af707f0a0

```

2. To rename the +DATA disk group to TEST1_DATA, and +REDO to TEST1_REDO, repeat the `renamedg` command.
3. Mount the disk groups.

The following example shows how to mount the +DATA, +REDO, and +FRA disk groups to the test server.

Note: Some columns have been removed to improve readability.

```

[grid@test-server][+ASM][~/home/grid]$ asmcmd
ASMCMDB> lsdg
State ... Total_MB ... Free_MB ... Name
MOUNTED ... 20789 ... 20789 ... OCR/

```

```

ASMCMDB> mount TEST1_DATA
ASMCMDB> mount TEST1_REDO
ASMCMDB> mount TEST1_FRA

```

```

ASMCMDB> lsdg
State ... Total_MB ... Free_MB ... Name
MOUNTED ... 20789 ... 20789 ... OCR/
MOUNTED ... 2097152 ... 1747045 ... TEST1_DATA/
MOUNTED ... 8389498 ... 8389248 ... TEST1_FRA/
MOUNTED ... 410074 ... 287072 ... TEST1_REDO/

```

Transfer the pfile From the Source Database

To perform this task, use the SQL *PLUS tool.

1. Log in to your source database and export the server parameter file (spfile) to a parameter file (pfile).
2. Transfer the pfile to the server where the database copy runs.
3. Change the disk group names.
4. To start the copy, use the pfile.

```

SQL> create pfile='/tmp/prod.pfile.txt' from spfile;
File created.

```

5. Transfer the pfile to the target server.

```

[oracle@prod-server]$ scp /tmp/prod.pfile.txt test-server:/tmp
oracle@test-server password:
prod.pfile.txt 100% 817 0.8KB/s 00:00

```

6. To reflect the new disk groups and the audit destination, modify the disk group names.

In the following example, parameters that do not require modification are omitted, and the changes are shown in bold.

```
[oracle@test-server]$ vi /tmp/prod.pfile.txt
*.audit_file_dest='/u01/app/oracle/admin/TEST01/adump'# Make sure disk group name
matches the new directory you created.
*.control_files='+TEST1_DATA/PROD/CONTROLFILE/current.261.879504783'
*.db_create_file_dest='+TEST1_DATA'
*.db_name='PROD'
*.db_recovery_file_dest='TEST1_+FRA'
*.service_names='PROD'
```

Rename the DATA, REDO, and TEMP Files

After you have Oracle ASM running with your disk groups, change the file paths in the database to point Oracle ASM to the new disk groups.

To perform these tasks, use the SQL *PLUS tool.

1. Mount the database using the production pfile.

```
[oracle@test-server]$ sqlplus / as sysdba

SQL*Plus: Release 12.1.0.2.0 Production on Thu Dec 3 15:47:16 2015

Copyright (c) 1982, 2014, Oracle. All rights reserved.

Connected to an idle instance.

SQL> startup mount pfile='/tmp/prod.pfile.txt';
ORACLE instance started.

Total System Global Area 1.0737E+11 bytes
  Fixed Size                7658152 bytes
  Variable Size             1.8254E+10 bytes
  Database Buffers          8.8852E+10 bytes
  Redo Buffers              260775936 bytes
Database mounted.

SQL> create spfile from pfile='/tmp/prod.pfile.txt';
File created.
```

2. Rename the DATA files.

```
SQL> select name from v$datafile;
NAME
-----
+DATA/PROD/DATAFILE/system.258.879504695
+DATA/PROD/DATAFILE/sysaux.257.879504653
+DATA/PROD/DATAFILE/users.259.879504739
+DATA/PROD/DATAFILE/undo.268.879511223

SQL> alter database rename file '+DATA/PROD/DATAFILE/system.258.879504695'~
to '+TEST1_DATA/PROD/DATAFILE/system.258.879504695' ;
Database altered.
```

```
SQL> alter database rename file '+DATA/PROD/DATAFILE/sysaux.257.879504653'~
to '+TEST1_DATA/PROD/DATAFILE/sysaux.257.879504653';
Database altered.
```

```
SQL> alter database rename file '+DATA/PROD/DATAFILE/users.259.879504739'~
to '+TEST1_DATA/PROD/DATAFILE/users.259.879504739';
Database altered.
```

```
SQL> alter database rename file '+DATA/PROD/DATAFILE/undo.268.879511223'~
to '+TEST1_DATA/PROD/DATAFILE/undo.268.879511223';
Database altered.
```

```
SQL> select name from v$datafile;
```

```
NAME
-----
+TEST1_DATA/PROD/DATAFILE/system.258.879504695
+TEST1_DATA/PROD/DATAFILE/sysaux.257.879504653
+TEST1_DATA/PROD/DATAFILE/users.259.879504739
+TEST1_DATA/PROD/DATAFILE/undo.268.879511223
```

3. Rename REDO log files.

```
SQL> select member from v$logfile;
MEMBER
-----
+REDO/PROD/ONLINELOG/group_1.450.883301211
+REDO/PROD/ONLINELOG/group_2.451.883301245
+REDO/PROD/ONLINELOG/group_3.452.883301277
```

```
SQL> alter database rename file '+REDO/PROD/ONLINELOG/group_1.450.883301211'~
to '+TEST1_REDO/PROD/ONLINELOG/group_1.450.883301211';
Database altered.
```

```
SQL> alter database rename file '+REDO/PROD/ONLINELOG/group_2.451.883301245'~
to '+TEST1_REDO/PROD/ONLINELOG/group_2.451.883301245';
Database altered.
```

```
SQL> alter database rename file '+REDO/PROD/ONLINELOG/group_3.452.883301277'~
to '+TEST1_REDO/PROD/ONLINELOG/group_3.452.883301277';
Database altered.
```

```
SQL> select member from v$logfile;
MEMBER
-----
+TEST1_REDO/PROD/ONLINELOG/group_1.450.883301211
+TEST1_REDO/PROD/ONLINELOG/group_2.451.883301245
+TEST1_REDO/PROD/ONLINELOG/group_3.452.883301277
```

4. Rename the TEMP files.

```
SQL> select name from v$tempfile;
```

```
NAME
```

```
-----  
+DATA/PROD/TEMPFILE/temp.264.879504785
```

```
SQL> alter database rename file '+DATA/PROD/TEMPFILE/temp.264.879504785'~  
to '+TEST1_DATA/PROD/TEMPFILE/temp.264.879504785';  
Database altered.
```

```
SQL> select name from v$tempfile;
```

```
NAME
```

```
-----  
+TEST1_DATA/PROD/TEMPFILE/temp.264.879504785
```

Rename the Database Copy

To perform this task, use the SQL *PLUS tool.

1. Recover and open the database.

```
SQL> recover database;  
Media recovery complete.
```

```
SQL> alter database open;  
Database altered.
```

2. Shut down the database and remount it.

```
SQL> shutdown immediate;  
Database dismounted.  
ORACLE instance shut down.
```

```
SQL> startup mount pfile='/tmp/prod.pfile.txt';  
ORACLE instance started.
```

```
Total System Global Area 1.0737E+11 bytes  
Fixed Size 7658152 bytes  
Variable Size 1.8254E+10 bytes  
Database Buffers 8.8852E+10 bytes  
Redo Buffers 260775936 bytes  
Database mounted.
```

```
SQL> exit  
Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -  
64bit Production  
With the Partitioning, Automatic Storage Management, OLAP, Advanced Analytics,  
and Real Application Testing options
```

3. To rename the database, use the nid utility.

```
[oracle@test-server]$ nid TARGET=SYS DBNAME=TEST1
DBNEWID: Release 12.1.0.2.0 - Production on Thu Dec 3 16:32:17 2015
Copyright (c) 1982, 2014, Oracle and/or its affiliates. All rights reserved.
Password:

Connected to database PROD (DBID=3038352206)
Connected to server version 12.1.0
Control Files in database:

+TEST1_DATA/PROD/CONTROLFILE/current.261.879504783
Change database ID and database name PROD to TEST1? (Y/[N]) => y
Proceeding with operation
Changing database ID from 3038352206 to 1274540500
Changing database name from PROD to TEST1
Control File +TEST1_DATA/PROD/CONTROLFILE/current.261.879504783 - modified
Datafile +TEST1_DATA/PROD/DATAFILE/system.258.87950469 - dbid changed, wrote
new name
Datafile +TEST1_DATA/PROD/DATAFILE/sysaux.257.87950465 - dbid changed, wrote
new name
Datafile +TEST1_DATA/PROD/DATAFILE/users.259.87950473 - dbid changed, wrote
new name
Datafile +TEST1_DATA/PROD/DATAFILE/undo.268.87951122 - dbid changed, wrote
new name
Datafile +TEST1_DATA/PROD/TEMPFILE/temp.264.87950478 - dbid changed, wrote
new name
Control File +TEST1_DATA/PROD/CONTROLFILE/current.261.879504783 - dbid
changed, wrote new name

Instance shut down

Database name changed to TEST1.

Modify parameter file and generate a new password file before restarting.
Database ID for database TEST1 changed to 1274540500.
All previous backups and archived redo logs for this database are unusable.
Database is not aware of previous backups and archived logs in Recovery Area.
Database has been shut down, open database with RESETLOGS option.
Successfully changed database name and ID.

DBNEWID - Completed successfully.
```

4. Modify the previous pfile with the following changes.

Note: Some parameters that do not require modification are omitted.

```
[oracle@test-server]$ vi /tmp/prod.pfile.txt

*.audit_file_dest='/u01/app/oracle/admin/TEST1/adump'
*.control_files='+TEST1_DATA/PROD/CONTROLFILE/current.261.879504783'
*.db_create_file_dest='+TEST1_DATA'
*.db_name='TEST1'
*.db_recovery_file_dest='+TEST1_FRA'
*.service_names='TEST1'

[oracle@test-server]$ sqlplus / as sysdba

SQL*Plus: Release 12.1.0.2.0 Production on Thu Dec 3 16:35:43 2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.

Connected to an idle instance.
```

```
SQL> startup mount pfile='/tmp/prod.pfile.txt';
ORACLE instance started.
```

```
Total System Global Area 1.0737E+11 bytes
Fixed Size                 7658152 bytes
Variable Size              1.8254E+10 bytes
Database Buffers          8.8852E+10 bytes
Redo Buffers               260775936 bytes
Database mounted.
```

```
SQL> create spfile from pfile='/tmp/prod.pfile.txt';
File created.
```

```
SQL> alter database open resetlogs;
Database altered.
```

5. Check that the database name has been changed.

```
SQL> select * from global_name;
```

```
GLOBAL_NAME
-----
TEST1
```

Conclusion

Using the Oracle FS System to create LUN copies allows you to quickly make a copy of your database and provides the following benefits:

- » You can make a copy of your database in minutes. A database copy taken with other database block-based tools, such as Oracle Recovery Manager (Oracle RMAN) or the Data Guard feature of the Oracle Database, takes hours.
- » Using the Oracle FS System to make LUN copies does not affect the users of the database. You do not need to shut down the database because the copy operations occur on the Oracle FS System.
- » Using the methods in this white paper, you can support multiple copies of your database on the same server.
- » You can assign unique QoS configuration settings to the LUNs. QoS configuration settings can increase or decrease performance, and can result in significant cost savings.

References

Refer to the following product documentation for information about the Oracle FS System and the Oracle Database.

[Oracle FS System documentation](#)

- » For information about LUN copies: *Oracle Flash Storage System Administrator's Guide*
- » For information about the LUN commands used in this white paper: *Oracle Flash Storage System CLI Reference*

[Oracle Database documentation](#)

- » For information about Oracle ASM: *Database Administrator's Guide*
- » For information about RAC Databases: *Oracle Real Application Clusters Administration and Deployment Guide*
- » For information about troubleshooting Oracle Database: *Oracle Database Backup and Recovery User's Guide*

[SQL *Plus documentation](#)

- » For information about SQL *Plus commands: *Oracle Database SQL Language Reference*



Oracle Corporation, World Headquarters

500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries

Phone: +1.650.506.7000
Fax: +1.650.506.7200

CONNECT WITH US

-  blogs.oracle.com/oracle
-  facebook.com/oracle
-  twitter.com/oracle
-  oracle.com


Hardware and Software. Engineered to Work Together

Copyright © 2016, 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.

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. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0216

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

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