ORACLE®
FLASH STORAGE
SYSTEMS

Oracle Database Protection Using Oracle Flash Storage System LUN Copies

ORACLE®

Table of Contents

Introduction to Backup and Recovery

As a backup administrator, your principal duty is to devise, to implement, and to manage a backup and recovery strategy. If data loss occurs, the purpose of a backup and recovery strategy is to protect a database against data loss and to reconstruct the database.

When implementing a backup and recovery strategy, you have the following solutions available:

- Recovery Manager (RMAN)

  RMAN is a built-in feature of the Oracle database server and does not require separate installation. RMAN is fully integrated with the Oracle database to perform a range of backup and recovery activities. The activities include maintaining an RMAN repository of historical data about backups. You can access RMAN through the command line or through Oracle Enterprise Manager.

- User-managed backup and recovery

  In this solution, you perform backup and recovery with a mixture of host operating system commands and SQL*Plus recovery commands. You are responsible for determining all aspects of when and how backups and recovery are done.

Source: *Oracle Database Backup and Recovery User's Guide*

For more information, refer to the following URL:
http://docs.oracle.com/cd/B10501_01/server.920/a96566/rcmintro.htm

"Choosing the best Oracle backup strategy can be a difficult decision. Oracle data backup choices include user-managed backups, Oracle Recovery Manager (RMAN), and import/export backups. Depending on your needs and your budget, all three Oracle backup strategies could be suitable for your environment."

-– W. Curtis Preston, executive editor at Tech Target and independent backup expert

As a user-managed backup and recovery solution, the Oracle FS System drastically reduces recovery time. The Oracle FS System takes a complete copy of your database without any downtime. The backups can be recovered quickly because they are mounted directly to the server and the backups are not subject to a lengthy restoration process. In addition, the database server is unaffected by the backups, because they run in the background on the Oracle FS System.

This white paper describes the following activities:

- How to configure your database and Oracle FS System for generating consistent copies of database LUNs and archived redo logs

- How to recover your database quickly while meeting the requirements for minimal data loss.

NOTE: This white paper applies to both the Oracle FS1-2 System and the Oracle All Flash FS System. Throughout this document, the product will generically be referred to as the Oracle FS System.

Use Case Scenario

In this section, we use a real example to understand the risks of using RMAN to back up a large database. Currently, the data is backed up once a week using an RMAN full backup, complemented by daily, incremental backups. So what happens in the catastrophic event of a complete loss of the database?

- The customer must be able to recover the database and to resume operations based on the Recovery Time Objective (RTO). The RTO is typically defined as the maximum amount of time that an IT-based business process can be down before the organization suffers significant material losses.

- The customer must not lose any more data based on the Recovery Point Objective (RPO). The RPO is typically defined as the maximum amount of data an IT-based business process could lose before causing harm to the organization.

In a cluster environment, the cluster comprises multiple interconnected computers or servers that appear to end users and applications as though they are one server. Oracle Real Application Clusters (RAC) enable you to cluster Oracle databases. Oracle RAC uses Oracle Clusterware for the infrastructure to bind multiple servers so that the servers operate as a single system.

In this example, the organization has a 1 TB Oracle 12c database that operates on an Automatic Storage Management (ASM) cluster. The cluster consists of two nodes, as shown in Figure 1.
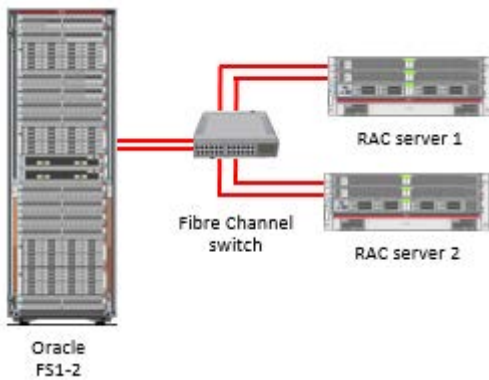

Figure 1. ASM-based RAC database with two nodes

The Problem

With a database of 1 TB, a full RMAN restoration of a database takes four hours or longer. Not only is the time required for full or incremental restore operation problematic, but performance is impacted as well. The weekly, full RMAN backup or incremental backups cause a noticeable slowdown of the database server.

The Solution

Use the Oracle FS System to create consistent LUN copies. With this solution, you can effectively utilize full copies of all database LUNs within minutes. There is no impact to the performance of the database servers.

The following steps for this solution are detailed in this white paper:

1. Create Clone LUNs to obtain a consistent copy of all database LUNs.

2. Create a LUN Copy to remove the dependency on the original LUN.

3. Mount the LUN copies to the database server.

4. Roll forward the archived redo logs to the current database.

Before you begin, review the following information about the key data protection components that are used in this solution. Key components include Clone LUNs, LUN copies, ASM disk groups, and redo logs.

Data Protection Concepts

Clone LUNs

In the Oracle FS System, a Clone LUN is a point-in-time, read-write, space-efficient snapshot of a source LUN. A Clone LUN can be accessed immediately

When the data in the source LUN changes, the original data is captured and stored in the Clone LUN. A clone is dependent on the source LUN for the data that has not been changed.

When you copy a Clone LUN, there is no dependency between the copy of the database LUNs and the original database LUNs. If the Clone LUN was used and the original LUN was lost, the Clone LUN would also be lost. Copying the Clone LUN avoids that. You copy a Clone LUN because you do not want to recover any changes made subsequently to the source LUN.

Important: Make sure that the clone repository for the source LUN does not consume the maximum capacity allocated in the repository. To allocate more clone capacity, monitor the amount of free capacity that remains and modify the source LUN. If a clone requires more space capacity from the repository, but space is not available, the Oracle FS System first takes the clone offline and then generates a system alert.

Deleting a source LUN or a Clone LUN affects the associated dependent clones:

- Deleting a source LUN also deletes any clones of the LUN.

- Deleting a Clone LUN retains any clones of the Clone LUN.

Refer to the *Oracle Flash Storage System Administrator's Guide* for complete information about Clone LUNs and LUN copies.

You can find this document at the following URL: http://docs.oracle.com/en/storage/#fla

Click the Oracle Flash Storage System Documentation Library link and locate the document in the System Management section.

Automatic Storage Management

Automatic Storage Management (ASM) provides a vertical integration of the file system and a volume manager that is built for Oracle database files. ASM distributes I/O load across all available resources to optimize performance. Load balancing removes the need for manual I/O tuning by spreading out the database files to avoid hotspots. ASM helps you manage a dynamic database environment. You can increase the size of a database without having to shut down the database to adjust the storage allocation.

ASM Disk Groups

A disk group is one or more ASM disks that are managed as a logical unit. ASM disks can be added or dropped from a disk group while the database is running. 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. Since the Oracle FS System uses RAID to protect the data on each ASM disk, external redundancy should be used.

ASM Disks

Storage is added and removed from disk groups in units of ASM disks. To obtain optimal I/O performance, 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 ASM as a single ASM disk.

To learn more about ASM, click the following URL: http://docs.oracle.com/database/121/OSTMG/GUID-24A04857-F691-433D-AC74-F7C999BE87B1.htm#OSTMG94061

Redo Logs

Redo logs protect database data in memory if the database data has been altered but has not yet been written to permanent storage. In `archivelog` mode, the Oracle database copies filled online redo logs to disk. The database must be in archive log mode for this procedure to work.

Redo logs are critical to the integrity of the Oracle database. To learn more about Redo logs, click the following URL: https://docs.oracle.com/cloud/latest/db121/ADMIN/onlineredo.htm#ADMIN007

Storage Domain, LUN, and ASM Layout

Oracle recommends placing LUNs for different ASM disk groups into different Storage Domains. Different Storage Domains isolate the effects of the unlikely loss of an Oracle Flash Storage System drive group. They also help to prevent disks in one disk group from impacting the performance of another disk group. At a minimum, separate the LUNs for the following disk groups into different Storage Domains.

- +DATA
- +FRA
- +ARC_LOGS

For the +FRA and +ARC_LOGS diskgroups, separating the LUNs into separate Storage Domains ensures that the needed Archived Redo Logs are available to roll forward during recovery.

When the redo logs are archived, they are written to the FRA Storage Domain diskgroup. The best practice is to place the FRA on its own ASM disk group. The archived redo logs are critical to your ability to meet your RPO. Keep at least one other copy of the redo logs in a safe, alternate location (+ARC_LOGS in this example).

NOTE: The LUN sizes in the following table are used for this example only. LUN sizes vary according to your installation.

Table 1. Storage Domain, LUN, and ASM Disk Group Layout

Storage domain	LUN name	LUN size	ASM disk group	Purpose
Data	OCR	20 GB	+OCR	Oracle Cluster Registry
	DATA_01	1500 GB	+DATA	Data and index tablespaces
	DATA_02	1500 GB		
	TEMP_01	1000 GB	+TEMP	Temporary tablespaces
	TEMP_02	1000 GB		
Redo	REDO_01	200 GB	+REDO	Online redo logs
	REDO_02	200 GB		
FRA	FRA_01	500 GB	+FRA	Fast Recovery Area is the primary location for archived redo logs
	FRA_02	500 GB		
Arc Logs	ARC_LOGS_01	500 GB	+ARC_LOGS	Secondary location for archived redo logs
	ARC_LOGS_02	500 GB		

Protecting an Oracle Database

The examples in this section describe how to create a consistent copy of all the LUNs in the database. You must also preserve a copy of the current archived redo logs, which are applied to the current database during recovery. This copy is stored in the +ARC_LOGS diskgroup. The best practice is to create a consistent copy of all the LUNs daily or according to the targeted RTO.

Important: The less frequently you make a copy of the LUNs, the more you are required to roll forward the redo logs, which extends recovery time. The following example makes a consistent copy of the database LUNs using the Oracle FS System and puts the database into hot-backup mode.

Note: As a best practice, Oracle recommends scripting and testing the FSCLI and SQL *PLUS commands to ensure accuracy.

1. Put the database into hot-backup mode. This action prepares the database for backup.

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

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

2. Create a collection of LUN clones. Before you perform this step, ensure that the Clone LUNs have sufficient repository space.

To perform this step, use the Oracle Flash Storage CLI.

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 LUN clones using the commands in the following example. The following commands provide a list of available LUNs to clone, prepare each clone, and then commit all clones at the same point in time.

```
# fscli lun –list
…
# fscli clone_lun -prepare -name CLONE_DATA_01 –source /MyHostGroup/DATA_01
    NewObject
Id: 414B303031393639A1045A24C2537D84
Fqn: /CLONE_DATA_01
ObjectType: PrepareLunClone

Command Succeeded

# 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_OCR     -source /MyHostGroup/OCR
…
# 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 -prepare -name CLONE_TEMP_01 -source /MyHostGroup/TEMP_01
…
# fscli clone_lun -prepare -name CLONE_TEMP_02 -source /MyHostGroup/TEMP_02
…
# fscli clone_lun -commit –cloneLUN~
/CLONE_DATA_01,/CLONE_DATA_02,/CLONE_FRA_01,/CLONE_FRA_02,/CLONE_OCR,/CLONE_REDO_0
~ 1,/CLONE_REDO_02,/CLONE_TEMP_01,/CLONE_TEMP_02
Command Succeeded
```

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

   The following example takes the 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 clones.

   Use the following command to initiate the copying of the LUN clones. These copies are used to recover the database in the event of an outage.

   To perform this step, use the Oracle Flash Storage CLI.

   Note: The clone_lun –copy command creates a LUN by copying the contents and the settings of an existing Clone LUN.

```
# fscli clone_lun -copy -source /CLONE_DATA_01 -name COPY_DATA_01
  NewObject
Id:  414B303031393639A1045BD151C33C6F
Fqn: /COPY_DATA_01
```

```
ObjectType: lun
Command Succeeded
# 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_OCR     -name COPY_OCR
…
# fscli clone_lun -copy -source /CLONE_TEMP_01 -name COPY_TEMP_01
…
# fscli clone_lun -copy -source /CLONE_TEMP_02 -name COPY_TEMP_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
…
```

The amount of time it takes to complete full copies of the LUN clones depends on how much data must be copied. After the commands finish running, you can mount the LUN copies. While the copy operation continues to run in the background, you can access the LUN copies before the copy is finished.

Do not make copies of the +ARC_LOGS secondary archived redo logs destination. In case a recovery is required, you want to have the most recently archived redo logs available.

5. Delete the LUN clones.

    After the full copies of the LUN clones are complete, you can delete the source clones to free up space in the Oracle FS System. To check the LUN copy status, use the following command. Make sure that all LUN copies were created successfully before you delete the clones. In the following example, the copy tasks for the LUN clones have not completed.

    To perform this step, use the Oracle Flash Storage CLI.

    Note: The `task -list` command displays a list of the tasks that are currently running on the Oracle FS System.

```
# 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.

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

Once the LUN copies are complete, delete the LUN clones using the following command.

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

```
# fscli clone_lun -delete -cloneLUN ~
/CLONE_DATA_01,/CLONE_DATA_02,/CLONE_FRA_01,/CLONE_FRA_02,/CLONE_OCR,/CLONE_REDO_0
~1,/CLONE_REDO_02,/CLONE_TEMP_01,/CLONE_TEMP_02
Command Succeeded
```

Preserve the Archived Redo Logs

Consider the following points when managing the archived redo logs:

- Ensure that the recent redo logs are archived so you can use them in the event of an outage.

- Store a copy of the archived redo logs in a different location than the rest of the database LUNs.

- Archive the redo logs to meet your RPO. If you lose all database LUNs and must restore them, you can only roll forward to the point that the last redo log was archived.

Set the ARCHIVE_LAG_TARGET Parameter

The following example sets the ARCHIVE_LAG_TARGET parameter so that the database automatically archives the redo logs every 3600 seconds (60 minutes).

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

```
SQL> alter system set ARCHIVE_LAG_TARGET=3600 scope=both sid='*';
System altered.
```

You are not required to restart the database.

1. Safely store the archived redo logs.

    When the redo logs are archived, they are written to the DB_RECOVERY_DEST, which is located in the FRA. The best practice is to place the FRA on its own ASM disk group. Because the archived redo logs are critical to your ability to meet your Recovery Point Objective (RPO), keep at least one other copy of the redo logs in a safe, alternate location.

    Oracle provides an easy way to store an extra copy of the redo logs by configuring a second Archive Log Destination (LOG_ARCHIVE_DEST_2). Each time a redo log is archived, the log is copied first to LOG_ARCHIVE_DEST_1 and then to LOG_ARCHIVE_DEST_2 in sequence.

2. Ensure that the secondary location for the archived redo logs meets these requirements:

- The secondary location for the archived redo logs is not located with any of the other database files or archived redo logs.

- The secondary location for the archived redo logs is available to every RAC node.

This second Archive Log Destination can be on a shared file system that is mounted to each RAC node or in another disk group.

Creating a Separate Diskgroup

In this example, we will use a separate diskgroup called `+ARC_LOGS`.

Configure the database using the following sequence of steps. To perform this task, use the SQL *PLUS tool.

1. Confirm where the archived redo logs are currently written.

```
SQL> show parameter db_recovery_file_dest

NAME                                 TYPE        VALUE
------------------------------------ ----------- ----------------------------
db_recovery_file_dest                string      +FRA
```

2. Set the primary LOG_ARCHIVE_DEST_1 to use the DB_RECOVERY_FILE_DEST (+FRA).

```
SQL> alter system set ~ log_archive_dest_1='LOCATION=USE_DB_RECOVERY_FILE_DEST'~
scope=both sid='*';
System altered.
```

3. To use the new diskgroup, set the secondary `LOG_ARCHIVE_DEST_2`.

Create a diskgroup named `+ARC_LOGS` and ensure that the diskgroup is large enough to accommodate the setting for `DB_RECOVERY_FILE_DEST_SIZE` that you have configured for your database. Now set the secondary Archive Log Destination:

```
SQL> alter system set log_archive_dest_2='LOCATION=+ARC_LOGS' scope=both~
sid='*';
System altered.
```

Confirming Your Settings

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

1. ```
SQL> show parameter db_recovery_file_dest

NAME   TYPE   VALUE
------------------------------------ ----------- ----------------------------
db_recovery_file_dest   string   +FRA
```

2. ```
SQL> show parameter log_archive_dest_1

NAME                                 TYPE      VALUE
------------------------------------ -------- ----------------------------
log_archive_dest_1                   string   LOCATION=USE_DB_RECOVERY_FILE_DEST
```

3. `SQL>` **`show parameter log_archive_dest_2`**

```
NAME                                 TYPE          VALUE
-------------------------------- ----------- -----------------------------
log_archive_dest_2               string      LOCATION=+ARC_LOGS
```

RESULT: The database is configured so that the redo logs are archived every 60 minutes to two different locations.

Recovering the Oracle Database

The examples in this section describe how to recover the original database from the LUN copies created when you created LUN copies from clones. Recover all LUN copies together, because they are treated as a set.

1. Disable the data path to the RAC nodes for source LUNs.

    To disable the data path, use the following command for the original LUNs so that the copies can be mapped for recovery.

    To perform this step, use the Oracle Flash Storage CLI.

    ```
    # fscli lun -modify -lun /MyHostGroup/DATA_01 -inactive
      NewObject
      Id: 414B303031393639A10451E0D40BD047
      Fqn: /MyHostGroup/DATA_01
      ObjectType: lun
    Command Succeeded
    # fscli lun -modify -lun /MyHostGroup/DATA_02 -inactive
    …
    # fscli lun -modify -lun /MyHostGroup/FRA_01 -inactive
    …
    # fscli lun -modify -lun /MyHostGroup/FRA_02 -inactive
    …
    # fscli lun -modify -lun /MyHostGroup/OCR -inactive
    …
    # fscli lun -modify -lun /MyHostGroup/REDO_01 -inactive
    …
    # fscli lun -modify -lun /MyHostGroup/REDO_02 -inactive
    …
    # fscli lun -modify -lun /MyHostGroup/TEMP_01 -inactive
    …
    # fscli lun -modify -lun /MyHostGroup/TEMP_02 -inactive
    …
    ```

2. Map the LUN copies to the RAC nodes.

    To map the LUN copies to the RAC nodes for recovery, use the following command set.

    To perform this step, use the Oracle Flash Storage CLI.

    Note: The `hostmap` command manages the mappings between Oracle FS System SAN LUNs and hosts.

    ```
    # fscli hostMap -add -lun /COPY_DATA_01 -lunNumber 100 -hostGroup /MyHostGroup
    …
    # fscli hostMap -add -lun /COPY_DATA_02 -lunNumber 101 -hostGroup /MyHostGroup
    …
    # fscli hostMap -add -lun /COPY_FRA_01  -lunNumber 102 -hostGroup /MyHostGroup
    …
    # fscli hostMap -add -lun /COPY_FRA_02  -lunNumber 103 -hostGroup /MyHostGroup
    …
    ```

```
# fscli hostMap -add -lun /COPY_OCR     -lunNumber 104 -hostGroup /MyHostGroup
…
# fscli hostMap -add -lun /COPY_TEMP_01 -lunNumber 105 -hostGroup /MyHostGroup
…
# fscli hostMap -add -lun /COPY_TEMP_02 -lunNumber 106 -hostGroup /MyHostGroup
…
```

3. Recognize the LUNs.

    Now that the original LUNs have been unmapped and the LUN copies have been mapped, the server must recognize the changes. Consult the instructions for recognizing LUNs on your operating system. Typically, rebooting each node causes the operating system to recognize the new LUNs. Less disruptive ways to recognize LUNs can exist, depending on your operating system.

4. Recover the database.

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

    a. Ensure that the grid infrastructure has started and then mount the database:

        ```
        SQL> startup mount;
        ORACLE instance started.

        Total System Global Area 8.5899E+10 bytes
        Fixed Size                   7654304 bytes
        Variable Size             1.7985E+10 bytes
        Database Buffers          6.7646E+10 bytes
        Redo Buffers               260780032 bytes
        Database mounted.
        ```

    b. Since the database was in hot-backup mode when the copies were made, take the database out of hot-backup mode:

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

    c. Point the DB_RECOVERY_FILE_DEST to the location where you preserved the archived redo logs earlier. The most recent archived redo logs will be available under the +ARC_LOGS disk group that we created earlier.

        ```
        SQL> alter system set db_recovery_file_dest='+ARC_LOGS' scope=memory sid='*';
        System altered.
        ```

    d. Issue the RECOVER command, which checks the database for consistency and applies the archived redo logs from the DB_RECOVERY_FILE_DEST (+ARC_LOGS).

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

    e. To finish recovery, open the database.

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

RESULT: The database is recovered and is ready for use in less time than the traditional RMAN restore and roll forward operations.

Conclusion

This white paper details how to perform backup and recovery operations using a combination of Oracle FSCLI commands and SQL*Plus recovery commands. Oracle recommends scripting and testing the commands to ensure accuracy.

The Oracle FS System user-managed backup solution protects an Oracle database using LUN copies. LUN copies are an alternative to taking full and incremental backups with Oracle's RMAN. Because recovery time is drastically reduced with the Oracle FS System, customers can meet the RTO and RPO for the data backup and recovery operations and, with this solution, there is less burden placed on the database server.

References

Refer to the following URL for complete Oracle Flash Storage and Oracle Database documentation:
http://docs.oracle.com/en/

• *Oracle Flash Storage System Administrator's Guide* for information about Clone LUNs and Storage Domains

• *Oracle Database Storage Administrator's Guide* for information about ASM

• *Oracle Real Application Clusters Administration and Deployment Guide* for information about RAC Databases

• *Oracle Database Backup and Recovery User's Guide*

The *Oracle Flash Storage System CLI Reference* contains information about the LUN and Host map commands used in this white paper. To learn more about the Oracle Flash Storage System CLI commands, click the following URL: http://docs.oracle.com/cd/E57980_15/index.htm

Refer to the *Oracle SQL Reference* for more information about the alter database, startup, and recovery commands used in this white paper. To learn more about the Oracle SQL Reference commands, click the following URL: http://docs.oracle.com/database/121/SQLRF/statements_1006.htm#SQLRF00802

ORACLE

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200