Technical Report

Oracle Archived Logs Management Best Practices

Padmanabhan Sadagopan, NetApp
March 2011 | TR-3901

TABLE OF CONTENTS

LIST OF TABLES

1 INTRODUCTION

Oracle® archived logs (archived redo logs) are copies of the online redo logs used by the Oracle Database to maintain consistency during shutdown and startup of the database. During database recovery, Oracle archived logs (ALs) from a backup will be used to replay the transaction.

Oracle archived logs need to be monitored and managed constantly for the database to continue its optimal operation. If the disk space available to Oracle archived logs is used up, then the whole database will "freeze" and not allow any update or insert operations.

This document provides information about how to manage Oracle archived logs in an Oracle Database on NetApp® storage. This document also addresses best practices and recommendations for implementing archived logs in Oracle Database environments.

1.1 SCOPE AND ASSUMPTIONS

This document is specifically for an environment where Oracle Databases are deployed on NetApp storage using NFS. This document assumes prior knowledge and understanding of Oracle Database and NetApp storage. Database administrators might need to work with storage administrators to perform some of the tasks mentioned in this document.

1.2 ARCHIVED LOGS

Archived redo logs are filled groups of online redo log files that have been archived to an offline destination. Archived logs are used to roll forward a database backup.

When recovering from a cold backup, the database may be started up and used immediately, or additional archived logs may be applied to bring the database to a specific point in time. When recovering from a hot backup, the application of the archived logs is mandatory to bring the database to a consistent usable state. Once at a consistent state, additional archived logs may be applied to bring a database up to a specific point in time. Note that recovery of all committed transactions will require further applying of unarchived redo logs. This document only deals with archived logs.

2 ARCHIVED LOGS

2.1 ARCHIVED LOGS DESTINATION

Oracle Archived logs can be archived to a single or multiple destinations that can be local or remote. You can setup the archived logs destination(s) using Oracle `init.ora` parameters such as `LOR_ARCHIVE_DEST_n`.

When you archive to multiple destinations, a copy of each filled redo log file is written to each destination. These redundant copies are to make sure that archived logs are always available in the event of a failure at one of the destinations.

Edit the `init.ora` file to include:

```
*.LOG_ARCHIVE_DEST_1='LOCATION=\u10\oradata\MYDB\archive\archive1'
*.LOG_ARCHIVE_DEST_2='LOCATION=\u20\oradata\MYDB\archive\archive2'
*.LOG_ARCHIVE_FORMAT='%t_%s_%r.dbf'
```

For details, see the Oracle Database Reference document.

Following are the acceptable archived logs destinations:

- Directories defined in the `LOG_ARCHIVE_DEST_n`
- Another directory in the same server or another server

To minimize the problems associated with Oracle archived logs destination failure, you can define the minimum number of successful destination by setting the `init.ora` parameter `LOG_ARCHIVE_MIN_SUCCEED_DEST=`*n* . The default value is 1. Valid values for *n* are 1 to 2 if you are using duplexing, or 1 to 10 if you are multiplexing.

This parameter determines the minimum number of destinations to which Oracle Database must successfully archive a redo log group before it can reuse online log files. By setting a value greater than the default value, you can make sure that the archived logs are successfully archived in multiple locations for better availability.

Archived logs location information can be obtained from `V$ARCHIVED_LOG, V$LOG_HIST, V$LOG` data dictionary views.

Note: For details about Oracle archived logs destination parameter and the archived logs location information, see the Oracle Database Reference document.

3 NETAPP STORAGE SYSTEM CONFIGURATION

This section explains the configuration needed to be done on NetApp storage systems as well as on the Oracle Database servers to manage the archived logs.

Archived logs management is a list of policy based activities performed on the archived logs; backup, recovery, and deletion of archived logs.

3.1 ORACLE DATABASE LAYOUT ON NETAPP

Table 1 shows the sample Oracle Database layout on NetApp storage systems. NetApp volumes are mounted to the DB servers. Two dedicated volumes are defined to provide recoverability and maximum availability for archived logs.

Table 1) Oracle Database layout.

Database Files	NetApp Volume	Physical Directory	Comments
Datafiles	ora_data01	/u10/app/<DBSID>/oradata	
Control files	ora_cntrl01	/u20/app/<DBSID>/control	
Redo logs	ora_redo01	/u30/app/<DBSID>/logs	
Archived logs	ora_arch10	/u40/app/<DBSID>/arch10	Primary archived log location
Archived logs	ora_arch20	/u50/app/<DBSID>/arch20	Secondary archived log location

3.2 ARCHIVED LOGS POLICIES ON DB SERVERS

You can define a policy to determine how long backups and archived logs need to be retained for media recovery. It is assumed that you have already defined the backup policy for the entire Oracle Database.

Policies can be defined as environment variables at the database servers and you can leverage those variables to manage the archived logs through automated scripts.

Table 2 defines NetApp specific policies for archived logs.

Table 2) Archived logs Backup and retention policies.

Policy	Description
ALS_BACKUP_POLICY	Determines how often to back up archived logs. The duration can be same as the database backup policy defined.
ALS_RETENTION_POLICY	Determines how often to delete the backup of the archived logs. The duration can be same as the database backup policy defined.

3.3 ARCHIVED LOGS PARAMETERS ON DB SERVERS

In addition to the Oracle provided archived logs `init.ora` parameters, you can define the below mentioned parameters in the database servers to manage the archived logs.

Table 3) Archived logs parameters.

Parameter	Description
AL_SNAPSHOT_KEEP_LOCAL_PRI	The number of days that the Snapshot copies of AL volumes need to be kept in a primary local destination.
AL_SNAPSHOT_KEEP_LOCAL_TMP	The number of days that the Snapshot copies of AL volumes need to be kept in a temporary local destination (other than the primary local destination)
AL_SNAPSHOT_DELETE_LOCAL	The number of days after which the Snapshot copies of AL volumes need to be deleted or moved to tape or secondary storage system at remote destination
ARCHIVEDLOGS_FILES_DELETE	Number of days' worth of AL files to be deleted from default location

For example:

- `AL_SNAPSHOT_KEEP_LOCAL_PRI=7`

  Retains the AL Snapshot copy for 7 days in the primary local destination.
- `AL_SNAPSHOT_KEEP_LOCAL_TMP=20`

  Retains the AL Snapshot copy for 20 days in the secondary destination.
- `AL_SNAPSHOT_DELETE_LOCAL = 25`

  Deletes the Snapshot copy from the primary after 25 days.
- `ARCHIVEDLOGS_FILES_DELETE=25`

  Deletes 25 days worth of AL files from the primary destination.

3.4 SAMPLE CONFIGURATION

1. Create two dedicated FlexVol volumes, one for the local destination (and the other one for a remote destination so that archived logs are multiplexed into two different storage systems.
2. NFS mount the volumes and export them from the database servers.

   Below is a sample of mount command output:

```
OS Prompt>> mount
Filer1:/vol/ MYDB_ARCH_VOL10      150G  7.3M  150G   1% /u10/oradata/MYDB/arch_dest/arch10
Filer2:/vol/MYDB_ARCH_VOL20       150G  7.3M  150G   1% /u20/oradata/M
```

3. Entry in the `init.ora` file:

```
*.LOG_ARCHIVE_DEST_10='LOCATION=/u10/oradata/MYDB/arch_dest/arch10'
*.LOG_ARCHIVE_DEST_20='LOCATION=/u20/oradata/MYDB/arch_dest/arch20'
*.LOG_ARCHIVE_FORMAT='%t_%s_%r.dbf'
```

4 ORACLE DATABASE BACKUP AND RECOVERY

4.1 BACKING UP AN ORACLE DATABASE RUNNING IN ARCHIVELOG MODE

This backup procedure assumes that all the NetApp volumes needed for datafiles, control files, and archived log files are mounted and accessible from the database servers. To backup the Oracle Database when running in ARCHIVELOG mode, do as follows:

1. Put the entire database or all the tablespaces in "hot backup" mode.
2. Issue either **`snapshot create`** or OS specific copy command (for example, **`cp`**) to back up the datafiles and control files to a local or remote location.
3. End database "hot backup" mode.
4. Perform a “log switch” or issue the **`archivelog current`** command to archive the latest redo log files:

   `ALTER SYSTEM ARCHIVE LOG CURRENT`

   `ALTER SYSTEM SWITCH LOGFILE`
5. Now issue either **`snapshot create`** or OS specific command to back up the archived log files.

4.2 RESTORING AN ORACLE DB RUNNING IN ARCHIVELOG MODE

1. Determine which database file needs recovery.
2. Put the database in the right mode (mount/open), whether single or multiple database file recovery.
3. Restore the necessary files by issuing the **`snap restore`** or OS specific copy command
4. Recover the restored files by logging in to sqlplus and issuing the **`recover database`** command.
5. Put the database in normal mode.

5 MANAGING ARCHIVED LOGS

5.1 BACKING UP ARCHIVED LOGS

Table 4 lists the NetApp data protection features that can be used to back up the Oracle archived log files.

Table 4) NetApp data protection features.

Feature	Description
Snapshot	Helps you back up data within a FlexVol volume where data resides.
SnapMirror®	Enables you to periodically create Snapshot copies of data on one volume or qtree; replicate that data to a partner volume or qtree usually on another storage system; and archive one or more iterations of that data as Snapshot copies. Replication on the partner volume or qtree ensures quick availability and restoration of data from the point of the last Snapshot copy should the storage system containing the original volume or qtree become disabled.
SnapVault®	Enables you to back up qtrees on multiple volumes and storage systems to a single SnapVault secondary storage system for quick backup and restore of its sources. In the event of data loss or corruption on a system, backed-up data can be restored from the SnapVault secondary system with less downtime and uncertainty than are associated with conventional tape backup and restore operations.
Tape backup	The **`dump`** and **`restore`** commands allow you to back up Snapshot copies to tape. The dump command creates a Snapshot copy of the volume and then copies that data to tape.

Before creating a backup of archived logs, make sure that the database is in a consistent state as follows:

1. Put the Oracle Database in hot backup mode:
   ```
   ALTER DATABASE BEGIN BACKUP;
   ```
2. Create a Snapshot copy of datafiles and control files.
3. End database hot backup mode:
   ```
   ALTER DATABASE END BACKUP;
   ```
4. Issue the **`switch logfile`** command to capture last redo.
5. Determine which archived redo log files need to be backed up by running the following query:
   ```
   SELECT THREAD#,SEQUENCE#,NAME
   FROM V$ARCHIVED_LOG;
   ```
6. Create a Snapshot copy of the archived logs by issuing the NetApp **`snapshot create`** command (for example, **`snap create) or`** an operating system specific copy command (for example, cp) or to back up the archived redo logs identified in the previous step.

Now you have a consistent backup of archived logs along with datafiles and control files.

5.2 RESTORING ARCHIVED LOGS

You can restore the archived logs from the backup that you created previously. Based on the type of recovery you need to perform you might need to restore datafiles and control files before you restore the archived logs.

RECOVERY WHEN ARCHIVED LOGS ARE IN THE DEFAULT LOCATION

If all the required archived log files are mounted at the LOG_ARCHIVE_DEST_1 destination, and if the value for LOG_ARCHIVE_FORMAT is never altered, then the database can suggest and apply log files to complete media recovery automatically.

RECOVERY WHEN ARCHIVED LOGS ARE IN A NONDEFAULT LOCATION

If all the required archived log files are mounted at a non-default location we have to use the **SET LOGSOURCE** command to start the recovery from the non-default location.

1. Using an operating system utility, copy the archived redo logs to an alternate location. For example, enter:

   ```
   OS Prompt>> cp  /u10/oradata/MYDB/arch_dest/arch10/*   /tmp
   ```

2. Using SQL*Plus, specify the alternative location for the recovery operation. Use the LOGSOURCE parameter of the SET statement. For example, start SQL*Plus and run:

   ```
   SET LOGSOURCE "/tmp"
   ```

3. Begin media recovery as usual. For example, enter:

   ```
   SQL>>RECOVER DATABASE
   ```

5.3 DELETING ARCHIVED LOGS

To delete the archived logs, use the ALS_BACKUP_POLICY and the ALS_RETENTION_POLICY parameters (see Table 2).

Note: Before you delete the backup of archived logs, whether it is Snapshot copies or actual files, make sure that they meet the defined retention policy.

NetApp also recommends deleting Snapshot copies that belong to the same backup set (backups taken at the same time), including datafiles, control files, and archived log files.

6 METHODS TO MANAGE ARCHIVED LOGS

You can perform backup/restore of the archived logs using multiple tools and technologies available from NetApp and Oracle:

- NetApp Snapshot and NetApp SnapRestore
- NetApp SnapVault
- NetApp Snap Creator Framework
- NetApp SnapManager® for Oracle (SMO)
- Oracle Recovery Manager (RMAN)
- Custom scripts (Snapshots, APIs, and shell scripts)

6.1 NETAPP SNAPSHOT

6.1.1 BACK UP USING SNAPSHOT

1. Make sure the database, datafiles, and control files are backed up first. Then issue the **`ALTER SYSTEM SWITCH LOGFILE`** command followed by the **`ALTER SYSTEM ARCHIVELOG ALL`** command to force archiving of all log files. Then create a Snapshot copy of the ARCH volumes.
2. Connect to the storage system and issue the **`snap create`** command.
3. Connect to the storage system and issue the **`snap delete`** command to delete the oldest Snapshot copies.
4. Connect to the storage system and issue the **`snap rename`** command to rename the second oldest to the oldest Snapshot copy (which you just deleted in the previous step) to reuse the Snapshot copies.
5. Assuming that you have a set of Snapshot copies (0-6 for 7 days' backup), whenever you create a particular Snapshot copy for a particular day, delete the oldest one, and then rename the current one with a previous name. This allows you to use the same Snapshot copy name and avoid reaching the maximum Snapshot copy limit.
6. Query the `V$PARAMETER` to get the `LOG_ARCH_DEST` parameter value.
7. Use OS command to delete the files located in `LOG_ARCH_DEST`, which are older than the value provided in `ARCH_FILES_KEEP` parameter:

   ```
   prompt> find $ORA_ADMIN/$ORACLE_SID/arch/*.log -mtime
   +<ARCHIVEDLOGS_FILES_DELETE> -exec rm {} \;
   ```

SAMPLE COMMANDS

Creating a Snapshot copy:

To create a Snapshot copy, enter:

```
rsh -n $FILER snap create $VOLUME $SNAPSHOT
```

Listing Snapshot copies:

To list Snapshot copies, enter:

```
rsh -n $FILER snap list $VOLUME
```

Use a naming convention for the archived log Snapshot copy, which is similar to the following:

```
<hostname_SID>_<sysdate>_<#s ranging from 0 to 6>.arch
```

Deleting Snapshot copies:

To delete Snapshot copies, enter:

```
rsh -n $FILER snap delete $VOLUME $SNAPSHOT
```

To delete the archived log volume Snapshot copy that is greater than five days old, enter:

```
rsh -n FILER1 snap delete MYDB_ARCH_VOL10 <MYHOST1_MYDB_ARCH10.01012011.06>
```

Renaming Snapshot copies:

To rename a Snapshot copy, enter:

```
rsh -n $FILER snap rename $VOLUME $FROM_SNAPSHOT $TO_SNAPSHOT
```

To rename an archived log volume Snapshot copy that is older than five days, enter:

```
rsh -n FILER1 snap rename MYDB_ARCH_VOL10 <MYHOST1_MYDB_ARCH10.01012011.05>
<MYHOST1_MYDB_ARCH10.01012011.06>
rsh -n FILER1 snap rename MYDB_ARCH_VOL10 <MYHOST1_MYDB_ARCH10.01012011.04>
<MYHOST1_MYDB_ARCH10.01012011.05>
```

Deleting Archived Logs

To delete archived logs from the archived log destination directory that is older than five days, issue the following command:

```
find  /u10/app/oracle/<ORA_SID>/arch/arch10/* -name <ORA_SID>_*.arch  -mtime +5 -exec
rm {}
```

Example:

```
find  /u10/app/oracle/<ORA_SID>/arch/arch10/* -name <ORA_SID>_*.arch  -mtime +5 -exec
ls -ltr {} \;

find    /u10/app/oracle/<ORA_SID>/arch/arch10/*    -name <ORA_SID>_*.arch  -mtime +5 –
exec   rm {} \;
```

If you used NetApp Snapshot to create a backup, you can use SnapRestore® to restore them.

6.1.2 RESTORE USING SNAPRESTORE

SnapRestore enables you to quickly revert a local volume or file to the state it was in when a particular Snapshot copy was created. In most cases, reverting a file or volume is much faster than restoring files from tape or copying files from a Snapshot copy to the active file system.

SnapRestore performs Snapshot copy restoration more quickly, using less disk space, than an administrator can achieve by manually copying objects to be restored from the Snapshot copy to the active file system.

For NFS volumes, using volume-based SnapRestore, unmount the files and directories in that particular volume. Unmounting is not required if you use single file SnapRestore.

Sample command:

```
Storage System>> snap restore -f -t vol <arch vol name>
Storage System>> snap restore -f -t vol -s <snapshot_name> <vol_name>
Storage System>> snap restore -f -t file -r <restore_as_new_path> <path_and_file_name>
Storage System>> snap restore -f -t file -s snapshot_name -r restore_as_path
path_and_file_name
```

Where:

-f option avoids warning messages and prompts to confirm your decision to revert the volume.

-s `snapshot_name` specifies the name of the Snapshot copy from which to revert the data.

-t file specifies that you are entering the name of a file to revert.

-r `restore_as_new_path` restores the file to a location different from (but in the same volume as) the location in the Snapshot copy.

`path_and_file_name` is the complete path to the name of the file to be reverted. You can enter only one path name.

A file can be restored only to the volume where it was originally located. The directory structure to which a file is to be restored must be the same as that specified in the path. If this directory structure does not exist, you must create it before restoring the file.

6.1.3 DELETE SNAPSHOT S AND FILES

DELETING THE ARCHIVED LOG SNAPSHOT COPIES

To delete the Snapshot copies that were used as a backups of archived logs, you can use the NetApp **SNAP DELETE** command or whatever archived logs management tools such as SnapVault, SMO, or RMAN or so on.

```
Storage System>> snap delete <primaryvolume> <Snapshot copy name>
```

DELETING THE ARCHIVED LOG FILES

You can also use manual scripts and OS commands to delete the archived logs that are older than the retention time specified for archived redo logs.

Leverage the archived logs variable defined earlier such as `ARCHIVEDLOGS_FILES_KEEP`, `ARCHIVEDLOGS_FILES_DELETE.`

To clean up archived log files that are more than five days old, issue the following command:

```
prompt> find /u10/oradata/MYDB/arch_dest/arch10/*.log -mtime
+<ARCHIVEDLOGS_FILES_DELETE> -exec rm {} \;
```

```
prompt> find /u10/oradata/MYDB/arch_dest/arch10/*.log -mtime +5 -exec rm {} \;
```

6.2 NETAPP SNAPVAULT

The SnapVault disk-based backup and restore system enables you to perform fast and simple data restore operations. The qtree is the basic unit of SnapVault backup and restore operations. SnapVault backs up specified qtrees on the primary system to associated qtrees on the SnapVault secondary system. If necessary, data is restored from the secondary qtrees back to their associated primary qtrees.

On primary systems, SnapVault backs up primary qtree data, nonqtree data, and entire volumes to qtree locations on the SnapVault secondary systems. The SnapVault secondary system is the central disk-based unit that receives and stores backup data from the system as Snapshot copies.

If data needs to be restored to the primary system, SnapVault transfers the specified versions of the qtrees back to the primary system that requests them.

6.2.1 BACKUP USING SNAPVAULT

SnapVault enables data stored on a primary system to be backed up to a secondary system quickly and efficiently as read-only Snapshot copies. After you have enabled SnapVault on both the primary and secondary storage systems and have given primary and secondary storage systems access to each other, specify the qtrees or volumes for the archived logs files whose data you want transferred from the primary storage system to the SnapVault secondary storage system. You must then perform a complete (baseline) transfer of data from the primary storage system to the secondary storage system.

Enter the following command from the secondary system:

```
snapvault start -S prim_system:prim_qtree_path sec_host:sec_qtree_path
snapvault start -S prim_system:/vol/volume_name /vol/volume_name/qtree_name
```

Where:

`-S prim_system:/vol/volume_name` specifies the volume on the primary system whose data you want to back up.

`/vol/volume_name/qtree_name` specifies the qtree in the secondary system where you want to store this data.

You can also use the "**`snapvault snap sched`**" command to schedule the Snapshot copy backup.

Note: Use this when you want to back up a volume that contains many qtrees or use the Snapshot copy management of SnapVault or to consolidate data from several volumes into single destination volume.

6.2.2 RESTORE USING SNAPVAULT

You use the SnapVault restore command to restore a backed-up qtree saved to the secondary system. You can restore the data to an existing qtree on the primary system using baseline restore or incremental restore.

After successfully restoring data, use the **`snapvault start -r`** command to resume the SnapVault relationship between the restored qtree and its backup qtree on the secondary system.

Example:

```
snapvault restore [-s snapname] -S sec_system:sec_qtree_path [prim_system:]prim_qtree_path
```

Where:

`-S` [sec_system:]sec_qtree_path specifies the secondary system and qtree path from which you want to restore the data.

`-s` option specifies that the restore operation must use the specified (`snapname`) Snapshot copy on the secondary system.

`prim_system` is the name of the primary system to which you want to restore. If specified, this name must match the name of the host system.

`prim_qtree_path` is the name of the primary system qtree to which you want to restore.

6.2.3 DELETE USING SNAPVAULT

You can use the **`snap delete`** command to delete any Snapshot copies in the particular volume:

```
Storage System>> snap delete <primaryvolume> <Snapshot copy name>
```

6.3 NETAPP SNAPMANAGER FOR ORACLE

SnapManager for Oracle (SMO) backs up datafiles, control files, and archived logs. Although SMO does not currently manage or restore archived logs, SMO includes them with each backup and uses them for cloning from hot (online) backups. SMO requires that archiving be enabled before creating an online backup of the database.

SMO does not remove archived logs that have been backed up and does not restore them. You can use either RMAN or a scripted solution to manage SMO backed-up archived logs.

SMO expects the archived logs to exist in the original location during the recovery process. If the archived logs are accidentally deleted or do not exist in the original location, first mount the appropriate SMO backups that contain the missing archived logs and manually copy the missing archived logs to the original location before initiating the SMO restore and recovery operation.

6.3.1 BACKUPS USING SMO

By default, SMO creates backups using Snapshot copies on the primary storage system. SMO organizes information into profiles which are then associated with databases, and then creates backups via a profile. All of the metadata about the profiles and their backups are then maintained within repositories. Profiles contain information about the database being managed, while the repository contains data about operations performed on profiles. The repository records when a backup took place, which files were backed up, and whether a clone was created from the backup. When DBAs restore a database or recover a portion of it, SMO queries the repository to determine what was backed up.

For a full backup, SMO backs up the entire database and creates Snapshot copies at the volume level (not at the tablespace level). When you specify a full online backup, SMO puts the database in "hot backup" mode, then takes Snapshot copies of datafiles. After the database is out of "hot backup" mode, there will be a ‘log switch’, and then it creates Snapshot copies of control and archived log files.

Sample command:

```
>> smo backup create -profile <profile_name> -auto -full -label <name> -protect -
retain –daily
```

Note: SMO does not backup online log files, temporary files, and other parameter files.

6.3.2 RESTORE USING SMO

SMO provides the ability to restore a database to the state it was in at the time a Snapshot copy was created. In addition to its file-based restore process, SMO leverages volume-based fast restore technology (available only in the *nix platforms only), which shortens the restore time significantly compared to traditional recovery methods. Since backups can now be created more frequently, the number of logs that need to be applied is drastically reduced, thus reducing the mean time to recovery (MTTR) for a database.

You can use SMO to automatically restore and recover the database as follows:

1. Determine if the archived logs required for recovery are in the original location.
2. If not, mount the corresponding SMO backup using the SMO CLI/GUI.
3. Manually copy the archived logs from the SMO backup to their original location.
4. Use SMO to automatically restore and recover the database.

Sample command:

```
>> smo backup restore -profile <profile_name> -label <name> -complete
```

6.3.3 DELETE USING SMO

Note: Currently, it is not possible to have different retention policies for datafiles and archived log files, as SMO treats the whole database as one logical unit. A retention policy defined at the database level is applicable for the archived logs as well. This holds true when you try to manage the backup Snapshot copies.

However, to delete the archived logs located in the default archived log destination, you can set up an environment variable to define the retention value and leverage the same in your script (see section 3.2 Archived LogS Policies on DB Servers).

To delete archived logs:

Before you delete the archived logs, you need to be aware of the Snapshot copies to which the archived logs belong. Assume you have three Snapshot copies, namely s1, s2, and s3, created at the same time for datafiles, control files, and archived log files respectively, all belonging to the same database.

1. Get the 'Start Date' and 'Label name' from SMO repository by running the **`smo backup list`** command:

   `>> smo backup list –profile  MYDB1  -verbose`

```
Start Date                  Status       Scope   Mode     Primary      Label      Retention
Protection
---------------------------  -------------- -------    ---------      --------    --------
-------------  ---------------
2011-01-10 14:31:27  SUCCESS FULL   ONLINE   EXISTS  backup1    DAILY      PROTECTED
```

2. Delete the actual archived log files from the LOG_ARCH_DEST directory, which are older than 'Start Date' time, using OS specific commands.

   `OS Prompt>> find /u10/oradata/MYDB/arch_dest/arch10/*.log -mtime +<ARCHIVEDLOGS_FILES_DELETE> -exec rm {} \;`

3. As SMO automatically deletes the backup (Snapshot copies) based on the database-level retention policy defined in the system, there is no need to manually delete the backup.
4. If SMO is not configured to delete the backup, invoke **`smo delete`** command to delete the label, which automatically deletes the backup/Snapshot copy for the data/control and archived log files associated with that label.
5. Make sure that every time the backup is deleted, the corresponding archived logs files are deleted from the default destination by running the OS specific commands.

   Note: You can leverage SMO pre and post scripting capabilities to automate deletion of the archived logs. For versions prior to SMO 3.1, there is no pre or post script capability during backup/recovery operation. It is available only during cloning operation.

Sample command:

Before deleting the backup, verify that operations are complete and then issue the delete command:

`>> smo operation list -profile profile_name -quiet -verbose`

`>> smo backup delete  -profile <profile_name> -label <name>`

6.4 NETAPP SNAP CREATOR FRAMEWORK PLUG-IN

We can leverage the Snap Creator Framework and its associated modules for Oracle Database and ARCHIVELOG to backup/restore and delete archived logs. NetApp recommends backing up the archived logs using SnapVault before deleting them through the Snap Creator archived logs module. Only if the backup operation is successful, will Snap Creator delete the archived logs on primary based on the retention policy.

The Snap Creator ARCHIVELOG module is not application aware, which means that it does not track which archived logs belong to which backup. Nor does it have a concept of a “recovery window”. It simply provides a mechanism for backing up data, getting data to secondary, and deleting archived logs on primary based on retention. The Snap Creator ARCHIVELOG module works with any database and file structure.

Note: For more information, see the TR-3841: SnapCreator 3.2 Installation and Administration Guide.

6.4.1 ORACLE MODULE

The Oracle Module only supports Oracle Database 10*g* or higher. The entire database is put into hot backup mode which is only supported starting with Oracle 10*g*. The Oracle Module uses SQL*Plus to communicate with the database.

Table 5) Snap Creator Framework Oracle Module

Parameter	Description
APP_NAME	The application name.
ORACLE_DATABASES	A list of Oracle Databases and the user name: this is, db1:user1;db2:user2.
SQLPLUS_CMD	The path to the `sqlplus` command.
CNTL_FILE_BACKUP_DIR	The path to the directory for storing backup control files (Oracle user must have permissions).
ORA_TEMP	The path to a directory for storing temp file, that is, `/tmp` (Oracle user must have permissions).
ARCHIVE_LOG_ONLY	Informs Oracle Module to only do a switch log. This setting is useful if you are handling archived logs separate from data backup.

6.4.2 ARCHIVELOG MODULE

As the ARCHIVELOG module is database independent, it can work with any database such as Oracle, DB2, and so on. This module does not communicate with the database. Its purpose is to delete the archived logs that are older than the configured threshold value (in days).

Table 6) Snap Creator Framework Archived log module

Parameter	Description
ARCHIVE_LOG_ENABLE	Enables archived log management (deletion of old archived logs).
ARCHIVE_LOG_RETENTION	Determines the archived logs retention period in days; epoch time is used and is precise to the second.
ARCHIVE_LOG_DIR	Path to the directory that contains the archived logs.
ARCHIVE_LOG_EXT	File extension of the archived logs: for example, if the archived logs are 10192091019.log, you would set this to “log”; the search pattern to use is <something>.<extension>

Sample commands:

```
./snapcreator --profile <Profile> --action snap --policy <Policy> <Optional Arguments>

./snapcreator --profile <Profile> --action arch <Optional Arguments>

./snapcreator --profile <Profile> --action restore –-policy <Policy> <Optional Arguments>
```

```
./snapcreator --profile <Profile> --action delete –-policy <Policy> <Optional Arguments>
```

6.5 ORACLE RECOVERY MANAGER

You can back up the whole database or a particular tablespace with Oracle Recovery Manager (RMAN). You can also define backup retention policies as follows:

```
'CONFIGURE RETENTION POLICY TO REDUNDANCY <#days>'
```

The following example shows the procedure for backing up a whole database to the default destination:

```
RMAN> BACKUP DATABASE;
RMAN> SQL 'ALTER SYSTEM ARCHIVE LOG CURRENT';
```

Note: By archiving the logs immediately after the backup, you can have a full set of archived logs through the time of the backup. By doing this, you can perform media recovery after restoring this backup.

You can back up logs with `BACKUP ARCHIVELOG` or back up logs while backing up datafiles and control files by specifying `BACKUP ... PLUS ARCHIVELOG`.

While backing up the archived logs, you can specify a range of archived redo logs by time, SCN, or log sequence number:

```
RMAN>> BACKUP ARCHIVELOG ALL;
RMAN>> BACKUP ARCHIVELOG FROM TIME 'SYSDATE-30' UNTIL TIME 'SYSDATE-7';
```

Make sure log switches happened (RMAN does that) to make sure that the backup contains all redo that was generated prior to the start of the command.

6.5.1 CONFIGURE THE DELETION POLICY USING RMAN

Configure the archived redo log deletion policy using the following commands:

```
CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF <#> DAYS;

CONFIGURE ARCHIVELOG DELETION POLICY <>.
```

6.5.2 CONDITIONS WHEN ARCHIVED LOGS GETS DELETED

Archived logs get deleted when they meet one of the following conditions:

- If there is not enough space in the flash recovery area
- When the RMAN archived log deletion policy is configured
- When retention policy of recovery window is configured and all unwanted archived log are backed up.
- When archived logs are deleted by RMAN commands or backed up using the **`delete input`** command.

6.5.3 RMAN COMMANDS TO DELETE

You can specify the `DELETE INPUT` or `DELETE ALL INPUT` clauses for the `BACKUP ARCHIVELOG` command to delete archived logs after they are backed up, eliminating the separate step of manually deleting the archived redo logs.

With `DELETE INPUT`, RMAN only deletes the specific copy of the archived redo log chosen for the backup set.

With `DELETE ALL INPUT`, RMAN will delete each backed-up archived redo log file from all log archiving destinations:

```
RMAN>> BACKUP DEVICE TYPE sbt   ARCHIVELOG ALL DELETE ALL INPUT;
```

7 GENERAL RECOMMENDATIONS AND GUIDELINES

7.1 GENERAL BEST PRACTICES

Best Practices
• Specify a custom name for the archived log backups that includes the DB SID and timestamp. • Provide a meaningful name for the Snapshot copies of the volumes used for archived logs as follows: `<hostname_SID>_<sysdate>_<#s ranging from 0 to 6>.arch` • Backups on another storage controller or on tape should exist in case total machine failure is encountered, but these should not be relied upon to provide fast recovery. • All archived logs from the point of the last backup until the current point in time should be available on local storage for fast recovery

7.2 STORAGE BEST PRACTICES

Best Practices
• Use a separate (or dedicated) volume for the archived logs within the same storage system. • Turn `auto-snapshot off` in the archived logs volume: `vol options <volume_name> nosnap on` • Reuse Snapshot copies. By renaming the Snapshot copies, you can reuse the Snapshot copies so that you won't reach the maximum Snapshot limit. • Create two dedicated FlexVol volumes, one for local destination and the other for a remote destination, so that archived logs are multiplexed into two different storage systems.

7.3 RMAN BEST PRACTICES

Best Practice
• Use RMAN to create an archived log backup (use RMAN with control file).

7.4 SMO BEST PRACTICES

Best Practices
• Use SMO CLI commands with operating system commands to manage the archived logs. • To use RMAN, register the SMO target database with RMAN. • Before purging the archived logs backup, make sure the metadata of that archived logs backup is purged from the SMO repository.

8 REFERENCES

SnapManager 3.0 for Oracle Installation and Administration Guide:
http://now.netapp.com/NOW/knowledge/docs/SnapManager/relsmoracle30/html/software/install/index.html

SnapManager 3.0 for Oracle Best Practices

http://media.netapp.com/documents/tr-3761.pdf

Backing Up Database Files and Archived Logs with RMAN:
http://download.oracle.com/docs/cd/B19306_01/backup.102/b14192/bkup003.htm

Snap Creator 3.2 Installation and Administration Guide:
http://media.netapp.com/documents/tr-3841.pdf

9 ACKNOWLEDGEMENTS

Greg Loughmiller

Jeffrey Steiner

Mike Doherty

Neil Gerren

Badrinarayanan Srinivasan

Antonio Jose Rodrigues Neto

Anand Ranganathan

Buddy Taylor

Habib Rangoonwala

Phil Clay

Keith Tenzer

Bill Heffelfinger

Esther Smitha

.

www.netapp.com

 TR-3901-0311