NetApp Plug-in 2.0 for Oracle Recovery Manager Media Management Library (RMAN MML)

Command Reference Guide

TABLE OF CONTENTS

LIST OF TABLES

1 Introduction

This document provides examples of various Oracle Recovery Manager (RMAN) operations that will take advantage of the NetApp Plug-in for Oracle Recovery Manager Media Management Library (RMAN MML), version 2.0.0.

The NetApp RMAN MML integrates NetApp storage technologies with RMAN. This allows Oracle Database Administrators (DBAs) to utilize NetApp technologies (Snapshot, FlexClone, SnapRestore), when backing up, restoring and recovering, and duplicating Oracle Database(s) via RMAN.

2 Prerequesites

The following prerequisites must be completed in order for the NetApp RMAN MML to function correctly.

1. Installation and configuration must be complete. See the Quick Start Guide, available on the NetApp RMAN MML Community Site, for installation and configuration instructions.

   Note: See the Installation and Administration Guide, also available on the NetApp RMAN MML Community Site, for an in-depth technical overview of the NetApp RMAN MML.

2. You must be logged in as the Oracle user.
3. The following environment variables must be set:

Environment Variable	Value
BACKUP_DIR	The path of the RMAN MML configuration directory. Example: "/oracle/rman_ntap_configuration"
CONF	The name of the RMAN MML configuration file. **Note:** The configuration file should be located in the directory specified by the BACKUP_DIR variable. Example: "rman_ntap.conf"
ORACLE_HOME	The path of the "Oracle Home" directory. This can be set by running '. oraenv'. Refer to Oracle documentation for details.
ORACLE_SID	The name (SID) of the Oracle instance on which you wish to perform RMAN operations. This can be set by running '. oraenv'. Refer to Oracle documentation for details.

3 Scripting RMAN Commands

This section describes one method for scripting RMAN commands to automate various RMAN operations. All example scripts that are referenced in this document were created in this manner.

To script RMAN commands, create a script titled '<SCRIPT_NAME>.sh' using the following template:

```
## USAGE: bash ./<SCRIPT_NAME>.sh INPUT_1 INPUT_2

# Check to see if ORACLE_HOME is set
if [ -z "$ORACLE_HOME" ]; then
        echo "ORACLE_HOME is empty..."
```

```
        echo "Please run '. oraenv'."
        exit
fi

# Check to see if ORACLE_SID is set
if [ -z "$ORACLE_SID" ]; then
        echo "ORACLE_SID is empty..."
        echo "Please run '. oraenv'."
        exit
fi

# Check to see if BACKUP_DIR is set
if [ -z "$BACKUP_DIR" ]; then
        echo "BACKUP_DIR is empty..."
        echo "You must set the BACKUP_DIR environment variable to the filepath of your RMAN MML
configuration directory."
        echo "e.g. export BACKUP_DIR=<RMAN_MML_CONFIGURATION_DIRECTORY_PATH>"
        echo "EXAMPLE: export BACKUP_DIR=/oracle/rman_mml_config"
        exit
fi

# Check to see if CONF is set
if [ -z "$CONF" ]; then
        echo "CONF is empty..."
        echo "You must set the CONF environment variable to the name of your RMAN MML
configuration file."
        echo "e.g. export BACKUP_DIR=<RMAN_MML_CONFIGURATION_DIRECTORY_PATH>"
        echo "EXAMPLE: export BACKUP_DIR=rman_ntap.conf"
        exit
fi

# Contents of RMAN script
rman_script="
# Add RMAN command(s) here.
# Note: Input parameters can be referenced as follows:
#              $1, for INPUT_1; $2, for INPUT_2; and so on.
"

# Display contents of RMAN script
cat << EOF
RMAN SCRIPT:----------------------------------------
${rman_script}
-------------------------------------------------
EOF

# Execute RMAN script
$ORACLE_HOME/bin/rman <<EOF
${rman_script}
EOF
```

4 Command Reference

This section explains and gives examples of various operations that can be performed using RMAN, that take advantage of the NetApp RMAN MML.

Note: All example scripts that are referenced in this section are included with the installation, and can be found in the following location: `'<EXTRACTION_LOCATION>/ntap-rman-<VERSION_SPECIFIC_NAME>-2.0.0/sample_scripts/'`.

4.1 Backup Database

The NetApp RMAN MML can be used to create a backup, utilizing Snapshot copies, of a database.

Required Parameters

The following parameters must be included in your RMAN MML configuration file. All others are optional. See Table 6 for possible parameters.

Table 1) Backup database - required parameters.

Item#	Parameter	Description
1	FILER	Lists the storage controllers and their respective usernames and passwords.

	Syntax:
	`FILER=<storage_ip_or_hostname_1>:<username>/<password>;<storage_ip_or_hostname_2>:<username>/<password>`
	
	Example:
	`FILER=mikeo_rmqa:vsadmin/Netapp12345`	

2	FILERPASS_ENCRYPTED	Denotes whether or not the provided storage system password is encrypted. **Note:** If the password is not encrypted, MML will automatically encrypt the password upon its first run. **Note:** An encrypted password can be created using 'mml_pass_encrypt_decrypt_x64.o'.

	Example:
	`FILERPASS_ENCRYPTED=NO`
	(if password is not encrypted)
	
	Example:
	`FILERPASS_ENCRYPTED=YES`
	(if password is encrypted)	

3	VOLUMES	Lists the NetApp volumes that you want to backup (Snapshot), restore (SnapRestore), and/or duplicate.

	Syntax:
	`VOLUMES=<storage_ip_or_hostname_1>:<volumename1>,<volumename2>;<storage_ip_or_hostname_2>:<volumename>`
	
	Example:
	`VOLUMES=mikeo_rmqa:rmqa_oradata0,rmqa_oradata1,rmqa_oradata2`	

4	PROTOCOL	Should be set to 'nfs'. This version of the MML only supports NFS.
	Syntax: `PROTOCOL=nfs`	
5	DB_MOUNTPOINT	Lists the NetApp volumes that you want to backup (Snapshot), restore (Snaprestore), and/or duplicate, and their corresponding mountpoints on the Oracle Database server.
	Syntax: `DB_MOUNTPOINT=<storage_ip_or_hostname_1>:<volumename1>:<mountpoint1>,<mountpoint2>;<storage_ip_or_hostname_2>:<volumename2>:<mountpoint3>;<storage_ip_or_hostname_3>:<volumename3>:<mountpoint4>`	

Item#	Parameter	Description
	Example: DB_MOUNTPOINT=mikeo_rmqa:rmqa_oradata0:/oracle/rmqa/oradata0;mikeo_rmqa:rmqa_oradata1:/oracle/rmqa/oradata1;mikeo_rmqa:rmqa_oradata2:/oracle/rmqa/oradata2	

RMAN Commands

The following RMAN commands will create a backup, utilizing Snapshot copies, of a database:

```
connect target /

sql 'alter system archive log current';

run {
allocate channel EFGH device type sbt_tape
PARMS='SBT_LIBRARY=$ORACLE_HOME/lib/libobk.so
ENV=(BACKUP_DIR=<PATH_TO_RMAN_MML_CONFIGURATION_DIRECTORY>,LD_LIBRARY_PATH=$ORACLE_HOME/lib,CONF=
<RMAN_MML_CONFIGURATION_FILE_NAME>.conf)' debug 1 trace 99;
backup proxy only database tag "<TAG>";
release channel EFGH;
}
```

Example Script

The script, `'backup_database.sh'`, will create a backup, utilizing Snapshot copies, of a database:

```
## USAGE: bash ./backup_database.sh <TAG>
## EXAMPLE: bash ./backup_database.sh test_backup_1
```

4.2 Restore Database

The NetApp RMAN MML can be used to restore a database from any available backup that was created using the NetApp RMAN MML.

Required Parameters

The same parameters that are presented in Section 4.1, “Backup Database”, must be included in the RMAN MML configuration file. All others are optional. See Table 6 for possible parameters.

RMAN Commands

The following RMAN commands will restore a backup of a database.

Note: The database must be started in mount mode.

Note: Providing a <TAG> is optional. If no <TAG> is provided, the RMAN MML will restore the latest backup.

```
connect target /

run {
allocate channel EFGH device type sbt_tape
PARMS='SBT_LIBRARY=$ORACLE_HOME/lib/libobk.so
ENV=(RESTORETYPE=volume,BACKUP_DIR=<PATH_TO_RMAN_MML_CONFIGURATION_DIRECTORY>,LD_LIBRARY_PATH=$OR
ACLE_HOME/lib,CONF=<RMAN_MML_CONFIGURATION_FILE_NAME>.conf)' debug 1 trace 99;
restore database from tag "<TAG>";
recover database;
release channel EFGH;
}

sql 'alter database open';
```

Example Script

The script, `'restore_database.sh'`, will restore a backup of a database:

```
## USAGE: bash ./restore_database.sh <TAG>
## EXAMPLE: bash ./restore_database.sh test_backup_1
## NOTE: Database must be started in "mount" mode.
```

4.3 Backup Database and Controlfiles

The NetApp RMAN MML can be used to create a backup, utilizing Snapshot copies, of a database and its controlfiles.

Prerequisites

To configure the controlfile backup, you must run the following commands from within RMAN.

Note: The controlfile backup location must be on a mounted NetApp volume.

```
RMAN> connect target /
RMAN> configure controlfile autobackup format for device type sbt to
'<CONTROLFILE_BACKUP_LOCATION>/cf_%F';
RMAN> configure controlfile autobackup format for device type disk clear;
RMAN> configure controlfile autobackup on;
```

Required Parameters

The same parameters that are presented in Section 4.1, “Backup Database”, must be included in the RMAN MML configuration file. In addition, the following parameters must also be included. All others are optional. See Table 6 for possible parameters.

Table 2) Backup database and controlfiles - required paremeters.

Item#	Parameter	Description
6	CONTROL_VOLUMES	Specifies the volume(s) to which controlfiles will be backed up if controlfile autobackup is configured. **Note:** Only required if you wish to use controlfile autobackup.

	Syntax:
	`CONTROL_VOLUMES=<storage_ip_or_hostname_1>:<controlvolunme1>`
	
	Example:
	`CONTROL_VOLUMES=mikeo_rmqa:rmqa_oractrl`	

7	CONTROL_MOUNTPOINT	Specifies the volume(s) to which controlfiles will be backed up if controlfile autobackup is configured, and their corresponding mountpoint(s). **Note:** Only required if you wish to use controlfile autobackup.

	Syntax:
	`CONTROL_MOUNTPOINT=<storage_ip_or_hostname_1>:<controlvolunme1>:<controlvolumemountpoint1>`
	
	Example:
	`CONTROL_MOUNTPOINT=mikeo_rmqa:rmqa_oractrl:/rmqa_oractrl`	

RMAN Commands

If the Prerequisites have been completed, the same RMAN commands that are presented in Section 4.1, “Backup Database”, will create a backup, utilizing Snapshot copies, of a database and its controlfiles.

Example Script

If the Prerequisites have been completed, the same bash script that is presented in Section 4.1, “Backup Database”, will create a backup, utilizing Snapshot copies, of a database and its controlfiles.

4.4 Restore Database and Controlfiles

The NetApp RMAN MML can be used to restore a database and its controlfiles from any available backup that was created using the NetApp RMAN MML.

Required Parameters

The same parameters that are presented in Section 4.3, “Backup Database and Controlfiles”, must be included in the RMAN MML configuration file. All others are optional. See Table 6 for possible parameters.

RMAN Commands

The following RMAN commands will restore a backup, including controlfiles, of a database.

Note: The database must be started in nomount mode.

```
connect target /

set dbid <DBID>;

run {
allocate channel EFGH device type sbt_tape
PARMS='SBT_LIBRARY=$ORACLE_HOME/lib/libobk.so
ENV=,BACKUP_DIR=<PATH_TO_RMAN_MML_CONFIGURATION_DIRECTORY>,LD_LIBRARY_PATH=$ORACLE_HOME/lib,CONF=
<RMAN_MML_CONFIGURATION_FILE_NAME>.conf, RESTORETYPE=controlvolume)' debug 1 trace 99;
set controlfile autobackup format for device type sbt to '<CONTROLFILE_BACKUP_LOCATION>/cf_%F';
restore controlfile from autobackup;
release channel EFGH;
}

alter database mount;

run {
allocate channel EFGH device type sbt_tape
PARMS='SBT_LIBRARY=$ORACLE_HOME/lib/libobk.so
ENV=,BACKUP_DIR=<PATH_TO_RMAN_MML_CONFIGURATION_DIRECTORY>,LD_LIBRARY_PATH=$ORACLE_HOME/lib,CONF=
<RMAN_MML_CONFIGURATION_FILE_NAME>.conf, RESTORETYPE=volume)' debug 1 trace 99;
restore database;
recover database;
release channel EFGH;
}

sql 'alter database open resetlogs';
```

Example Script

The script, `'restore_database_and_ctrlfiles'`, will restore a backup, including controlfiles, of a database:

```
## USAGE: bash ./restore_database_and_ctrlfiles.sh <DBID> <CONTROLFILE_AUTOBACKUP_FORMAT>
## EXAMPLE: bash ./restore_database_and_ctrlfiles.sh 3604801818 /oracle/rmqa/ctrlbak/rmqa/cf_%F
## NOTE: Database must be started in "nomount" mode.
```

4.5 Backup Pluggable Database (Oracle Database 12*c*)

The NetApp RMAN MML can be used to create a backup, utilizing Snapshot copies, of a pluggable database.

Note: Oracle Database 12*c* only.

Required Parameters

The same parameters that are presented in Section 4.1, “Backup Database”, must be included in the RMAN MML configuration file. All others are optional. See Table 6 for possible parameters.

RMAN Commands

The following RMAN commands will create a backup, utilizing Snapshot copies, of a pluggable database:

```
connect target /

sql 'alter system archive log current';

run {
allocate channel EFGH device type sbt_tape
PARMS='SBT_LIBRARY=$ORACLE_HOME/lib/libobk.so
ENV=(BACKUP_DIR=<PATH_TO_RMAN_MML_CONFIGURATION_DIRECTORY>,LD_LIBRARY_PATH=$ORACLE_HOME/lib,CONF=
<RMAN_MML_CONFIGURATION_FILE_NAME>.conf)' debug 1 trace 99;
backup proxy only pluggable database <PDB_NAME> tag "<TAG>";
release channel EFGH;
}
```

Example Script

The script, `‘backup_pdb.sh’`, will create a backup, utilizing Snapshot copies, of a pluggable database:

```
## USAGE: bash ./backup_pdb.sh <PDB_NAME> <TAG>
## EXAMPLE: bash ./backup_pdb.sh plugdb test_backup_1
```

4.6 Restore Pluggable Database (Oracle Database 12c)

The NetApp RMAN MML can be used to restore a pluggable database from any available backup that was created using the NetApp RMAN MML.

Note: Oracle Database 12c only.

Required Parameters

The same parameters that are presented in Section 4.1, “Backup Database”, must be included in the RMAN MML configuration file. All others are optional. See Table 6 for possible parameters.

RMAN Commands

The following RMAN commands will restore a backup of a pluggable database.

Note: The database must be started in mount mode.

Note: Providing a <TAG> is optional. If no <TAG> is provided, the RMAN MML will restore the latest backup.

```
connect target /

run {
allocate channel EFGH device type sbt_tape
PARMS='SBT_LIBRARY=$ORACLE_HOME/lib/libobk.so
ENV=(RESTORETYPE=file,BACKUP_DIR=<PATH_TO_RMAN_MML_CONFIGURATION_DIRECTORY>,LD_LIBRARY_PATH=$ORAC
LE_HOME/lib,CONF=<RMAN_MML_CONFIGURATION_FILE_NAME>.conf)' debug 1 trace 99;
```

```
restore pluggable database <PDB_NAME> from tag "<TAG>";
recover pluggable database <PDB_NAME>;
release channel EFGH;
}

sql 'alter database open';
```

Example Script

The script, `'restore_pdb.sh'`, will restore a backup of a pluggable database:

```
## USAGE: bash ./restore_pdb.sh <PDB_NAME> <TAG>
## EXAMPLE: bash ./restore_pdb.sh plugdb test_backup_1
## NOTE: Database must be started in "mount" mode.
```

4.7 Backup Tablespace

The NetApp RMAN MML can be used to create a backup, utilizing Snapshot copies, of a tablespace.

Required Parameters

The same parameters that are presented in Section 4.1, “Backup Database”, must be included in the RMAN MML configuration file. All others are optional. See Table 6 for possible parameters.

RMAN Commands

The following RMAN commands will create a backup, utilizing Snapshot copies, of a tablespace:

```
connect target /

run {
allocate channel EFGH device type sbt_tape
PARMS='SBT_LIBRARY=$ORACLE_HOME/lib/libobk.so
ENV=(BACKUP_DIR=<PATH_TO_RMAN_MML_CONFIGURATION_DIRECTORY>,LD_LIBRARY_PATH=$ORACLE_HOME/lib,CONF=
<RMAN_MML_CONFIGURATION_FILE_NAME>.conf)' debug 1 trace 99;
backup proxy only tablespace <TABLESPACE_NAME> tag "<TAG>";
release channel EFGH;
}
```

Example Script

The script, `'backup_tablespace.sh'`, will create a backup, utilizing Snapshot copies, of a tablespace:

```
## USAGE: bash ./backup_tablespace.sh <TABLESPACE_NAME> <TAG>
## EXAMPLE: bash ./backup_tablespace.sh tbs tbs_bk1
```

4.8 Restore Tablespace

The NetApp RMAN MML can be used to restore a tablespace from any available backup that was created using the NetApp RMAN MML.

Required Parameters

The same parameters that are presented in Section 4.1, “Backup Database”, must be included in the RMAN MML configuration file. All others are optional. See Table 6 for possible parameters.

RMAN Commands

The following RMAN commands will restore a backup of a tablespace.

Note: Providing a <TAG> is optional. If no <TAG> is provided, the RMAN MML will restore the latest backup.

```
connect target /

run {
allocate channel EFGH device type sbt_tape
PARMS='SBT_LIBRARY=$ORACLE_HOME/lib/libobk.so
ENV=(BACKUP_DIR=<PATH_TO_RMAN_MML_CONFIGURATION_DIRECTORY>,LD_LIBRARY_PATH=$ORACLE_HOME/lib,CONF=
<RMAN_MML_CONFIGURATION_FILE_NAME>.conf, RESTORETYPE=file)' debug 1 trace 99;
sql 'alter tablespace <TABLESPACE_NAME> offline immediate';
restore tablespace <TABLESPACE_NAME> from tag "<TAG>";
recover tablespace <TABLESPACE_NAME>;
sql 'alter tablespace <TABLESPACE_NAME> online';
release channel EFGH;
}
```

Example Script

The script, `'restore_tablespace.sh'`, will restore a backup of a tablespace:

```
## USAGE: bash ./restore_tablespace.sh <TABLESPACE_NAME> <TAG>
## EXAMPLE: bash ./restore_tablespace.sh tbs tbs_bk1
```

4.9 Backup Pluggable Database Tablespace (Oracle Database 12c)

The NetApp RMAN MML can be used to create a backup, utilizing Snapshot copies, of a tablespace that is located within a pluggable database.

Note: Oracle Database 12c only.

Required Parameters

The same parameters that are presented in Section 4.1, “Backup Database”, must be included in the RMAN MML configuration file. All others are optional. See Table 6 for possible parameters.

RMAN Commands

The following RMAN commands will create a backup, utilizing Snapshot copies, of a tablespace that is located within a pluggable database:

```
connect target sys/<SYS_PASSWORD>@localhost:1521/<PDB_NAME>

run {
allocate channel EFGH device type sbt_tape
PARMS='SBT_LIBRARY=$ORACLE_HOME/lib/libobk.so
ENV=(BACKUP_DIR=<PATH_TO_RMAN_MML_CONFIGURATION_DIRECTORY>,LD_LIBRARY_PATH=$ORACLE_HOME/lib,CONF=
<RMAN_MML_CONFIGURATION_FILE_NAME>.conf)' debug 1 trace 99;
backup proxy only tablespace <TABLESPACE_NAME> tag "<TAG>";
release channel EFGH;
}
```

Example Script

The script, `'backup_pdb_tablespace.sh'`, will create a backup, utilizing Snapshot copies, of a tablespace that is located within a pluggable database:

```
## USAGE: bash ./backup_pdb_tablespace.sh <PDB_NAME> <PDB_SYS_PASSWD> <TABLESPACE_NAME> <TAG>
## EXAMPLE: bash ./backup_pdb_tablespace.sh plugdb Password1 ptbs ptbs_bk1
```

4.10 Restore Plugable Database Tablespace (Oracle Database 12c)

The NetApp RMAN MML can be used to restore a tablespace (that is located within a pluggable database) from any available backup that was created using the NetApp RMAN MML.

Note: Oracle Database 12c only.

Required Parameters

The same parameters that are presented in Section 4.1, “Backup Database”, must be included in the RMAN MML configuration file. All others are optional. See Table 6 for possible parameters.

RMAN Commands

The following RMAN commands will restore a backup of a tablespace that is located within a pluggable database.

Note: Providing a <TAG> is optional. If no <TAG> is provided, the RMAN MML will restore the latest backup.

```
connect target sys/<SYS_PASSWORD>@localhost:1521/<PDB_NAME>

run {
allocate channel EFGH device type sbt_tape
PARMS='SBT_LIBRARY=$ORACLE_HOME/lib/libobk.so
ENV=(BACKUP_DIR=<PATH_TO_RMAN_MML_CONFIGURATION_DIRECTORY>,LD_LIBRARY_PATH=$ORACLE_HOME/lib,CONF=
<RMAN_MML_CONFIGURATION_FILE_NAME>.conf, RESTORETYPE=file)' debug 1 trace 99;
sql 'alter tablespace <TABLESPACE_NAME> offline immediate';
restore tablespace <TABLESPACE_NAME> from tag "<TAG>";
recover tablespace <TABLESPACE_NAME>;
sql 'alter tablespace <TABLESPACE_NAME> online';
release channel EFGH;
}
```

Example Script

The script, `‘restore_pdb_tablespace.sh’`, will restore a backup of a tablespace that is located within a pluggable database:

```
## USAGE: bash ./restore_pdb_tablespace.sh <PDB_NAME> <PDB_SYS_PASSWD> <TABLESPACE_NAME> <TAG>
## EXAMPLE: bash ./restore_pdb_tablespace.sh plugdb Password1 ptbs ptbs_bk1
```

4.11 Backup Datafile(s)

The NetApp RMAN MML can be used to create a backup, utilizing Snapshot copies, of datafile(s).

Required Parameters

The same parameters that are presented in Section 4.1, “Backup Database”, must be included in the RMAN MML configuration file. All others are optional. See Table 6 for possible parameters.

RMAN Commands

The following RMAN commands will create a backup, utilizing Snapshot copies, of datafile(s):

```
connect target /

sql 'alter system archive log current';

run {
allocate channel EFGH device type sbt_tape
```

```
PARMS='SBT_LIBRARY=$ORACLE_HOME/lib/libobk.so
ENV=(BACKUP_DIR=<PATH_TO_RMAN_MML_CONFIGURATION_DIRECTORY>,LD_LIBRARY_PATH=$ORACLE_HOME/lib,CONF=
<RMAN_MML_CONFIGURATION_FILE_NAME>.conf)' debug 1 trace 99;
backup proxy only datafile <DATAFILE_NUMBER>,<DATAFILE_NUMBER>,… tag "<TAG>";
release channel EFGH;
}
```

Example Script

The script, 'backup_datafiles.sh', will create a backup, utilizing Snapshot copies, of datafile(s):

```
## USAGE: bash ./backup_datafiles.sh <DATAFILE_#>,<DATAFILE_#>,... <TAG>
## EXAMPLE: bash ./backup_datafiles.sh 1,2,6 dbfile_bk1
```

4.12 Restore System Datafile(s)

The NetApp RMAN MML can be used to restore system datafile(s) (e.g. 'system01.dbf', 'sysaux01.dbf', etc.) from any available backup that was created using the NetApp RMAN MML.

Required Parameters

The same parameters that are presented in Section 4.1, "Backup Database", must be included in the RMAN MML configuration file. All others are optional. See Table 6 for possible parameters.

RMAN Commands

The following RMAN commands will restore a backup of system datafile(s).

Note: Providing a <TAG> is optional. If no <TAG> is provided, the RMAN MML will restore the latest backup.

Note: The database must be started in mount mode.

```
connect target /

run {
allocate channel EFGH device type sbt_tape
PARMS='SBT_LIBRARY=$ORACLE_HOME/lib/libobk.so
ENV=(BACKUP_DIR=<PATH_TO_RMAN_MML_CONFIGURATION_DIRECTORY>,LD_LIBRARY_PATH=$ORACLE_HOME/lib,CONF=
<RMAN_MML_CONFIGURATION_FILE_NAME>.conf, RESTORETYPE=file)' debug 1 trace 99;
restore datafile <DATAFILE_NUMBER>,<DATAFILE_NUMBER>,… from tag "<TAG>";
recover datafile <DATAFILE_NUMBER>,<DATAFILE_NUMBER>,…;
release channel EFGH;
}

sql 'alter database open';
```

Example Script

The script, 'restore_system_datafiles.sh', will restore a backup of system datafile(s):

```
## USAGE: bash ./restore_system_datafiles.sh <DATAFILE_#>,<DATAFILE_#>,... <TAG>
## EXAMPLE: bash ./restore_system_datafiles.sh 1,2 dbfile_bk1
## NOTE: Database must be started in "mount" mode.
```

4.13 Restore Nonsystem Datafiles

The NetApp RMAN MML can be used to restore nonsystem datafile(s) (i.e. datafiles for user-created tablespaces, etc.) from any available backup that was created using the NetApp RMAN MML.

Required Parameters

The same parameters that are presented in Section 4.1, “Backup Database”, must be included in the RMAN MML configuration file. All others are optional. See Table 6 for possible parameters.

RMAN Commands

The following RMAN commands will restore a backup of nonsystem datafile(s).

Note: Providing a <TAG> is optional. If no <TAG> is provided, the RMAN MML will restore the latest backup.

```
connect target /

run {
allocate channel EFGH device type sbt_tape
PARMS='SBT_LIBRARY=$ORACLE_HOME/lib/libobk.so
ENV=(BACKUP_DIR=<PATH_TO_RMAN_MML_CONFIGURATION_DIRECTORY>,LD_LIBRARY_PATH=$ORACLE_HOME/lib,CONF=
<RMAN_MML_CONFIGURATION_FILE_NAME>.conf, RESTORETYPE=file)' debug 1 trace 99;
sql 'alter database datafile <DATAFILE_NUMBER>,<DATAFILE_NUMBER>,… offline';
restore datafile <DATAFILE_NUMBER>,<DATAFILE_NUMBER>,… from tag "<TAG>";
recover datafile <DATAFILE_NUMBER>,<DATAFILE_NUMBER>,…;
sql 'alter database datafile <DATAFILE_NUMBER>,<DATAFILE_NUMBER>,… online';
release channel EFGH;
}
```

Example Script

The script, `‘restore_nonsystem_datafiles.sh’`, will restore a backup of nonsystem datafile(s):

```
## USAGE: bash ./restore_nonsystem_datafiles.sh <DATAFILE_#>,<DATAFILE_#>,... <TAG>
## EXAMPLE: bash ./restore_nonsystem_datafiles.sh 5,6,8 dbfile_bk1
```

4.14 Delete Backup

The NetApp RMAN MML can be used to delete any backup that was created using the NetApp RMAN MML.

Required Parameters

The same parameters that are presented in Section 4.1, “Backup Database”, must be included in the RMAN MML configuration file. All others are optional. See Table 6 for possible parameters.

RMAN Commands

The following RMAN commands will delete a backup.

```
connect target /

run {
allocate channel EFGH device type sbt_tape
PARMS='SBT_LIBRARY=$ORACLE_HOME/lib/libobk.so
ENV=(BACKUP_DIR=<PATH_TO_RMAN_MML_CONFIGURATION_DIRECTORY>,LD_LIBRARY_PATH=$ORACLE_HOME/lib,CONF=
<RMAN_MML_CONFIGURATION_FILE_NAME>.conf, DELETETYPE=snap)' debug 1 trace 99;
delete noprompt backup tag "<TAG>";
release channel EFGH;
}
```

Example Script

The script, `‘delete_backup.sh’`, will delete a backup:

```
## USAGE: bash ./delete_backup.sh <TAG>
```

```
## EXAMPLE: bash ./delete_backup.sh test_backup_1
```

4.15 Delete Obsolete Backups Greater than Five Minutes Old

The NetApp RMAN MML can be used to delete any backups that were created using the NetApp RMAN MML, that are both obsolete (i.e newer backups are available) and greater than five minutes old.

Required Parameters

The same parameters that are presented in Section 4.1, “Backup Database”, must be included in the RMAN MML configuration file. All others are optional. See Table 6 for possible parameters.

RMAN Commands

The following RMAN commands will delete obsolete backups that are greater than five minutes old.

```
connect target /

run {
allocate channel EFGH device type sbt_tape
PARMS='SBT_LIBRARY=$ORACLE_HOME/lib/libobk.so
ENV=(BACKUP_DIR=<PATH_TO_RMAN_MML_CONFIGURATION_DIRECTORY>,LD_LIBRARY_PATH=$ORACLE_HOME/lib,CONF=
<RMAN_MML_CONFIGURATION_FILE_NAME>.conf, DELETETYPE=snap)' debug 1 trace 99;
delete noprompt obsolete until time 'sysdate-05/60/24';
release channel EFGH;
}
```

Example Script

The script, ‘delete_obsolete.sh’, will delete obsolete backups that are greater than five minutes old:

```
## USAGE: bash ./delete_obsolete.sh
## NOTE: Deletes obsolete backups that are greater than 5 mins old.
```

4.16 Duplicate Database

The NetApp RMAN MML can be used to duplicate (clone) a database.

Prerequisites

In order to duplicate a database, you must have the following prerequisites completed:

1. You must create an orapwd file for the new (duplicate) database.
   Example command for creating orapwd file (database name = ‘clrmqa0’):

   ```
   orapwd file=$ORACLE_HOME/dbs/orapwclrmqa0 password=Password1 entries=5 force=y
   ```

2. You must create a pfile for the new (duplicate) database.
   Example pfile (database name = ‘clrmqa0’):

   ```
   [oracle@mikeo-ora01]$ vi $ORACLE_HOME/dbs/initclrmqa0.ora
   clrmqa0.__oracle_base='/u01/app/oracle'#ORACLE_BASE set from environment
   ### Basic Configuration Parameters
   ### -------------------------------
   sga_target=600M
   sga_max_size=700M
   ### Traces, Dumps and Passwordfile
   ### -------------------------------
   *.audit_file_dest='/u01/app/oracle/admin/clrmqa0/adump'
   *.audit_trail='db'
   ### Basic Configuration Parameters
   ### -------------------------------
   *.compatible='11.2.0'
   ```

```
*.control_files='/oracntl/clrmqa0/control01.ctl','/oracntl/clrmqa0/control02.ctl'
*.db_block_size=8192
*.db_domain=''
*.db_name='clrmqa0'
*.db_unique_name='clrmqa0'
*.db_recovery_file_dest='/u01/app/oracle/fast_recovery_area/clrmqa0'
*.db_recovery_file_dest_size=4322230272
*.diagnostic_dest='/u01/app/oracle'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=clrmqa0XDB)'
### REDO Logging without Data Guard
### ------------------------------
*.log_archive_dest_1='LOCATION=/oraarch/clrmqa0'
*.log_archive_format='clrmqa0_%t_%s_%r.dbf'
#*.memory_target=10092544000
*.remote_login_passwordfile='EXCLUSIVE'
### System Managed Undo
### --------------------
*.undo_tablespace='UNDOTBS1'
*.undo_management=auto
*.undo_retention=10800

### Duplicate Database
### ---------------------------------------------
# This is only used when you duplicate the database
# on the same host to avoid name conflicts
# In ourexample we have both log members in the sigle location. we can mention multiple pairs
in LOG_FILE_NAME_CONVERT if we have different locations
DB_FILE_NAME_CONVERT=('/oradata1','/oradata1_clrmqa0')
LOG_FILE_NAME_CONVERT=('/oralog','/oralog/clrmqa0')
db_file_multiblock_read_count     = 32
db_files                          = 512

event="10298 trace name context forever, level 1"
event="19392 trace name context forever, level 8" # kgnfs
event="19393 trace name context forever, level 1" # skgnfs
event="19394 trace name context forever, level 8" # skgnfs
event="19396 trace name context forever, level 1" # kgodm
```

3. You must create the directories that you specify in the pfile.

   Example (based on pfile shown above):

```
[oracle@mikeo-ora01]$ mkdir -p /u01/app/oracle/admin/clrmqa0/adump
[oracle@mikeo-ora01]$ mkdir -p /u01/app/oracle/fast_recovery_area/clrmqa0
[oracle@mikeo-ora01]$ mkdir -p /oraarch/clrmqa0
[oracle@mikeo-ora01]$ mkdir -p /oracntl/clrmqa0
[oracle@mikeo-ora01]$ mkdir -p /oralog/clrmqa0
[oracle@mikeo-ora01]$ mkdir -p /oradata1_clrmqa0
```

4. You must start the new (duplicate) database in nomount mode, using the pfile that you just created.

   Example (based on pfile shown above):

```
[oracle@mikeo-ora01]$ export ORACLE_SID=clrmqa0
[oracle@mikeo-ora01]$ sqlplus / as sysdba
SQL> startup force nomount pfile='$ORACLE_HOME/dbs/initclrmqa0.ora';
```

5. You must create an entry in `'tnsnames.ora'` for the new (duplicate) database:

   Example (based on pfile shown above):

```
[oracle@mikeo-ora01]$ vi $ORACLE_HOME/network/admin/tnsnames.ora
CLRMQA0 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = mikeo-ora01)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = clrmqa0)
    )
  )
```

Required Parameters

The same parameters that are presented in Section 4.1, “Backup Database”, must be included in the RMAN MML configuration file. In addition, the following parameter must also be included. All others are optional. See Table 6 for possible parameters.

Table 3) Duplicate database - required parameters.

Item#	Parameter	Description
21	DUPLICATEDB_MOUNTPOINT	When cloning a database, this parameter specifies where you would like the MML to mount the clone of each of the source database volumes. **Note:** Only required if you are performing a clone/duplicate operation.

	Syntax:
	`DUPLICATEDB_MOUNTPOINT=<storage_ip_or_hostname_1>:<sourcedb_volume1>:<clonedb_volume1_mountpoint>;<storage_ip_or_hostname_2>:<sourcedb_volume2>:<clonedb_volume2_mountpoint>`
	
	Example:
	`DUPLICATEDB_MOUNTPOINT=10.61.168.18:sdb:/mml_11g_oradata_clondb1`
	
	Example:
	`DUPLICATEDB_MOUNTPOINT=mikeo_rmqa:rmqa_oradata0:/oracle/rmqa/oradata0_clrmqa0;mikeo_rmqa:rmqa_oradata1:/oracle/rmqa/oradata1_clrmqa0;mikeo_rmqa:rmqa_oradata2:/oracle/rmqa/oradata2_clrmqa0`	

RMAN Commands

If the required Prerequisites have been completed, the following RMAN commands will duplicate a database.

```
connect target sys/<SOURCE_DB_SYS_PASSWD>@<SOURCE_DB_NAME>
connect AUXILIARY /
run {
allocate channel ABCD type sbt_tape
PARMS='SBT_LIBRARY=$ORACLE_HOME/lib/libobk.so
ENV=(BACKUP_DIR=$BACKUP_DIR,LD_LIBRARY_PATH=$ORACLE_HOME/lib,CONF=$CONF,RESTORETYPE=clone)' debug
1 trace 99;

allocate auxiliary channel EFGH type sbt_tape
PARMS='SBT_LIBRARY=$ORACLE_HOME/lib/libobk.so
ENV=(BACKUP_DIR=$BACKUP_DIR,LD_LIBRARY_PATH=$ORACLE_HOME/lib,CONF=$CONF,RESTORETYPE=clone)' debug
1 trace 99;

DUPLICATE TARGET DATABASE TO <CLONE_DB_NAME>
PFILE='<CLONE_DB_PFILE>'
NOFILENAMECHECK
LOGFILE
      GROUP 1 ( '<CLONE_DB_REDO_LOG_DIRECTORY>/redo01.log' ) SIZE 10M REUSE,
      GROUP 2 ( '<CLONE_DB_REDO_LOG_DIRECTORY>/redo02.log' ) SIZE 10M REUSE,
      GROUP 3 ( '<CLONE_DB_REDO_LOG_DIRECTORY>/redo03.log' ) SIZE 10M REUSE;

release channel ABCD;
}
```

Example Script

The script, `'duplicate_database.sh'`, if the required Prerequisites have been completed, will duplicate a database:

```
## USAGE: bash ./duplicate_database.sh <SOURCE_DB_NAME> <SOURCE_DB_SYS_PASSWD> <CLONE_DB_NAME>
<CLONE_DB_PFILE> <CLONE_DB_REDO_LOG_DIRECTORY>
## EXAMPLE: bash ./duplicate_database.sh rmqa Password1 clrmqa0 $ORACLE_HOME/dbs/initclrmqa0.ora
/oracle/rmqa/oraconf/clrmqa0
```

Postrequisites

The following steps should be completed after the database is duplicated, to ensure proper functioning:

1. Comment out (or remove) the following two lines in the duplicate database pfile by adding a '#' to the beginning of the line, as shown:

   Note: Based on example provided above.

   ```
   #DB_FILE_NAME_CONVERT=('/oradata1','/oradata1_clrmqa0')
   #LOG_FILE_NAME_CONVERT=('/oralog','/oralog/clrmqa0')
   ```

2. Create an spfile from the pfile. Refer to Oracle documentation for details.

   For example:

   Note: The database must be shutdown.

   ```
   [oracle@mikeo-ora01]$ sqlplus / as sysdba
   SQL> startup pfile='$ORACLE_HOME/dbs/initclrmqa0.ora'
   SQL> create spfile from pfile='$ORACLE_HOME/dbs/initclrmqa0.ora';
   ```

3. Run `'df -kh'` to discover mounting information about the duplicate database volume(s).

   For example:

   ```
   [oracle@mikeo-ora01]$ df -kh
   mikeo_rmqa:/TAG20140306T174305_NetApp_rmqa_x_0np2gtgj_5_1_20140306172747_CL
       29G   5.0G        23G    18%    /oradata1_clrmqa0
   ```

4. If you are using Linux, update `'/etc/fstab'` to include the new duplicate database volume(s) so that this volume(s) will automatically mount when the system is started. If you are using Solaris, you will update `'/etc/vfstab'` instead.

   For example:

   ```
   [root@mikeo-ora01]$ vi /etc/fstab
   …
   mikeo_rmqa:/TAG20140306T174305_NetApp_rmqa_x_0np2gtgj_5_1_20140306172747_CL /oradata1_clrmqa0
     nfs rsize=65536,wsize=65536,vers=3,hard,nointr,tcp,timeo=600,actimeo=0    0 0
   ```

5. If you are using Linux, update `'/etc/oratab'` to include the new duplicate database. If you are using Solaris, you will update `'/var/opt/oracle/oratab'` instead. Refer to Oracle documentation for details.

   For example:

   ```
   [oracle@mikeo-ora01]$ vi /etc/oratab
   …
   clrmqa0:/u01/app/oracle/product/11.2.0.3/db_1:N
   ```

4.17 Add SnapVault Functionality

All file-level restore operations (the operations in Sections 4.1, 4.5, 4.6, 4.7, 4.8, 4.9, 4.10, 4.11, 4.12, and 4.13) can be performed with added SnapVault functionality. To add SnapVault functionality, add the following parameters to your RMAN MML configuration file. See Table 6 for all possible parameters. See the Installation and Administration Guide, available on the NetApp RMAN MML Community Site, for further details on SnapVault functionality.

Note: All SnapVault storage relationships must have already been configured. See NetApp Documentation for details on SnapVault.

Note: The environment variable, `RESTORETYPE`, must be set to '`file`' for all SnapVault operations.

Table 4) SnapVault - required parameters.

Item#	Parameter	Description
10	NTAP_SNAPVAULT_UPDATE	Determines whether or not the MML will perform a SnapVault update when creating a backup.

	Syntax:
	`NTAP_SNAPVAULT_UPDATE=YES|NO`
	
	Example:
	`NTAP_SNAPVAULT_UPDATE=no`	

11	SNAPVAULT_VOLUMES	Specifies the source and destination volumes for SnapVault backups and restores. **Note:** Only required if you would like to perform SnapVault updates when taking backups or if you would like to have the ability to restore from a SnapVault destination. **Note:** If you are using 7-mode storage, the destination volumes are required to each contain a qtree, and you must specify the qtree here (see examples).

	Syntax:
	`SNAPVAULT_VOLUMES=<source_storage_ip_or_hostname>:<sv source1>,<sv_source2>,<sv_source3>;<dest_storage_ip_or_hostname _1>:<sv_dest1>/<qtreename>;<dest_storage_ip_or_hostname_2>:<sv_dest2>/<qtreename >,<sv_dest3>/<qteename>`
	
	Example:
	`SNAPVAULT_VOLUMES=10.61.173.191:svoradata_src1,svoradata_src2,svoradata_src3;10. 61.173.191:svoradata_dest1/qt1,svoradata_dest2/qt1,svoradata_dest3/qt1`
	(7-mode, all destination volumes residing on the same storage controller)
	
	Example:
	`SNAPVAULT_VOLUMES=10.61.173.191:svoradata_src1,svoradata_src2,svoradata_src3;10. 61.173.192:svoradata_dest1/qt1;10.61.173.193:svoradata_dest2/qt1,svoradata_dest3 /qt1`
	(7-mode, destination volumes residing on different storage controllers)
	
	Example:
	`SNAPVAULT_VOLUMES=mikeo_rmqa:rmqa_oradata0,rmqa_oradata1,rmqa_oradata2;mikeo_ora racnfs:sv_oradata0,sv_oradata1,sv_oradata2`
	(Clustered Data ONTAP, destination volumes all residing on a different SVM within the same cluster)	

12	NTAP_SNAPVAULT_RETENTIONS	Determines the number of NetApp Snapshot copies on the SnapVault secondary (destination) volume that you would like to retain for a given retention policy (i.e. hourly, daily, weekly, monthly). **Note:** Only required if you are utilizing the MML's SnapVault functionality.

	Syntax:
	`NTAP_SNAPVAULT_RETENTIONS =<hourly|daily|weekly|monthly>:<number>`
	
	Example:
	`NTAP_SNAPVAULT_RETENTIONS=hourly:23`	

Item#	Parameter	Description
	Example: `NTAP_SNAPVAULT_RETENTIONS=daily:12`	
13	SNAP_TYPE	Specifies the “type” of snapshot (i.e. hourly, daily, weekly, or monthly) that the RMAN MML will create when taking a snapshot. This corresponds to retention policies on the storage controller

	Syntax:
	`SNAP_TYPE=<hourly|daily|weekly|monthly>`
	
	Examples:
	`SNAP_TYPE=HOURLY`
	`SNAP_TYPE=DAILY`
	`SNAP_TYPE=WEEKLY`
	`SNAP_TYPE=MONTHLY`	

14	NTAP_SNAPVAULT_RETENTION_AGE	Specifies the retention age (in days) for SnapVault Snapshot copies. **Note:** A snapshot will only be deleted if the number of snapshots exceeds the count specified in NTAP_SNAPVAULT_RETENTIONS and the age of the snapshot exceeds the value specified here. **Note:** Only required if you are utilizing the MML’s SnapVault functionality.

	Syntax:
	`NTAP_SNAPVAULT_RETENTION_AGE=<number of days>`
	
	Example:
	`NTAP_SNAPVAULT_RETENTION_AGE=2`	

15	RESTORE_FROM_SNAPVAULT_SECONDARY	Determines whether or not the MML will attempt to restore the database files from a SnapVault destination volume Snapshot. **Note:** When restoring from a SnapVault destination volume Snapshot, the MML will perform a file-level restore using an operating system copy command.

	Syntax:
	`RESTORE_FROM_SNAPVAULT_SECONDARY=yes|no`
	
	Example:
	`RESTORE_FROM_SNAPVAULT_SECONDARY=yes`	

16	NTAP_SNAPVAULT_WAIT	Specifies the time (in minutes) that the MML should wait for the SnapVault update process to complete. **Note:** Default value is 0. A value of 0 directs the MML to wait until the SnapVault update is complete, however long that may be. **Note:** Only required if you are utilizing the MML’s SnapVault functionality.

	Syntax:
	`NTAP_SNAPVAULT_WAIT=<number of minutes>`
	
	Example:
	`NTAP_SNAPVAULT_WAIT=2`	

4.18 Add SnapMirror Functionality

All file-level restore operations (the operations in Sections 4.1, 4.5, 4.6, 4.7, 4.8, 4.9, 4.10, 4.11, 4.12, and 4.13) can be performed with added SnapMirror functionality. To add SnapMirror functionality, add the following parameters to your RMAN MML configuration file. See Table 6 for all possible parameters. See the Installation and Administration Guide, available on the NetApp RMAN MML Community Site, for further details on SnapMirror functionality.

Note: All SnapMirror storage relationships must have already been configured. See NetApp Documentation for details on SnapMirror.

Note: The environment variable, `RESTORETYPE`, must be set to '`file`' for all SnapMirror operations.

Table 5) SnapMirror - required parameters.

Item#	Parameter	Description
17	NTAP_SNAPMIRROR_UPDATE	Determines whether or not the MML will perform a SnapMirror update when creating a backup.

	Syntax:
	`NTAP_SNAPMIRROR_UPDATE=YES|NO`
	
	Example:
	`NTAP_SNAPMIRROR_UPDATE=yes`	

18	SNAPMIRROR_VOLUMES	Specifies the source and destination volumes for SnapMirror backups and restores. Only required if you would like to perform SnapMirror updates when creating backups or if you would like to have the ability to restore from a SnapMirror destination.

	Syntax:
	`SNAPMIRROR_VOLUMES=<source_storage_ip_or_hostname>:<sm source1>,<sm_source2>,<sm_source3>;<dest_storage_ip_or_hostname _1>:<sm_dest1>;<dest_storage_ip_or_hostname_2>:<sm_dest2>,<sm_dest3>`
	
	Example:
	`SNAPMIRROR_VOLUMES=10.61.173.191:smoradata_source,smoradata_src2,smoradata_src3; 10.61.173.191:smoradata_dest,smoradata_dest2,smoradata_dest3`
	(7-mode, all destination volumes residing on the same storage controller)
	
	Example:
	`SNAPMIRROR_VOLUMES=10.61.173.191:smoradata_source,smoradata_src2,smoradata_src3; 10.61.173.192:smoradata_dest;10.61.173.193:smoradata_dest2,smoradata_dest3`
	(7-mode, destination volumes residing on different storage controllers)
	
	Example:
	`SNAPMIRROR_VOLUMES=mikeo_rmqa:rmqa_oradata0,rmqa_oradata1,rmqa_oradata2;mikeo_or aracnfs:sm_oradata0,sm_oradata1,sm_oradata2`
	(Clustered Data ONTAP, destination volumes all residing on a different SVM within the same cluster)	

19	RESTORE_FROM_SNAPMIRROR_SECONDARY	Determines whether or not the MML will attempt to restore the database files from a SnapMirror destination volume. **Note:** When restoring from a SnapMirror destination volume, the MML will perform a file-level restore using an operating system copy command.
	Syntax:	

Item#	Parameter	Description
	RESTORE_FROM_SNAPMIRROR_SECONDARY=YES|NO Example: RESTORE_FROM_SNAPMIRROR_SECONDARY=no	
20	NTAP_SNAPMIRROR_WAIT	Specifies the time (in minutes) that the MML should wait for the SnapMirror update process to complete. **Note:** Default value is 0. A value of 0 directs the MML to wait until the SnapMirror update is complete, however long that may be. **Note:** Only required if you are utilizing the MML's SnapMirror functionality.
	Syntax: NTAP_SNAPMIRRORT_WAIT=<number of minutes> Example: NTAP_SNAPMIRROR_WAIT=2	

5 Log Location

If you are using Oracle Database 11g or 12c, the RMAN MML log will be located at the following location:

```
$ORACLE_BASE/diag/rdbms/<DATABASE_NAME>/<DATABASE_INSTANCE>/trace/sbtio.log
```

If you are using Oracle Database 10g, the RMAN MML log will be located at the following location:

```
$ORACLE_BASE/admin/<DATABASE_NAME>/udump/sbtio.log
```

Appendixes

Table 6) Parameters for NetApp RMAN MML configuration file.

Item#	Parameter	Description
1	FILER	Lists the storage controllers and their respective usernames and passwords.
	Syntax: FILER=<storage_ip_or_hostname_1>:<username>/<password>;<storage_ip_or_hostname_2>:<username>/<password> Example: FILER=mikeo_rmqa:vsadmin/Netapp12345	
2	FILERPASS_ENCRYPTED	Denotes whether or not the provided storage system password is encrypted. **Note:** If the password is not encrypted, MML will automatically encrypt the password upon its first run. **Note:** An encrypted password can be created using 'mml_pass_encrypt_decrypt_x64.o'.
	Example: FILERPASS_ENCRYPTED=NO	

Item#	Parameter	Description
	(if password is not encrypted) Example: `FILERPASS_ENCRYPTED=YES` (if password is encrypted)	
3	VOLUMES	Lists the NetApp volumes that you want to backup (Snapshot), restore (SnapRestore), and/or duplicate.

	Syntax:
	`VOLUMES=<storage_ip_or_hostname_1>:<volumename1>,<volumename2>;<storage_ip_or_hostname_2>:<volumename>`
	
	Example:
	`VOLUMES=mikeo_rmqa:rmqa_oradata0,rmqa_oradata1,rmqa_oradata2`	

4	PROTOCOL	Should be set to 'nfs'. This version of the MML only supports NFS.
	Syntax: `PROTOCOL=nfs`	
5	DB_MOUNTPOINT	Lists the NetApp volumes that you want to backup (Snapshot), restore (Snaprestore), and/or duplicate, and their corresponding mountpoints on the Oracle Database server.

	Syntax:
	`DB_MOUNTPOINT=<storage_ip_or_hostname_1>:<volumename1>:<mountpoint1>,<mountpoint2>;<storage_ip_or_hostname_2>:<volumename2>:<mountpoint3>;<storage_ip_or_hostname_3>:<volumename3>:<mountpoint4>`
	
	Example:
	`DB_MOUNTPOINT=mikeo_rmqa:rmqa_oradata0:/oracle/rmqa/oradata0;mikeo_rmqa:rmqa_oradata1:/oracle/rmqa/oradata1;mikeo_rmqa:rmqa_oradata2:/oracle/rmqa/oradata2`	

6	CONTROL_VOLUMES	Specifies the volume(s) to which controlfiles will be backed up if controlfile autobackup is configured. **Note:** Only required if you wish to use controlfile autobackup.

	Syntax:
	`CONTROL_VOLUMES=<storage_ip_or_hostname_1>:<controlvolunme1>`
	
	Example:
	`CONTROL_VOLUMES=mikeo_rmqa:rmqa_oractrl`	

7	CONTROL_MOUNTPOINT	Specifies the volume(s) to which controlfiles will be backed up if controlfile autobackup is configured, and their corresponding mountpoint(s). **Note:** Only required if you wish to use controlfile autobackup.
	Syntax: `CONTROL_MOUNTPOINT=<storage_ip_or_hostname_1>:<controlvolunme1>:<controlvolumemountpoint1>`	

Item#	Parameter	Description
	Example: `CONTROL_MOUNTPOINT=mikeo_rmqa:rmqa_oractrl:/rmqa_oractrl`	
8	VALIDATE_VOLUMES	Specifies whether or not you want the RMAN MML, for each operation (backup, restore, duplicate), to validate whether or not the data that it is about to perform an operation on is located on compatible NetApp storage. **Note:** Optional.

	Example:
	`VALIDATE_VOLUMES=DATA`
	(if you would like the RMAN MML to perform the validation)
	
	Example:
	`VALIDATE_VOLUMES=`
	(if you do not want RMAN MML to perform the validation)	

9	SNAME	The name specified here will be used as part of the name of each snapshot that is created by the RMAN MML. Snapshot copies will be deleted according to this naming convention. **Note:** This is a required parameter for SnapVault. **Note:** It is recommended to use the database name.

	Syntax:
	`SNAME=<text>`
	
	Example:
	`SNAME=rmqa`	

10	NTAP_SNAPVAULT_UPDATE	Determines whether or not the MML will perform a SnapVault update when creating a backup.

	Syntax:
	`NTAP_SNAPVAULT_UPDATE=YES|NO`
	
	Example:
	`NTAP_SNAPVAULT_UPDATE=no`	

11	SNAPVAULT_VOLUMES	Specifies the source and destination volumes for SnapVault backups and restores. **Note:** Only required if you would like to perform SnapVault updates when taking backups or if you would like to have the ability to restore from a SnapVault destination. **Note:** If you are using 7-mode storage, the destination volumes are required to each contain a qtree, and you must specify the qtree here (see examples).
	Syntax: `SNAPVAULT_VOLUMES=<source_storage_ip_or_hostname>:<sv source1>,<sv_source2>,<sv_source3>;<dest_storage_ip_or_hostname _1>:<sv_dest1>/<qtreename>;<dest_storage_ip_or_hostname_2>:<sv_dest2>/<qtreename >,<sv_dest3>/<qteename>` Example:	

Item#	Parameter	Description
	SNAPVAULT_VOLUMES=10.61.173.191:svoradata_src1,svoradata_src2,svoradata_src3;10.61.173.191:svoradata_dest1/qt1,svoradata_dest2/qt1,svoradata_dest3/qt1 (7-mode, all destination volumes residing on the same storage controller) Example: SNAPVAULT_VOLUMES=10.61.173.191:svoradata_src1,svoradata_src2,svoradata_src3;10.61.173.192:svoradata_dest1/qt1;10.61.173.193:svoradata_dest2/qt1,svoradata_dest3/qt1 (7-mode, destination volumes residing on different storage controllers) Example: SNAPVAULT_VOLUMES=mikeo_rmqa:rmqa_oradata0,rmqa_oradata1,rmqa_oradata2;mikeo_oracnfs:sv_oradata0,sv_oradata1,sv_oradata2 (Clustered Data ONTAP, destination volumes all residing on a different SVM within the same cluster)	
12	NTAP_SNAPVAULT_RETENTIONS	Determines the number of NetApp Snapshot copies on the SnapVault secondary (destination) volume that you would like to retain for a given retention policy (i.e. hourly, daily, weekly, monthly). **Note:** Only required if you are utilizing the MML's SnapVault functionality.
	Syntax: NTAP_SNAPVAULT_RETENTIONS =<hourly|daily|weekly|monthly>:<number> Example: NTAP_SNAPVAULT_RETENTIONS=hourly:23 Example: NTAP_SNAPVAULT_RETENTIONS=daily:12	
13	SNAP_TYPE	Specifies the "type" of snapshot (i.e. hourly, daily, weekly, or monthly) that the RMAN MML will create when taking a snapshot. This corresponds to retention policies on the storage controller
	Syntax: SNAP_TYPE=<hourly|daily|weekly|monthly> Examples: SNAP_TYPE=HOURLY SNAP_TYPE=DAILY SNAP_TYPE=WEEKLY SNAP_TYPE=MONTHLY	
14	NTAP_SNAPVAULT_RETENTION_AGE	Specifies the retention age (in days) for SnapVault Snapshot copies. **Note:** A snapshot will only be deleted if the number of snapshots exceeds the count specified in NTAP_SNAPVAULT_RETENTIONS and the age of the snapshot exceeds the value specified here. **Note:** Only required if you are utilizing the MML's SnapVault functionality.
	Syntax: NTAP_SNAPVAULT_RETENTION_AGE=<number of days>	

Item#	Parameter	Description
	Example: `NTAP_SNAPVAULT_RETENTION_AGE=2`	
15	RESTORE_FROM_SNAPVAULT_SECONDARY	Determines whether or not the MML will attempt to restore the database files from a SnapVault destination volume Snapshot. **Note:** When restoring from a SnapVault destination volume Snapshot, the MML will perform a file-level restore using an operating system copy command.

	Syntax:
	`RESTORE_FROM_SNAPVAULT_SECONDARY=yes|no`
	
	Example:
	`RESTORE_FROM_SNAPVAULT_SECONDARY=yes`	

16	NTAP_SNAPVAULT_WAIT	Specifies the time (in minutes) that the MML should wait for the SnapVault update process to complete. **Note:** Default value is 0. A value of 0 directs the MML to wait until the SnapVault update is complete, however long that may be. **Note:** Only required if you are utilizing the MML's SnapVault functionality.

	Syntax:
	`NTAP_SNAPVAULT_WAIT=<number of minutes>`
	
	Example:
	`NTAP_SNAPVAULT_WAIT=2`	

17	NTAP_SNAPMIRROR_UPDATE	Determines whether or not the MML will perform a SnapMirror update when taking a backup.

	Syntax:
	`NTAP_SNAPMIRROR_UPDATE=YES|NO`
	
	Example:
	`NTAP_SNAPMIRROR_UPDATE=yes`	

18	SNAPMIRROR_VOLUMES	Specifies the source and destination volumes for SnapMirror backups and restores. **Note:** Only required if you would like to perform SnapMirror updates when taking backups or if you would like to have the ability to restore from a SnapMirror destination.

	Syntax:
	`SNAPMIRROR_VOLUMES=<source_storage_ip_or_hostname>:<sm source1>,<sm_source2>,<sm_source3>;<dest_storage_ip_or_hostname_1>:<sm_dest1>;<dest_storage_ip_or_hostname_2>:<sm_dest2>,<sm_dest3>`
	
	Example:
	`SNAPMIRROR_VOLUMES=10.61.173.191:smoradata_source,smoradata_src2,smoradata_src3;10.61.173.191:smoradata_dest,smoradata_dest2,smoradata_dest3`
	(7-mode, all destination volumes residing on the same storage controller)
	
	Example:
	`SNAPMIRROR_VOLUMES=10.61.173.191:smoradata_source,smoradata_src2,smoradata_src3;`	

Item#	Parameter	Description

	`10.61.173.192:smoradata_dest;10.61.173.193:smoradata_dest2,smoradata_dest3`
	(7-mode, destination volumes residing on different storage controllers)
	
	Example:
	`SNAPMIRROR_VOLUMES=mikeo_rmqa:rmqa_oradata0,rmqa_oradata1,rmqa_oradata2;mikeo_oraracnfs:sm_oradata0,sm_oradata1,sm_oradata2`
	(Clustered Data ONTAP, destination volumes all residing on a different SVM within the same cluster)	

19	RESTORE_FROM_SNAPMIRROR_SECONDARY	Determines whether or not the MML will attempt to restore the database files from a SnapMirror destination volume. **Note:** When restoring from a SnapMirror destination volume, the MML will perform a file-level restore using an operating system copy command.

	Syntax:
	`RESTORE_FROM_SNAPMIRROR_SECONDARY=YES|NO`
	
	Example:
	`RESTORE_FROM_SNAPMIRROR_SECONDARY=no`	

20	NTAP_SNAPMIRROR_WAIT	Specifies the time (in minutes) that the MML should wait for the SnapMirror update process to complete. **Note:** Default value is 0. A value of 0 directs the MML to wait until the SnapMirror update is complete, however long that may be. **Note:** Only required if you are utilizing the MML's SnapMirror functionality.

	Syntax:
	`NTAP_SNAPMIRRORT_WAIT=<number of minutes>`
	
	Example:
	`NTAP_SNAPMIRROR_WAIT=2`	

21	DUPLICATEDB_MOUNTPOINT	When cloning a database, this parameter specifies where you would like the MML to mount the clone of each of the source database volumes. **Note:** Only required if you are performing a clone/duplicate operation.

	Syntax:
	`DUPLICATEDB_MOUNTPOINT=<storage_ip_or_hostname_1>:<sourcedb_volume1>:<clonedb_volume1_mountpoint>;<storage_ip_or_hostname_2>:<sourcedb_volume2>:<clonedb_volume2_mountpoint>`
	
	Example:
	`DUPLICATEDB_MOUNTPOINT=10.61.168.18:sdb:/mml_11g_oradata_clondb1`
	
	Example:
	`DUPLICATEDB_MOUNTPOINT=mikeo_rmqa:rmqa_oradata0:/oracle/rmqa/oradata0_clrmqa0;mikeo_rmqa:rmqa_oradata1:/oracle/rmqa/oradata1_clrmqa0;mikeo_rmqa:rmqa_oradata2:/oracle/rmqa/oradata2_clrmqa0`	

22	HTTPS_PORT	SSL is the default protocol used by the MML to communicate with the storage controller. The default port is 443. If you are not using port 443, you can specify your port number here.

Item#	Parameter	Description
		Note: Only required if you are using a port other than 443 for SSL.

	Syntax:
	`HTTPS_PORT=<portnumber>`
	
	Example:
	`HTTPS_PORT=444`	

Version History

Version	Date	Document Version History
Version 1.0	April 2014	Initial release.

Refer to the Interoperability Matrix Tool (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

NetApp provides no representations or warranties regarding the accuracy, reliability, or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information in this document is distributed AS IS, and the use of this information or the implementation of any recommendations or techniques herein is a customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. This document and the information contained herein may be used solely in connection with the NetApp products discussed in this document.

Go further, faster®

www.netapp.com