**Procedures for Hot and Cold Backups and Restores with RMAN using LiteSpeed Engine for Oracle**

The basic RMAN procedures for full database backups and restores are the same with and without LiteSpeed, but when restoring it is necessary to use SBT\_TAPE channels. This is a critical point in a restore if RMAN’s recovery catalog is not available. The recovery catalog is often stored in the database controlfile, and if the controlfile itself is part of the restore, then the restore steps for both the pfile and controlfile require specific allocations of SBT\_TAPE channels. Once the controlfile has been restored the recovery catalog will be intact but it may be necessary to continue to allocate the SBT\_TAPE channel in the datafile restore and recovery steps.

This solution note provides example of backing up and restoring databases in mounted and open states, and concludes with comments on:

* How to restore from an old LiteSpeed RMAN backup that is expired
* How to restore from an old LiteSpeed backup that is no longer present in RMAN’s recovery catalog
* Clarification on the right time to remove a LiteSpeed backup from RMAN’s recovery catalog

The backup and restore commands are just examples. Certain commands pertaining to Oracle and RMAN may need to be changed depending on exactly what the customer needs to do. The key points for LiteSpeed are pointed out and should be heeded despite minor modifications.

To do a full database RMAN restore of a database that cannot be mounted or which RMAN cannot connect to for some reason, one must have the DBID of the database being restored before starting the restore procedure. RMAN needs the DBID to connect to the database. It is easily obtained from a functional database by starting RMAN when the database is mounted or open. RMAN will report the DBID when it automatically connects to the database whose SID is given by environment variable ORACLE\_SID. Whether using LiteSpeed or not, DBAs should record the DBIDs of databases they are backing up using RMAN.

***Cold Backup and Restore, Full Database, RMAN without LiteSpeed***

This procedure, provided for contrast with the subsequent procedure which uses LiteSpeed, backs up a mounted database in NOARCHIVELOG mode using native RMAN backup capability without LiteSpeed, and then does a restore from the backup, including the controlfile, to the same database. It uses RMAN’s controlfile autobackup capability to simplify restoration of the controlfile. The example supposes that the recovery catalog is maintained in the controlfile and that, prior to the restore, the controlfile has not been damaged, so the recovery catalog information it contains is fully intact. Thus RMAN can determine from its recovery catalog whether the backup was made using a DISK or SBT\_TAPE channel and will automatically allocate a channel of the correct type for reading from the backup file. If the controlfile had been damaged, it would be necessary to use an additional step to restore the controlfile from the backup, and a DISK channel should be explicitly allocated for that step of the restore in an RMAN run block, in order to assure RMAN uses the correct device type to read the controlfile backup. An example of allocating channels in run blocks is included in the example of a hot backup and restore.

// Configure RMAN. Assume Fast Recovery Area (FRA) is used, so

// no ‘format’ parameters needed.

RMAN>configure controlfile autobackup on;

RMAN>configure default device type to disk;

// Perform the backup. No archivelog backup, since this is a cold backup.

// Database is in NOARCHIVELOG mode. The shutdown and startup commands

// can also be performed from the RMAN prompt, but they are actually SQL

// commands that get passed to the SQL command parser. This shows them

// executed from SQL\*Plus after logging in with ‘/ as sysdba’.

SQL> shutdown immediate;

SQL> startup mount;

SQL> alter database noarchivelog;

RMAN> backup database;

// Go back to archivelog mode, if desired

SQL> alter database archivelog;

// Now some time passes, and it is desired to return the database to the same

// state from the backup. To be able to restore from a backup, RMAN must

// be able to display the backup in its list of backups which can be obtained by

// typing “list backupset” at the RMAN command prompt. For this example,

// RMAN stores its backup list information in the controlfile. If we didn’t restore

// the controlfile itself from the backup first, and this were an old backup,

// we might have to take steps to update RMAN’s recorded status of the

// backup to “available,” or even to add the backup to the recovery catalog in

// the controlfile with RMAN’s “catalog” command. However, since we are

// restoring the controlfile with the backup, we are actually restoring the

// recovery catalog to its original state along with the datafile, and since we

// restore the controlfile first and know the backup will be the last backup

// in the controlfile after the controlfile is restored, we don’t have to worry

// the backup’s status in the catalog.

// Restore the database, starting with the controlfile

SQL> startup nomount;

C:> rman (i.e. start RMAN from the command line with no parameters)

RMAN> set dbid <DBID of the database>;

RMAN> connect target;

RMAN> restore controlfile from autobackup;

// Mount the database and run the restore command.

SQL> alter database mount;

RMAN> restore database;

RMAN> exit;

// The “resetlogs” is needed when opening the database at this point. But once that

// command is issued, the same instance of the database cannot be restored again

// from old backups. Thus, it is important to perform another backup of the

// database immediately after this line has successfully been executed.

SQL> alter database open resetlogs;

SQL> shutdown immediate;

SQL> startup mount;

C:> rman target /

RMAN> backup database;

RMAN> <delete the old backup which can no longer be restored from, if desired>

This procedure does the same tasks as the previous example, but uses LiteSpeed for the backup and restore. Differences from the above procedure are in **bold**. The main thing to note is allocation of SBT\_TAPE channels for restore steps.

**// Configure LiteSpeed if needed.**

**C:> cd $ORACLE\_HOME/Quest/LiteSpeed**

**C:> leo configure**

**C:> etc. …continue until finished configuring LiteSpeed**

// Configure RMAN. Backups to SBT\_TAPE device type do not use the

// FRA, so ‘format’ parameters are used to place the backup in a specific

// location, otherwise LiteSpeed places the backups in a default location.

// Note, Oracle 9i needs “SBT\_TAPE” instead of “sbt” for the device

// type. Later versions of Oracle accept the “sbt” abbreviation.

// If using parallelism greater than 1, additional channels up to the degree of

// parallelism would need to be configured here in the same way, or all can

// be allocated with format parameters in RMAN run blocks. Arbitrarily place

// LiteSpeed backups in the directory “e:\backups” for the example.

RMAN>configure controlfile autobackup on;

RMAN>**configure default device type to sbt;**

RMAN>**configure channel 1 device type sbt format ‘e:\backups\%U.leo’;**

// Perform the backup. No archivelog backup, since this is a cold backup.

// Database is in NOARCHIVELOG mode. The shutdown and startup commands

// can also be performed from the RMAN prompt, but they are actually SQL

// commands that get passed to the SQL command parser. This shows them

// executed from SQL\*Plus after logging in with ‘/ as sysdba’.

SQL> shutdown immediate;

SQL> startup mount;

SQL> alter database noarchivelog;

RMAN> backup database;

// Go back to archivelog mode, if desired

SQL> alter database archivelog;

// Now some time passes, and it is desired to return the database to the same

// state from the backup. To be able to restore from a backup, RMAN must

// be able to display the backup in its list of backups which can be obtained by

// typing “list backupset” at the RMAN command prompt. For this example,

// RMAN stores its backup list information in the controlfile. If we didn’t restore

// the controlfile itself from the backup first, and this were an old backup,

// we might have to take steps to update RMAN’s recorded status of the

// backup to “available,” or even to add the backup to the recovery catalog in

// the controlfile with RMAN’s “catalog” command. However, since we are

// restoring the controlfile with the backup, we are actually restoring the

// recovery catalog to its original state along with the datafile, and since we

// restore the controlfile first and know the backup will be the last backup

// in the controlfile after the controlfile is restored, we don’t have to worry

// the backup’s status in the catalog.

// Restore the database, starting with the controlfile

SQL> startup nomount;

C:> rman (i.e. start RMAN from the command line with no parameters)

RMAN> set dbid <DBID of the database>;

RMAN> connect target;

RMAN> **run{ allocate channel t1 device type sbt;**

restore controlfile from autobackup;

**}**

// Mount the database and run the restore command.

SQL> alter database mount;

RMAN> **run{ allocate channel t1 device type sbt;**

restore database;

**}**

RMAN> exit;

// The “resetlogs” is needed when opening the database at this point. But once that

// command is issued, the same instance of the database cannot be restored again

// from old backups. Thus, it is important to perform another backup of the

// database immediately after this line has successfully been executed.

SQL> alter database open resetlogs;

SQL> shutdown immediate;

SQL> startup mount;

C:> rman target /

RMAN> backup database;

RMAN> <delete the old backup which can no longer be restored from, if desired>

***Hot Backup and Restore, Full Database, RMAN Including LiteSpeed***

The following procedure sets up a database for archivelog mode and performs a hot backup (i.e. a backup while the database is open) using RMAN with LiteSpeed. It then simulates a catastrophe by assuming deletion or corruption of all the key database files except the spfile. It restores the database from the backup. The procedure assumes the recovery catalog is stored in the controlfile rather than a separate recovery catalog database.

The example restores the datafiles to new locations, and so this can be used as a model for restoring to a database residing on a different machine. In the latter case, though, the database name on both machines must be identical for the procedure to work. (If the databases have different names, an export/import would be needed to transfer data from one database to the other.)

This procedure uses commands that provide the paths to the backupset that contains the spfile and controlfile backups, so there is no dependency on the contents of the spfile or controlfile before they have been restored. The recovery catalog tells which backuppieces contain the spfile and controlfile backups. If the catalog is in the control file and that is part of the restore, that information may be lacking. In that case, you would have a collection of backup files to restore from. If you don’t know which contains the spfile or controlfile, you can just try them until you find the right one – Oracle simply gives an error if there is no spfile or controlfile in the backuppiece whose path is provided.

Create database, SID=10gR26. No Fast Recovery Area. DBID = 205007197 Then:

SQL> shutdown immediate;

SQL> startup mount;

SQL> alter database archivelog;

SQL> alter database open;

SQL> alter system switch logfile;

SQL> alter system switch logfile;

SQL> alter system switch logfile;

SQL> shutdown immediate;

SQL> startup;

// Configure LiteSpeed if needed.

C:> cd $ORACLE\_HOME/Quest/LiteSpeed

C:> leo configure

C:> etc. …continue with LiteSpeed configuration.

// Configure RMAN for a LiteSpeed backup, placing backup files in the

// directory “f:\temp”. If using parallelism greater than one, additional

// SBT\_TAPE channels would need to be configured, with similar

// “format” parameters.

RMAN> configure default device type to sbt;

RMAN> configure channel 1 device type sbt format 'f:\temp\%U.leo';

// With the database in archivelog mode, automatic controlfile backup off.

// Because this is a hot backup, the archived redo logs have to be included

// in the backup.

RMAN> backup database plus archivelog;

// In a test run, these four backup pieces were created and will be referred to in

// subsequent steps:

// f:\temp\01lof6fg\_1\_1.leo (archivelog, seq 2,3,4,5,6)

// f:\temp\02lof6fi\_1\_1.leo (4 datafiles)

// f:\temp\03lof6g2\_1\_1.leo (controlfile and spfile)

// f:\temp\04lof6g5\_1\_1.leo (archivelog, seq 7)

// Shutdown database

SQL> shutdown immediate;

// To simulate a disaster, delete all the 10gR26 controlfiles and datafiles, and the

// pfile in %ORACLE\_HOME%\database directory.

// Restore to database 10gR26, DBID = 205007197 (the same database)

sqlplus / as sysdba

SQL> startup nomount;

// First restore the pfile. The pfile will be used to create an spfile in a subsequent

// step. The database will not start at all, even in “nomount” mode, unless an spfile

// is present. That’s why we didn’t delete the spfile above. If no spfile is available

// or it is corrupted, then the first step in the restore is to find a backup of the spfile

// which was made independently of the RMAN backup, which you can just copy

// into place and use to start up the database in nomount mode for the following

// steps. If you don’t have such a backup of the spfile, then you have to create one

// using the recommended Oracle procedure.

C:> rman target /

RMAN> set dbid = 205007197

RMAN> run{

RMAN> allocate channel c1 device type sbt;

RMAN> restore spfile to pfile 'c:\oraclesoftware\versions\10.2.0\oracle\product\db\_1\database\init10gR26.ora'

RMAN> from 'f:\temp\03lof6g2\_1\_1.leo';

RMAN> }

// Create an spfile from the pfile. This makes sure that the database will start up

// with the correct information in the future, in a state consistent with the rest of

// the backup being restored from. It is possible to overwrite the spfile directly

// with the restore spfile command above. We restored to the pfile first, to create

// the pfile for possible editing. Some people do not want a pfile around and

// would create the spfile directly, skipping the restore to pfile step.

SQL>create spfile from pfile;

C:> rman target /

RMAN> set dbid = 205007197

RMAN> run{

RMAN> allocate channel c1 device type sbt;

RMAN> restore controlfile from 'f:\temp\03lof6g2\_1\_1.leo';

RMAN> }

// Restore the datafiles to a different location, E:\10gR26. (If we were restoring

// to the original location, we would omit the “set newname” and “switch datafile

// all” commands.)

SQL> alter database mount;

C:> rman target /

RMAN>run{

RMAN>allocate channel c1 device type sbt;

RMAN>set newname for datafile 'f:\10gr26\10gr26\users01.dbf' to 'e:\10gr26\users01.dbf';

RMAN>set newname for datafile 'f:\10gr26\10gr26\undotbs01.dbf' to 'e:\10gr26\undotbs01.dbf';

RMAN>set newname for datafile 'f:\10gr26\10gr26\system01.dbf' to 'e:\10gr26\system01.dbf';

RMAN>set newname for datafile 'f:\10gr26\10gr26\sysaux01.dbf' to 'e:\10gr26\sysaux01.dbf';

RMAN>restore database;

RMAN>switch datafile all;

RMAN>recover database;

RMAN>}

// At this point, expect a possible error like:

// RMAN-03002: failure of recover command at 09/21/2010 15:45:58

// RMAN-11003: failure during parse/execution of SQL statement: alter database

// recover logfile

// 'C:\ORACLESOFTWARE\VERSIONS\10.2.0\ORACLE\PRODUCT\DB\_1\

// RDBMS\ARC00007\_0730306528.001'

// ORA-00310: archived log contains sequence 7; sequence 8 required

// ORA-00334: archived log:

// 'C:\ORACLESOFTWARE\VERSIONS\10.2.0\ORACLE\PRODUCT\DB\_1\

// RDBMS\ARC00007\_0730306528.001'

// The error occurs because the online redo logs are not included in the recovery

// operation. Online redo logs are never backed up by RMAN, by design, and

// hence the LiteSpeed backup does not contain them either. The restore is

// to the time of the last backup. Despite the possible appearance of the error

// message above, the database is in a consistent state and can be opened.

// The following command will succeed without an error and is the end of the

// recovery operation.

// Due to use of “resetlogs”, another backup should be made immediately.

SQL>alter database open resetlogs;

*Restoring from Very Old LiteSpeed Backups*

RMAN retains its records of old backups indefinitely. When a backup expires according to the retention policy, an examination of backups through RMAN’s “list backupset” or “list backup” command simply shows the backup has status “expired,” but the record of the backup remains in the recovery catalog. If the backup is “expired,” a restore can still be done from that backup, as long as the backup files can still be accessed at their original location. For a LiteSpeed backup to target “disk”, the backup files must be in the location RMAN reports them as being in if a path is shown, or else must be in LiteSpeed’s default backup directory if RMAN shows no path for the backup files. (If no path is shown but the ‘backupdir’ parameter was present in the param.txt file at the time of the backup, the parameter must also be present in the param.txt for the restore, and the files must be placed in the directory specified by the backupdir parameter.) If it was a backup to TSM or Netbackup target, the files must be on the TSM or Netbackup server or tape library in the location LiteSpeed originally sent them to. If the backup files are present in their original locations but the backup is shown as “expired,” then *make sure the default device type is set to SBT\_TAPE*, and take the backups back to “available” status by executing RMAN’s “crosscheck” command on the backupset(s) you want to be able to restore from.

If your old LiteSpeed backups were to target ‘disk’ and recent backups were to target ‘tsm’ or ‘netbackup’ or *vice versa,* then you need to temporarily reconfigure LiteSpeed for the same target type used to make the older backup. Otherwise, LiteSpeed will not be able to find the old backups. If an RMAN crosscheck command has to be performed to bring the backups to “available” status, the temporary LiteSpeed reconfiguration must be done before running the crosscheck command.

RMAN records can be intentionally deleted from the recovery catalog with RMAN commands such as “change backupset <number> delete,” and some backup scripts run these commands. If a record of an old backupset is removed from RMAN’s recovery catalog, the only way to restore from it is to tell RMAN to put it back in the catalog again, using RMAN’s “catalog” command. However, **the RMAN “catalog” command only works for backups made with device type DISK, not SBT\_TAPE. The implication is, if you want to restore from a LiteSpeed backup, then either the backup must currently be in the existing recovery catalog (either with status “available” or “expired”), or else the recovery catalog must be restored to the older date as well, before restoring from the old LiteSpeed backup.** Usually recovery catalogs are maintained in the controlfile. In this case, simply restoring the controlfile from the old backup before restoring the datafiles ensures that the backup is in the recovery catalog and RMAN will be able to restore from it. If there is a separate recovery catalog or you do not want to restore the controlfile, then you must not have RMAN delete its records of old backups until you are 110% certain you will never need to restore from them.