11. Oracle Recovery Manager Overview and Configuration.

Abstract: This lesson provides an overview of RMAN, including the capabilities and components of the RMAN tool. The RMAN utility attempts to move away from the highly customized OS backup scripts (user-managed) discussed in earlier lessons, to a highly standardized backup and recovery process. Thus, as of Oracle version 8, you can reduce backup and recovery mistakes associated with the highly customized OS backup scripts used before RMAN's release. In this lesson, you will walk through a practical example of connecting to the RMAN utility without using the optional, but recommended, recovery catalog. We will also demonstrate multiple ways to configure the RMAN environment to automate and set up manual RMAN settings. Each of the topics covered in this lesson is important; you can apply this knowledge to your real-world DBA work in addition to your preparations for the test. When you know how to use the RMAN repository, channel allocation, and MML, and you can configure the RMAN environment, you will be better able to utilize RMAN in your organization.

Contents

Objectives:

- Identify the features and components of RMAN.
- Describe the RMAN repository and control file usage.
- Describe channel allocation.
- Describe the Media Management Library interface.
- Connect to RMAN without the recovery catalog.
- Configure the RMAN environment.

1. Identifying Features of Oracle Recovery Manager (RMAN)

Oracle Recovery Manager (*RMAN*) has many features that can be used to facilitate the backup and recovery process. This tool comes in both GUI and command-line versions. In general, RMAN performs and standardizes the backup and recovery process, and by doing this, it can reduce mistakes made by DBAs during this process. A list of some of the major RMAN features follows:

Backs up databases, tablespaces, data files, control files, and archived logs

The RMAN tool is capable of backing up Oracle databases in multiple ways to allow for flexibility in backup and recovery methods.

Compresses backups by determining which blocks have changed and backing up only those blocks

One way RMAN improves performance of the backup is by compressing backups. RMAN identifies blocks that have changed and only backs up those blocks. Also, empty blocks are not backed up.

Performs incremental backups

RMAN has the ability to perform incremental and full backups. Incremental backups include only the changes that have been made since the last backup. This type of backup can improve performance by allowing you to take a full backup one day a week and incremental backups on the rest of the days.

Provides scripting capabilities to combine tasks

One way RMAN can improve the efficiency of your backup, restoration, and recovery operations is by allowing RMAN commands to be scripted. The scripts can consist of multiple RMAN commands that can be stored within the recovery catalog. These scripts can be called and executed to perform tasks repetitively.

Logs backup operations

RMAN has the ability to log the status of backups as they progress. This information is stored in log and trace files.

Integrates with third-party tape media software

The RMAN tool has APIs with many third-party tape media software. These allow RMAN to be executed within other non-Oracle backup utilities and to be integrated in a common backup strategy for an organization.

Provides reports and lists of catalog information

Information about backups that is stored within the recovery catalog can be queried with RMAN LIST and REPORT commands. These commands provide useful ways to display information.

Stores information about backups in a catalog within an Oracle database

Information about backups is stored in the recovery catalog. This information can be retrieved at a later date or whenever desired.

Offers performance benefits, such as parallel processing of backup and restore operations

Backup and restore operations can be set in parallel. This can split workloads onto different tape heads and disk devices, which will improve performance.

Creates duplicate databases for testing and development purposes

Duplicate databases can be created from RMAN backups and can be used for testing purposes.

Tests whether backups can be restored successfully

RMAN provides the VALID commands that will check to see whether the backup is valid.

Determines whether backups are still available in media libraries

RMAN provides the CROSSCHECK command to determine if the backup media and catalog information match.

Figure 1 illustrates some of the differences between RMAN and the customized backup scripts and commands used in the earlier lessons.

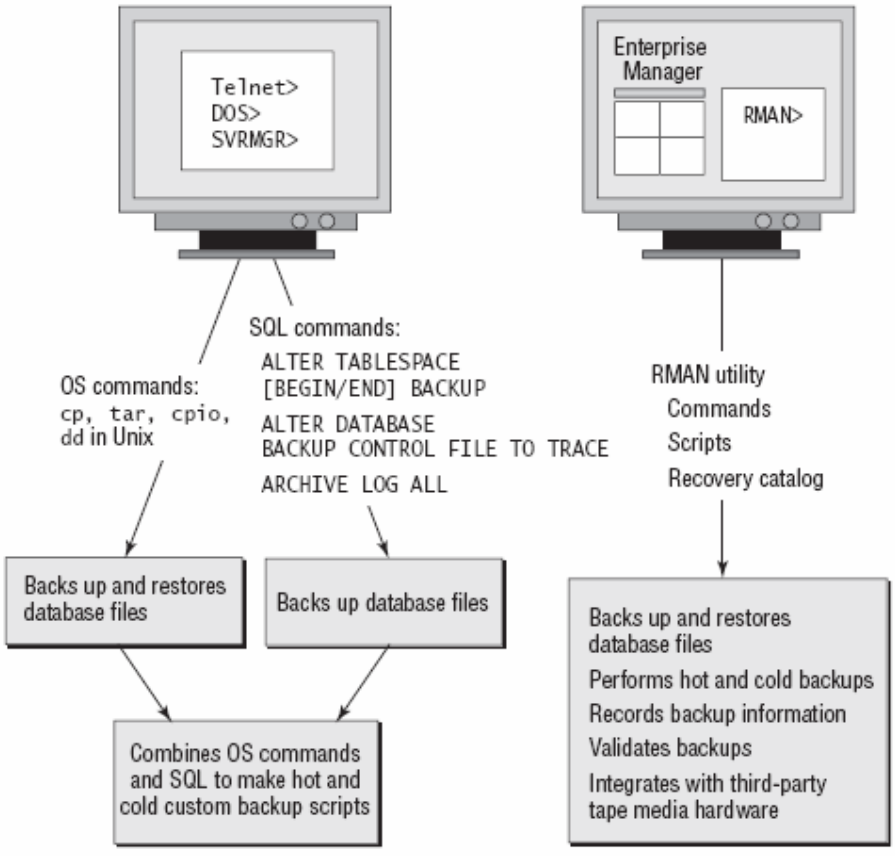

FIGURE 1. Differences between backup scripts and the RMAN utility.

2. Exploring RMAN Components

The main components of RMAN are GUI or command-line access, the optional recovery catalog, the RMAN commands and scripting, and tape media connectivity. These components enable a DBA to automate and standardize the backup and recovery process. Each component is described below:

GUI or command-line access method

This method provides access to Recovery Manager, which in turn provides access to RMAN. The GUI method spawns off of server sessions that connect to the target database—the database targeted by RMAN for backup or recovery actions. GUI access is provided through the Oracle Enterprise Manager (OEM) tool, which is a DBA tool that performs backups, exports/imports, data loads, performance monitoring/tuning, job and event scheduling, and standard DBA management, to mention a few. Command-line access can be run in a standard Unix Telnet or X Windows session as well as in a DOS shell in the Windows environment.

Optional recovery catalog

This special data dictionary of backup information is stored in a set of tables, in much the same way that the data dictionary stores information about databases. The recovery catalog provides a method for storing information about backups, restores, and recoveries. This information provides status updates on the success or failure of backups, the OS backup, data file copies, tablespace copies, control file copies, archived log copies, full database backups, and the physical structures of a database.

RMAN commands

These commands enable different actions to be performed that facilitate the backup and restoration of the database. These commands can be organized logically into scripts, which can then be stored in the recovery catalog database. The scripts can be reused for other backups, thus keeping consistency among different target database backups.

Tape media connectivity

This component provides a method for interfacing with various third-party tape hardware vendors to store and track backups in automated tape libraries (ATLs). An ATL is a large tape cabinet that stores many tapes and can write to multiple tapes at the same time to improve performance. ATLs use robotic arms to load, unload, and store tapes. By default, Oracle provides media management libraries (MML) for the Legato Storage Management (LSM) software, which manages an ATL device.

Figure 2 shows an example of how the RMAN utilities' components fit together to form a complete backup and recovery package.

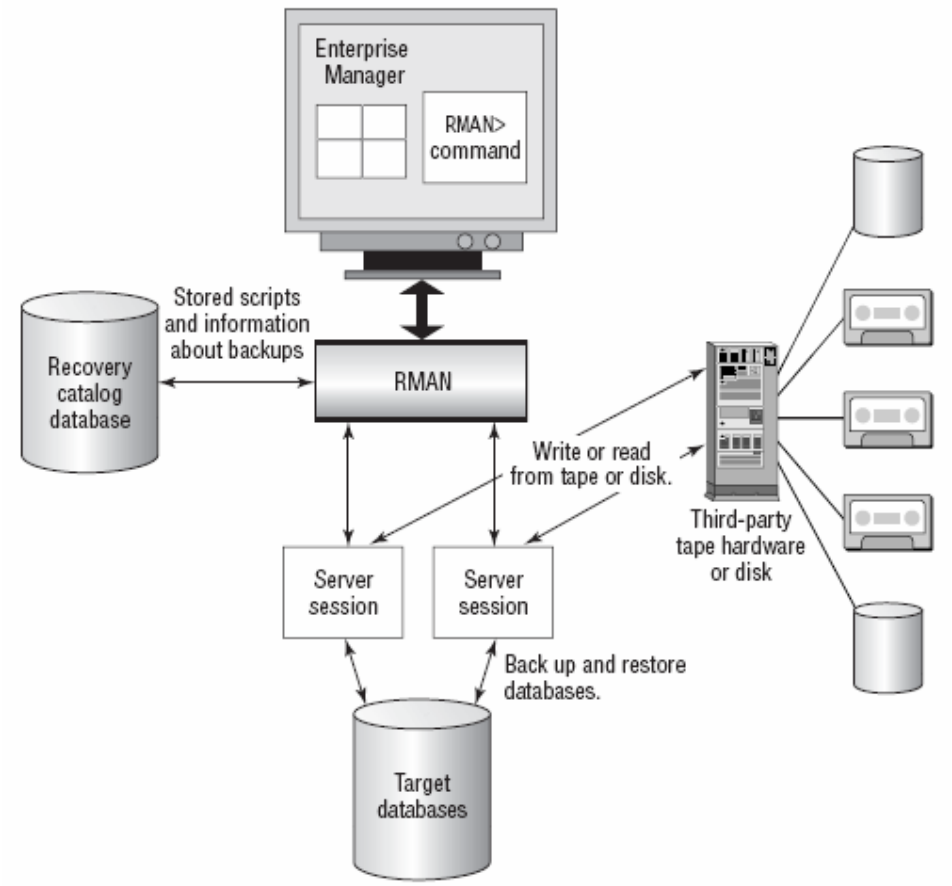

FIGURE 2. The components of RMAN.

3. Storing Information Using the RMAN Repository and Control Files

The RMAN utility uses two methods to store information about the target databases that are backed up. Each method is called the *RMAN repository*. When RMAN uses the first method, it accesses an optional RMAN catalog or recovery catalog of information about backups. In the second method, it accesses the necessary information about backups in the target database's control file. If the optional recovery catalog database is not used, then the target database's control file will be used as the RMAN repository.

The information that RMAN needs to function in the recovery catalog database or in the target database's control file is also called the RMAN repository. The target database's control is always updated, whether the catalog is used or not. In the next paragraph, we will discuss the recovery catalog as the RMAN repository for storing information, and then we will list the commands you can use in this method. Following this list, we will focus on using the control file as the RMAN repository.

Oracle recommends that you store RMAN backup data in the recovery catalog database as opposed to in the control file of the target database. If you store the data in this manner, the RMAN utility will have full functionality. This recovery catalog database is another Oracle database that has special RMAN catalog tables that store metadata about backups in much the same way that a data dictionary stores data about objects in the database. When this database is used, activities such as cross checking backups and available tapes can be performed. Also, using this method, backup scripts can be created and stored in the recovery catalog database for later use. This database can also be backed up so that the information it contains is made safe.

Before we move on to discuss the second method in more detail, take a look at these commands. They are allowed only if you use the RMAN recovery catalog as the RMAN repository.

- CREATE CATALOG
- UPGRADE CATALOG
- DROP CATALOG
- CREATE SCRIPT
- DELETE SCRIPT
- REPLACE SCRIPT
- PRINT SCRIPT
- LIST STORED FILES
- LIST INCARNATION
- REGISTER DATABASE
- REPORT SCHEMA AT TIME
- RESET DATABASE
- RESYNC CATALOG

As mentioned earlier, the RMAN utility also enables you to connect to a target database without using this recovery catalog database. Though this approach is not recommended by Oracle, it does have its uses. (For instance, you might use this approach if the overhead of creating and maintaining the recovery catalog were too great for your organization.)

If you use RMAN without the recovery catalog, you are storing most of the necessary information about each target database in the target database's control file, which serves as the RMAN repository. Thus, you must manage the target database's control file to support this data. The init.ora file's CONTROL_FILE_RECORD_KEEP_TIME parameter determines how long information that can be used by RMAN is kept in the control file. The default value for this parameter is 7 days, but it can be as many as 365 days. The greater the number, the larger the control file becomes so that it can store more information. The control file can only be as large as the OS allows, so be aware of this. The information that is stored within the control file is stored in the *reusable sections*. These sections can grow if the value of the CONTROL_FILE_RECORD_KEEP_TIME parameter is 1 or more. The reusable sections are made up of the following categories:

- Archived log
- Backup data file
- Backup redo log
- Copy corruption
- Deleted object
- Offline range
- Backup corruption
- Backup piece
- Backup set
- Data file copy
- Log history

4. Describing Channel Allocation

Channel allocation is a method you can use to connect RMAN and target databases. While you are doing this, you can also determine the type of I/O device that the server process will use to perform the backup or restore operation. Figure 3 displays this example. The I/O device can be either tape or disk (in Figure 3, you can see both).

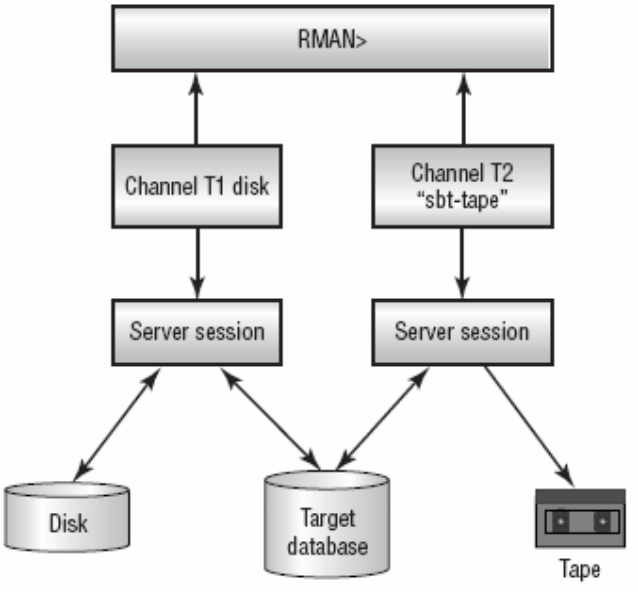

FIGURE 3. Channel allocation.

Channels can be allocated manually or automatically.

Manual channel allocation is performed any time you issue the ALLOCATE CHANNEL command, which starts a server process on the server of the target database. To manually write to a disk file system, you would use the ALLOCATE CHANNEL <channel name> TYPE DISK command. Similarly, to write to a tape backup system, you would use the ALLOCATE CHANNEL <channel name> TYPE 'SBT_TAPE'command. These are the most common manual channel allocation usages.

Automatic channel allocation is performed when you set the RMAN configuration at the RMAN command prompt. You do this by using the CONFIGURE DEFAULT DEVICE or CONFIGURE DEVICE command. This type of allocation is automatic when you are executing the BACKUP, RESTORE, or DELETE commands. By using the CONFIGURE commands you can eliminate the need to use the ALLOCATE CHANNEL <channel name> TYPE DISK or 'SBT_TAPE' command every time you perform a BACKUP, RESTORE, or DELETE. The complete listing of automatic channel allocation commands is as follows:

- CONFIGURE DEVICE TYPE PARALLELISM
- CONFIGURE DEFAULT DEVICE TYPE
- CONFIGURE CHANNEL DEVICE TYPE
- CONFIGURE CHANNEL n DEVICE TYPE

There are some default naming conventions for the disk and tape device types—ORA_MAINT_DISK_n and ORA_SBT_TAPE_n. The example below shows the default device type set to DISK and parallelism set to 1. This means that if you don't allocate a channel manually, you will get the parameters listed below.

```
RMAN> show all;

RMAN configuration parameters are:

…

CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default

CONFIGURE DEVICE TYPE DISK PARALLELISM 1; # default

RMAN>
```

5. Exploring the Media Management Library Interface

The media management library (MML) interface is an API that interfaces RMAN and different hardware vendors' ATLs. All tape hardware vendors that wish to work with Oracle RMAN will make their own MML. This is necessary because most tape hardware devices are proprietary and require different program calls. The MML is then linked in with the Oracle database kernel so that the RMAN server process and MML can read and write the Oracle data to the tape device. Figure 4 describes this concept.

NOTE: Oracle provides a third-party media management library (MML) that is included in its

software installation by default with RMAN. This MML is used with Legato Storage Manager (LSM) software, which will manage an automated tape library (ATL).

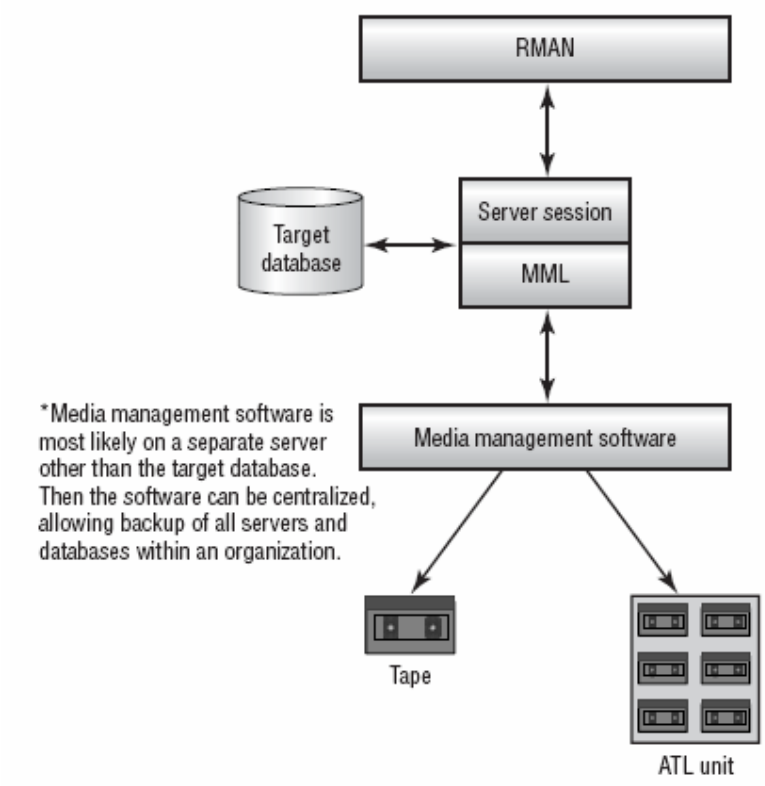

FIGURE 4. MML and RMAN server process.

When you set up the MML with RMAN, you need to use OS commands to replace an existing shared library with the new media management vendor's library. Here is a generic example of this being done in the Unix environment.

1. If an old libobk.so symbolic link already exists in $ORACLE_HOME/ lib, then remove it before installing the media manager. For example:

   oracle@octilli:/oracle/product/9.0.1/rdbms/lib > rm libobk.so

2. There are two ways to access the new media management library from the vendor. Either create a symbolic link,

   oracle@octilli: > ln -s /vendor/lib/oracle_lib.so

   $ORACLE_HOME/rdbms/lib/libobk.so

or move the library into the $ORACLE_HOME/lib directory.

   oracle@octilli: > mv /vendor/lib/oracle_lib.so

```
$ORACLE_HOME/rdbms/lib/libobk.so
```

REAL WORLD SCENARIO **Configuring and Using the Media Manager Vendor Software**
In your role as a DBA, you have configured RMAN and made sure that it is operational. You also have a backup script and method that you feel comfortable with. But you are not out of the woods, yet. This is because there is often a whole new setup, configuration, usage, and testing effort associated with the software and hardware solutions that you choose to integrate with RMAN. For instance, if you interface with an ATL, you will encounter software that manages the tapes, and most likely, the RMAN backups as well. In some cases, these software solutions can be simple agents performing shell scripts that then call RMAN scripts. But in some cases, you may end up dealing with full-featured software applications that execute RMAN from their own GUI screens. As a result, you should make sure that you plan for the extra time you will need to configure and work with these software tools. You will also need to take into account the time you will need to work with system administrators, backup coordinators, and other database administrators to assure that your complete backup package is properly implemented.

6. Connecting to RMAN without a Recovery Catalog

In order to connect to the target database in RMAN, you must set the ORACLE_SID to the appropriate target database. In this example, it is orc9.

This example uses the oraenv shell script provided by Oracle with the 9i database software to change database environments. Next, you initiate the RMAN utility. Once the RMAN utility is started, issue the CONNECT TARGET command with the appropriate SYSDBA privileged account. This performs the connection to the target database.

Let's walk through this step by step:

1. Set the ORACLE_SID to the appropriate target database that you wish to connect to.

   ```
   oracle@octilli:/oracle/product/9.0.1/bin > . oraenv
   ORACLE_SID = [orc9] orc9
   ```

2. Execute the RMAN utility by typing RMAN and pressing Enter.

   ```
   oracle@octilli:/oracle/product/9.0.1/bin > rman
   Recovery Manager: Release 9.0.1.0.0 - Production
   (c) Copyright 2001 Oracle Corporation. All rights
   reserved.
   ```

```
RMAN>
```

3. Issue the CONNECT TARGET command with the appropriate DBA privileged account.

```
RMAN> connect target /

connected to target database: ORC9 (DBID=3960695)

RMAN>
```

Here are two other methods of connecting to the target database without the recovery catalog:

```
oracle@octilli:~ > rman target / nocatalog
```

and

```
oracle@octilli:~ > rman target SYS/CHANGE_ON_INSTALL@tst9
NOCATALOG
```

7. Configuring the RMAN Environment

Configuring the RMAN environment consists mainly of executing the CONFIGURE command at some point while you are using the RMAN prompt.

The existing configuration can be seen by executing the SHOW ALL command, as we did in the "Describing Channel Allocation" section earlier. Let's start by first looking at the output from the SHOW ALL command in the target database TST9.

```
RMAN> show all;

using target database controlfile instead of recovery catalog

RMAN configuration parameters are:

CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 5 DAYS;

CONFIGURE BACKUP OPTIMIZATION OFF; # default

CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default

CONFIGURE CONTROLFILE AUTOBACKUP OFF; # default

CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE

DISK TO '%F'; # default

CONFIGURE DEVICE TYPE DISK PARALLELISM 1; # default

CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO

1; # default

CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE DISK TO

1; # default
```

```
CONFIGURE MAXSETSIZE TO UNLIMITED; # default
CONFIGURE SNAPSHOT CONTROLFILE NAME TO '/oracle/product/
9.0.1/dbs/snapcf_tst9.ft
RMAN>
```

Now let's look at a specific example of what happens while you are configuring the AUTOBACKUP control file. This process begins when you execute the CONFIGURE CONTROLFILE AUTOBACKUP ON command at the RMAN prompt, as shown here:

```
RMAN> configure controlfile autobackup on;
new RMAN configuration parameters:
CONFIGURE CONTROLFILE AUTOBACKUP ON;
new RMAN configuration parameters are successfully stored
RMAN>
```

If you want to revert to the default entry for the AUTOBACKUP control file or some other configuration setting, you can use the CLEAR command, as demonstrated here:

```
RMAN> configure controlfile autobackup clear;
old RMAN configuration parameters:
CONFIGURE CONTROLFILE AUTOBACKUP ON;
RMAN configuration parameters are successfully reset to
default value
RMAN>
```

There are other configuration parameters that can be used in RMAN. Each of these configuration parameters can be used in a similar way as the previous configuration example. The other configuration parameters can be broken into the following major categories:

- Configuring automatic channels
- Configuring the AUTOBACKUP control file
- Configuring the backup retention policy
- Configuring the maximum size of backup sets
- Configuring backup optimization
- Configuring the number of backup copies
- Configuring tablespaces for exclusion from whole database backups
- Configuring the snapshot control file location

8. Summary

This lesson discussed the features and capabilities of the RMAN utility. From it, you should have gotten a sense of some of the basic functions of RMAN. You should have learned how RMAN can be used with or without the recovery catalog and in what environments this practice would be most beneficial. In addition, you learned what the effects of using RMAN on the control file would be if you are not using the recovery catalog. You also learned of manual and automatic channel allocation and explored specific examples of each.

Having an understanding of the components and features of RMAN is important for the workplace. When you know how to work with the RMAN repository, use channel allocation and the MML, and configure the RMAN environment, you will be able to make appropriate decisions when you implement and use RMAN in the workplace. This level of understanding will definitely be beneficial in the testing process.

References

[1] Oracle9i DBA Fundamentals II.