8

Backup and Recovery Enhancements

CERTIFICATION OBJECTIVES

8.01 Using the Flash Recovery Area

8.02 Using Incremental Backups

8.03 Enhanced RMAN Commands

8.04 Miscellaneous Backup and Recovery Enhancements

✓ Two-Minute Drill

Q&A Self Test

Oracle Database 10*g* contains several interesting innovations in the backup and recovery area. You can now store all recovery-related files, such as archive logs and database backups, in a unified location on disk. This unified storage location is the new *flash recovery area*, and it enables DBAs to simplify file management for all recovery-related files. Cheaper and faster modern disk storage means that you can recover more quickly from a disk-based recovery strategy than from a tape-based backup strategy.

The RMAN can now help you recover image copies using incremental backup files. Oracle Database 10*g* contains new methods for making faster incremental backups. This chapter also looks at several changes in the RMAN backup commands, including the deprecation of the `COPY` command. You'll also learn how to perform a hot recovery without needing to restore backup files first, by using the `SWITCH DATABASE` command. This capability will enable you to implement a faster database recovery.

You can also create compressed backup sets in Oracle Database 10*g* to save on storage space. You'll learn about simplified database recovery after using `RESETLOGS`, without needing the usual full backup. You'll learn how to drop a database using the `DROP DATABASE` command. You'll review the new features that let you specify limits on your backup windows.

Let's start with a detailed look at the interesting and very useful flash recovery area feature.

CERTIFICATION OBJECTIVE 8.01

Using the Flash Recovery Area

The *flash recovery area* serves as the default storage area for all files related to backup and restore operations. You can use an Oracle-managed directory, an operating system file system, or an Automatic Storage Management (ASM) disk group for your flash recovery area. Traditionally, Oracle DBAs have needed to manage the backup-related storage areas, making sure that there was enough storage space to save their backup-related files. Oracle recommends that you now let the database take care of these chores, by using the new strategy of automatic disk-based backup and recovery.

To enable this feature, you simply designate sufficient disk space as the flash recovery area, specify the area's maximum size, and let Oracle know how long you want to retain the backup-related information. Oracle will then manage the backup of related files, including archive log files, control files, and other files. Oracle will delete the files that your database no longer needs because the files have become obsolete or they have been backed up to tape. Thus, your main tasks will be provisioning adequate space for the flash recovery area and selecting the appropriate retention period for keeping the related files.

Once you create the flash recovery area, you can direct all you backup files, including the archived redo log files, to that area. Oracle will manage these files for you, deleting the files that either are backed up to tape or are older than the backup retention interval. In order to automatically delete unwanted files, the flash recovery area concept relies on the Oracle Managed Files (OMF) system. OMF automates file management in Oracle databases by automatically creating and managing the operating system files that you allocate for your database. All you need to do to get an OMF file system going is to configure a couple of OMF-related initialization parameters: `DB_CREATE_FILE_DEST` and `DB_CREATE_ONLINE_LOG_DEST_`*n*. As you are aware, OMF has the capability to automatically create and delete Oracle files, without the DBA's intervention. The RMAN capitalizes on this capability of the OMF file system to perform its backup- and recovery-related functions through using the flash recovery area.

You can also use a flash recovery area on top an ASM file system (see Chapter 10 for a discussion of ASM file systems). ASM file systems consolidate disks into easily manageable disk groups, and provide striping and mirroring capabilities based on the Oracle database itself.

The flash recovery area provides the following benefits:

- Single storage location for all recovery-related files
- Automatic management of recovery-related disk space
- Faster backup and restore operations, since you don't need to restore tape backups
- Increased reliability of backups, since disks are generally safer storage devices than tapes

exam watch

Several databases can share a central flash recovery area.

What's in the Flash Recovery Area?

You keep all the files you need for media recovery in the flash recovery area. Ideally, your flash recovery area must hold a complete backup of all your datafiles, any incremental backups you may have, control file backups, and all the archived redo logs that are necessary for media recovery. You can also use the flash recovery area as a disk cache for tape.

exam watch

All files that you need to recover from a media failure are part of your flash recovery area. You should be able to completely recover your database by using the files in the flash recovery area. In addition to these recovery files, you may also store other database files in the flash recovery area.

The flash recovery area may contain the following files:

- **Datafile copies** You can make copies of the database datafiles using the new RMAN command BACKUP AS COPY. You can use this command to create image copies of all datafiles, which the RMAN will automatically store in the flash recovery area.
- **Control file autobackups** The flash recovery area serves as the default location for all control file autobackups made by the RMAN. The database places any control file backups it generates in the flash recovery area.
- **Archived redo log files** Since Oracle automatically deletes all obsolete files and files that have been transferred to tape, the flash recovery area is an ideal location for archived redo log files. You don't need to manually delete the old archived redo log files, because Oracle will automatically delete the files if you store them in the flash recovery area.
- **Online redo log files** Oracle recommends that you save a multiplexed copy of your online redo log files in the flash recovery area. Oracle will generate its own names for the online redo log files. The default size of an online redo file that's created in the flash recovery area is 100MB. The following statements can create online redo logs in the flash recovery area: CREATE DATABASE, ALTER DATABASE ADD LOGFILE, ALTER DATABASE ADD STANDBY LOGFILE, and ALTER DATABASE OPEN RESETLOGS.
- **Current control files** Oracle also recommends that you store a multiplexed current control file in the flash recovery area.

- **RMAN files** If you specify a location for the flash recovery area by using the DB_RECOVERY_FILE_DEST initialization parameter, the RMAN will, by default, use the flash recovery area for storing its backup-related files. Oracle will also automatically use OMF files and will generate the filenames. As noted, Oracle saves all control file backups and autobackups made by the RMAN in the flash recovery area as well.
- **Flashback logs** Oracle stores all *flashback logs* in the flash recovery area. If you enable the flashback database feature (discussed in Chapter 9), Oracle copies images of each altered block in every datafile into flashback logs stored in the flash recovery area. The database will use these flashback logs to reconstruct the datafile contents as of any moment at which it captured the logs.

Oracle calls the multiplexed redo log files and control files in the flash recovery area permanent files, since they should never be deleted and are part of the live database. Oracle terms all the other files in the flash recovery area (recovery-related files) transient files, since Oracle will delete them eventually after they have become obsolete or have already been copied to tape.

The background process Archiver (ARC*n*) automatically creates a copy of all archived redo logo files in the flash recovery area if you specify the flash recovery area as the archive log destination. You can specify the flash recovery area as the destination for your archived redo log files by using the LOG-ARCHIVE_DEST_10 parameter. If you create a flash recovery area and do not set any other local archiving destinations, LOG_ARCHIVE_DEST_10 is implicitly set to the USE_DB_RECOVERY_FILE_DEST. (That is, archived redo log files will be sent automatically to the flash recovery area location.)

If you've also configured other archive log locations by using the LOG_ARCHIVE_DEST_*n* parameter, Oracle will also place copies of archived redo logs in those other locations. In the following example, I turned on archiving for the database nina without configuring an explicit archive log location. On issuing the ARCHIVE LOG LIST command, this is what I got:

```
SQL> archive log list
Database log mode              Archive Mode
Automatic archival             Enabled
Archive destination            USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence     825
Next log sequence to archive   827
Current log sequence           827
SQL>
```

I am using a flash recovery area for the database nina. The `USE_DB_RECOVERY_FILE_DEST` location for the archive logs points to the flash recovery area for the database, which is the C:\Oracle\Product\10.1.0\flsh_recovery_area\nina\ directory on my Windows XP server.

Sizing the Flash Recovery Area

How large should your flash recovery area be? Oracle recommends that your flashback area size be equal to the sum of the size of your database, any incremental backups that you may have made, and all the archived redo logs on disk. If you have a 100GB sized database, and you have about forty 0.5GB sized redo logs on disk, your flash recovery area needs more than 120GB of space. Your flash recovery area should be large enough for you to save a set of the following files:

- A copy of all datafiles
- Incremental backups
- Online redo logs
- Archived redo logs not yet backed up to tape
- Control files
- Control file autobackups (these will include copies of the control file and the SPFILE)

Note that this list includes both online redo log files and control files. Oracle recommends that you save both a duplexed redo log file and a current control file in your flash recovery area, along with all the other recovery-related files (database file copies and the archived redo log files).

At the minimum, Oracle recommends that you keep the archived logs not yet saved to tape in the flash recovery area.

The size of your database, of course, will be the main determinant of the size of the flash recovery area. The size of the flash recovery area also depends on the following factors:

- Your RMAN backup retention policy
- The type of storage device you are using for your backups (tape and disk or just a disk device)
- The amount of data block changes in your database

Creating a Flash Recovery Area

You can create a flash recovery area in several ways:

- You can configure the flash recovery area during database creation using the Database Creation Assistant (DBCA).
- You need to configure two initialization parameters to configure a flash recovery area. Since these initialization parameters are dynamic, you can also create a flash recovery area while the database is running.
- Of course, you can also use the OEM Database Control to configure a flash recovery area.

Configuring a Flash Recovery Area

You use the DB_RECOVERY_FILE_DEST and DB_RECOVERY_FILE_DEST_SIZE initialization parameters to configure a flash recovery area in your database.

The DB_RECOVERY_FILE_DEST parameter specifies the location of the flash recovery area, such as the /u01/App/Oracle/flsh_rcv directory.

exam watch

When you use the DB_RECOVERY_FILE_DEST parameter to specify the destination of your flash recovery area, you can use a directory, file system, or ASM disk group as your destination. Oracle will actually let you use a file server device for the flash recovery area, provided that it is one that has been verified by the Oracle Storage Compatibility Program.

The DB_RECOVERY_FILE_DEST_SIZE parameter specifies the maximum size of the flash recovery area. Note that the database isn't going to assign or allocate this space to the flash recovery area immediately. If you assign a 2GB chunk of file space to the flash recovery area, Oracle will use this space when it needs to store new files in it. If no files are stored in the flash recovery area, all the space allocated to it remains under the control of the operating system, although the space is technically assigned to the flash recovery area.

on the job

You must first specify the DB_RECOVERY_FILE_DEST_SIZE parameter before you can specify the DB_RECOVERY_FILE_DEST parameter.

Dynamically Defining the Flash Recovery Area

You can dynamically create or modify the flash recovery area using the two initialization parameters DB_RECOVERY_FILE_DEST and DB_RECOVERY_FILE_DEST_SIZE in the following manner:

```
SQL>  alter system set
  2*  db_recovery_file_dest_size = 2G scope=both
System altered.
SQL>  alter system set
  2   db_recovery_file_dest = 'C:\oracle\recovery_area' scope=both;
System altered.
SQL>
```

You must always specify the size parameter, DB_RECOVERY_FILE_DEST_SIZE, before specifying the location parameter, DB_RECOVERY_FILE_DEST. Also, make sure that you first create the specific flash recovery area directory (in this example, it is the recovery_area directory) before using the DB_RECOVERY_FILE_DEST parameter.

The SCOPE=BOTH clause ensures that any changes you make will be applicable immediately, in addition to being written permanently to your SPFILE.

on the job

The flash recovery area and DB_RECOVERY_FILE_DEST location are synonymous.

Disabling the Current Flash Recovery Area

You can disable the current flash recovery area by setting the DB_RECOVERY_FILE_DEST parameter to blank (''), which has the effect of unsetting the current destination for the flash recovery area files. Note that you can use the V$RECOVERY_FILE_DEST view to check the location of the flash recovery area. The following example illustrates how you can disable your flash recovery area.

```
SQL> select name from v$recovery_file_dest;
NAME
---------------------------
C:\oracle\recovery_area
SQL> alter system set
  2  db_recovery_file_dest = '';
System altered.
SQL> select name from v$recovery_file_dest;
NAME
---------------------------
SQL>
```

Note that even after you disable the flash recovery area, the RMAN will continue to access the files located in the flash recovery area for backup and recovery purposes. However, the RMAN won't have access to the automatic space management features of the flash recovery area.

Default File Location and the Flash Recovery Area

Using the flash recovery area implies that you are using OMF. If you are configuring a flash recovery area in your database, you can't use the usual LOG_ARCHIVE_DEST and LOG_ARCHIVE_DUPLEX_DEST parameters to specify redo log archive destinations. You must instead use the newer LOG_ARCHIVE_DEST_*n* parameters. When you use OMF files, Oracle determines the default location for the datafiles, control files, and redo log files, based on the values of the initialization parameters DB_CREATE_FILE_DEST and DB_CREATE_ONLINE_LOG_DEST_*n*, which determine the location of all OMF files. When you create the flash recovery area on top of the OMF files, the default location of the control files and redo log files will be based on your choice among the two initialization parameters, as well as a third parameter that specifies the location of the flash recovery area. This additional parameter is the new DB_RECOVERY_FILE_DEST parameter.

Thus, the three important initialization parameters—DB_CREATE_FILE_DEST, DB_CREATE_ONLINE_LOG_DEST_*n*, and DB_RECOVERY_FILE_DEST—determine where and how many control files and redo log files Oracle will create when you use a flash recovery area built on top of an OMF-based file system. I summarize the various possibilities in the following sections.

on the job

Oracle Corporation recommends that your DB_RECOVERY_FILE_DEST location not be the same as the DB_CREATE_FILE_DEST location or any of the DB_CREATE_ONLINE_LOG_DEST_n locations.

Control Files

If you set the CONTROL_FILE parameter manually in your initialization file before starting the instance to create a new database, Oracle will create the control files in that location. If you haven't set the CONTROL_FILES parameter, Oracle will create the control files in various default locations, according to the following rules:

- If you specify the DB_CREATE_ONLINE_LOG_DEST_*n* parameter, Oracle will create an OMF-based control file in *n* number of locations, with the first directory holding the primary control file.

exam watch

If the database creates an OMF control file, and it is using a server parameter file, then the database sets the `CONTROL_FILES` initialization parameter in the server parameter file.

- If you specify the `DB_CREATE_FILE_DEST` and `DB_RECOVERY_FILE_DEST` parameters, Oracle will create an OMF-based control file in both of these locations.
- If you just specify the `DB_RECOVERY_FILE_DEST` parameter, Oracle will create an OMF-based control file in the flash recovery area only.
- If you omit all three of the initialization parameters, Oracle will create a non-OMF-based control file in the system-specific default location.

Redo Log Files

If you omit the `LOGFILE` clause during database creation, Oracle will create the redo log files according to the following rules:

- If you specify the `DB_CREATE_ONLINE_LOG_DEST_`*n* parameter, Oracle will create an online redo log member in *n* number of locations, up to the `MAXLOGMEMBERS` limit for the database.
- If you specify the `DB_CREATE_FILE_DEST` and `DB_RECOVERY_FILE_DEST` parameters only, Oracle will create an online redo log member in both of these locations up to the `MAXLOGMEMBERS` limit for the database.
- If you just specify the `DB_RECOVERY_FILE_DEST` parameter, Oracle will create an online redo log member in the flash recovery area only.
- If you omit all three initialization parameters, Oracle will create a non-OMF online redo log file in the system-specific default location.

Backing Up the Flash Recovery Area

When you configure the flash recovery area, the RMAN will automatically place its backup files in that area. Oracle will use the OMF files and automatically generate the filenames itself. Before going further, let's clarify the difference between the two ways you can store database backups using RMAN:

- **Image copies** These are identical to your normal operating system file copies. Image copies are byte-by-byte copies of Oracle database files. However, the big difference between the copies made by the operating system and the RMAN image copies is that the RMAN records information about its image copies in the RMAN repository. Since you can't use a backup unless it is part of the RMAN recovery catalog, you can't use normal operating system backup copies inside RMAN.

- **Backup sets** These are logical entities, consisting of individual components (files) called *backup pieces*. Backup pieces store the contents of one or several Oracle database files. You can't access the backup pieces on an individual basis; you must access them as part of a backup set.

You can back up the flash recovery area itself by using simple RMAN backup commands. In order for the commands to work, you must set `CONFIGURE BACKUP OPTIMIZATION` to `ON`. You can back up the flash recovery area only to a tape device using these backup commands.

The RMAN command `BACKUP RECOVERY AREA` backs up all flash recovery files in the current or previous flash recovery area destinations. This command will back up only those files that have never been backed up to tape before. The files that the command will back up include full backups, incremental backups, control file autobackups, archive logs, and datafile copies.

The RMAN command `BACKUP RECOVERY FILES` command backs up all the files that the `BACKUP RECOVERY AREA` command does, but from *all areas* on your file system, not just from the flash recovery area.

exam watch

Neither of the two commands, `BACKUP RECOVERY AREA` or `BACKUP RECOVERY FILES`, will back up any permanent files in the flash recovery area, such as the current control file and the current redo log file. These commands will also not back up the flashback logs.

on the job

You can use the RMAN command `BACKUP RECOVERY FILE DESTINATION` to move disk backups created in the flash recovery area to tape.

Managing the Flash Recovery Area

There really isn't a whole lot you need to do in the way of managing the flash recovery area. You must take care to size the flash recovery area large enough so there is no space pressure in that area. Let's first discuss how the flash recovery area manages its space, and then see the dictionary view that helps you manage the flash recovery area.

Space Management

Once you create the flash recovery area, Oracle will start storing recovery-related files in that area, until it uses up all the space you allocated to the flash recovery area. Whenever you add a new file to the flash recovery area, Oracle automatically updates its list of backup files that are candidates for deletion. This list consists of those files that either have been backed up to tape or are obsolete as per your retention rules. Oracle automatically deletes files on this list when it needs space for incoming new recovery-related files.

When the flash recovery area runs low on space and it can't delete any files to relieve the space pressure, Oracle issues an automatic *warning* alert, once the flash recovery are is 85 percent full. When the area uses up 97 percent of its space, Oracle will send a *critical* alert. Oracle sets these warning and critical thresholds, and you can't change them.

Oracle will add entries to both your alert log file and the DBA_OUTSTANDING_ALERTS view to mark any flash recovery area space alerts. Oracle will continue to place new recovery-related files in the flash recovery area, until it hits the 100 percent space usage barrier, at which point it will issue an error stating that your flash recovery area is out of room.

If you ever receive the out-of-space warning and critical alerts because of space pressure in you flash recovery area, you have the following options:

- Consider changing your backup retention and archive log retention policies.
- Increase the size of the DB_RECOVERY_FILE_DEST_SIZE parameter to allocate more space to your current flash recovery area.
- Use the BACKUP RECOVERY AREA command in the RMAN to back up the contents of the flash recovery area to a tape device.
- Use the RMAN to delete unnecessary backup files. The RMAN commands CROSSCHECK and DELETE EXPIRED come in handy during this deletion process.

Note that if the database can't archive any redo log files because the flash recovery area is full, the database will hang, as it always does when your traditional archive destination fills up.

Data Dictionary Views

The V$RECOVERY_FILE_DEST view is the main source to help you manage the flash recovery area. This view contains information about the flash recovery area. Here's its structure:

```
SQL> desc v$recovery_file_dest
 Name                                      Null?    Type
 ----------------------------------------- -------- ---------------------
 NAME                                               VARCHAR2(513)
 SPACE_LIMIT                                        NUMBER
 SPACE_USED                                         NUMBER
 SPACE_RECLAIMABLE                                  NUMBER
 NUMBER_OF_FILES                                    NUMBER
SQL>
```

In the v$recovery_file_dest view, the SPACE_LIMIT column tells you how much space has been allocated to the flash recovery area (by the DB_RECOVERY_FILE_DEST_SIZE parameter). The SPACE_RECLAIMABLE column tells you how much space you can reclaim by getting rid of obsolete and redundant files in the flash recovery area.

A new column, named IS_RECOVERY_DEST_FILE, has been added to the V$LOGFILE, V$CONTROLFILE, V$ARCHIVED_LOG, V$DATAFILE_COPY, and V$BACKUP_PIECE views. The column can take a value of YES or NO. If the column value is YES, that means that the respective file in that view is in the flash recovery area.

Moving the Flash Recovery Area

If you need to move your flash recovery area to a different location, you can do so by using the DB_RECOVERY_FILE_DEST initialization parameter, as in this example:

```
ALTER SYSTEM SET DB_RECOVERY_FILE_DEST='/u01/app/oracle/new_
area' SCOPE=BOTH;
```

Oracle will immediately start creating all the new flash recovery area files in your new flash recovery area location.

You can leave the permanent files (control files and online redo log files), flashback logs, and transient files in their original location. Eventually, Oracle will delete all the transient files from the previous flash recovery area location, when each of them become eligible for deletion. However, if you want to move your current permanent files, transient files, or flashback logs to the new flash recovery area, you can do so by using the standard file-moving procedures.

CERTIFICATION OBJECTIVE 8.02

Using Incremental Backups

A *full backup* includes every block in the datafiles. *Incremental backups* capture only those data blocks that have changed since the last backup. Incremental backups help you avoid spending huge amounts of time and resources that you'll need for performing frequent full database backups. The RMAN can perform an incremental backup at the database, tablespace, or datafile level. In Oracle Database 10*g*, there are several enhancements pertaining to incremental backups. The main enhancements are the ability to recover with incrementally updated backups and the fast incremental backup capability. Let's first look at how you can reduce recovery time by using incrementally updated backups.

Recovering with Incrementally Updated Backups

Oracle Database 10*g* gives you the ability to recover databases using incrementally updated backup files. Currently, the standard operating procedure is for you to apply archived redo logs to your datafile backup copies in order to perform a media recovery. For example, if your database needs a recovery on Friday and your last backup was done on Sunday night, you must apply all the archive logs that your database generated since Sunday night.

In Oracle Database 10*g*, you can apply incremental backups to your *datafile image copies* when you use the RMAN. This application of incremental backups to image copies enables you to roll forward or recover the image copy to a specified point in time. Here's how this efficient recovery strategy works:

1. **Apply the incremental backups to datafile image copies.** Use the `RECOVER COPY OF DATAFILE` command of the RMAN to perform the recovery. This will update all of your datafile image copies, all the way up to the system change number (SCN) in the latest incremental backup. The RMAN will treat this updated datafile image copy as a normal datafile image copy.
2. **Then apply the archive logs since the last incremental backup only.** Apply these archive logs to the updated datafile image copies. Note that here is where the incrementally updated backups provide their value, since you don't need to apply all the archive logs since the last backup, but only those logs since the last incremental backup! You don't need to perform a full image copy after incrementally restoring your database files.

exam watch

Incremental backups help restore datafiles by applying changes at the database block level. Archived redo logs apply changes at the transaction level. Incrementally updated backups lead to a faster recovery, since you need to apply only data block changes, and that is always much faster than applying individual transactions. Thus, Oracle prefers that you use incremental backups over archived logs during a recovery, whenever you have a choice.

If you have incremental backups available, along with the archived redo logs, the RMAN will always choose incremental backups over archived redo logs during recoveries, because it is a much faster way to recover your database. However, note that for a level 0 incremental backup, Oracle may not choose incremental backups over the archived redo logs.

on the job

Oracle recommends that you make whole backups of your database regularly, and make daily (or more frequent) incremental backups. You must retain all of these backups on disk. You can then perform a quicker full database or point-in-time recovery using the incrementally updated backup approach, and make minimal use of the slower archived log-based recovery technique. Thus, during a recovery, the incremental backups will have already been applied, and the only archive logs needed will be those since the last incremental backup.

You can always restart the recovery process, if the process fails while it is applying the incremental backup files to the datafile image copies. If you have multiple versions of an image copy in the RMAN catalog, the RMAN will automatically use the latest version. If there are overlapping incremental backups, the RMAN will choose the one covering the longer period.

Fast Incremental Backups

During an incremental backup, Oracle must scan the entire datafile, even if only a very small part of the file has changed. This makes for unnecessarily long incremental backup times. Oracle Database 10*g* contains enhancements that help you make much faster incremental backups.

Oracle Database 10*g* maintains a new file, the *change tracking file*, to track the physical location of all database changes. The RMAN simply reads this file to find out which data blocks to read and copy during an incremental backup process. Thus, RMAN avoids needing to read entire datafiles during incremental backups, and your backup times are dramatically shorter than before.

A new background process, the *change tracking writer* (CTWR), is in charge of writing the block change information to the change tracking file. The size of the change tracking file will depend on the size of the database, the amount of DML activity, the number of old backups maintained by the change tracking file, and the number of active instances, if you are using Real Application Clusters (RAC).

Oracle recommends that you place the change tracking file on the same disks as the datafiles.

Enabling Block Change Tracking

Oracle Database 10*g* doesn't track block changes by default. You must explicitly enable the feature, by using the following command:

```
SQL> alter database
  2  enable block change tracking
  3  using file 'C:\ORACLE\RECOVERY_AREA\CHANGETRACK.LOG';
Database altered.
SQL>
```

If you store the change tracking file along with you database files, the file will be deleted automatically when you disable block change tracking. You can rename or relocate a change tracking file by using the `ALTER DATABASE RENAME FILE` command, as shown here:

```
SQL> alter database rename file
  2  'C:\ORACLE\RECOVERY_AREA\CHANGETRACK.LOG'
  3* to 'C:\ORACLE\NEWCHANGE.LOG';
```

You must make sure that your database is in the mount stage before you can rename the change tracking file.

You may disable block change tracking by using the following command:

```
SQL> alter database disable block change tracking;
Database altered.
SQL>
```

exam watch

The change tracking file contains the physical location of all database changes. The minimum size is 10MB. Oracle creates the file automatically and allocates additional space in 10MB increments. The file's size depends on your database size, number of instances in an RAC, and the number of old backups the file maintains.

Monitoring Block Change Tracking

You can monitor the block change tracking feature in your database (as noted in the previous section, this feature is turned off by default) by using the V$BLOCK_CHANGE_TRACKING and V$BACKUP_CONTROLFILE views.

The V$BLOCK_CHANGE_TRACKING view shows the name, size, and status of your change tracking file, as shown in this example:

```
SQL> select filename, status, bytes
  2  from v$block_change_tracking;
FILENAME                                   STATUS       BYTES
-----------------------------------------------------------------
C:\ORACLE\RECOVERY_AREA\CHANGETRACK.LOG    ENABLED      11599872
SQL>
```

You can use the V$BACKUP_DATAFILE view to see how the block change feature is helping you minimize the file scanning done by the incremental backup process. You can use the ratio between the value of the BLOCKS_READ column and the DATAFILE_BLOCKS column to figure out the percentage of blocks Oracle is reading in order to perform its incremental backups. If the BLOCKS_READ/DATAFILE_BLOCKS ratio is too high, you may need to take more frequent incremental backups to reduce the ratio.

CERTIFICATION OBJECTIVE 8.03

Enhanced RMAN Commands

Oracle Database 10*g* has several new RMAN commands, and some older commands have been deprecated. You'll also find some enhancements in the RMAN scripting feature. Let's review these changes, starting with the BACKUP AS COPY command.

Using the BACKUP AS COPY Command

The RMAN COPY command has been deprecated in Oracle Database 10*g*. The new command that replaces it is BACKUP AS COPY. In the older backup commands, when you created a backup set, you needed to use RMAN later on to extract individual datafiles from the backup set. The new BACKUP AS COPY command creates image copies, so you don't need to extract them from a backup set. Using a single BACKUP AS COPY command, you can now copy all of the following:

- A database
- One or multiple tablespaces
- One or multiple datafiles
- One or more multiple archived redo logs
- One or more control files

You can use the BACKUP AS COPY command to create image copies of datafiles, as shown here:

```
RMAN> BACKUP AS COPY DATABASE;
```

If you want RMAN to create image copies by default, use the following RMAN configuration command:

```
RMAN> configure device type disk
2> backup type to copy;
old RMAN configuration parameters:
CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO COMPRESSED BACKUPSET;
new RMAN configuration parameters:
CONFIGURE DEVICE TYPE DISK BACKUP TYPE TO COPY PARALLELISM 1;
new RMAN configuration parameters are successfully stored
released channel: ORA_DISK_1
RMAN>
```

To create a backup set, rather than an image copy, you can use the following command:

```
RMAN> backup as backupset database;
```

Performing Backups

You can back up various database elements as follows:

- **Entire database backup** You can now perform an image file backup of the entire database, by using the following command:

  ```
  RMAN> backup database;
  ```

- **Previous database image copy** If you wish to back up a previous image copy of all the database files and control files in your database, use the following command:

  ```
  RMAN> backup copy of database;
  ```

- **Tablespace** You can back up all datafiles belonging to a tablespace by using the following command (USERS is the tablespace in this example and the next one):

  ```
  RMAN> backup tablespace users;
  ```

- **Previous tablespace image copy** If you want to make a backup of a previous image copy or a backup set of a tablespace, you can do so by using the following command:

  ```
  RMAN> backup copy of tablespace users;
  ```

- **Single datafile** You use the following command to perform a backup of a single datafile (datafile 10 in this example):

  ```
  RMAN> backup datafile 10;
  ```

- **Previous datafile backup** The following command backs up a previously backed up datafile:

  ```
  RMAN> backup copy of datafile 10;
  ```

- **Current control file** To back up a current control file, use the following command (equivalent to the SQL command `ALTER DATABASE BACKUP CONTROLFILE`):

  ```
  RMAN> backup current controlfile;
  ```

- **Previous control file copies** In order to copy all the previously made control file copies, use the following command.

  ```
  RMAN> backup controlfilecopy all;
  ```

on the job

The output of a backup command may be an image copy or a backup set, depending on whether your `CONFIGURE DEVICE TYPE DISK BACKUP TYPE` parameter is set to `COPY` or `BACKUPSET`.

Using the CATALOG Command

In Oracle Database 10*g*, you can issue the `CATALOG` command to make the RMAN start cataloging backup files immediately. You can use the command when you are searching for new files or when you want to let the RMAN know that you are using a new backup directory.

The `CATALOG` command has the following syntax:

```
RMAN> catalog backuppiece 'filename';
```

Instead of the keyword BACKUPPIECE, you can use the keyword DATAFILECOPY, if you are cataloging an image copy of a datafile instead of a backup piece from a backup set.

Here is an example of the use of the CATALOG command:

```
RMAN> catalog datafilecopy  “C:\ORACLE\PRODUCT\10.1.0\FLASH_RECOVERY_AREA\
NINA\DATAFILE\01_MF_SYSAUX_OGOGDVDC_.DBF”;
catalogued datafile copy
datafile copy
filename= C:\ORACLE\PRODUCT\10.1.0\FLASH_RECOVERY_AREA\
NINA\DATAFILE\01_MF_SYSAUX_OGOGDVDC_.DBF recid=4 stamp=530016304
RMAN>
```

You can also use the reverse command, UNCATALOG, to get rid of a backup piece from the RMAN catalog, as shown here:

```
RMAN> change backuppiece  'file_name' uncatalog;
```

The following example uncatalogs the datafile copy cataloged in the previous example:

```
RMAN> change datafilecopy
"C:\ORACLE\PRODUCT\10.1.0\FLASH_RECOVERY_AREA\
NINA\DATAFILE\01_MF_SYSAUX_OGOGDVDC_.DBF" UNCATALOG;
Uncatalogued datafile copy
datafile copy
filename= C:\ORACLE\PRODUCT\10.1.0\FLASH_RECOVERY_AREA\
NINA\DATAFILE\01_MF_SYSAUX_OGOGDVDC_.DBF recid=4 stamp=530016304
Uncatalogued 1 objects
RMAN>
```

exam watch

If you eliminate all backup pieces from the RMAN catalog, the RMAN will remove the backup set information as well.

Let's say you suspect that there are some backup files in various directories that aren't yet a part of the RMAN catalog. Some of the filenames may be hard to remember, since they may be OMF-based files. Instead of worrying about where all the backup files might be, you can simply ask the RMAN to search in a certain directory for all backup files that aren't part of the catalog already. You use the CATALOG START WITH command to perform this job, as shown in the following example.

```
RMAN> catalog start with
2> "C:\ORACLE\PRODUCT\10.1.0\FLASH_RECOVERY_AREA\NINA\DATAFILE";
searching for all files that match the pattern C:\ORACLE\PRODUCT\10.1.0\
FLASH_RECOVERY_AREA\NINA\DATAFILE
List of Files Unknown to the Database
=====================================
File Name: C:\ORACLE\PRODUCT\10.1.0\FLASH_RECOVERY_AREA\NINA\DATAFILE\
O1_MF_SYSAUX_0G0GDVDC_.DBF
Do you really want to catalog the above files (enter YES or NO)? YES
cataloging files...
cataloging done
List of Cataloged Files
=======================
File Name: C:\ORACLE\PRODUCT\10.1.0\FLASH_RECOVERY_AREA\NINA\DATAFILE\
O1_MF_SYSAUX_0G0GDVDC_.DBF
RMAN>
```

The CATALOG START WITH command comes in very handy if you happen to lose your RMAN recovery catalog. In earlier releases, there was no way to tell the RMAN about a backup set. This was a complete showstopper for many DBAs. If your recovery catalog was destroyed, even though all the backup sets were perfect and you knew exactly where they were and what was in them, there was no way to tell a newly created catalog to use them. Now you can just point it at the tape library and let it find everything. This is an important and long overdue enhancement.

Enhancements in RMAN Scripts

There are two important enhancements in the RMAN scripting feature. The first is the *convertibility* of text files into stored scripts and vice versa. The second is the *shareability* of RMAN scripts among databases.

Convertibility of RMAN Scripts

The RMAN lets you use scripts to perform backups and recovery, to save you the drudgery of typing the commands in repeatedly, using the command line. RMAN offers you two kinds of scripts: *stored scripts* and *text scripts*. Stored scripts are stored in the RMAN recovery catalog, and the text scripts are kept in regular text files. Stored scripts offer the advantage that any user who logs in to RMAN can access them easily, comparing to accessing the text files, which may be stored anywhere.

In previous versions of Oracle, you couldn't convert a text script into a stored script and vice versa. In Oracle Database 10*g*, this limitation goes away: you can now change a stored script into a text script and a text script to a stored script.

Here's an RMAN command that sends the contents of a stored script to a text file:

```
RMAN> print script full_backup to file 'my_script_file.txt';
```

Global RMAN Scripts

In previous Oracle versions, you could execute a stored script only if you were connected to the target database where the script was originally created, thus making the scripts *local* in nature. Oracle Database 10*g* enhances stored scripts by providing the new concept of *global* scripts, which you can execute against any database registered in the recovery catalog, as long as your RMAN client is connected to the recovery catalog and a target database simultaneously.

Multiple databases can share the same RMAN scripts, provided the databases connect to the database with the RMAN catalog. Of course, you must use the RMAN recovery catalog in order to use this feature.

The following two statements show the syntax for creating a local and a global script, respectively.

```
RMAN> create script full_backup
      {
      backup database plus archivelog;
      delete obsolete;
      }
RMAN> create global script global_full_backup
      {
      backup database plus archivelog;
      delete obsolete;
      }
```

The `EXECUTE SCRIPT` command will let you execute a global or local script, as shown here:

```
RMAN> run { execute script full_backup; }
```

The following PRINT SCRIPT command prints out your global scripts.

```
RMAN> print global script full_backup;
```

CERTIFICATION OBJECTIVE 8.04

Miscellaneous Backup and Recovery Enhancements

This section briefly covers miscellaneous enhancements in the database backup and recovery area. Some of the enhancements apply to the RMAN interface only, and some apply to both the RMAN and user-performed backup and recovery.

Using the Database Control to Configure Backups

To use the OEM Database Control to configure your backup procedures, on the Database Control home page, click the Maintenance tab. In the Backup/Recovery region of the Maintenance page, choose Configure Backup Settings. You can use one of the following choices to tell RMAN where to place its target files:

- In the location you specify for the `FORMAT` option in a backup command
- In the location you specify with the `CONFIGURE CHANNEL FORMAT` option
- In the location you specify for the `DB_RECOVERY_FILE_DEST` (flash recovery area) parameter

If you don't use any of these alternatives, the backup files will go into an operating system-specific default directory.

Automatic Channel Failover

If one of the channels on which the RMAN is performing a backup fails, the RMAN will automatically try to use an alternative channel, provided you are using multiple channels. The backup process will continue without a hitch, although the RMAN reports the channel problem in the log files and saves the information in the `V$RMAN_OUTPUT` view.

Implementing Fast Recovery

During a normal recovery process in previous versions, you needed to restore the backup files first, before starting the recovery process. In Oracle Database 10*g*, you can use the backup files directly during a recovery process, instead of needing to restore them first. You can still perform the traditional restore/recovery in the RMAN, by using the `RESTORE DATABASE` command. However, for those special times when you really need a fast recovery, Oracle Database 10*g* offers the `SWITCH DATABASE` command.

The `SWITCH DATABASE` command doesn't restore your backup files from the backup location. The RMAN simply adjusts the pointers for the datafiles in the control file, so they now point to the backup files in your flash recovery area. Thus, RMAN will convert the backup files into the current datafiles, and then start the recovery process with these files.

on the job

You must use image copies rather than backups sets in order to utilize the fast recovery process with the `SWITCH DATABASE` command.

The `SWITCH DATABASE` command is straightforward:

```
RMAN> switch database to copy;
```

Since you don't need to copy any files, only rename the recovery files, the recovery process is very fast. However, it comes with a price: you need to make sure that you have additional backups made as soon as possible, to make up for the fact that you have used up one of your backups to recover your database.

Recovering Datafiles Without Backups

If you lose a datafile and it turns out that you don't have a backup, you can still use the RMAN to recover that datafile. You must, however, have a copy of the control file with references to the lost datafile, since RMAN needs the information from the control file to recover the datafile. In addition, you must have all the archived redo log files from the time the datafile was lost, up to the time you want to recover.

The ability to recover a file that has never been backed up has always been available from SQL*Plus, with the help of the `CREATE DATAFILE` *`X`* `AS` *`Y`* statement. Now, in Oracle Database 10*g*, you can create the lost file as part of an RMAN `RESTORE DATABASE` command.

Compressed Backups

Oracle Database 10*g* lets you compress RMAN backups to save on storage. However, there is a cost to the compression feature: your recovery times will be longer when you use compressed backup sets, since there is now the additional step of uncompressing the backup sets before starting the recovery. Nevertheless, all recovery procedures remain the same when you use a compressed backup set. The compression factor will depend on the nature of the data in your datafiles. Oracle recommends using this in preference to an external compression utility, and you certainly shouldn't use both utilities together.

exam watch

You can compress only a database that's using the Oracle Database 10g version server software. You must set the `COMPATIBLE` initialization parameter to a minimum of 10.0.0. You can't compress an image copy; you can compress a backup only if you are using backup sets.

Here is the RMAN command that lets you compress a backup set:

```
RMAN> backup as compressed backupset database;
```

RMAN disables backup compression by default. If you want to make backup compression the default mode, you may do so by using the following command.

```
RMAN> configure device type disk parallelism  4
2> backup type to compressed backupset;
old RMAN configuration parameters:
CONFIGURE DEVICE TYPE DISK BACKUP TYPE TO COPY PARALLELISM 1;
new RMAN configuration parameters:
CONFIGURE DEVICE TYPE DISK PARALLELISM 4 BACKUP TYPE TO COMPRESSED BACKUPSET;
new RMAN configuration parameters are successfully stored
released channel: ORA_DISK_1
RMAN>
```

The V$BACKUP_FILES view contains information on backup filenames and file sizes, and will also tell you about the compression status of the files. Here's a simple query that shows the size of a file, as well as whether RMAN will compress the file during a backup:

```
SQL> select fname, compressed, backup_type
     from v$backup_files;
```

Simplified Recovery Through RESETLOGS

When you perform an incomplete recovery or a recovery using a backup control file, you must open the database with the RESETLOGS option. The RESETLOGS option reinitializes the redo log files, and thus changes the database incarnation. You'll start a new sequence of redo logs after using the RESETLOGS option to open the database. In previous versions of Oracle, this meant that you couldn't use any backups from the previous incarnations (before the RESETLOGS operation) to recover the database. This is why you needed to perform a whole database backup immediately after a RESTLOGS operation.

In Oracle Database 10*g*, you can use backups taken before an incomplete recovery operation; that is, you can use backups from older incarnations of the database. So, you don't need to make a full backup after you use the RESETLOGS option. You can perform an incomplete recovery with the RESETLOGS option and immediately open the database, thereby substantially reducing your downtime following a media problem.

Since you can use older archived redo logs created prior to using the RESETLOGS option, you may now end up with situations where there may be multiple archived logs from various database incarnations with identical sequence numbers. In order to distinguish the archived redo logs from several database incarnations, Oracle Database 10*g* uses a *new format* for the archived redo log files. The initialization parameter LOG_ARCHIVE_FORMAT uses a text string and a set of variables to specify the default archive log format. Oracle applies the string generated by using this format to the string generated from the LOG_ARCHIVE_DEST parameter to come up with the complete path and filename for archived redo log files.

Previously, you used the following format for the LOG_ARCHIVE_FORMAT parameter to specify the archived redo log format:

```
LOG_ARCHIVE_FORMAT="log%t_%s.arc"
```

In this format specification, the variable t stands for thread number, and the variable s stands for the log sequence number.

The new archive redo log format in Oracle Database 10*g* is of the following form:

```
LOG_ARCHIVE_FORMAT="log%t_%s_%r.arc"
```

> exam watch
>
> ***The change in the LOG_ARCHIVE_FORMAT initialization parameter enables the new feature of a recovery through RESETLOGS.***

In the new archive log format, there is an additional variable r, which stands for the RESETLOGS identifier.

The V$DATABASE view contains several additional columns to help you manage information regarding all RESETLOGS operations. These columns include RESETLOGS_CHANGE#, RESETLOGS_TIME, and RESETLOGS_ID. The V$LOG_HISTORY and V$OFFLINE_RANGE views provide information about redo logs belonging to different database incarnations. Here's a simple query on the V$LOG_HISTORY view:

```
SQL> select sequence#, resetlogs_change#,
  2* resetlogs_time from v$log_history;
SEQUENCE# RESETLOGS_CHANGE# RESETLOGS
---------- ----------------- ---------
       831            318842 23-MAR-04
       832            318842 23-MAR-04
       833            318842 23-MAR-04
…
454 rows selected.
SQL>
```

Dropping a Database

Sometimes, you may want to drop test database or drop your old database after you successfully migrate to a new version of Oracle. It's very easy to drop an Oracle

Database 10*g* database. You can use the new command `DROP DATABASE` to remove a database, in both the RMAN as well as the SQL*Plus interface. You may also use the DBCA to drop the database.

Here is how you drop a database in the RMAN:

```
RMAN> drop database;
```

In SQL*Plus, you use an identical command:

```
SQL> drop database;
```

on the job

Your database must be mounted `EXCLUSIVE` and not be open in order to use the `DROP DATABASE` command, whether you're using the RMAN or SQL*Plus interface to drop a database. In addition, you must have SYSDBA privileges.

exam watch

The `DROP DATABASE` command is irrevocable—you can't roll back the command. Also, remember that Oracle automatically drops only the control fields and datafiles, not the archived redo log files and backup files.

Here are some features of the `DROP DATABASE` command:

- Oracle will drop all *control files and datafiles* automatically, whether you use the SQL*Plus, RMAN, or DBCA interface to drop a database.
- Oracle *doesn't* remove *archived redo logs and backups*.
- If you are using an SPFILE, Oracle will remove it automatically.

exam watch

After you drop a database, no matter which one of the three methods you use (RMAN, SQL*Plus, or DBCA), the RMAN catalog continues to list the dropped database information. Regardless of which method you use to drop a database, you need to use the following RMAN command to wipe out all traces of the dropped database from the RMAN registry:

```
RMAN> unregister database;
```

The `DROP  DATABASE` command removes all datafiles and control files. Oracle doesn't remove any backups or archived redo log files when you drop a database.

To make the RMAN remove all database backup copies as well as the archived redo log files, you can add the `INCLUDING BACKUPS` clause to your `DROP DATABASE` command in the RMAN, as shown here:

```
RMAN> drop database including backups;
```

Specifying Limits for Backup Duration

Sometimes, a DBA may run into a situation where a nightly backup is interfering with the performance of a critical database job. In order to avoid major spikes in resource use, you can dictate the database to "take it easy," by taking a longer time to finish the backup. The new `DURATION` command in the RMAN provides this capability.

exam watch

The `DURATION` option during backups replaces the old `RATE` and `READRATE` options used to modulate the RMAN file-reading speed.

You can use the `DURATION` command as an option for your regular backup commands, such as `BACKUP AS COPY`, to specify the time (in hours and minutes) a backup job can take. You specify the `DURATION` clause in the backup command in the following manner.

```
DURATION <hrs>:<mins>  [PARTIAL]  [MINIMIZE  {TIME|LOAD}]
```

In the `DURATION` clause, the options work as follows:

- **`PARTIAL`** Normally, when your database backup jobs run past the time interval you specify through the `DURATION` parameter, the RMAN job errors out and the backup is canceled. This is the RMAN's default behavior when it runs past any specified duration. You can override this default behavior by specifying the `PARTIAL` clause, which will prevent the issuing of any RMAN error messages.
- **`MINIMIZE TIME`** This option tells the RMAN to "hurry up" and finish as fast as it can.
- **`MINIMIZE LOAD`** This option tells the RMAN to "slow down" if it is well within its allotted time for a backup job.

on the job

You must use disks if you want to use the `MINIMIZE TIME` option, since you generally want a tape backup to end as quickly as possible.

Here is a simple example illustrating the use of this very interesting clause during a database backup:

```
RMAN> backup as copy
2>    duration 02:00
3>    minimize time database;
```

This BACKUP AS COPY command example specifies the following:

- Limit the backup time to two hours (DURATION 02:00)
- Run the backup at full speed, enabling it to possibly finish under the two-hour limit (MINIMIZE TIME)
- Back up the entire database (DATABASE)

exam watch

Remember that the DURATION clause's PARTIAL option leads to an error if the backup exceeds its time limit. The MINIMIZE TIME option gets the job done the fastest. The MINIMIZE LOAD option minimizes resource use.

Placing All Datafiles in Backup Mode

Oracle Database 10*g* allows you to place all datafiles in a backup mode during an online database backup. It also enables you to take all the datafiles out of the backup mode at the same time. Let's look at the new backup commands for performing these tasks.

Starting the Online Backup

Previously, during an online (hot) database backup, you needed to place each tablespace into the backup mode separately by using the BEGIN BACKUP TABLESPACE command. In Oracle Database 10*g*, you don't need to specify tablespaces individually during an online backup. You can use the ALTER DATABASE command to specify that all datafiles in the entire database be placed in the backup mode simultaneously. Your database must be open and in the ARCHIVELOG mode in order to use this command. Here's the syntax of the command:

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

on the job

You need to place the datafiles in an online backup mode only if you are performing a user-managed recovery, in order to guarantee that any fractured blocks can be recovered (assure consistency). You don't need this command in the RMAN, since an RMAN backup does not contain any fractured blocks.

If the backup process encounters any nonexistent, read-only, or offline files, Oracle skips them, and the backup will continue. No errors are issued.

If you encounter any of the following situations, your attempt to start an online backup will fail, with a "cannot start online backup error" message.

- A user-managed online backup is in progress.
- An RMAN online backup is in progress.
- A file is in the process of being made a read-only file.
- A file fails verification.

Ending the Online Backup

You must use the `ALTER DATABASE END BACKUP` command to end the online backup. You can issue this command when the database is in a mounted or an open state. The status of a datafile determines how the database treats that file. Here is a summary of the relationship between file status and the `END BACKUP` command:

- Offline files will cause a warning message, but the `END BACKUP` command itself will complete successfully.
- Read-only files are skipped by the database.
- Nonexistent datafiles are also skipped.
- You can't use the `END BACKUP` command while the database is going through an RMAN backup.

Automatic Auxiliary Instance Creation

Oracle Database 10*g*, like its predecessors, lets you perform a tablespace point-in-time recovery (TSPITR) to recover from certain database errors. You may specify the previous point in time you want to revert to by using a sequence number or an SCN.

The biggest headache in the previous Oracle versions while performing a TSPITR was that you had to create an *auxiliary database instance* and remove it after finishing the TSPITR process. Setting up the auxiliary instance was complex, making this method of recovery unappetizing to many DBAs. In Oracle Database 10*g*, you don't need to create the auxiliary instance. Oracle will now create the auxiliary instance and remove it after the recovery is over.

on the job

Since you are creating your auxiliary instance in the same location as your primary database, you need to be aware of the possibility of some performance degradation during the TSPITR operation.

You must provide the locations for all files of the auxiliary instance to the RMAN, so it can create the instance. Here are the various ways in which you can submit the auxiliary file location to the RMAN, ranked from the most important to the least important:

- AUXILIARY DESTINATION
- SET NEWNAME FOR DATAFILE
- CONFIGURE AUXNAME FOR DATAFILE
- DB_FILE_NAME_CONVERT

Oracle recommends that you must use the AUXILIARY DESTINATION argument to provide the file locations for a RECOVER TABLESPACE operation.

INSIDE THE EXAM

The exam will certainly query your knowledge of the flash recovery area. You must know the relevant initialization parameters for creating the flash recovery area and the kinds of files you can store in it. What are the differences between recovery-related files and other files in the flash recovery area? You may also see a question on how control file and redo log file placement will depend on the combination of the flash recovery area and the OMF file destination parameters (DB_CREATE_ONLINE_DEST_*n* and DB_CREATE_FILE_DEST).

You must understand the new RMAN commands like BACKUP AS COPY and RECOVER COPY OF DATAFILE. What is the new RMAN command to back up an entire database? What is the command to implement a fast recovery using RMAN? What do the DURATION clause and the PARTIAL, MINIMIZE LOAD, and MINIMIZE TIME options mean during a backup command?

Expect a question on the new recovery technique that uses incrementally updated backups. What command do you use to recover with an incrementally updated backup? Expect some questions on the new fast incremental backup feature and how you enable it (by enabling block change tracking). What is the change tracking file? What does the new CTWR background process do? Know the details of the new compressed backups feature. You most certainly will face questions on using the simplified recovery process through RESETLOGS. What changes in the initialization parameters make this possible? You must remember the commands that enable you to implement a fast recovery (SWITCH DATABASE TO COPY).

Know the steps in dropping a database. What files are removed by Oracle when you issue the DROP DATABASE command?

CERTIFICATION SUMMARY

This chapter dealt with several interesting enhancements related to backup and recovery. You were introduced to the new flash recovery area concept, which helps centralize all your recovery-related storage efforts. You then looked at the new techniques to reduce your incremental backup times. You also learned how to recover faster by applying incremental backups to datafile image copies.

Several new RMAN commands were explained in this chapter, including the `BACKUP AS COPY` and `CATALOG` commands. You also learned about the new scripting enhancements in the RMAN.

You learned about the new `SWITCH DATABASE` command, which helps you cut back on restore/recovery times in a crisis. You also learned how to compress your RMAN backups. You learned about the enhancements that let you avoid needing to back up your databases after a `RESETLOGS` operation. You also learned how you can use archived redo logs from a previous database incarnation. You learned how to drop the database with the `DROP DATABASE` command. You saw how you could use the `DURATION` option to control backup windows during certain times. You also saw how Oracle Database 10*g* can help you during a TSPITR operation, by automatically creating and removing the auxiliary instance.

✓ TWO-MINUTE DRILL

Using the Flash Recovery Area

□ The flash recovery area is the default storage location for all recovery-related files.

□ You may use an Oracle-managed directory, operating system file system, or an ASM disk group for your flash recovery area.

□ Oracle automatically deletes obsolete files and files that have been backed up to tape.

□ If you don’t use OMF files, Oracle can’t automatically delete unnecessary recovery files from the flash recovery area.

□ The flash recovery area contains datafile copies, control file autobackups, archived redo log files, RMAN backup files, and flashback logs.

□ Oracle recommends that you store a duplexed copy of the online redo log files and the current control file in your flash recovery area.

□ The size of the flash recovery area should be greater than the sum of the database size, any incremental backups, and all the archived logs on disk.

□ The size of the flash recovery area also depends on your RMAN retention policy.

□ You use the two initialization parameters DB_RECOVERY_FILE_DEST and DB_RECOVERY_FILE_DEST_SIZE to configure the flash recovery area.

□ Both the flash recovery area initialization parameters are dynamic.

□ You must first set the DB_RECOVERY_FILE_DEST_SIZE parameter before you can set the DB_RECOVERY_FILE_DEST parameter.

□ You can disable the flash recovery area by using the DB_RECOVERY_FILE_DEST parameter and setting it to a null location.

□ You can back up the flash recovery area by using the command BACKUP RECOVERY AREA, using the RMAN interface.

□ The BACKUP RECOVERY FILES command in the RMAN backs up all recovery files, irrespective of their location.

□ Oracle sends you both warning and critical alerts when the flash recovery area space starts to fill up.

□ The V$RECOVERY_FILE_DEST view is your main source for information regarding the flash recovery area.

Using Incremental Backups

❑ In Oracle Database 10*g*, you can apply incremental backups to image copies of datafiles.

❑ Oracle prefers using the incremental backup approach to using archived logs.

❑ Oracle Database 10*g* maintains a change tracking file, where the change tracking writer (CTWR) background process writes information about block changes.

❑ The change tracking file is automatically deleted when you disable block change tracking.

❑ The ratio between the value of the BLOCKS_READ column and the DATAFILE_BLOCKS column, shown in the `V$BACKUP_DATAFILE` view, will tell you if you should make more frequent incremental backups.

Enhanced RMAN Commands

❑ The RMAN `COPY` command is deprecated in Oracle Database 10*g*.

❑ The `BACKUP AS COPY` command creates image copies of the database, tablespaces, and datafiles.

❑ The `CATALOG` command helps you search for new files that need to be cataloged.

❑ The `UNCATALOG` command lets you uncatalog items from the RMAN recovery catalog.

❑ The `CATALOG START WITH` command lets you specify a directory for RMAN to search for recovery-related files.

❑ You can convert RMAN stored scripts into text file-based scripts and vice versa.

❑ The global scripts feature lets several databases use common scripts. You must use a recovery catalog for this feature to work.

Miscellaneous Backup and Recovery Enhancements

❑ If an RMAN channel fails during a backup, the RMAN will automatically switch over to an alternative channel, provided you have configured multiple channels.

- □ Using the `SWITCH DATABASE` command, you can recover directly from backup files, without restoring them first.
- □ You can recover a datafile even if you don’t have a backup for it, as long as you can get the pertinent information from the relevant control file.
- □ The RMAN lets you compress backup sets to save on space requirements.
- □ You can now recover through a `RESETLOGS` operation, because backups/archives from previous incarnations are usable.
- □ The new `LOG_ARCHIVE_FORMAT` parameter contains a new `RESETLOGS` identifier to help identify the redo logs from different database incarnations.
- □ You can now drop a database using either the RMAN or SQL*Plus interface, with the command `DROP DATABASE`.
- □ When you drop a database, Oracle will remove datafiles, control files, and the SPFILE automatically.
- □ Oracle doesn’t automatically remove the archived redo logs and backup files when you drop a database.
- □ No matter which method you use to drop the database, you must use the RMAN command `UNREGISTER DATABASE` to remove database information from the RMAN catalog.
- □ Using the `DURATION` clause, you can set limits on your backup job duration.
- □ The `PARTIAL` option of the `DURATION` clause prevents errors when you cross the time interval for a backup.
- □ The `MINIMIZE TIME` option of the `DURATION` clause will speed up your backups.
- □ The `MINIMIZE LOAD` option of the `DURATION` clause may slow down your backups.
- □ You can put all datafiles in the backup mode simultaneously with the command `ALTER DATABASE BEGIN BACKUP`.
- □ The RMAN can now automatically create and remove the auxiliary instance necessary during a TSPITR operation.

SELF TEST

The following questions will help you measure your understanding of the material presented in this chapter. Read all the choices carefully, because there might be more than one correct answer. Choose all correct answers for each question.

Using the Flash Recovery Area

1. Which of the following items are stored in the flash recovery area?

   A. Datafiles

   B. Undo segments

   C. Datafile copies

   D. Archived redo log files

2. When can you create a flash recovery area?

   A. At database creation time

   B. Anytime, since you can create it using dynamic initialization parameters

   C. Only after you shut down the database and start it up in the mount state

   D. Only if the database is already running

3. The DBA is running low on space and can provide only the least possible space to the flash recovery area. The DBA should, at a minimum, have enough space in the flash recovery area to store which of the following items?

   A. Archived logs not yet saved to tape

   B. Online redo log files and current control files

   C. Datafile copies and archived redo log files

   D. Archived redo log files

4. What happens once the DBA disables the flash recovery area?

   A. The RMAN will continue to be able to access the files in the flash recovery area.

   B. The RMAN will be unable to access the files in the flash recovery area.

   C. The RMAN will be unable to access the automatic space management features of the flash recovery area.

   D. The RMAN can use only the automatic space management features of the flash recovery area.

5. If the DBA specifies the `DB_CREATE_FILE_DEST` and the `DB_RECOVERY_FILE_DEST` parameters at database creation time, what will Oracle do?

A. Create an OMF control file in both locations specified by the two parameters

B. Create an OMF control file in the flash recovery area only

C. Create an OMF control file in the system-specified default location

D. Create an OMF control file in the location specified by the `DB_RECOVERY_FILE_DEST` parameter only

Using Incremental Backups

6. What does using incrementally updated backups involve?

A. The updating of archived redo log files with incremental backups

B. The direct updating of datafiles with incremental backups

C. The updating of image copies of datafiles with incremental backups

D. The updating of the control files with incremental backups

7. To apply incremental backups, which of the following commands must you use?

A. `RECOVER BACKUP OF DATABASE`

B. `RECOVER DATAFILE`

C. `RECOVER COPY OF DATAFILE`

D. `RECOVER COPY OF DATABASE`

8. Incremental backups apply changes at what level?

A. Transaction level

B. Oracle block level

C. Operating system block level

D. Datafile level

9. What does the CTWR process do?

A. Updates the incremental backups

B. Updates the change tracking file

C. Applies the database block changes

D. Checks for space in the flash recovery area

10. To enable block change tracking in your database, what must you do?

A. Use the `ENABLE BLOCK CHANGE TRACKING` command

B. Do nothing, since Oracle enables block change tracking by default

C. First create the change tracking file in the default location

D. First restart the database in the mount state, before creating the change tracking file

Enhanced RMAN Commands

11. What can you do using the new `BACKUP AS COPY` command?

A. Extract image copies directly from backup sets

B. Make image copies without needing to extract them from backup sets

C. Make copies at all levels except the database level

D. Make an image copy of only a single file at a time

12. Which of the following commands backs up the entire database?

A. `BACKUP DATABASE`

B. `BACKUP COPY OF DATABASE`

C. `BACKUP ALL`

D. `BACKUP BACKUPSET DATABASE`

13. What does the RMAN `CATALOG` command do?

A. Helps you create image copies of datafiles

B. Helps you create image copies of current control files

C. Helps you catalog any new files that aren't a part of the recovery catalog

D. Helps you catalog only the old files

14. What does the RMAN `BACKUP CONTROLFILECOPY ALL` command do?

A. Backs up all previously made control file copies

B. Backs up all control files that were never backed up before

C. Makes a copy of all current control files

D. Stores the backed up control file copies on tape

15. What happens when you uncatalog all the metadata about backup pieces from the RMAN catalog?

A. The RMAN automatically removes the catalog.

B. The RMAN removes the backup set information from the catalog as well.

C. You can't remove all of the backup pieces from the catalog.

D. The RMAN replaces the backup sets with image copies.

Miscellaneous Backup and Recovery Enhancements

16. The DBA needs to recover the production database in the shortest time possible. What is the DBA's best choice?

A. Use the `RESTORE DATABASE` command
B. Use the `RECOVER DATABASE` command
C. Use the `SWITCH BACKUP` command
D. Use the `SWITCH DATABASE` command

17. Which of the following backups can you compress?

A. Only backup sets
B. Only image copies
C. Both image copies and backup sets
D. Only single datafile copies

18. Which of the following is true about using the `RESETLOGS` option?

A. You don't need to back up the database.
B. You must back up the database right away.
C. You can use archive redo logs from an older incarnation of the database.
D. You can't use archive redo logs from an older incarnation of the database.

19. What happens when you use the `DROP DATABASE` command?

A. Only the datafiles, control files, and archived redo logs are removed.
B. Only the datafiles, control files, and backups are removed.
C. Only the datafiles, control files, and online redo log files are removed.
D. Only the datafiles, archived redo log files, and backups are removed.

20. Which of the following commands tells the database to slow down if it is within its allotted time for a database backup?

A. `MINIMIZE PARTIAL`
B. `MAXIMIZE LOAD`
C. `MINIMIZE TIME`
D. `MINIMIZE LOAD`

LAB QUESTIONS

1. Show how you would drop a database using the RMAN. Show the step-by-step procedures in detail.

2. Show how you would catalog all the files in a disk location using the RMAN.

SELF TEST ANSWERS

Using the Flash Recovery Area

1. ☑ **C** and **D.** Oracle stores datafile copies and archived redo log files in the flash recovery area.
   ☒ **A** is incorrect since Oracle doesn't store datafiles in the flash recovery area—it stores datafile *copies*. **B** is incorrect since there are no undo-related files in the flash recovery area. All undo-related information is stored in the undo tablespace.

2. ☑ **A** and **B. A** is correct since you can choose the flash recovery area option during database creation when you use the DBCA tool. **B** is correct as well, because the initialization parameters `DB_RECOVERY_FILE_DEST` and `DB_RECOVERY_FILE_DEST_SIZE` are dynamic parameters.
   ☒ **C** is wrong since you don't need to restart the database to create a flash recovery area. **D** is incorrect because, while you can configure the flash recovery area with dynamic initialization parameters, it doesn't imply that you can configure the area *only* when the database is running.

3. ☑ **A.** Oracle recommends that, at a minimum, you must keep the archived logs not yet saved to tape in the flash recovery area.
   ☒ **B, C,** and **D** offer invalid recommendations.

4. ☑ **A** and **C. A** is correct because, even after you disable the flash recovery area, the RMAN continues to be able to access any recovery-related files it stored in the flash recovery area. **C** is correct because you'll be unable to access the automatic space management features of the flash recovery area if you disable the feature.
   ☒ **B** is incorrect since you continue to be able to access the files even after the flash recovery area is disabled. **D** is wrong since you can't use any automatic features of the flash recovery area after you disable it.

5. ☑ **A.** Oracle will create an OMF-based control file in both the locations.
   ☒ **B, C,** and **D** identify the wrong locations.

Using Incremental Backups

6. ☑ **C.** Incrementally updated backups involve the updating of image copies of datafiles with incremental backups.
   ☒ **A** is incorrect since you never update the archived redo logs with incremental backups. **B** is wrong since you never update datafiles with incremental backups—you update datafile *copies*. **D** is wrong because you don't update any control files with incremental backups.

7. ☑ **C** and **D.** To apply incremental backups, you use the RECOVERY COPY OF DATAFILE or RECOVERY COPY OF DATABASE command.
☒ **A** and **B** offer the wrong commands.

8. ☑ **B.** Incremental changes apply changes to datafiles at the Oracle block level.
☒ **A** is wrong because it is the archived redo logs that apply changes at the transaction level. In fact, this is why the application of block changes is much faster during a recovery. **C** is wrong since it is the Oracle block level, not the operating system block level, that Oracle applies changes to the datafiles. **D** is wrong since changes aren't applied at the granularity of a datafile, but at the granularity of an Oracle data block.

9. ☑ **B.** The CTWR process writes block change information to the change tracking file.
☒ **A** is wrong since the CTWR process doesn't update the incremental backups. **C** is wrong since the CTWR process doesn't apply block level changes—it merely *records* them. **D** is wrong because the CTWR isn't in charge of checking for space in the flash recovery area.

10. ☑ **A.** To enable block change tracking in your database, you must enable it explicitly, by using the ENABLE BLOCK CHANGE TRACKING command.
☒ **B** is wrong since Oracle *disables* block change tracking by default. **C** and **D** are wrong because you don't need to create any files to enable block change tracking. It is the job of Oracle to create the change tracking file automatically when you enable block change tracking.

Enhanced RMAN Commands

11. ☑ **B.** The BACKUP AS COPY command enables you to make direct image copies of files, without needing to extract them from backup sets as before.
☒ **A** is wrong since you don't need to extract image copies from backup sets anymore, once you use this command. **C** is wrong because the command will make image copies at all levels, including the database level. **D** is wrong since the command can make copies of one or a set of files simultaneously.

12. ☑ **A.** The BACKUP DATABASE command backs up the entire database.
☒ **B** is wrong since the BACKUP COPY OF DATABASE command will back up a *previous* image copy of all the database and control files. **C** is wrong because the command is made up to mislead you. **D** is wrong since the command given there has a syntax error.

13. ☑ **C.** The CATALOG command helps you recatalog the recovery area in search of new files.
☒ **A** and **B** are wrong since the command doesn't help you *make* copies, it only catalogs them. **D** is wrong because the command isn't limited to cataloging old files.

14. ☑ **A.** The BACKUP CONTROLFILE ALL command backs up all previously made control file copies.
☒ **B** and **C** are wrong since the command doesn't back up the current control files, even if they have never been backed up before. **D** is wrong because the command has nothing to do with tape backup.

15. ☑ **B.** When you remove all backup pieces for a backup set, the RMAN will remove the backup set as well from its recovery catalog.
☒ **A** is wrong since RMAN never removes the catalog automatically. **C** is wrong because you *can* remove all the backup pieces from the catalog. **D** is wrong since RMAN doesn't replace any backup sets with image copies.

Miscellaneous Backup and Recovery Enhancements

16. ☑ **D.** The SWITCH DATABASE commands lets you recover in the shortest time possible, because all you do is simply point the control file to the backup files instead of the old datafiles. You can immediately start the recovery process, since there is no real restore process involved.
☒ **A** will perform a database restore and start the recovery process but is very time consuming and therefore will not be the fastest way to recover. **B** is wrong because the RECOVER DATABASE command needs a restore of the database files from your backup location before you start the recovery. Thus, it takes much longer to recover with this method. **C** is incorrect since it points to a nonexistent command.

17. ☑ **A.** You can compress only a backup set, not an image copy.
☒ **B** and **C** are wrong because they state that you can compress image copies, which isn't true. **D** is wrong since there is no limitation that states that you can compress only single datafile copies; you may copy entire backup sets.

18. ☑ **A** and **C. A** is correct since in Oracle Database 10*g*, you don't need to back up your database after a RESETLOGS operation. **C** is correct because you can use older archive logs from a different incarnation.
☒ **B** is wrong because you don't need to back up your database after a RESTLOGS operation in Oracle Database 10*g*. **D** is wrong since you *can* use older archive logs from a different database incarnation.

19. ☑ **B.** When you use the DROP DATABASE command, all datafiles, current control files, and backups are removed.
☒ **A, C,** and **D** list the wrong sets of files that the DROP DATABASE command will cause to be removed from your server.

20. ☑ **D.** The MINIMIZE LOAD option tells the database to slow down if it appears that its going to make its target interval of time.
☒ **A** and **B** are wrong since MINIMIZE PARTIAL and MAXIMIZE LOAD are nonexistent options. **C** is wrong because MINIMIZE TIME tells the database to work harder.

LAB ANSWERS

1. To drop a database using the RMAN, follow this procedure:

Connect to the target database through the RMAN, making sure you are using the recovery catalog:

```
rman target/catalog  rman/rman@testdb
```

Catalog all the database backups, both in the flash recovery area and in other locations:

```
RMAN> catalog start with '/u01/app/oracle/rcv_area'; # flash recovery
area
RMAN> catalog start with '/u09/app/oracle/arch_dest2'; # second archive
dest
```

Delete all backups and copies associated with the database:

```
RMAN> delete backupset; # deletes all backups
RMAN> delete copy; # delete all image copies (including archived logs)
```

Finally, drop the database and automatically unregister it from the recovery catalog:

```
RMAN> drop database;
```

2. To catalog all files in a disk location, first connect to the target database as before and provide the RMAN with the names of the directories you want it to catalog:

```
RMAN> catalog start with '/u01/app/oracle/datafiles/';
```

The following command will catalog all files in the recovery area:

```
RMAN> catalog recovery area;
```