

# CHAPTER 12

Using Recovery  
Manager (RMAN)



In Chapter 11, we discussed a number of different ways in which we can back up our data and protect the database from accidental, inadvertent, or deliberate corruption. Physical backups of the database ensure that no committed transaction is lost, and we can restore the database from any previous backup to the current point in time or any point in between; logical backups allow the DBA or a user to capture the contents of individual database objects at a particular point in time, providing an alternative recovery option when a complete database-restoration operation would have too big an impact on the rest of the database.

Oracle's Recovery Manager (RMAN) takes backup and recovery to a new level of protection and ease of use. Since RMAN's appearance in Oracle version 8, there have been a number of major improvements and enhancements that can make RMAN a "one-stop shopping" solution for nearly every database environment. In addition to the RMAN command-line interface improvements in Oracle 10g, all the RMAN functionality has been included in the web-based Oracle Enterprise Manager (OEM) interface as well, allowing a DBA to monitor and perform backup operations when only a web browser connection is available.

In this chapter, we'll use a number of examples of RMAN operations, both using command-line syntax and the OEM web interface. The examples will run the gamut from RMAN environment setup to back up, and the recovery and validation of the backup itself. We'll go into some detail about how RMAN manages the metadata associated with the database and its backups. Finally, we'll cover a number of miscellaneous topics, such as using RMAN to catalog backups made outside of the RMAN environment.

Oracle Database 11g brings even more functionality to an RMAN environment. The Data Recovery Advisor operates in both a proactive manner by detecting problems with the database before an application failure, as well as a reactive manner to analyze a failure and provide at least one repair option that will minimize downtime, if any! In virtually all scenarios, Data Recovery Advisor uses RMAN for its repair operation, incorporating new commands such as **list failure**, **change failure**, **advise failure**, and **repair failure**. In addition, RMAN supports a failover location for archived redo log files when the flash recovery area is full. Finally, the **duplicate** command now supports creating a duplicate database or physical standby database without using database backup files; this dramatically reduces the amount of time needed to create a copy of a database.

Due to the wide variety of tape backup management systems available, discussing any particular hardware configuration would be beyond the scope of this book. Instead, the focus in this chapter will be on using the flash recovery area, a dedicated area allocated on disk to store disk-based copies of all types of objects that RMAN can back up. The flash recovery area is new to Oracle 10g.

For all the examples in this chapter, we will use a recovery catalog with RMAN. Although most of the functionality of RMAN is available by only using the control file of the target database, benefits such as being able to store RMAN scripts and additional recovery capabilities far outweigh the relatively low cost of maintaining an RMAN user account in a different database.

## RMAN Features and Components

RMAN is more than just a client-side executable that can be used with a web interface. It comprises a number of other components, including the database to be backed up (the target database), an optional recovery catalog, an optional flash recovery area, and media management software to support tape backup systems. We will review each of these briefly in this section.

Many features of RMAN do not have equivalents in the backup methods presented in Chapter 11. We'll contrast the advantages and disadvantages of using RMAN versus the more traditional methods of backups.

## RMAN Components

The first, and minimal, component in the RMAN environment is the RMAN executable. It is available along with the other Oracle utilities in the directory \$ORACLE\_HOME/bin, and it's installed by default with both the Standard and Enterprise Editions of Oracle 11g. From a command-line prompt, you can invoke RMAN with or without command-line arguments; in the following example, we're starting up RMAN using operating system authentication without connecting to a recovery catalog:

```
[oracle@dw ~]$ rman target /
RMAN>
```

The command-line arguments are optional; we can specify our target database and a recovery catalog from the **RMAN>** prompt also. In Figure 12-1, you can see how to access RMAN features from Oracle Enterprise Manager.

RMAN would not be of much use unless we have a database to back up. One or more target databases can be cataloged in the recovery catalog; in addition, the control file of the database being backed up contains information about backups performed by RMAN. From within the RMAN client, you can also issue SQL commands for those operations you cannot perform with native RMAN commands.

The RMAN recovery catalog, whether using the target database control file or a dedicated repository in a separate database, contains the location of recovery data, its own configuration settings, and the target database schema. At a minimum, the target database control file contains this data; to be able to store scripts and to maintain a copy of the target database control file, a recovery catalog is highly recommended. In this chapter, all examples will use a recovery catalog.

The screenshot displays the Oracle Enterprise Manager 11g Database Control interface. At the top, it shows 'ORACLE Enterprise Manager 11g Database Control' with 'Help' and 'Logout' links. The user is logged in as 'As RJB'. The main navigation bar includes 'Home', 'Performance', 'Availability', 'Server', 'Schema', 'Data Movement', and 'Software and Support'. The 'Availability' section is expanded to show 'Backup/Recovery' and 'Oracle Secure Backup'. Under 'Backup/Recovery', there are two sub-sections: 'Setup' and 'Manage'. 'Setup' includes links for 'Backup Settings', 'Recovery Settings', and 'Recovery Catalog Settings'. 'Manage' includes links for 'Schedule Backup', 'Manage Current Backups', 'Backup Reports', 'Manage Restore Points', 'Perform Recovery', and 'View and Manage Transactions'. A mouse cursor is hovering over 'View and Manage Transactions'. Below these are 'Related Links' for 'Access', 'Advisor Central', 'Alert Log Contents', 'All Metrics', 'Blackouts', 'EM SQL History', 'Metric Collection Errors', 'Monitoring Configuration', 'Policy Groups', 'Scheduler Central', 'Target Properties', 'User-Defined Metrics', 'Alert History', 'Baseline Metric Thresholds', 'Metric and Policy Settings', 'Monitor in Memory Access Mode', and 'SQL Worksheet'. At the bottom, there are 'Database', 'Help', and 'Logout' links, and a copyright notice: 'Copyright © 1996, 2007, Oracle. All rights reserved. Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners. About Oracle Enterprise Manager'.

FIGURE 12-1 Accessing RMAN functionality from OEM

As of Oracle 10g, the *flash recovery area* simplifies disk-based backup and recovery by defining a location on disk to hold all RMAN backups. Along with the location, the DBA can also specify an upper limit to the amount of disk space used in the flash recovery area. Once a retention policy is defined within RMAN, RMAN will automatically manage the backup files by deleting obsolete backups from both disk and tape. The initialization parameters related to the flash recovery area are covered in the next section.

To access all non-disk-based media, such as tape and DVD-ROM, RMAN utilizes third-party media management software to move backup files to and from these offline and near-line devices, automatically requesting the mount and dismount of the appropriate media to support backup and restore operations. Most major media management software and hardware vendors have device drivers that directly support RMAN.

## RMAN vs. Traditional Backup Methods

There are very few reasons not to use RMAN as your main tool for managing backups. Here are some of the major features of RMAN that are either not available with traditional backup methods or have significant restrictions using traditional backup methods:

- **Skip unused blocks** Blocks that have never been written to, such as blocks above the high water mark (HWM) in a table, are not backed up by RMAN when the backup is an RMAN backupset. Traditional backup methods have no way to know which blocks have been used.
- **Backup compression** In addition to skipping blocks that have never been used, RMAN can also use an Oracle-specific binary compression mode to save space on the backup device. Although operating system-specific compression techniques are available with traditional backup methods, the compression algorithm used by RMAN is customized to maximize the compression for the typical kinds of data found in Oracle data blocks. Although there is a slight increase in CPU time during an RMAN compressed backup or recovery operation, the amount of media used for backup may be significantly reduced, as well as network bandwidth if the backup is performed over the network. Multiple CPUs can be configured for an RMAN backup to help alleviate the compression overhead.
- **Open database backups** Tablespace backups can be performed in RMAN without using the **begin/end backup** clause with **alter tablespace**. Whether using RMAN or a traditional backup method, however, the database must be in ARCHIVELOG mode.
- **True incremental backups** For any RMAN incremental backup, unchanged blocks since the last backup will not be written to the backup file. This saves a significant amount of disk space, I/O time, and CPU time. For restore and recovery operations, RMAN supports *incrementally updated backups*. Data blocks from an incremental backup are applied to a previous backup to potentially reduce the amount of time and number of files that need to be accessed to perform a recovery operation. We will cover an example of an incrementally updated backup later in this chapter.
- **Block-level recovery** To potentially avoid downtime during a recovery operation, RMAN supports *block-level recovery* for recovery operations that only need to restore or repair a small number of blocks identified as being corrupt during the backup operation. The rest of the tablespace and the objects within the tablespace can remain online while RMAN repairs the damaged blocks. The rows of a table not being repaired by RMAN are even available to applications and users.

- **Multiple I/O channels** During a backup or recovery operation, RMAN can utilize many I/O channels, via separate operating system processes, to perform concurrent I/O. Traditional backup methods, such as a Unix **cp** command or an Oracle **export**, are typically single-threaded operations.
- **Platform independence** Backups written with RMAN commands will be syntactically identical regardless of the hardware or software platform used, with the only difference being the media management channel configuration. On the other hand, a Unix script with lots of **cp** commands will not run very well if the backup script is migrated to a Windows platform!
- **Tape manager support** All major enterprise backup systems are supported within RMAN by a third-party media management driver provided by a tape backup vendor.
- **Cataloging** A record of all RMAN backups is recorded in the target database control file, and optionally in a recovery catalog stored in a different database. This makes restore and recovery operations relatively simple compared to manually tracking operating system–level backups using “copy” commands.
- **Scripting capabilities** RMAN scripts can be saved in a recovery catalog for retrieval during a backup session. The tight integration of the scripting language, the ease of maintaining scripts in RMAN, and the Oracle scheduling facility make it a better choice than storing traditional operating system scripts in an operating system directory with the operating system’s native scheduling mechanisms.
- **Encrypted backups** RMAN uses backup encryption integrated into Oracle Database 11g to store encrypted backups. Storing encrypted backups on tape requires the Advanced Security Option.

In a few limited cases, a traditional backup method may have an advantage over RMAN; for example, RMAN does not support the backup of password files and other non-database files such as `tnsnames.ora`, `listener.ora`, and `sqlnet.ora`. However, these files are relatively static in nature, and they can easily be backed up and restored using a traditional backup method such as the Unix **cp** command.

## Backup Types

RMAN supports a number of different backup methods, depending on your availability needs, the desired size of your recovery window, and the amount of downtime you can endure while the database or a part of the database is involved in a recovery operation.

### Consistent and Inconsistent Backups

A physical backup can be classified by being a *consistent* or an *inconsistent* backup. In a consistent backup, all datafiles have the same SCN; in other words, all changes in the redo logs have been applied to the datafiles. Because an open database with no uncommitted transactions may have some dirty blocks in the buffer cache, it is rare that an open database backup can be considered consistent. As a result, consistent backups are taken when the database is shut down normally or in a MOUNT state.

In contrast, an inconsistent backup is performed while the database is open and users are accessing the database. Because the SCNs of the datafiles typically do not match when an inconsistent backup is taking place, a recovery operation performed using an inconsistent backup

must rely on both archived and online redo log files to bring the database into a consistent state before it is opened. As a result, a database must be in ARCHIVELOG mode to use an inconsistent backup method.

### Full and Incremental Backups

Full backups include all blocks of every datafile within a tablespace or a database; it is essentially a bit-for-bit copy of one or more datafiles in the database. Either RMAN or an operating system command can be used to perform a full backup, although backups performed outside of RMAN must be cataloged with RMAN before they can be used in an RMAN recovery operation.

In Oracle 11g, incremental backups can be level 0 or level 1. A level 0 backup is a full backup of all blocks in the database that can be used in conjunction with differential, incremental, or cumulative incremental level 1 backups in a database recovery operation. A distinct advantage to using an incremental backup in a recovery strategy is that archived and online redo log files may not be necessary to restore a database or tablespace to a consistent state; the incremental backups may have some or all of the blocks needed. An example of using level 0 and level 1 incremental backups is presented later in this chapter. Incremental backups can only be performed within RMAN.

### Image Copies

Image copies are full backups created by operating system commands or RMAN **backup as copy** commands. Although a full backup created with a Unix **cp** command can be later registered in the RMAN catalog as a database backup, doing the same image copy backup in RMAN has the advantage of checking for corrupt blocks as they are being read by RMAN and recording the information about the bad blocks in the data dictionary. Image copies are the default backup file format in RMAN.

This is a great feature of Oracle 11g's RMAN for the following reason: If you add another datafile to a tablespace, you need to also remember to add the new datafile to your Unix script **cp** command. By creating image copies using RMAN, all datafiles will automatically be included in the backup. Forgetting to add the new datafile to a Unix script will make a recovery situation extremely inconvenient at best and a disaster at worst.

### Backupsets and Backup Pieces

In contrast to image copies, which can be created in most any backup environment, backupsets can be created and restored only with RMAN. A *backupset* is an RMAN backup of part or all of a database, consisting of one or more *backup pieces*. Each backup piece belongs to only one backupset, and can contain backups of one or many datafiles in the database. All backupsets and pieces are recorded in the RMAN repository, the same as any other RMAN-initiated backup.

### Compressed Backups

For any Oracle11g RMAN backup creating a backupset, compression is available to reduce the amount of disk space or tape needed to store the backup. Compressed backups are only usable by RMAN, and they need no special processing when used in a recovery operation; RMAN

automatically decompresses the backup. Creating compressed backups is as easy as specifying **as compressed backupset** in the RMAN **backup** command.

## Overview of RMAN Commands and Options

In the next few sections, we'll review the basic set of commands you'll use on a regular basis. We'll show you how to make your job even easier by persisting some of the settings in an RMAN session; in addition, we'll set up the retention policy and the repository we'll use to store RMAN metadata.

At the end of this section, we'll review the initialization parameters related to RMAN backups and the flash recovery area.

## Frequently Used Commands

Table 12-1 provides a list of the most common RMAN commands you'll use on a regular basis, along with some common options and caveats for each command. For the complete list of all RMAN commands and their syntax, see the *Oracle Database Backup and Recovery Reference, 11g Release 1*.

If backups use a flash recovery area (I presented the flash recovery area in Chapter 11), you can back up the database without any other explicit RMAN configuration by running the following command:

```
RMAN> backup database;
```

Note that this is a full backup and can be used with archived redo log files to recover a database. However, this is not a level 0 backup and cannot be used as part of an incremental backup strategy. See the "Backup Operations" section later in this chapter.

## Setting Up a Repository

Whether you use a repository for the metadata from one database or a hundred, the repository setup is very straightforward and needs to be done only once. The examples that follow assume that we have a default installation of an Oracle 11g database; the repository database itself can be used for other applications if there is no significant performance degradation when RMAN needs to update metadata in the repository.

### CAUTION

*Using an RMAN target database for the repository is strongly discouraged. Loss of the target database prevents any chance of a successful recovery of the database using RMAN because the repository metadata is lost along with the target database.*

The following sequence of commands creates a tablespace and a user to maintain the metadata in the repository database. In this and all subsequent examples, a database with a SID of **rac**, the RAC database we created in Chapter 10, is used for all repository operations.

---

<b>RMAN Command</b>	<b>Description</b>
@	Runs an RMAN command script at the pathname specified after the @. If no path is specified, the path is assumed to be the directory from which RMAN was invoked.
ADVISE FAILURE	Displays repair options for the failure found.
BACKUP	Performs an RMAN backup, with or without archived redo logs. Backs up datafiles, datafile copies, or performs an incremental level 0 or level 1 backup. Backs up an entire database, or a single tablespace or datafile. Validates the blocks to be backed up with the VALIDATE clause.
CATALOG	Adds information about file copies and user-managed backups to the repository.
CHANGE	Changes the status of a backup in the RMAN repository. Useful for explicitly excluding a backup from a restore or recovery operation, or to notify RMAN that a backup file was inadvertently or deliberately removed by an operating system command outside of RMAN.
CONFIGURE	Configures the persistent parameters for RMAN. The parameters configured are available during every subsequent RMAN session unless they are explicitly cleared or modified.
CONVERT	Converts datafile formats for transporting tablespaces or entire databases across platforms.
CREATE CATALOG	Creates the repository catalog containing RMAN metadata for one or more target databases. It is strongly recommended that this catalog not be stored in one of the target databases.
CROSSCHECK	Checks the record of backups in the RMAN repository against the actual files on disk or tape. Objects are flagged as EXPIRED, AVAILABLE, UNAVAILABLE, or OBSOLETE. If the object is not available to RMAN, it is marked UNAVAILABLE.
DELETE	Deletes backup files or copies and marks them as DELETED in the target database control file. If a repository is used, the record of the backup file is removed.
DROP DATABASE	Deletes the target database from disk and unregister it.
DUPLICATE	Uses backups of the target database (or use the live database) to create a duplicate database.
FLASHBACK DATABASE	Performs a Flashback Database operation, new to Oracle 10g. The database is restored to a point in the past by SCN or log sequence number using flashback logs to undo changes before the SCN or log sequence number, and then archived redo logs are applied to bring the database forward to a consistent state.

---

**TABLE 12-1** *Common RMAN Commands*

---

RMAN Command	Description
LIST	Displays information about backupsets and image copies recorded in the target database control file or repository. See REPORT for identifying complex relationships between backupsets.
RECOVER	Performs a complete or incomplete recovery on a datafile, a tablespace, or the entire database. Can also apply incremental backups to a datafile image copy to roll it forward in time.
REGISTER DATABASE	Registers a target database in the RMAN repository.
REPAIR FAILURE	Repairs one or more failures recorded in the automated diagnostic repository (ADR).
REPORT	Performs a detailed analysis of the RMAN repository. For example, this command can identify which files need a backup to meet the retention policy or which backup files can be deleted.
RESTORE	Restores files from image copies or backupsets to disk, typically after a media failure. Can be used to validate a restore operation without actually performing the restore by specifying the PREVIEW option.
RUN	Runs a sequence of RMAN statements as a group between { and }. Allows you to override default RMAN parameters for the duration of the execution of the group.
SET	Sets RMAN configuration settings for the duration of the RMAN session, such as allocated disk or tape channels. Persistent settings are assigned with CONFIGURE.
SHOW	Shows all or individual RMAN configured settings.
SHUTDOWN	Shuts down the target database from within RMAN. Identical to the SHUTDOWN command within SQL*Plus.
STARTUP	Starts up the target database. This command has the same options and function as the SQL*Plus STARTUP command.
SQL	Runs SQL commands that cannot be accomplished directly or indirectly using standard RMAN commands; for example, it can run <b>sql 'alter tablespace users offline immediate'</b> ; before restoring and recovering the USERS tablespace.
TRANSPORT TABLESPACE	Creates transportable tablespace sets from backup for one or more tablespaces.
VALIDATE	Examines a backup set and report whether its data is intact and consistent.

---

**TABLE 12-1** Common RMAN Commands (continued)

The tablespace that holds the repository database requires at least 125MB to hold recovery catalog entries; here is a space requirements breakdown by tablespace:

- 90MB in the SYSTEM tablespace
- 5MB in the TEMP tablespace
- 5MB in the UNDO tablespace
- 15MB in RMAN's default tablespace for each database registered in the recovery catalog
- 1MB for each online redo log file

Starting out with available free space of 125MB will in most cases be sufficient for the first year, and enabling additional extents of 50MB each will be sufficient in the long term depending on how many databases you manage in the recovery catalog. Overall, a very small amount of disk space compared to your terabyte data warehouse!

Connect to the repository database with SYSDBA privileges and create the RMAN account and the recovery catalog in the RMAN tablespace as follows:

```
[oracle@oc1 ~]$ sqlplus / as sysdba

SQL*Plus: Release 11.1.0.6.0 -
                Production on Tue Aug 28 20:56:24 2007

Copyright (c) 1982, 2007, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 -
  Production
With the Partitioning, Real Application Clusters, OLAP,
  Data Mining and Real Application Testing options

SQL> create tablespace rman datafile '+data1'
  2 size 125m autoextend on next 50m maxsize 500m;

Tablespace created.

SQL> grant recovery_catalog_owner to rman identified by rman;

Grant succeeded.

SQL> alter user rman default tablespace rman
  2 quota unlimited on rman;

User altered.

SQL>
```

Now that the RMAN user account exists in the repository database, we can start RMAN, connect to the catalog, and initialize the repository with the **create catalog** command:

```
[oracle@dw ~]$ rman catalog rman/rman@rac

Recovery Manager: Release 11.1.0.6.0 -
    Production on Tue Aug 28 21:24:30 2007

Copyright (c) 1982, 2007, Oracle. All rights reserved.

connected to recovery catalog database

RMAN> create catalog;

recovery catalog created

RMAN>
```

From this point on, using a repository is as easy as specifying the repository username and password on the RMAN command line with the **catalog** parameter or using the **connect catalog** command in an RMAN session. Within Oracle Enterprise Manager, you can persist the repository credentials as demonstrated in Figure 12-2.

In future OEM sessions, any RMAN backup or recovery operations will automatically use the recovery catalog.

---

ORACLE Enterprise Manager 11g 
Help Logout  
Database

---

Database Instance: dw.world > Recovery Catalogs >
Cancel Next

### Add Recovery Catalog: Database

Specify the host, port, and SID of a database with an existing recovery catalog along with the recovery catalog username and password. The recovery catalog should be stored in a dedicated database that is on a different disk than the target database. Protect the recovery catalog by backing it up.

≡ Host	<input type="text" value="oc1"/>
≡ Port	<input type="text" value="1521"/>
≡ SID	<input type="text" value="rac"/>
≡ Recovery Catalog Username	<input type="text" value="rman"/>
≡ Recovery Catalog Password	<input type="password" value="****"/>

Cancel Next

[Database](#) | [Help](#) | [Logout](#)

Copyright © 1996, 2007, Oracle. All rights reserved.  
 Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.  
[About Oracle Enterprise Manager](#)

**FIGURE 12-2** Persisting RMAN repository credentials

## Registering a Database

For each database for which RMAN will perform a backup or recovery, we must *register* the database in the RMAN repository; this operation records information such as the target database schema and the unique database ID (DBID) of the target database. The target database need only be registered once; subsequent RMAN sessions that connect to the target database will automatically reference the correct metadata in the repository.

```
[oracle@dw ~]$ rman target / catalog rman@rac

Recovery Manager: Release 11.1.0.6.0 -
    Production on Tue Aug 28 21:34:08 2007

Copyright (c) 1982, 2007, Oracle. All rights reserved.

connected to target database: DW (DBID=3048318127)
recovery catalog database Password: *****
connected to recovery catalog database

RMAN> register database;

database registered in recovery catalog
starting full resync of recovery catalog
full resync complete

RMAN>
```

In the preceding example, we connect to the target database using operating system authentication and to the repository with password authentication. All databases registered with the repository must have unique DBIDs; trying to register the database again yields the following error message:

```
RMAN> register database;

RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03009: failure of register command on default channel
             at 08/28/2007 21:38:44
RMAN-20002: target database already registered in recovery catalog

RMAN>
```

## Persisting RMAN Settings

To make the DBA's job easier, a number of RMAN settings can be *persisted*. In other words, these settings will stay in effect between RMAN sessions. In the example that follows, we use the **show** command to display the default RMAN settings:

```
RMAN> show all;

RMAN configuration parameters for database with db_unique_name DW are:
CONFIGURE RETENTION POLICY TO REDUNDANCY 1; # default
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 BACKUP TYPE TO BACKUPSET; # 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 ENCRYPTION FOR DATABASE OFF; # default
CONFIGURE ENCRYPTION ALGORITHM 'AES128'; # default
CONFIGURE COMPRESSION ALGORITHM 'BZIP2'; # default
CONFIGURE ARCHIVELOG DELETION POLICY TO NONE; # default
CONFIGURE SNAPSHOT CONTROLFILE NAME TO
'/u01/app/oracle/product/11.1.0/db_1/dbs/snapcf_dw.f'; # default
RMAN>

```

Any parameters that are set to their default values have **# default** at the end of the configuration setting. These parameters are easy to review and change using OEM, as demonstrated in Figure 12-3.

In the next few sections, we'll review a few of the more common RMAN persistent settings.

**ORACLE Enterprise Manager 11g** Database Control Help Logout Database

Database Instance: dw.world >

### Backup Settings

Device Backup Set Policy

#### Backup Policy

Automatically backup the control file and server parameter file (SPFILE) with every backup and database structural change

Autobackup Disk Location   
An existing directory or diskgroup name where the control file and server parameter file will be backed up. If you do not specify a location, the files will be backed up to the flash recovery area location.

Optimize the whole database backup by skipping unchanged files such as read-only and offline datafiles that have been backed up

Enable block change tracking for faster incremental backups

Block Change Tracking File   
Specify a location and file, otherwise an Oracle managed file will be created in the database area.

#### Tablespaces Excluded From Whole Database Backup

Populate this table with the tablespaces you want to exclude from a whole database backup. Use the Add button to add tablespaces to this table.

Select	Tablespace Name	Tablespace Number	Status	Contents
No Items Selected				

**TIP** These tablespaces can be backed up separately using tablespace backup.

#### Retention Policy

Retain All Backups  
You must manually delete any backups

Retain backups that are necessary for a recovery to any time within the specified number of days (point-in-time recovery)

Days   
Recovery Window

Retain at least the specified number of full backups for each datafile

Backups   
Redundancy

**FIGURE 12-3** RMAN persistent parameters in OEM

## Retention Policy

Backups can be automatically retained and managed using one of two methods: by a *recovery window* or by *redundancy*. Using a recovery window, RMAN will retain as many backups as necessary to bring the database to any point in time within the recovery window. For example, with a Recovery Window of seven days, RMAN will maintain enough image copies, incremental backups, and archived redo logs to ensure that the database can be restored and recovered to any point in time within the last seven days. Any backups that are not needed to support this recovery window are marked as OBSOLETE and are automatically removed by RMAN if a flash recovery area is used and disk space is needed for new backups.

In contrast, a redundancy retention policy directs RMAN to retain the specified number of backups or copies of each datafile and control file. Any extra copies or backups beyond the number specified in the redundancy policy are marked as OBSOLETE. As with a recovery window, obsolete backups are automatically removed if disk space is needed and a flash recovery area is used. Otherwise, you can use the **delete obsolete** command to remove the backup files and update the catalog.

If the retention policy is set to NONE, no backups or copies are ever considered obsolete, and the DBA must manually remove unneeded backups from the catalog and from disk.

In the following example, we will set the retention policy to a recovery window of four days (from a default redundancy policy of 1 copy):

```
RMAN> configure retention policy to recovery window of 4 days;
```

```
new RMAN configuration parameters:
CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 4 DAYS;
new RMAN configuration parameters are successfully stored
RMAN>
```

## Device Type

If the default device type is set to DISK and no pathname parameter is specified, RMAN uses the flash recovery area for all backups; you can easily override the disk backup location in OEM, as you can see in Figure 12-4. As with many of the simplified administration tasks from Oracle 11g, there is no need to allocate or deallocate a specific channel for backups unless you're using a tape device.

Although configuring a tape device is specific to your installation, in general terms we configure a tape device as follows:

```
RMAN> configure channel device type sbt
2> parms='ENV=(<vendor specific arguments>)';
```

### NOTE

**sbt** is the device type used for any tape backup subsystem, regardless of vendor.

Although we can use the flash recovery area to restore and recover our database entirely from disk, at some point it becomes inefficient to keep all our backups on disk, especially if we have a large recovery window. As a result, we can make copies of our backup files to tape, and RMAN will dutifully keep track of where the backups are in case we need to restore or recover the database from tape, or restore archived redo logs to roll forward an image copy in the flash recovery area.

The screenshot shows the Oracle Enterprise Manager 11g Database Control interface. At the top, it displays 'ORACLE Enterprise Manager 11g Database Control' and 'Database Instance: dw.world >'. The 'Backup Settings' section is active, with tabs for 'Device', 'Backup Set', and 'Policy'. Under 'Disk Settings', the 'Parallelsim' is set to 1. The 'Disk Backup Location' is set to '/u01/orabak/dw'. The 'Disk Backup Type' is set to 'Backup Set'. Below this, there are three radio button options: 'Backup Set' (selected), 'Compressed Backup Set', and 'Image Copy'. A 'Test Disk Backup' button is visible. Below the 'Disk Settings' section, there is a 'Tape Settings' section with a 'Test Tape Backup' button. A note states: 'Tape drives must be mounted before performing a backup. You should verify that the tape settings are valid by clicking on 'Test Tape Backup', before saving them.'

**FIGURE 12-4** *Configuring backup destination using OEM*

## Control File Autobackup

Because of the importance of the control file, we want to back it up at least as often as it changes due to modifications in the structure of the database. By default, the backup of the control file does not occur automatically. This is a strange default, considering the importance of the control file and how little disk space it takes to back it up. Fortunately, RMAN can easily be configured to back up the control file automatically, either any time a successful backup must be recorded in the repository or when a structural change affects the contents of the control file (in other words, cases when a control file backup must occur to ensure a successful recovery if and when a recovery operation is required).

```
RMAN> configure controlfile autobackup on;
```

```
new RMAN configuration parameters:
CONFIGURE CONTROLFILE AUTOBACKUP ON;
new RMAN configuration parameters are successfully stored
```

```
RMAN>
```

Every RMAN backup from this point on will automatically include a copy of the control file; the control file is also backed up whenever a new tablespace is created or another datafile is added to an existing tablespace.

## Backup Compression

If disk space is at a premium, you have a very large database, and you have some extra CPU capacity, it makes sense to compress the backups to save space. The files are decompressed automatically during a restore or recovery operation.

```

RMAN> configure device type disk backup type to compressed backupset;

```

```

new RMAN configuration parameters:
CONFIGURE DEVICE TYPE DISK BACKUP TYPE TO
      COMPRESSED BACKUPSET PARALLELISM 1;
new RMAN configuration parameters are successfully stored

```

```

RMAN>

```

Compressing backupsets may not be necessary if the operating system's file system has compression enabled or if the tape device hardware automatically compresses backups; however, RMAN's compression algorithm is tuned to efficiently back up Oracle data blocks, and as a result it may do a better job of compressing the backupsets.

## Initialization Parameters

A number of initialization parameters are used to control RMAN backups. We'll cover some of the more important parameters in this section.

### CONTROL\_FILE\_RECORD\_KEEP\_TIME

A record of all RMAN backups is kept in the target control file. This parameter specifies the number of days that RMAN will attempt to keep a record of backups in the target control file. After this time, RMAN will begin to reuse records older than this retention period. If RMAN needs to write a new backup record, and the retention period has not been reached, RMAN will attempt to expand the size of the control file. Usually, this is successful because the size of the control file is relatively small compared to other database objects. However, if space is not available for the expansion of the control file, RMAN will reuse the oldest record in the control file and write a message to the alert log.

As a rule of thumb, you should set CONTROL\_FILE\_RECORD\_KEEP\_TIME to several days beyond your actual recovery window to ensure that backup records are retained in the control file. The default is 7 days.

### DB\_RECOVERY\_FILE\_DEST

This parameter specifies the location of the flash recovery area. It should be located on a file system different from any database datafiles, control files, or redo log files, online or archived. If you lose the disk with the datafiles, the flash recovery area is gone too, mitigating the advantages of using a flash recovery area.

### DB\_RECOVERY\_FILE\_DEST\_SIZE

The parameter DB\_RECOVERY\_FILE\_DEST\_SIZE specifies an upper limit to the amount of space used for the flash recovery area. The underlying file system may have less or more than this amount

of space; the DBA should ensure that at least this amount of space is available for backups. Note that this is the amount of recovery space for this database only; if multiple databases share the same ASM disk group for their flash recovery area, the sum of all values for DB\_RECOVERY\_FILE\_DEST\_SIZE must not exceed the available space in the disk group.

In our data warehouse database, **dw**, a flash recovery area is defined in the disk group **+RECOV** with a maximum size of 8GB. As this limit is reached, RMAN will automatically remove obsolete backups and generate an alert in the alert log when the amount of space occupied by nonobsolete backups is within 10 percent of the value specified in DB\_RECOVERY\_FILE\_DEST\_SIZE.

The parameters DB\_RECOVERY\_FILE\_DEST and DB\_RECOVERY\_FILE\_DEST\_SIZE are both dynamic; they can be changed on the fly while the instance is running to respond to changes in disk space availability.

## Data Dictionary and Dynamic Performance Views

A number of Oracle data dictionary and dynamic performance views contain information specific to RMAN operations, on both the target database and the catalog database. In Table 12-2 are the key views related to RMAN. Each of these views will be covered in more detail later in this chapter.

---

View	Description
RC_*	RMAN Recovery Catalog views. Only exist in the RMAN repository database and contain recovery information for all target databases.
V\$RMAN_STATUS	Displays finished and in-progress RMAN jobs.
V\$RMAN_OUTPUT	Contains messages generated by RMAN sessions and each RMAN command executed within the session.
V\$SESSION_LONGOPS	Contains the status of long-running administrative operations that run for more than six seconds; includes statistics gathering and long-running queries, in addition to RMAN recovery and backup operations.
V\$DATABASE_BLOCK_CORRUPTION	Corrupted blocks detected during an RMAN session.
V\$FLASH_RECOVERY_AREA_USAGE	The percentage of space used, by object type, in the flash recovery area.
V\$RECOVERY_FILE_DEST	The number of files, space used, space that can be reclaimed, and space limit for the flash recovery area.
V\$RMAN_CONFIGURATION	RMAN configuration parameters with non-default values for this database.

---

**TABLE 12-2** RMAN Data Dictionary and Dynamic Performance Views

The RC\_\* views only exist in a database that is used as an RMAN repository; the V\$ views exist and have rows in any database that is backed up using RMAN. To highlight this difference, we'll look at the view V\$RMAN\_CONFIGURATION in the target database:

```
SQL> connect rjb/rjb@dw
Connected.
SQL> select * from v$rman_configuration;
```

CONF#	NAME	VALUE
1	RETENTION POLICY	TO RECOVERY WINDOW OF 4 DAYS
2	CONTROLFILE AUTOBACKUP	ON
3	DEVICE TYPE	DISK BACKUP TYPE TO COMPRESSED BACKUPSET PARALLELISM 1

SQL>

Note that these are the three RMAN persistent parameters that we changed earlier. The recovery catalog database keeps these non-default values in the view RC\_RMAN\_CONFIGURATION for all databases registered with RMAN:

```
SQL> connect rman/rman@rac
Connected.
SQL> select db_key, db_unique_name, name, value
2      from rman.rc_rman_configuration;
```

DB_KEY	DB_UNIQUE_NAME	NAME	VALUE
1	dw	CONTROLFILE AUTOBACKUP	ON
1		RETENTION POLICY	TO RECOVERY WINDOW O F 4 DAYS
1	dw	DEVICE TYPE	DISK BACKUP TYPE TO COMPRESSED BACKUPSET PARALLELISM 1

3 rows selected.

If we were using RMAN to back up another database, this view would contain other values for DB\_KEY and DB\_UNIQUE\_NAME for other target databases with non-default RMAN parameters.

Because we are not using RMAN to back up the rac database, the views V\$RMAN\_\* are empty.

## Backup Operations

In this section, we'll run through some examples to back up the target database in a variety of ways: We'll perform two kinds of full backups, create image copies of selected database files, investigate how incremental backups work, and delve further into backup compression, incremental backup optimization, and the flash recovery area.

We'll continue to use our data warehouse database, **dw**, as the target database, with the database **rac** as the RMAN repository.

## Full Database Backups

In our first example of a full database backup, we'll use backupsets to copy all database files, including the SPFILE, to the flash recovery area:

```

RMAN> backup as backupset database spfile;

```

```

Starting backup at 29-AUG-07
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=111 device type=DISK
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00002 name=+DATA/dw/datafile/sysaux.257.630244581
input datafile file number=00001 name=+DATA/dw/datafile/system.256.630244579
input datafile file number=00006
      name=+DATA/dw/datafile/users_crypt.267.630456963
input datafile file number=00005
      name=+DATA/dw/datafile/example.265.630244801
input datafile file number=00003      name=+DATA/dw/datafile/undotbs1.258.630244583
input datafile file number=00004 name=+DATA/dw/datafile/users.259.630244583
channel ORA_DISK_1: starting piece 1 at 29-AUG-07
channel ORA_DISK_1: finished piece 1 at 29-AUG-07
piece handle=+RECOV/dw/backupset/2007_08_29/
      nnndf0_tag20070829t181238_0.292.63190
      775 tag=TAG20070829T181238 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:02:26
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
including current SPFILE in backup set
channel ORA_DISK_1: starting piece 1 at 29-AUG-07
channel ORA_DISK_1: finished piece 1 at 29-AUG-07
piece handle=+RECOV/dw/backupset/2007_08_29/
      nnsnf0_tag20070829t181238_0.293.631908931 tag=TAG20070829T181238
      comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:02
Finished backup at 29-AUG-07

Starting Control File and SPFILE Autobackup at 29-AUG-07
piece handle=+RECOV/dw/autobackup/2007_08_29/
      s_631908935.294.631908947 comment=NONE
Finished Control File and SPFILE Autobackup at 29-AUG-07

RMAN> sql 'alter system archive log current';
sql statement: alter system archive log current
RMAN>

```

The **alter system** command ensures that we have archived logs for all transactions, including those that occurred while the backup was taking place; this ensures that we can perform media recovery after restoring this backup.

Note that the SPFILE is backed up twice, the second time along with the control file. Because we set **configure controlfile autobackup to on**, we automatically back up the control file and SPFILE whenever we do any other kind of backup or the structure of the database changes. As a result, we don't need to specify SPFILE in the **backup** command.

Taking a peek into the flash recovery area using the **asmcmd** tool, we see a lot of cryptic filenames for the recent archived redo logs and the full database backup we just performed:

```
SQL> connect / as sysdba
```

```
Connected.
```

```
SQL> show parameter db_recovery
```

NAME	TYPE	VALUE
db_recovery_file_dest	string	+RECOV
db_recovery_file_dest_size	big integer	8G

```
SQL> select name from v$database;
```

```
NAME
```

```
-----  
DW
```

```
SQL> exit
```

```
[oracle@dw ~]$ asmcmd
```

```
ASMCMDB> ls
```

```
DATA/
```

```
RECOV/
```

```
ASMCMDB> cd recov/dw
```

```
ASMCMDB> ls
```

```
ARCHIVELOG/
```

```
AUTOBACKUP/
```

```
BACKUPSET/
```

```
CONTROLFILE/
```

```
ONLINELOG/
```

```
ASMCMDB> ls -l backupset
```

Type	Redund	Striped	Time	Sys	Name
				Y	2007_08_25/
				Y	2007_08_29/

```
ASMCMDB> ls -l backupset/2007_08_29
```

Type	Redund	Striped	Time	Sys	Name
BACKUPSET	MIRROR	COARSE	AUG 29 18:00:00	Y	
nnndf0_TAG20070829T181238_0.292.631908775					
BACKUPSET	MIRROR	COARSE	AUG 29 18:00:00	Y	
nnsnf0_TAG20070829T181238_0.293.631908931					

```
ASMCMDB> ls -l archivelog
```

Type	Redund	Striped	Time	Sys	Name
				Y	2007_08_25/
				Y	2007_08_26/
				Y	2007_08_27/
				Y	2007_08_28/
				Y	2007_08_29/

```
ASMCMDB> ls -l archivelog/2007_08_29
```

Type	Redund	Striped	Time	Sys	Name
ARCHIVELOG	MIRROR	COARSE	AUG 29 00:00:00	Y	
thread_1_seq_100.289.631843233					
ARCHIVELOG	MIRROR	COARSE	AUG 29 07:00:00	Y	
thread_1_seq_101.290.631869317					

```

ARCHIVELOG MIRROR COARSE AUG 29 14:00:00 Y
                                thread_1_seq_102.291.631893633
ARCHIVELOG MIRROR COARSE AUG 29 18:00:00 Y
                                thread_1_seq_103.295.631908977
ASMCMD> ls -l autobackup
Type Redund Striped Time Sys Name
Y 2007_08_29/
ASMCMD> ls -l autobackup/2007_08_29
Type Redund Striped Time Sys Name
AUTOBACKUP MIRROR COARSE AUG 29 18:00:00 Y s_631908935.294.631908947
ASMCMD>

```

As an alternative, you can use RMAN's **list** command to see these backups as they are cataloged in the target database control file and the RMAN repository. There are four backupsets, one for a previous full database backup, and three others: a more recent full backup containing the datafiles themselves, one for the explicit SPFILE backup, and one for the implicit SPFILE and control file backup.

```
RMAN> list backup by backup;
```

```
List of Backup Sets
```

```
=====
```

```

BS Key Type LV Size Device Type Elapsed Time Completion Time
-----
163 Full 1.25G DISK 00:03:57 25-AUG-07
BP Key: 165 Status: AVAILABLE Compressed: NO Tag:
TAG20070825T215501

```

```
Piece Name:
```

```
+RECOV/dw/backupset/2007_08_25/
nnndf0_tag20070825t215501_0.271.631576525
```

```
List of Datafiles in backup set 163
```

```

File LV Type Ckp SCN Ckp Time Name
-----
1 Full 2315404 25-AUG-07
+DATA/dw/datafile/system.256.630244579
2 Full 2315404 25-AUG-07
+DATA/dw/datafile/sysaux.257.630244581
3 Full 2315404 25-AUG-07
+DATA/dw/datafile/undotbs1.258.630244583
4 Full 2315404 25-AUG-07
+DATA/dw/datafile/users.259.630244583
5 Full 2315404 25-AUG-07
+DATA/dw/datafile/example.265.630244801
6 Full 2315404 25-AUG-07
+DATA/dw/datafile/users_crypt.267.630456963

```

```

BS Key Type LV Size Device Type Elapsed Time Completion Time
-----
164 Full 9.36M DISK 00:00:19 25-AUG-07
BP Key: 166 Status: AVAILABLE Compressed: NO Tag:
TAG20070825T215501

```

# 438 Oracle Database 11g DBA Handbook

```

Piece Name:
+RECOV/dw/backupset/2007_08_25/
ncsnf0_tag20070825t215501_0.272.631576759
  SPFILE Included: Modification time: 25-AUG-07
  SPFILE db_unique_name: DW
  Control File Included: Ckp SCN: 2315588      Ckp time: 25-AUG-07

```

```

BS Key  Type LV Size          Device Type Elapsed Time Completion Time
-----
229     Full   1.28G      DISK        00:02:33    29-AUG-07
        BP Key: 231   Status: AVAILABLE Compressed: NO Tag:
                                           TAG20070829T181238

```

```

Piece Name:
+RECOV/dw/backupset/2007_08_29/
nnndf0_tag20070829t181238_0.292.631908775
  List of Datafiles in backup set 229

```

```

File LV Type Ckp SCN      Ckp Time  Name
-----
1      Full 2741782  29-AUG-07
+DATA/dw/datafile/system.256.630244579
2      Full 2741782  29-AUG-07
+DATA/dw/datafile/sysaux.257.630244581
3      Full 2741782  29-AUG-07
+DATA/dw/datafile/undotbs1.258.630244583
4      Full 2741782  29-AUG-07
+DATA/dw/datafile/users.259.630244583
5      Full 2741782  29-AUG-07
+DATA/dw/datafile/example.265.630244801
6      Full 2741782  29-AUG-07
+DATA/dw/datafile/users_crypt.267.630456963

```

```

BS Key  Type LV Size          Device Type Elapsed Time Completion Time
-----
230     Full   80.00K     DISK        00:00:12    29-AUG-07
        BP Key: 232   Status: AVAILABLE Compressed: NO Tag:
                                           TAG20070829T181238

```

```

Piece Name: +RECOV/dw/backupset/2007_08_29/
nnsnf0_tag20070829t181238_0.293.631908931
  SPFILE Included: Modification time: 28-AUG-07
  SPFILE db_unique_name: DW

```

```

BS Key  Type LV Size          Device Type Elapsed Time Completion Time
-----
244     Full   9.48M     DISK        00:00:11    29-AUG-07
        BP Key: 246   Status: AVAILABLE Compressed: NO Tag:
                                           TAG20070829T181535

```

```

Piece Name:
+RECOV/dw/autobackup/2007_08_29/
s_631908935.294.631908947
  SPFILE Included: Modification time: 28-AUG-07

```

```
SPFILE db_unique_name: DW
Control File Included: Ckp SCN: 2741907      Ckp time: 29-AUG-07
```

RMAN>

One of the full backups can be used in conjunction with the archived redo logs (stored by default in the flash recovery area residing in the ASM disk group +RECOV) to recover the database to any point in time up to the last committed transaction.

Figure 12-5 shows a whole database backup configured to run using OEM. Notice that you can view, copy, or edit the RMAN script that OEM generates.

Displaying the contents of the catalog is just as easy in OEM. Figure 12-6 shows results equivalent to the **list backup by backup** command.

The **list** and **report** commands are covered in more detail later in this chapter.

## Tablespace

After adding a tablespace to the database, performing an immediate backup of the tablespace will shorten the time it will take to restore the tablespace later in the event of media failure. In addition, you might back up an individual tablespace in a database that is too large to back up all at once; again, creating a backupset or image copy of a tablespace at frequent intervals will reduce the amount of redo that would need to be applied to an older backup of the tablespace in the event

ORACLE Enterprise Manager 11g Database Control Help Logout Database

Options Settings Schedule **Review**

---

**Schedule Customized Backup: Review**

Database: **dw.world** [Cancel] [Edit RMAN Script] [Back] Step 4 of 4 [Submit Job]

Backup Strategy: **Customized Backup**

Object Type: **Whole Database**

Recovery Catalog Username: **rman**

Recovery Catalog Database: **oc1:1521:rac1**

---

**Settings**

Destination: **Disk**

Backup Type: **Full Backup**

Backup Mode: **Online Backup**

Flash Recovery Area: **+RECOV**

---

**RMAN Script**

The RMAN script below is generated based on the user input from previous pages.

```
backup device type disk tag '%TAG' database;
backup device type disk tag '%TAG' archivelog all not backed up;
```

[Return to Schedule Backup] [Cancel] [Edit RMAN Script] [Back] Step 4 of 4 [Submit Job]

FIGURE 12-5 Configure backup job with OEM

ORACLE® Enterprise Manager 11g Database Control

Database Instance: dw.world > Help Logout Database

Manage Current Backups Logged in As RJB

This backup data was retrieved from the recovery catalog.

Buttons: Catalog Additional Files, Crosscheck All, Delete All Obsolete, Delete All Expired

Search

Status: Available

Contents:  Datafile  Archived Redo Log  SPFILE  Control File

Completion Time: Within a month Go

Results

Buttons: Crosscheck, Change to Unavailable, Delete, Validate

Select All | Select None

Select	Key Tag	Completion Time	Contents	Device Type	Status	Keep	Pieces
<input type="checkbox"/>	244 TAG20070829T181535	Aug 29, 2007 6:15:46 PM	CONTROLFILE, SPFILE	DISK	AVAILABLE	NO	1
<input type="checkbox"/>	230 TAG20070829T181238	Aug 29, 2007 6:15:31 PM	SPFILE	DISK	AVAILABLE	NO	1
<input type="checkbox"/>	229 TAG20070829T181238	Aug 29, 2007 6:15:12 PM	DATAFILE	DISK	AVAILABLE	NO	1
<input type="checkbox"/>	164 TAG20070825T215501	Aug 25, 2007 9:59:20 PM	CONTROLFILE, SPFILE	DISK	AVAILABLE	NO	1
<input type="checkbox"/>	163 TAG20070825T215501	Aug 25, 2007 9:58:59 PM	DATAFILE	DISK	AVAILABLE	NO	1

FIGURE 12-6 Display backupset information with OEM

of media failure. For example, in an environment with three large tablespaces—USERS, USERS2, and USERS3—along with the default tablespaces SYSTEM and SYSAUX, you might back up the SYSTEM and SYSAUX tablespaces on Sunday, USERS on Monday, USERS2 on Wednesday, and USERS3 on Friday. Failures of any media containing datafiles from one of these tablespaces will use a tablespace backup that is no more than a week old plus the intervening archived and online redo log files for recovery.

In our next example, we’re adding a tablespace to the **dw** database to support a new set of star schemas:

```
SQL> create tablespace inet_star
2 datafile '+DATA' size 100m
3 autoextend on next 50m maxsize 500m;
Tablespace created.
```

From an RMAN session, we will back up the tablespace along with the control file. In this case, it’s critical that we back up the control file because it contains the definition for the new tablespace.

```
RMAN> report schema;
```

```
starting full resync of recovery catalog
full resync complete
Report of database schema for database with db_unique_name DW
```

## List of Permanent Datafiles

```

=====
File Size(MB) Tablespace          RB segs Datafile Name
-----
1    750      SYSTEM                YES      +DATA/dw/datafile/system.256.630244579
2    803      SYSAUX                  NO       +DATA/dw/datafile/sysaux.257.630244581
3    60       UNDOTBS1                YES      +DATA/dw/datafile/undotbs1.258.630244583
4    5        USERS                   NO       +DATA/dw/datafile/users.259.630244583
5    100     EXAMPLE                 NO       +DATA/dw/datafile/example.265.630244801
6    500     USERS_CRYPT             NO       +DATA/dw/datafile/users_crypt.267.630456963
7    100     INET_STAR               NO       +DATA/dw/datafile/inet_star.268.632004213

```

## List of Temporary Files

```

=====
File Size(MB) Tablespace          Maxsize(MB) Tempfile Name
-----
1    60       TEMP                    32767    +DATA/dw/tempfile/temp.264.630244787

```

RMAN> **backup as backupset tablespace inet\_star;**

```

Starting backup at 30-AUG-07
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=152 device type=DISK
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00007
name=+DATA/dw/datafile/inet_star.268.632004213
channel ORA_DISK_1: starting piece 1 at 30-AUG-07
channel ORA_DISK_1: finished piece 1 at 30-AUG-07
piece handle=+RECOV/dw/backupset/2007_08_30/
nnndf0_tag20070830t204700_0.302.63200435
tag=TAG20070830T204700 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:03
Finished backup at 30-AUG-07

```

```

Starting Control File and SPFILE Autobackup at 30-AUG-07
piece handle=+RECOV/dw/autobackup/2007_08_30/s_632004440.303.632004453
comment=NONE
Finished Control File and SPFILE Autobackup at 30-AUG-07

```

RMAN>

In Figure 12-7, you can see the new RMAN backup records in the repository—one for the tablespace (recorded as a single datafile backupset) and one for the controlfile/SPFILE autobackup.

ORACLE Enterprise Manager 11g Database Control

Database Instance: dw.world > Logged in As RJB

Manage Current Backups

Catalog Additional Files Crosscheck All Delete All Obsolete Delete All Expired

This backup data was retrieved from the recovery catalog.

Backup Sets Image Copies

Search

Status: Available

Contents:  Datafile  Archived Redo Log  SPFILE  Control File

Completion Time: Within a month Go

Results

Crosscheck Change to Unavailable Delete Validate

Select All | Select None

Select	Key	Tag	Completion Time	Contents	Device Type	Status	Keep	Pieces
<input type="checkbox"/>	310	TAG20070830T204720	Aug 30, 2007 8:47:34 PM	CONTROLFILE, SPFILE	DISK	AVAILABLE	NO	1
<input type="checkbox"/>	300	TAG20070830T204700	Aug 30, 2007 8:47:15 PM	DATAFILE	DISK	AVAILABLE	NO	1
<input type="checkbox"/>	289	TAG20070830T204449	Aug 30, 2007 8:45:03 PM	CONTROLFILE, SPFILE	DISK	AVAILABLE	NO	1
<input type="checkbox"/>	244	TAG20070829T181535	Aug 29, 2007 6:15:46 PM	SPFILE, CONTROLFILE	DISK	AVAILABLE	NO	1
<input type="checkbox"/>	230	TAG20070829T181238	Aug 29, 2007 6:15:31 PM	SPFILE	DISK	AVAILABLE	NO	1

FIGURE 12-7 Tablespace backup files in OEM

## Datafiles

Individual datafiles can be backed up as easily as we can back up a tablespace. If it's impractical to back up an entire tablespace within an RMAN session, you can back up individual datafiles within a tablespace over a period of days, and the archived redo log files will take care of the rest during a recovery operation. Here is an example of a datafile backup of a single datafile within a non-ASM tablespace:

```
RMAN> backup as backupset datafile
2>      '/u04/oradata/ord/oe_trans_06.dbf';
```

## Image Copies

Up until this point, we have been using backupset backups; in contrast, image copies make bit-for-bit copies of the specified tablespace or entire database. There are a couple of distinct advantages for using RMAN to perform image copy backups: First, the backup is automatically recorded in the RMAN repository. Second, all blocks are checked for corruption as they are read and copied to the backup destination. Another side benefit to making image copies is that the copies can be used "as is" outside of RMAN if, for some reason, a recovery operation must occur outside of RMAN.

In the example that follows, we make another backup of the INET\_STAR tablespace, this time as an image copy:

```

RMAN> backup as copy tablespace inet_star;
Starting backup at 30-AUG-07
using channel ORA_DISK_1
channel ORA_DISK_1: starting datafile copy
input datafile file number=00007
name=+DATA/dw/datafile/inet_star.268.632004213
output file name=+RECOV/dw/datafile/inet_star.304.632005047
tag=TAG20070830T205713 RECID=2 STAMP=632005118
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:01:14
Finished backup at 30-AUG-07

Starting Control File and SPFILE Autobackup at 30-AUG-07
piece handle=+RECOV/dw/autobackup/2007_08_30/
s_632005123.305.632005135 comment=NONE
Finished Control File and SPFILE Autobackup at 30-AUG-07

RMAN>

```

Image copies can only be created on DISK device types. In Figure 12-8, we perform the same backup with OEM.

Because we had earlier configured the default backup type to **compressed backupset**, we overrode the default value in an earlier setup screen for this backup.

The screenshot displays the Oracle Enterprise Manager 11g Database Control interface. At the top, there are navigation tabs for 'Tablespaces', 'Options', 'Settings', 'Schedule', and 'Review', with 'Review' being the active tab. Below the navigation, the main content area is titled 'Schedule Customized Backup: Review'. It shows the following configuration:

- Database: **dw.world**
- Backup Strategy: **Customized Backup**
- Object Type: **Tablespaces**
- Recovery Catalog Username: **rman**
- Recovery Catalog Database: **oc1:1521:rac1**

Below this, the 'Settings' section is expanded, showing:

- Destination: **Disk**
- Backup Type: **Full Backup**
- Backup Mode: **Online Backup**
- Flash Recovery Area: **+RECOV**

The 'Tablespaces' section shows a list containing 'INET\_STAR'. At the bottom, the 'RMAN Script' section displays the generated RMAN script:

```

run {
allocate channel oem_backup_disk1 type disk ;
backup as COPY tag '%TAG' tablespace 'INET_STAR' ;
backup as COPY archivelog all not backed up;
release channel oem_backup_disk1;
}

```

FIGURE 12-8 Image copy backup of a tablespace using OEM

## Control File, SPFILE Backup

To back up the control file and SPFILE manually, use the following RMAN command:

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

```
Starting backup at 30-AUG-07
starting full resync of recovery catalog
full resync complete
using channel ORA_DISK_1
channel ORA_DISK_1: starting compressed full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
including current control file in backup set
channel ORA_DISK_1: starting piece 1 at 30-AUG-07
channel ORA_DISK_1: finished piece 1 at 30-AUG-07
piece handle=+RECOV/dw/backupset/2007_08_30/
ncnnf0_tag20070830t220903_0.311.632009363 tag=TAG20070830T220903
comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:07
channel ORA_DISK_1: starting compressed full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
including current SPFILE in backup set
channel ORA_DISK_1: starting piece 1 at 30-AUG-07
channel ORA_DISK_1: finished piece 1 at 30-AUG-07
piece handle=+RECOV/dw/backupset/2007_08_30/
nnsnf0_tag20070830t220903_0.312.632009383 tag=TAG20070830T220903
comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 30-AUG-07
```

```
Starting Control File and SPFILE Autobackup at 30-AUG-07
piece handle=+RECOV/dw/autobackup/2007_08_30/
s_632009385.313.632009397 comment=NONE
Finished Control File and SPFILE Autobackup at 30-AUG-07
```

```
RMAN>
```

Note that because we already had **autobackup** set to **on**, we actually performed two backups of the control file and the SPFILE. The second backup of the control file, however, has a record of the first control file and SPFILE backup.

## Archived Redo Logs

Even when archived redo logs are sent to multiple destinations, including the flash recovery area, due to the critical nature of the archived redo logs, we want to back up copies of the logs to tape or another disk destination. Once the backup is completed, we have the option to leave the logs in place, to delete only the logs that RMAN used for the backup, or to delete all copies of the archived logs that were backed up to tape.

In the following example, we back up all the archived log files in the flash recovery area and then remove them from disk:

```
RMAN> backup device type sbt archivelog all delete input;
```

If archived log files are being sent to multiple locations, then only one set of the archived redo log files are deleted. If we want all copies to be deleted, we use **delete all input** instead of **delete input**. As of Oracle Database 11g, corrupt or missing archived log files do not prevent a successful RMAN backup of the archived logs as in previous releases; as long as one of the archive log file destinations has a valid log file for a given log sequence number, the backup is successful.

Backing up and deleting only older archived redo log files can be accomplished by specifying a date range in the **backup archivelog** command:

```
RMAN> backup device type sbt
2>     archivelog from time 'sysdate-30' until time 'sysdate-7'
3>     delete all input;
```

In the preceding example, all archived redo logs older than one week, going back for three weeks, are copied to tape and deleted from disk. In addition, you can specify a range using SCNs or log sequence numbers.

## Incremental Backups

An alternative strategy to relying on full backups with archived redo logs is to use *incremental backups* along with archived redo logs for recovery. The initial incremental backup is known as a *level 0* incremental backup. Each incremental backup after the initial incremental backup (also known as a *level 1* incremental backup) contains only changed blocks and as a result takes less time and space. Incremental level 1 backups can either be *cumulative* or *differential*. A cumulative backup records all changed blocks since the initial incremental backup; a differential backup records all changed blocks since the last incremental backup, whether it was a level 0 or a level 1 incremental backup.

When a number of different types of backups exist in the catalog, such as image copies, tablespace backupsets, and incremental backups, RMAN will choose the best combination of backups to most efficiently recover and restore the database. The DBA still has the option to prevent RMAN from using a particular backup (for example, if the DBA thinks that a particular backup is corrupt and will be rejected by RMAN during the recovery operation).

The decision whether to use cumulative or differential backups is based partly on where you want to spend the CPU cycles, and how much disk space you have available. Using cumulative backups means that each incremental backup will become progressively larger and take longer until another level 0 incremental backup is performed, but during a restore and recover operation, only two backupsets will be required. On the other hand, differential backups only record the changes since the last backup, so each backupset might be smaller or larger than the previous one, with no overlap in data blocks backed up. However, a restore and recover operation may take longer if you have to restore from several backupsets instead of just two.

Following our example with the **dw** database, we find out that several files are outside of our retention policy of four days; in other words, files that need more than four days worth of archived redo logs to recover the database:

```
RMAN> report need backup;
```

```
RMAN retention policy will be applied to the command
RMAN retention policy is set to recovery window of 4 days
Report of files whose recovery needs more than 4 days of archived logs
```

File	Days	Name
1	5	+DATA/dw/datafile/system.256.630244579
2	5	+DATA/dw/datafile/sysaux.257.630244581
3	5	+DATA/dw/datafile/undotbs1.258.630244583
4	5	+DATA/dw/datafile/users.259.630244583
5	5	+DATA/dw/datafile/example.265.630244801
6	5	+DATA/dw/datafile/users_crypt.267.630456963
7	5	+DATA/dw/datafile/inet_star.268.632004213
8	5	+DATA/dw/datafile/inet_intl_star.269.632009933

RMAN>

To remedy this situation, we can do another full backup, or we can pursue an incremental backup policy, which might be easier to implement and maintain. To set up our incremental policy, we need to perform a level 0 incremental backup first:

```
RMAN> backup incremental level 0
```

```
2> as compressed backupset database;
```

```
Starting backup at 30-AUG-07
using channel ORA_DISK_1
channel ORA_DISK_1: starting compressed incremental level 0
datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00002 name=+DATA/dw/datafile/sysaux.257.630244581
input datafile file number=00001 name=+DATA/dw/datafile/system.256.630244579
input datafile file number=00006
name=+DATA/dw/datafile/users_crypt.267.630456963
input datafile file number=00005
name=+DATA/dw/datafile/example.265.630244801
input datafile file number=00007
name=+DATA/dw/datafile/inet_star.268.632004213
input datafile file number=00003
name=+DATA/dw/datafile/undotbs1.258.630244583
input datafile file number=00008
name=+DATA/dw/datafile/inet_intl_star.269.632009933
input datafile file number=00004
name=+DATA/dw/datafile/users.259.630244583
channel ORA_DISK_1: starting piece 1 at 30-AUG-07
channel ORA_DISK_1: finished piece 1 at 30-AUG-07
piece handle=+RECOV/dw/backupset/2007_08_30/
nnndn0_tag20070830t222903_0.315.632010557
tag=TAG20070830T222903 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:01:36
Finished backup at 30-AUG-07
```

```
Starting Control File and SPFILE Autobackup at 30-AUG-07
piece handle=+RECOV/dw/autobackup/2007_08_30/
s_632010654.316.632010665 comment=NONE
Finished Control File and SPFILE Autobackup at 30-AUG-07
```

RMAN>

At any point in the future after this level 0 backup, we can perform an incremental level 1 differential backup:

```
RMAN> backup as compressed backupset
2> incremental level 1 database;
```

The default incremental backup type is differential; the keyword **differential** is neither needed nor allowed. However, to perform a cumulative backup, we add the **cumulative** keyword:

```
RMAN> backup as compressed backupset
2> incremental level 1 cumulative database;
```

How much database activity is performed may also dictate whether you use cumulative or differential backups. In an OLTP environment with heavy insert and update activity, incremental backups may be more manageable in terms of disk space usage. For a data warehouse environment with infrequent changes, a differential backup policy may be more suitable. Compared to using redo log files, both types of incremental backups are far superior in terms of the time to recover a database. In any case, we have addressed RMAN's retention policy:

```
RMAN> report need backup;
```

```
starting full resync of recovery catalog
full resync complete
RMAN retention policy will be applied to the command
RMAN retention policy is set to recovery window of 4 days
Report of files whose recovery needs more than 4 days of archived logs
File Days Name
-----
```

```
RMAN>
```

## Incrementally Updated Backups

An *incrementally updated* backup can potentially make a recover and restore operation even more efficient by rolling the changes from a level 1 incremental backup to a level 0 incremental image backup. If the incrementally updated backup is run on a daily basis, then any recovery operation would require at most the updated image copy, one incremental level 1 backup, and the most recent archived and online redo logs. The following example uses an RMAN script that can be scheduled to run at the same time every day to support an incrementally updated backup strategy:

```
run
{
  recover copy of database with tag 'incr_upd_img';
  backup incremental level 1
    for recover of copy with tag 'incr_upd_img' database;
}
```

The key part of both commands within the **run** script is the **recover copy** clause. Rather than doing a recovery of the actual database datafiles, we are recovering a *copy* of a database datafile by applying incremental backups. Using a *tag* with an RMAN backup allows us to apply the incremental backup to the correct image copy. Tags allow DBAs to easily refer to a specific backup for recovery or catalog cleanup operations; if the **backup** command does not provide

a tag, one is automatically generated for the backupset and is unique within the backupsets for the target database.

The basics of standard recovery operations and RMAN scripting capabilities are covered later in this chapter.

The OEM backup wizards make it easy to automate an incrementally updated backup strategy. In the figures that follow, we'll cover the steps needed to configure this strategy within OEM.

In Figure 12-9, we're specifying the strategy for backing up our database.

The database is open, archive log mode is enabled, and backups will follow the Oracle-suggested guidelines for a backup strategy. The other option in the pull-down menu is Customized. Figure 12-10 shows the next step in the backup configuration process: a summary of the database name, the strategy selected, where the backups will be sent, the recovery catalog in use, and a brief explanation as to how the backup will be performed.

In Figure 12-11, we specify when the backups will start, and what time of day they will run. Although the backup job can run any time during the day, because we are performing a hot backup (the database is open and users can process transactions), we want to minimize the possible impact on query and DML response time by scheduling the job during a time period with low activity.

Figure 12-12 gives us one more chance to review how the backup will be performed and where it will reside.

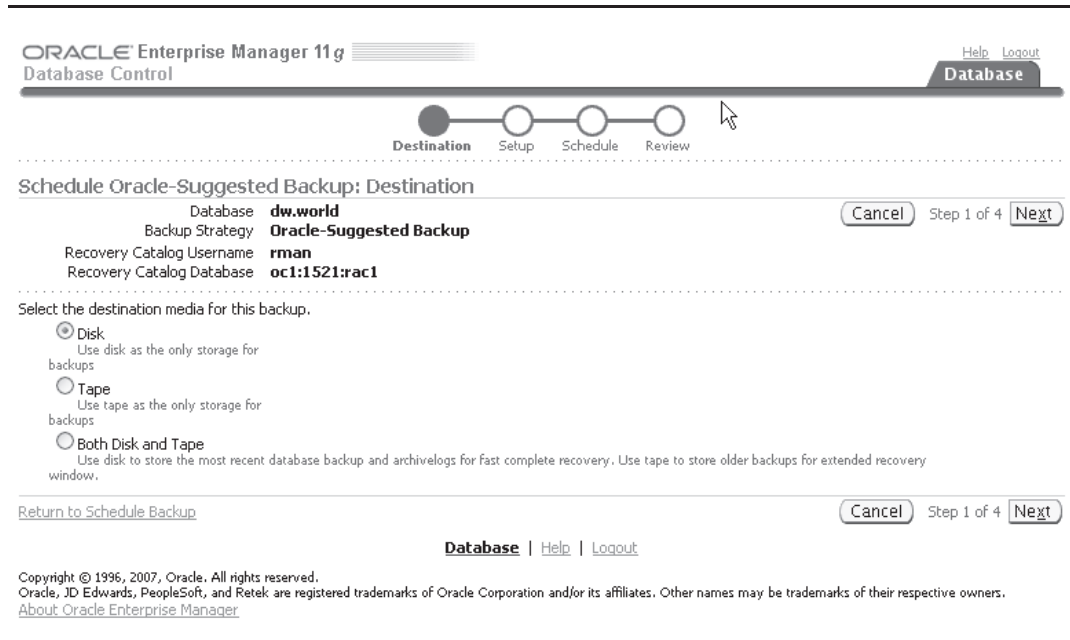


FIGURE 12-9 OEM backup strategy selection

ORACLE Enterprise Manager 11g Database Control

Help Logout Database

Destination Setup Schedule Review

Schedule Oracle-Suggested Backup: Setup

Database **dw.world** Cancel Back Step 2 of 4 Next

Backup Strategy **Oracle-Suggested Backup**

Recovery Catalog Username **rman**

Recovery Catalog Database **oc1:1521:rac1**

**Daily Backup**

A full database copy will be performed during the first backup. Subsequently, an incremental backup to disk will be performed everyday. The backups on disk will be retained so that you can always perform a full database recovery or a point-in-time recovery to any time within the past day.

**Disk Settings**

Flash Recovery Area **+RECOV**

TIP Disk backups that are necessary for a recovery to any time within the past day are retained.

**Encryption**

Return to Schedule Backup Cancel Back Step 2 of 4 Next

Database | Help | Logout

Copyright © 1996, 2007, Oracle. All rights reserved.  
Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.  
[About Oracle Enterprise Manager](#)

FIGURE 12-10 OEM backup setup summary

ORACLE Enterprise Manager 11g Database Control

Help Logout Database

Destination Setup Schedule Review

Schedule Oracle-Suggested Backup: Schedule

Database **dw.world** Cancel Back Step 3 of 4 Next

Backup Strategy **Oracle-Suggested Backup**

Recovery Catalog Username **rman**

Recovery Catalog Database **oc1:1521:rac1**

**Daily Backup Time**

Specify a date to start the backup. The first backup could be time consuming as it is a whole database backup. Consider starting the backup when the database is least active.

Start Date Sep 3, 2007 (example: Sep 3, 2007)

Specify a time to start the backup. Consider starting the backup when the database is least active during the day.

Time Zone (UTC-06:00) Canada Central Time (CST)

Daily Backup Time 2:00 AM

Return to Schedule Backup Cancel Back Step 3 of 4 Next

Database | Help | Logout

Copyright © 1996, 2007, Oracle. All rights reserved.  
Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.  
[About Oracle Enterprise Manager](#)

FIGURE 12-11 OEM backup schedule

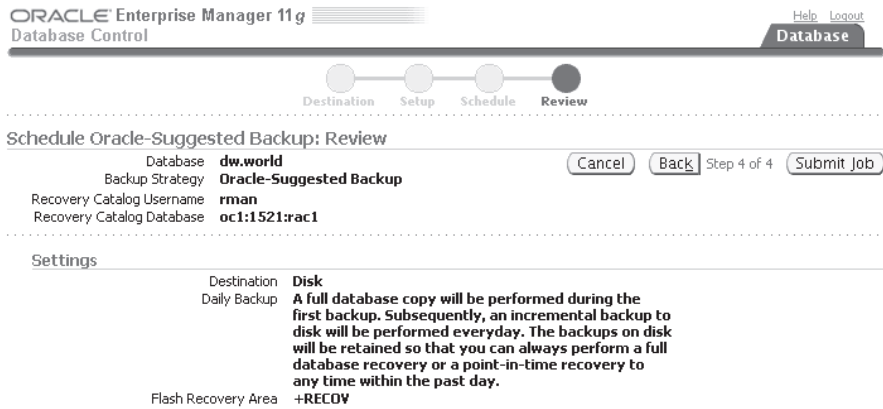


FIGURE 12-12 OEM backup summary

At the bottom of the browser window is the actual RMAN script that will be scheduled to run on a daily basis (see Figure 12-13). Coincidentally, it strongly resembles the RMAN script we presented earlier in this section.

## Incremental Backup Block Change Tracking

Another way to improve the performance of incremental backups is to enable *block change tracking*. For a traditional incremental backup, RMAN must inspect every block of the tablespace or datafile to be backed up to see if the block has changed since the last backup. For a very large database, the time it takes to scan the blocks in the database can easily exceed the time it takes to perform the actual backup.

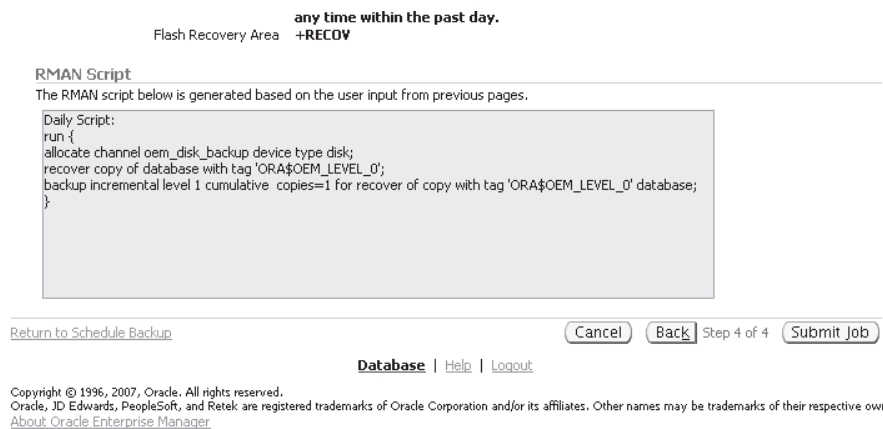


FIGURE 12-13 OEM backup script

By enabling block change tracking, RMAN knows which blocks have changed by using a *change tracking file*. Although there is some slight overhead in space usage and maintenance of the tracking file every time a block is changed, the tradeoff is well worth it if frequent incremental backups are performed on the database. In the following example, we create a block change tracking file in the DATA disk group and enable block change tracking:

```
SQL> alter database enable block change tracking
      2      using file '+DATA';
```

Database altered.

```
SQL> exit
[oracle@dw ~]$ asmcmd
ASMCMD> cd data/dw
ASMCMD> ls
CHANGETRACKING/
CONTROLFILE/
DATAFILE/
ONLINELOG/
PARAMETERFILE/
TEMPFILE/
spfiledw.ora
ASMCMD> cd changetracking
ASMCMD> ls -s
Block_Size  Blocks      Bytes      Space  Name
         512    22657    11600384   25165824  ctf.270.632356105
ASMCMD>
```

The next time a backup is performed, RMAN will only have to use the contents of the file ctf.270.632356105 (an OMF-named file in the DW/CHANGETRACKING directory of the DATA disk group) to determine which blocks need to be backed up. The space needed for the block change tracking file is approximately 1/250,000 the size of the database.

The dynamic performance view V\$BLOCK\_CHANGE\_TRACKING contains the name and size of the block change tracking file as well as whether change tracking is enabled:

```
SQL> select filename, status, bytes from v$block_change_tracking;
```

FILENAME	STATUS	BYTES
+DATA/dw/changetracking/ctf.270.632356105	ENABLED	11599872

SQL>

## Backup Compression

As you learned earlier in this chapter, we can either configure backup compression as the default for backupsets or explicitly specify compression in an RMAN **backup** command (for the purposes of this example, the backup was performed on September 3, 2007):

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

Comparing the size of the actual datafiles with the backupset, we can see just how much compression we can achieve for some additional CPU overhead:

```

ASMCMDB> cd +DATA
ASMCMDB> du DW
Used_MB      Mirror_used_MB
   2675             5361
ASMCMDB> cd +RECOV/dw/backupset
ASMCMDB> ls
2007_08_25/
2007_08_29/
2007_08_30/
2007_09_03/
ASMCMDB> du 2007_09_03
Used_MB      Mirror_used_MB
   241             483
ASMCMDB>

```

The database files occupy about 2.7GB in the +DATA disk group; the compressed backupset from RMAN comes in at 241MB, which is well over a 90 percent compression rate.

## Using a Flash Recovery Area

Earlier in this chapter, we covered the initialization parameters required to set up the flash recovery area: `DB_RECOVERY_FILE_DEST` and `DB_RECOVERY_FILE_DEST_SIZE`. Both of these parameters are dynamic, allowing the DBA to change either the RMAN destination for backups or the amount of space allowed for backups in the flash recovery area without restarting the instance.

To facilitate a completely disk-based recovery scenario, the flash recovery area should be big enough for a copy of all datafiles, incremental backup files, online redo logs, archived redo logs not on tape, control file autobackups, and SPFILE backups. Using a larger or smaller recovery window or adjusting the redundancy policy will require an adjustment in the size of the flash recovery area. If the flash recovery area is limited in size due to disk space constraints, at a minimum there should be enough room to hold the archived log files that have not yet been copied to tape. The dynamic performance view `V$RECOVERY_FILE_DEST` displays information about the number of files in the flash recovery area, how much space is currently being used, and the total amount of space available in the flash recovery area.

The flash recovery area automatically uses OMF. As part of Oracle 11g's simplified management structure, you do not need to explicitly set any of the `LOG_ARCHIVE_DEST_n` initialization parameters if you only need one location for archived redo log files; if the database is in ARCHIVELOG mode, and a flash recovery area is defined, then the initialization parameter `LOG_ARCHIVE_DEST_10` is implicitly defined as the flash recovery area.

As you have seen in many previous examples, RMAN uses the flash recovery area in a very organized fashion—with separate directories for archived logs, backupsets, image copies, block change tracking files, and automatic backups of the control file and SPFILE. In addition, each subdirectory is further subdivided by a timestamp, making it easy to find a backupset or image copy when the need arises.

Multiple databases can share the same flash recovery area, even a primary and a standby database. Even with the same `DB_NAME`, as long as the `DB_UNIQUE_NAME` parameter is different, there will not be any conflicts. RMAN uses the `DB_UNIQUE_NAME` to distinguish backups between databases that use the same flash recovery area.

## Validating Backups

Having multiple image backups or enough archived redo log files to support a recovery window is of less value if there are problems with the live database files or control files. The RMAN command **backup validate database** will simulate a backup, checking for the existence of the specified files, ensuring that they are not corrupted. No backup files are created. This command would be useful in a scenario where you can check for problems with the database or archived redo logs proactively, giving you an opportunity to fix problems before the actual backup operation or for scheduling additional time overnight to repair problems found during the day.

In the following example, we will validate the entire database along with the archived redo logs after one of the redo log files is accidentally lost:

```

ASMCMD> cd 2007_08_27
ASMCMD> ls
thread_1_seq_91.280.631681211
thread_1_seq_92.281.631708219
thread_1_seq_93.282.631730443
thread_1_seq_94.283.631749691
thread_1_seq_95.284.631750213
ASMCMD> rm *95.*
You may delete multiple files and/or directories.
Are you sure? (y/n) y
ASMCMD>
. . .
RMAN> backup validate database archivelog all;

Starting backup at 04-SEP-07
using channel ORA_DISK_1
archived log +RECOV/dw/archivelog/2007_08_27/
  thread_1_seq_95.284.631750213
  not found or out of sync with catalog
trying alternate file for archived log of thread 1 with sequence 95
channel ORA_DISK_1: starting compressed archived log backup set
channel ORA_DISK_1: specifying archived log(s) in backup set
input archived log thread=1 sequence=77 RECID=1 STAMP=631556992
. . .
List of Control File and SPFILE
=====
File Type      Status Blocks Failing Blocks Examined
-----
SPFILE         OK          0             2
Control File OK          0             602
Finished backup at 04-SEP-07
RMAN>

```

The **backup validate** command has identified an archived redo log file that is no longer in the flash recovery area. It may have been archived to tape outside of RMAN, or it may have been inadvertently deleted (in this case, it appears that the log file was intentionally deleted). Looking at the datestamp of the log file, we see that it is outside of our recovery window of four days, so it is not a critical file in terms of recoverability.

Synchronizing the flash recovery area and the catalog with the **crosscheck** command is covered later in this chapter; once we have fixed the cross-reference problem we have just discovered, we can perform the rest of the validation:

```
RMAN> backup validate database archivelog all;
```

```
Starting backup at 04-SEP-07
using channel ORA_DISK_1
channel ORA_DISK_1: starting compressed archived log backup set
channel ORA_DISK_1: specifying archived log(s) in backup set
input archived log thread=1 sequence=77 RECID=1 STAMP=631556992
input archived log thread=1 sequence=78 RECID=3 STAMP=631556992
input archived log thread=1 sequence=79 RECID=5 STAMP=631556995
input archived log thread=1 sequence=80 RECID=7 STAMP=631556998
input archived log thread=1 sequence=81 RECID=9 STAMP=631557001
input archived log thread=1 sequence=82 RECID=11 STAMP=631557001
input archived log thread=1 sequence=83 RECID=13 STAMP=631566057
input archived log thread=1 sequence=84 RECID=16 STAMP=631578707
input archived log thread=1 sequence=85 RECID=18 STAMP=631590167
input archived log thread=1 sequence=86 RECID=20 STAMP=631607160
input archived log thread=1 sequence=87 RECID=22 STAMP=631623660
input archived log thread=1 sequence=88 RECID=24 STAMP=631643444
. . .
channel ORA_DISK_1: starting compressed full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00002 name=+DATA/dw/datafile/sysaux.257.630244581
input datafile file number=00001 name=+DATA/dw/datafile/system.256.630244579
input datafile file number=00006
      name=+DATA/dw/datafile/users_crypt.267.630456963
input datafile file number=00005
      name=+DATA/dw/datafile/example.265.630244801
input datafile file number=00007
      name=+DATA/dw/datafile/inet_star.268.632004213
input datafile file number=00003
      name=+DATA/dw/datafile/undotbs1.258.630244583
input datafile file number=00008
      name=+DATA/dw/datafile/inet_intl_star.269.632009933
input datafile file number=00004 name=+DATA/dw/datafile/users.259.630244583
channel ORA_DISK_1: backup set complete, elapsed time: 00:01:05
. . .
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
List of Control File and SPFILE
=====
File Type      Status Blocks Failing Blocks Examined
-----
SPFILE         OK          0          2
Control File OK          0         602
Finished backup at 04-SEP-07
RMAN>
```

No errors were found during the validation; RMAN read every block of every archived redo log file and datafile to ensure that they were readable and had no corrupted blocks. However, no backups were actually written to a disk or tape channel.

## Recovery Operations

Every good backup plan includes a disaster recovery plan so that we can retrieve the datafiles and logs from the backups and recover the database files. In this section, we'll review several different aspects of RMAN recovery operations.

RMAN can perform restore and recovery operations at various levels of granularity, and most of these operations can be performed while the database is open and available to users. We can recover individual blocks, tablespaces, datafiles, or even an entire database. In addition, RMAN has various methods of validating a restore operation without performing an actual recovery on the database datafiles.

## Block Media Recovery

When there are only a small handful of blocks to recover in a database, RMAN can perform *block media recovery* rather than a full datafile recovery. Block media recovery minimizes redo log application time, and it drastically reduces the amount of I/O required to recover only the block or blocks in question. While block media recovery is in progress, the affected datafiles can remain online and available to users.



### NOTE

*Block media recovery is only available from within the RMAN application.*

There are a number of ways in which block corruption is detected. During a read or write operation from an **insert** or **select** statement, Oracle may detect a block is corrupt, write an error in a user trace file, and abort the transaction. An RMAN **backup** or **backup validate** command can record corrupted blocks in the dynamic performance view V\$DATABASE\_BLOCK\_CORRUPTION. In addition, the SQL commands **analyze table** and **analyze index** could uncover corrupted blocks.

To recover one or more data blocks, RMAN must know the datafile number and block number within the datafile. This information is available in a user trace file, as in the following example:

```
ORA-01578: ORACLE data block corrupted (file # 6, block # 403)
ORA-01110: data file 6: '/u09/oradata/ord/oe_trans01.dbf'
```

Alternatively, the block may appear in the view V\$DATABASE\_BLOCK\_CORRUPTION after an RMAN **backup** command; the columns FILE# and BLOCK# provide the information needed to execute the **recover** command. The column CORRUPTION\_TYPE identifies the type of corruption in the block, such as FRACTURED, CHECKSUM, or CORRUPT. Fixing the block is easily accomplished in RMAN:

```
RMAN> recover datafile 6 block 403;
```

```
Starting recover at 04-SEP-07
using channel ORA_DISK_1
```

```
starting media recovery
```

```
media recovery complete, elapsed time: 00:00:01
```

```
Finished recover at 04-SEP-07
```

```
RMAN>
```

A corrupted block must be restored completely; in other words, all redo operations up to the latest SCN against the data block must be applied before the block can be considered usable again.

#### NOTE

The **blockrecover** command, available in previous releases of RMAN, has been deprecated in Oracle Database 11g in favor of the **recover** command; the syntax of the command is otherwise the same.

## Restoring a Control File

In the rare event that you lose all copies of your control file, it is easy to restore the control file when a recovery catalog is used; start the instance with **nomount** (since we don't have a control file to read with **mount**) and issue the following RMAN command:

```
RMAN> restore controlfile;
```

If you are not using a recovery catalog, you can add the **from '<filename>'** clause to the command to specify where the latest control file exists:

```
RMAN> restore controlfile from '/u11/oradata/ord/bkup.ctl';
```

After restoring the control files, you must perform complete media recovery of your database and open the database with the **resetlogs** option. Complete media recovery can be performed using RMAN or the methods described in Chapter 11.

## Restoring a Tablespace

If the disk containing the datafiles for a tablespace fails or becomes corrupted, recovery of the tablespace is possible while the database remains open and available. The exception to this is the SYSTEM tablespace. In our **dw** database, suppose the disk containing the datafiles for the USERS tablespace has crashed. After the first phone call from the users (which happened even before OEM notified us of the error), we can check the dynamic performance view V\$DATAFILE\_HEADER to see which datafiles need recovery:

```
SQL> select file#, status, error, tablespace_name, name
2         from v$datafile_header;
```

FILE#	STATUS	ERROR	TABLESPACE_NAME	NAME
1	ONLINE		SYSTEM	+DATA/dw/datafile/system. 256.630244579
2	ONLINE		SYSAUX	+DATA/dw/datafile/sysaux. 257.630244581
3	ONLINE		UNDOTBS1	+DATA/dw/datafile/undotbs 1.258.630244583
4	ONLINE	FILE NOT FOUND		
5	ONLINE		EXAMPLE	+DATA/dw/datafile/example

```

      .265.630244801
6 ONLINE          USERS_CRYPT      +DATA/dw/datafile/users_c
      ryp.267.630456963
7 ONLINE          INET_STAR        +DATA/dw/datafile/inet_st
      ar.268.632004213
8 ONLINE          INET_INTL_STAR   +DATA/dw/datafile/inet_in
      tl_star.269.632009933

```

8 rows selected.

Incidentally, the alert log (and the session where you performed the **startup** command) would give you another clue the next time you tried to start the database with a missing or corrupted datafile:

```

ORA-01157: cannot identify/lock data file 4 - see DBWR trace file
ORA-01110: data file 4: '+DATA/dw/datafile/users.259.630244583'

```

After replacing the disk drive, we initiate an RMAN session and find out that file number 4 corresponds to the USERS tablespace:

```
RMAN> report schema;
```

Report of database schema for database with db\_unique\_name DW

List of Permanent Datafiles

=====

File	Size (MB)	Tablespace	RB segs	Datafile Name
1	750	SYSTEM	YES	+DATA/dw/datafile/system.256.630244579
2	826	SYSAUX	NO	+DATA/dw/datafile/sysaux.257.630244581
3	60	UNDOTBS1	YES	+DATA/dw/datafile/undotbs1.258.630244583
4	5	USERS	NO	+DATA/dw/datafile/users.259.630244583
5	100	EXAMPLE	NO	+DATA/dw/datafile/example.265.630244801
6	500	USERS_CRYPT	NO	+DATA/dw/datafile/users_crypt.267.630456963
7	100	INET_STAR	NO	+DATA/dw/datafile/inet_star.268.632004213
8	50	INET_INTL_STAR	NO	+DATA/dw/datafile/inet_intl_star.269.632009933

List of Temporary Files

=====

File	Size (MB)	Tablespace	Maxsize (MB)	Tempfile Name
1	60	TEMP	32767	+DATA/dw/tempfile/temp.264.630244787

RMAN>

To restore and recover the tablespace, we force the tablespace offline, restore and recover the tablespace, and bring it back online:

```

RMAN> sql 'alter tablespace users offline immediate';
sql statement: alter tablespace users offline immediate

RMAN> restore tablespace users;

Starting restore at 04-SEP-07
using channel ORA_DISK_1

channel ORA_DISK_1: starting datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
channel ORA_DISK_1: restoring datafile 00004 to
                    +DATA/dw/datafile/users.259.630244583
channel ORA_DISK_1: reading from backup piece
+RECOV/dw/backupset/2007_09_04/
                    nnndf0_tag20070904t215119_0.266.632440295
channel ORA_DISK_1: piece handle=+RECOV/dw/backupset/2007_09_04/
                    nnndf0_tag20070904t215119_0.266.632440295
                    tag=TAG20070904T215119
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:03
Finished restore at 04-SEP-07
starting full resync of recovery catalog
full resync complete

RMAN> recover tablespace users;

Starting recover at 04-SEP-07
using channel ORA_DISK_1

starting media recovery
media recovery complete, elapsed time: 00:00:01
Finished recover at 04-SEP-07

RMAN> sql 'alter tablespace users online';

sql statement: alter tablespace users online

RMAN>

```

The **restore** command copied the latest image or backupset copy of the datafiles in the USERS tablespace to their original locations; the **recover** command applied redo from either redo log files or incremental backups to bring the objects in the tablespace back up to the latest SCN. Once the tablespace is back online, it is available for use again, without the loss of any committed transactions to tables in the tablespace.

## Restoring a Datafile

Restoring a datafile is a very similar operation to restoring a tablespace. Once the missing or corrupted datafile is identified using the V\$DATAFILE\_HEADER view, the RMAN commands are very similar to the previous example; the tablespace is taken offline, the datafile(s) are restored and recovered, and the tablespace is brought back online. If only file number 7 was lost, the **recover** and **restore** commands are as simple as this:

```
RMAN> restore datafile 7;
RMAN> recover datafile 7;
```

Using OEM, the procedure is also very straightforward. In Figure 12-14, we configure the datafile restore by selecting the missing datafile (#7) in the INET\_STAR tablespace.

In Figure 12-15, we have the option to restore the datafile to an alternate location; in this case, we want to restore it to the original location. If the failed disk drive containing the datafile could not be repaired, we would specify an alternate location for the datafile.

Before the RMAN job is submitted, we have one more chance to review the configuration of the job in Figure 12-16. Figure 12-17 shows the actual RMAN commands that will be run to perform the requested operation.

ORACLE Enterprise Manager 11g Database Control

Help Logout Database

Datafiles Rename Schedule Review

Perform Object Level Recovery: Datafiles

Database: **dw.world** (Cancel) Step 1 of 4 (Next)

Recovery Scope: **Datafiles**

Operation Type: **Restore and Recover**

Recovery Catalog Username: **rman**

Recovery Catalog Database: **oc1:1521:rac1**

Populate this table with the datafiles you want to recover. (Add)

Select	Datafile Name	Datafile Number	Status	Need Recovery	Error
<input type="checkbox"/>	+DATA/dw/datafile/inet_star.268.632004213	7	ONLINE	NO	

Return to Perform Recovery (Cancel) Step 1 of 4 (Next)

Database | Help | Logout

Copyright © 1996, 2007, Oracle. All rights reserved.  
 Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.  
 About Oracle Enterprise Manager

FIGURE 12-14 Selecting the datafile to restore

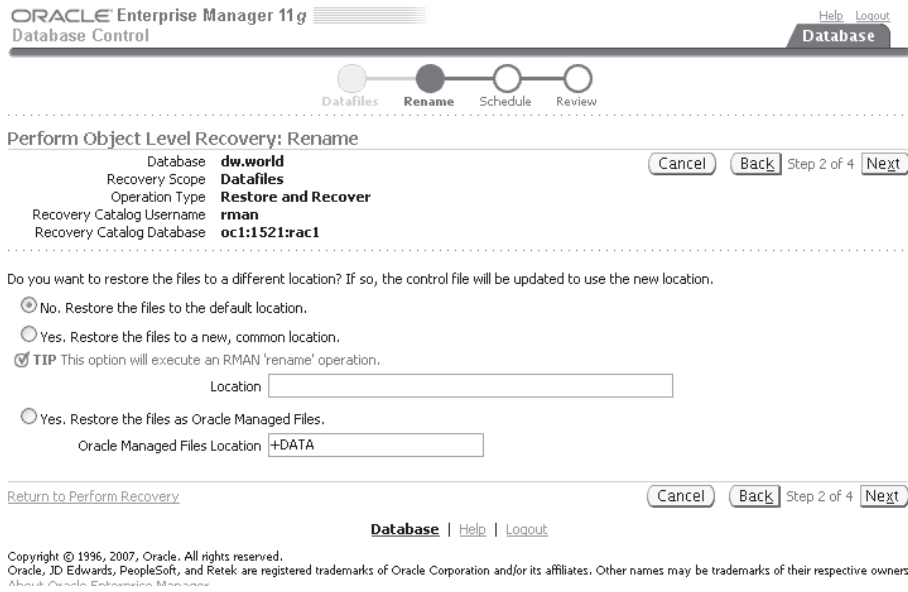


FIGURE 12-15 Specifying a location for the restored datafile

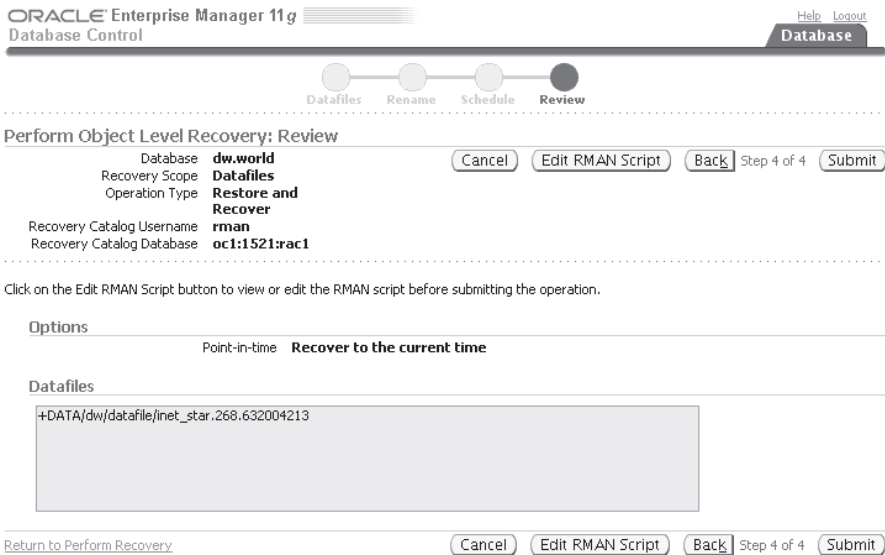


FIGURE 12-16 Review the restore options

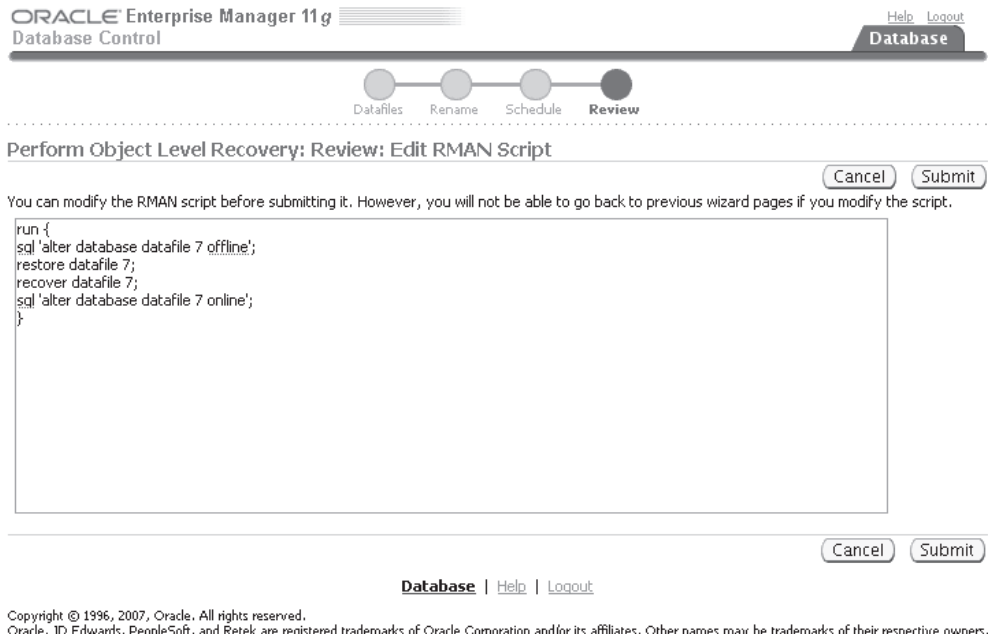


FIGURE 12-17 RMAN commands for the restore options

## Restoring an Entire Database

Although the loss of an entire database is a serious and disastrous event, having a solid backup and recovery policy, as we've described previously in this chapter, can bring the database back up to the most recent committed transaction with a minimum of effort. In the following scenario, we have lost all datafiles. However, because we have multiplexed the control file and online redo log files on many different disks, we will have them available during the RMAN restore and recovery operation. Alternatively, you can restore the control files or copy the online redo log files to the other destinations before mounting the database. If this is not feasible because the alternate disk locations are not available, you can alter your parameter file or SPFILE to indicate which files are still available.

The entire restore and recovery operation can be performed within RMAN; first, we start up RMAN and open the database in **mount** mode, just as if we used the **startup mount** command at a SQL\*Plus prompt:

```
[oracle@oltp oracle]$ rman target / catalog rman/rman@rac
```

```
Recovery Manager: Release 11.1.0.6.0 -
Production on Tue Sep 4 22:36:05 2007
```

```
Copyright (c) 1982, 2007, Oracle. All rights reserved.
```

```
connected to target database (not started)
```

```
recovery catalog database Password:
connected to recovery catalog database
```

```
RMAN> startup mount
```

```
Oracle instance started
database mounted
```

```
Total System Global Area      422670336 bytes
Fixed Size                     1300352 bytes
Variable Size                  360712320 bytes
Database Buffers               54525952 bytes
Redo Buffers                   6131712 bytes
starting full resync of recovery catalog
full resync complete
RMAN>
```

RMAN connects to the database and identifies it as being unavailable; this is the same as connecting to SQL\*Plus and seeing the message “Connected to an idle instance.” The next steps are to restore and recover the database:

```
RMAN> restore database;
```

```
Starting restore at 04-SEP-07
using channel ORA_DISK_1
```

```
channel ORA_DISK_1: starting datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
channel ORA_DISK_1: restoring datafile 00001 to
+DATA/dw/datafile/system.256.630244579
channel ORA_DISK_1: restoring datafile 00002 to
+DATA/dw/datafile/sysaux.257.630244581
channel ORA_DISK_1: restoring datafile 00003 to
+DATA/dw/datafile/undotbs1.258.630244583
channel ORA_DISK_1: restoring datafile 00004 to
+DATA/dw/datafile/users.259.632441707
channel ORA_DISK_1: restoring datafile 00005 to
+DATA/dw/datafile/example.265.630244801
channel ORA_DISK_1: restoring datafile 00006 to
+DATA/dw/datafile/users_crypt.267.630456963
channel ORA_DISK_1: restoring datafile 00007 to
+DATA/dw/datafile/inet_star.268.632004213
channel ORA_DISK_1: restoring datafile 00008 to
+DATA/dw/datafile/inet_intl_star.269.632009933
channel ORA_DISK_1: reading from backup piece
+RECOV/dw/backupset/2007_09_04/
nnndf0_tag20070904t215119_0.266.632440295
channel ORA_DISK_1: piece
handle=+RECOV/dw/backupset/2007_09_04/
nnndf0_tag20070904t215119_0.266.63244295
tag=TAG20070904T215119
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:03:11
```

```

Finished restore at 04-SEP-07
RMAN> recover database;
Starting recover at 04-SEP-07
using channel ORA_DISK_1

starting media recovery

archived log for thread 1 with sequence 142
  is already on disk as file
  +RECOV/dw/archivelog/2007_09_04/
  thread_1_seq_142.265.632440415
archived log for thread 1 with sequence 143
  is already on disk as file
  +RECOV/dw/archivelog/2007_09_04/
  thread_1_seq_143.349.632442295
archived log for thread 1 with sequence 144
  is already on disk as file
  +RECOV/dw/archivelog/2007_09_04/
  thread_1_seq_144.350.632442297
archived log for thread 1 with sequence 145
  is already on disk as file
  +RECOV/dw/archivelog/2007_09_04/
  thread_1_seq_145.351.632442303
archived log file
  name=+RECOV/dw/archivelog/2007_09_04/
  thread_1_seq_142.265.632440415 thread=1 sequence=142
archived log file
  name=+RECOV/dw/archivelog/2007_09_04/
  thread_1_seq_143.349.632442295 thread=1 sequence=143
media recovery complete, elapsed time: 00:00:35
Finished recover at 04-SEP-07
starting full resync of recovery catalog
full resync complete
RMAN> alter database open;
database opened
RMAN>

```

The database is now open and available for use. RMAN will pick the most efficient way to perform the requested operation, minimizing the number of files accessed or the number of disk I/Os to bring the database back to a consistent state in as short a time as possible. In the previous example, RMAN used a full database backupset and archived redo log files to recover the database.

During a recovery operation, RMAN may need to restore archived redo logs from tape; to limit the amount of disk space used during a recovery operation, the **recover** command used in the previous example could use the following options instead:

```
RMAN> recover database delete archivelog maxsize 2gb;
```

The parameter **delete archivelog** directs RMAN to remove archived log files from disk that were restored from tape for this recovery option; the **maxsize 2gb** parameter restricts the amount of space that can be occupied by restored archived log files at any point in time to 2GB. In our **dw** database, these two parameters are not needed; all archived log files needed to recover the database are kept in the flash recovery area on disk to support the defined retention policy.

## Validating Restore Operations

Earlier in this chapter, we validated the data blocks in the datafiles that we want to back up. In this section, we'll take the opposite approach and instead validate the backups that we have already made. We'll also find out from RMAN which backupsets, image copies, and archived redo logs would be used in a recovery operation without actually performing the recovery.

### RESTORE PREVIEW

The command **restore preview** will provide a list of the files that RMAN will use to perform the requested operation; the preview will also indicate if a tape volume will be requested, for example. No files are actually restored; only the recovery catalog is queried to determine which files are needed. In the following example, we want to find out what RMAN will need if we need to recover the USERS tablespace:

```
RMAN> restore tablespace users preview;
```

```
Starting restore at 04-SEP-07
using channel ORA_DISK_1
```

```
List of Backup Sets
```

```
=====
```

BS Key	Type	LV	Size	Device	Type	Elapsed Time	Completion Time
744	Full		239.34M	DISK		00:01:45	04-SEP-07
	BP Key: 747		Status: AVAILABLE	Compressed: YES		Tag:	
TAG20070904T215119							
Piece Name:							
+RECOV/dw/backupset/2007_09_04/nnndf0_tag20070904t215119_0.266.632440295							
List of Datafiles in backup set 744							
File	LV	Type	Ckp	SCN	Ckp Time	Name	
4		Full	3472960		04-SEP-07	+DATA/dw/datafile/users.259.632441707	

```
List of Archived Log Copies for database with db_unique_name DW
```

```
=====
```

Key	Thrd	Seq	S	Low	Time
806	1	142	A		04-SEP-07
Name: +RECOV/dw/archivelog/2007_09_04/thread_1_seq_142.265.632440415					
833	1	143	A		04-SEP-07
Name: +RECOV/dw/archivelog/2007_09_04/thread_1_seq_143.349.632442295					
831	1	144	A		04-SEP-07
Name: +RECOV/dw/archivelog/2007_09_04/thread_1_seq_144.350.632442297					
835	1	145	A		04-SEP-07
Name: +RECOV/dw/archivelog/2007_09_04/thread_1_seq_145.351.632442303					

```
Media recovery start SCN is 3472960
```

```
Recovery must be done beyond SCN 3472960 to clear datafile fuzziness
```

```
Finished restore at 04-SEP-07
```

```
RMAN>
```

For the restore operation, RMAN will need to use one backupset for the single datafile in the tablespace; archived redo log files will be used to bring the tablespace up to the current SCN.

If a restore operation needs to be performed immediately, and one of the files that RMAN will request to perform the operation is offsite, you can use the **change . . . unavailable** command to mark a backupset as unavailable and then run the **restore tablespace . . . preview** command again to see if RMAN can use disk-based backupsets to fulfill the request.

## RESTORE VALIDATE

The **restore . . . preview** command does not read the actual backupsets, only the catalog information; if we want to validate whether the backupsets themselves are readable and not corrupted, we use the **restore . . . validate** command. As with most other RMAN commands, we can perform the validation for a datafile, a tablespace, or the entire database. In the following example, we'll perform a validation on the same backupsets that RMAN reported in the previous example for the USERS tablespace:

```
RMAN> restore tablespace users validate;
Starting restore at 04-SEP-07
using channel ORA_DISK_1
channel ORA_DISK_1: starting validation of datafile backup set
channel ORA_DISK_1: reading from backup piece
+RECOV/dw/backupset/2007_09_04/
  nnndf0_tag20070904t230656_0.354.632444895
channel ORA_DISK_1: piece
handle=+RECOV/dw/backupset/2007_09_04/
  nnndf0_tag20070904t230656_0.354.632444895 tag=TAG20070904T230656
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: validation complete, elapsed time: 00:02:13
Finished restore at 04-SEP-07
RMAN>
```

All blocks of the backupsets were read to ensure that they are usable for a restore operation for the USERS tablespace.

## Point in Time Recovery

RMAN can be used to implement *point in time recovery*, or restoring and recovering a database up to a timestamp or SCN before the point at which a database failure occurred. As you found out in Chapter 11, a point in time recovery (PITR) may be useful for recovering from a user error where a table was dropped yesterday, but the error was not detected until today. Using PITR, we can recover the database to a point in time right before the table was dropped.

Using PITR has the disadvantage of losing all other changes to the database from the point at which the database was restored; this disadvantage needs to be weighed against the consequences of the dropped table. If both options are undesirable, then another method such as Flashback Table, Flashback Database, or tablespace point in time recovery (TSPITR) should be considered as an alternative for recovering from these types of user errors.

## Data Recovery Advisor

In a busy DBA's schedule, you may be aware of a database failure (usually from a user's phone call or e-mail), but you don't know the specific cause; using the Data Recovery Advisor, new to Oracle Database 11g, you can zoom in on the failure without checking the alert log or trace files.

The Data Recovery Advisor is available at the RMAN command line or in Oracle Enterprise Manager, as you'll see in the following paragraphs.

In this scenario, the datafile for the tablespace XPORT\_DW was accidentally deleted by the system administrator; the next time one of the users tries to create a table in this tablespace, they get this message:

```
SQL> create table daily_lineitem
  2 tablespace xport_dw
  3 as select * from oe.lineitem_table;
as select * from oe.lineitem_table
      *
ERROR at line 3:
ORA-01658: unable to create INITIAL extent for segment
in tablespace XPORT_DW
SQL>
```

You get an instant message from the user notifying you of the failure, and since you're already at the RMAN command-line interface, you use the **list failure** command to see what the problem might be:

```
RMAN> list failure;

List of Database Failures
=====

Failure ID Priority Status      Time Detected Summary
-----
1022      HIGH      OPEN        05-SEP-07   One or more non-system
                                     datafiles are corrupt
```

There is only one failure, so you drill down into the failure using the Failure ID and the **detail** option of the **list failure** command:

```
RMAN> list failure 1022 detail;

List of Database Failures
=====

Failure ID Priority Status      Time Detected Summary
-----
1022      HIGH      OPEN        05-SEP-07   One or more non-system
                                     datafiles are corrupt

Impact: See impact for individual child failures
List of child failures for parent failure ID 1022

Failure ID Priority Status      Time Detected Summary
-----
1025      HIGH      OPEN        05-SEP-07   Datafile 9:
                                     '/u02/oradata/xport_dw.dbf' is corrupt
Impact: Some objects in tablespace
XPORT_DW might be unavailable
```

Similar information is available in OEM; Figure 12-18 shows the OEM page when you click the Perform Recovery link, as you did earlier in this chapter.

ORACLE Enterprise Manager 11g Database Control

Database Instance: dw.world >

Database Failures - 1

Perform Recovery

**Oracle Advised Recovery**

The Data Recovery Advisor has detected failures. Click on "Advise and Recover" to have Oracle analyze and produce recovery advice. **Advise and Recover**

Failures Detected **Critical: 0 High: 1 Low: 0**  
 Failure Description **One or more non-system datafiles are corrupt**

**User Directed Recovery**

Recovery Scope: **Whole Database** **Recover**

Operation Type

- Recover to the current time or a previous point-in-time  
Datafiles will be restored from the latest usable backup as required.
- Restore all datafiles  
Specify Time, SCN or log sequence. The backup taken at or prior to that time will be used. No recovery will be performed in this operation.
- Recover from previously restored datafiles

**Decrypt Backups**

Host Credentials

**Overview**

- ◆ Recover database failures as advised by Oracle
- ◆ Restore and/or recover the entire database or selected objects
- ◆ Restore files to a new location
- ◆ Recover tablespaces to a point-in-time based on a timestamp, system change number (SCN), or log sequence number
- ◆ Recover datafile data blocks that are marked as corrupted, or based on datafile block IDs or tablespace block addresses
- ◆ Flashback database, tables, or transactions to a specific system change number (SCN) or timestamp

**FIGURE 12-18** OEM Recovery Advisor interface

In your RMAN session, you use the **advise failure** command to see a possible course of action:

```
RMAN> advise failure;
```

```
List of Database Failures
```

```
=====
```

Failure ID	Priority	Status	Time Detected	Summary
1022	HIGH	OPEN	05-SEP-07	One or more non-system datafiles are corrupt

```
Impact: See impact for individual child failures
List of child failures for parent failure ID 1022
```

Failure ID	Priority	Status	Time Detected	Summary
1025	HIGH	OPEN	05-SEP-07	Datafile 9: '/u02/oradata/xport_dw.dbf' is corrupt Impact: Some objects in tablespace XPORT_DW might be unavailable

```
analyzing automatic repair options; this may take some time
using channel ORA_DISK_1
analyzing automatic repair options complete
```

```
Mandatory Manual Actions
=====
no manual actions available

Optional Manual Actions
=====
no manual actions available
```

```
Automated Repair Options
=====
Option Repair Description
-----
```

```
1      Restore and recover datafile 9
      Strategy: The repair includes complete media recovery with no data loss
      Repair script: /u01/app/oracle/diag/rdbms/dw/dw/hm/reco_3725543542.hm
```

RMAN>

Clicking the Advise And Recover button in Figure 12-19 gives you the same advice, as you might expect; Figure 12-20 shows the RMAN script that OEM will execute to recover from the media failure.

The screenshot shows the Oracle Enterprise Manager 11g Database Control interface. At the top, it says 'ORACLE Enterprise Manager 11g Database Control' and 'Database Instance: dw.world > View and Manage Failures'. The last refresh is 'September 5, 2007 11:37:38 PM CDT'. Below this, there are filters for Failure Description, Impact, Priority (CRITICAL or HIGH), Status (OPEN), and Time Detected (All). A table titled 'Select failures and ...' shows a list of failures. The first failure is 'Data Failures' with a dropdown arrow. The second failure is 'One or more non-system datafiles are corrupt' with an impact of 'See impact for individual child failures', priority of 'HIGH', status of 'OPEN', and time detected of '2007-09-05 23:20:31.0'. The third failure is 'Datafile 9: /u02/oradata/xport\_dw.dbf is corrupt' with an impact of 'Some objects in tablespace XPORT\_DW might be unavailable', priority of 'HIGH', status of 'OPEN', and time detected of '2007-09-05 23:20:31.0'. Below the table, there is a tip: 'TIP All CRITICAL failures must be selected before "Advise". All CRITICAL failures must be unselected before "Set Priority High" or "Set Priority Low".' There are also 'Related Links' for 'Checkers'.

ORACLE Enterprise Manager 11g Database Control

Database Instance: dw.world > View and Manage Failures

Last Refresh September 5, 2007 11:37:38 PM CDT

Select dropdown values and optionally enter failure description and impact strings to filter the data that is displayed in your results set.

Failure Description Impact Priority Status Time Detected

CRITICAL or HIGH OPEN All Go

Select failures and ... Advise Close Set Priority High Set Priority Low

Select All Select None Expand All Collapse All

Select	Failure Description	Impact	Priority	Status	Time Detected
<input type="checkbox"/>	Data Failures				
<input checked="" type="checkbox"/>	One or more non-system datafiles are corrupt	See impact for individual child failures	HIGH	OPEN	2007-09-05 23:20:31.0
<input checked="" type="checkbox"/>	Datafile 9: /u02/oradata/xport_dw.dbf is corrupt	Some objects in tablespace XPORT_DW might be unavailable	HIGH	OPEN	2007-09-05 23:20:31.0

TIP All CRITICAL failures must be selected before "Advise". All CRITICAL failures must be unselected before "Set Priority High" or "Set Priority Low".

Related Links

Checkers

Database | Setup | Preferences | Help | Logout

Copyright © 1996, 2007, Oracle. All rights reserved.

FIGURE 12-19 OEM Recovery Advisor recommended actions

ORACLE Enterprise Manager 11g Database Control

Setup Preferences Help Logout

Database

Database Instance: dw.world >

Recovery Advice

The repair includes complete media recovery with no data loss

Cancel Continue

RMAN Script

```
# restore and recover datafile
sql 'alter database datafile 9 offline';
restore datafile 9;
recover datafile 9;
sql 'alter database datafile 9 online';
```

Cancel Continue

Database | Setup | Preferences | Help | Logout

Copyright © 1996, 2007, Oracle. All rights reserved.  
Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.  
[About Oracle Enterprise Manager](#)

**FIGURE 12-20** OEM Recovery Advisor RMAN command summary

In any case, you perform a straightforward tablespace recovery in RMAN using the recommendations in `/u01/app/oracle/diag/rdbms/dw/dw/hm/reco_3725543542.hm`, which not uncoincidentally are the same recommendations you see in Figure 12-20:

```
RMAN> sql 'alter database datafile 9 offline';
```

```
sql statement: alter database datafile 9 offline
```

```
RMAN> restore datafile 9;
```

```
Starting restore at 05-SEP-07
using channel ORA_DISK_1
```

```
channel ORA_DISK_1: starting datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
channel ORA_DISK_1: restoring datafile 00009 to /u02/oradata/xport_dw.dbf
channel ORA_DISK_1: reading from backup piece
+RECOV/dw/backupset/2007_09_05/nnndf0_tag20070905t230653_0.304.632531225
channel ORA_DISK_1: piece
handle=+RECOV/dw/backupset/2007_09_05/
nnndf0_tag20070905t230653_0.304.632531225 tag=TAG20070905T230653
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:03
Finished restore at 05-SEP-07
```

```

RMAN> recover datafile 9;

Starting recover at 05-SEP-07
using channel ORA_DISK_1

starting media recovery
media recovery complete, elapsed time: 00:00:02

Finished recover at 05-SEP-07

RMAN> sql 'alter database datafile 9 online';

sql statement: alter database datafile 9 online

RMAN>

```

## Miscellaneous Operations

In the next few sections, we'll cover some of the other capabilities of RMAN, beyond the backup, restore, and recovery operations. We'll show how to record the existence of other backups made outside of the database and perform some catalog maintenance. We'll also give a couple more examples of the **list** and **report** commands.

## Cataloging Other Backups

On occasion, we want the recovery catalog to include backups made outside of RMAN, such as image copies made with operating system commands or with the **asmcmd** command, as in this example:

```

ASMCMDB> pwd
+DATA/dw/datafile
ASMCMDB> ls
EXAMPLE.265.630244801
INET_INTL_STAR.269.632009933
INET_STAR.268.632004213
SYSAUX.257.630244581
SYSTEM.256.630244579
UNDOTBS1.258.630244583
USERS.259.632441707
USERS_CRYPT.267.630456963
ASMCMDB> cp USERS.259.632441707 /u02/oradata/USERS.259.632441707
source +DATA/dw/datafile/USERS.259.632441707
target /u02/oradata/USERS.259.632441707
copying file(s)...
file, /u02/oradata/USERS.259.632441707, copy committed.
ASMCMDB>

```

### CAUTION

*Image copies created with operating system commands must be performed either while the database is shut down or by using the **alter tablespace . . . begin/end backup** commands.*

Recording this image copy of the USERS tablespace is easy in RMAN using the **catalog** command:

```
RMAN> catalog datafilecopy '/u02/oradata/USERS.259.632441707';
starting full resync of recovery catalog
full resync complete
cataloged datafile copy
datafile copy file name=/u02/oradata/
USERS.259.632441707 RECID=11 STAMP=632447886
RMAN>
```

Now that the image copy is recorded in the RMAN repository, it may be considered for use in restore and recovery operations for the USERS tablespace.

## Catalog Maintenance

Earlier in this chapter, we discussed the use of the **backup validate** command to ensure that all the files that could be used in a backup operation were available, readable, and not corrupted. In that example, we found out that we had a mismatch between what the catalog reported and the archived redo logs on disk; some old archived redo logs were inadvertently removed from disk during a cleanup operation. In this section, we'll step through some of the maintenance operations we would need to perform to bring the catalog in synch with what actually exists on disk.

```
RMAN> backup validate database archivelog all;
Starting backup at 05-SEP-07
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=121 device type=DISK
archived log +RECOV/dw/archivelog/2007_08_27/
thread_1_seq_95.284.631750213 not found or out of sync with catalog
trying alternate file for archived log of thread 1 with sequence 95
. . .
```

Our first attempt to fix the problem was to remove all obsolete files outside of our recovery window of four days, keeping an extra day's worth of logs, since we have plenty of space in the flash recovery area.

```
RMAN> delete obsolete recovery window of 5 days;

using channel ORA_DISK_1
Deleting the following obsolete backups and copies:
Type           Key      Completion Time  Filename/Handle
-----
Archive Log    131     25-AUG-07
/u01/app/oracle/product/11.1.0/db_1/dbs/arch1_83_630244724.dbf
Archive Log    133     25-AUG-07
/u01/app/oracle/product/11.1.0/db_1/dbs/arch1_84_630244724.dbf
Archive Log    134     25-AUG-07
+RECOV/dw/archivelog/2007_08_25/thread_1_seq_84.273.631578703
Archive Log    135     26-AUG-07
/u01/app/oracle/product/11.1.0/db_1/dbs/arch1_85_630244724.dbf
. . .
Archive Log    976     05-SEP-07
/u01/app/oracle/product/11.1.0/db_1/dbs/arch1_81_630244724.dbf
Archive Log    977     05-SEP-07
/u01/app/oracle/product/11.1.0/db_1/dbs/arch1_82_630244724.dbf
```

```

Do you really want to delete the above objects (enter YES or NO)? yes
deleted archived log
archived log file
name=/u01/app/oracle/product/11.1.0/db_1/dbs/arch1_83_630244724.dbf RECID=13 STAMP=631566057
deleted archived log
archived log file
name=/u01/app/oracle/product/11.1.0/db_1/dbs/arch1_84_630244724.dbf RECID=15
STAMP=631578707
deleted archived log
. . .
Deleted 79 objects
RMAN-06207: WARNING: 1 objects could not be deleted for DISK channel(s) due
RMAN-06208:          to mismatched status. Use CROSSCHECK command to fix
status
RMAN-06210: List of Mismatched objects
RMAN-06211: =====
RMAN-06212: Object Type   Filename/Handle
RMAN-06213: -----
RMAN-06214: Archivelog
+RECOV/dw/archivelog/2007_08_27/thread_1_seq_95.284.631750213
RMAN>

```

Although we did remove a lot of the obsolete files from the flash recovery area, the catalog and the contents of the disk were still not in synch; RMAN suggests that we use the **crosscheck** command to remedy the problem:

```
RMAN> crosscheck archivelog all;
```

```

released channel: ORA_DISK_1
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=121 device type=DISK
validation failed for archived log
archived log file
name+=RECOV/dw/archivelog/2007_08_27
/thread_1_seq_95.284.631750213 RECID=38 STAMP=631750217
validation succeeded for archived log
archived log file
name=/u01/app/oracle/product/11.1.0/db_1/dbs/arch1_114_630244724.dbf RECID=75 STAMP=632020373
. . .
archived log file name=/u01/app/oracle/product/11.1.0/db_1/dbs/arch1_147_630244724.dbf
RECID=141 STAMP=632444769
validation succeeded for archived log
archived log file
name+=RECOV/dw/archivelog/2007_09_04/thread_1_seq_147.353.632444769
RECID=142 STAMP=632444769
Crosschecked 69 objects
RMAN>

```

The missing archived redo logs are now marked as EXPIRED in the catalog, and they won't be considered when validating backups or for performing restore or recovery operations.

All datafiles that RMAN could consider for a backup operation, including archived redo logs, are available and readable.

## REPORT and LIST

All throughout this chapter, I've provided a number of examples of how to extract information from the recovery catalog, whether it resides in the target database control file or in a catalog database repository. We've used both the **list** and **report** commands. The primary difference

between these commands is in their complexity: The **list** command displays information about backupsets and image copies in the repository as well as lists the contents of scripts stored in the repository catalog:

```
RMAN> list backup summary;
```

```
List of Backups
```

```
=====
```

Key	TY	LV	S	Device	Type	Completion	Time	#Pieces	#Copies	Compressed	Tag
487	B	0	A	DISK		30-AUG-07		1	1	YES	
											TAG20070830T222903
509	B	F	A	DISK		30-AUG-07		1	1	NO	
											TAG20070830T223054
624	B	F	A	DISK		03-SEP-07		1	1	NO	
											TAG20070903T133622
661	B	F	A	DISK		03-SEP-07		1	1	YES	
											TAG20070903T224442
677	B	F	A	DISK		03-SEP-07		1	1	NO	
											TAG20070903T224753
744	B	F	A	DISK		04-SEP-07		1	1	YES	
											TAG20070904T215119
768	B	F	A	DISK		04-SEP-07		1	1	NO	
											TAG20070904T215311
889	B	F	A	DISK		04-SEP-07		1	1	YES	
											TAG20070904T230656
915	B	F	A	DISK		04-SEP-07		1	1	NO	
											TAG20070904T230932

```
RMAN>
```

In contrast, the **report** command performs a more detailed analysis of the information in the recovery catalog; as in one of our previous examples, we used **report** to identify which database files needed backups to comply with our retention policy. In the following example, we find out what the datafiles looked like back on 8/30/07, and then we query the current status of the datafiles:

```
RMAN> report schema at time='30-aug-07';
```

```
Report of database schema for database with db_unique_name DW
```

```
List of Permanent Datafiles
```

```
=====
```

File	Size (MB)	Tablespace	RB segs	Datafile Name
1	750	SYSTEM	YES	
				+DATA/dw/datafile/system.256.630244579
2	829	SYSAUX	NO	
				+DATA/dw/datafile/sysaux.257.630244581
3	60	UNDOTBS1	YES	
				+DATA/dw/datafile/undotbs1.258.630244583
4	5	USERS	NO	
				+DATA/dw/datafile/users.259.632441707

## 474 Oracle Database 11g DBA Handbook

```
5    100    EXAMPLE          NO
                                     +DATA/dw/datafile/example.265.630244801
6    500    USERS_CRYPT       NO
                                     +DATA/dw/datafile/users_crypt.267.630456963
```

### List of Temporary Files

=====

File	Size (MB)	Tablespace	Maxsize (MB)	Tempfile Name
1	60	TEMP	32767	+DATA/dw/tempfile/temp.264.630244787

### RMAN> report schema;

Report of database schema for database with db\_unique\_name DW

### List of Permanent Datafiles

=====

File	Size (MB)	Tablespace	RB segs	Datafile Name
1	750	SYSTEM	YES	+DATA/dw/datafile/system.256.630244579
2	829	SYSAUX	NO	+DATA/dw/datafile/sysaux.257.630244581
3	60	UNDOTBS1	YES	+DATA/dw/datafile/undotbs1.258.630244583
4	5	USERS	NO	+DATA/dw/datafile/users.259.632441707
5	100	EXAMPLE	NO	+DATA/dw/datafile/example.265.630244801
6	500	USERS_CRYPT	NO	+DATA/dw/datafile/users_crypt.267.630456963
7	100	INET_STAR	NO	+DATA/dw/datafile/inet_star.268.632004213
8	50	INET_INTL_STAR	NO	+DATA/dw/datafile/inet_intl_star.269.632009933

### List of Temporary Files

=====

File	Size (MB)	Tablespace	Maxsize (MB)	Tempfile Name
1	60	TEMP	32767	+DATA/dw/tempfile/temp.264.630244787

RMAN>

At some point between 8/30/07 and today, we created the tablespaces INET\_STAR and INET\_INTL\_STAR.