Oracle® Database

Backup and Recovery Quick Start Guide

10*g* Release 2 (10.2)

B14193-02

August 2005

Oracle Database Backup and Recovery Quick Start Guide has three purposes:

- To introduce the basic concepts of Oracle database backup and recovery, and Recovery Manager (RMAN), the tool recommends that you use for your backup and recovery
- To orient you on the rest of the backup and recovery documentation set
- To serve as a convenient quick reference for the most frequently used RMAN commands, options, and views

This document is organized into the following sections:

- Overview of Backup and Recovery Documentation Set
- Overview of the RMAN Environment
- Starting and Exiting RMAN
- Configuring Persistent Settings for the RMAN Environment
- Backing Up Database Files
- Restoring and Recovering Database Files
- Reporting on RMAN Operations
- Managing the RMAN Repository
- Repetitive Tasks: RMAN and Scripting
- RMAN Syntax Quick Reference
- Backup and Recovery Views

ORACLE

Overview of Backup and Recovery Documentation Set

In addition to this quick start guide, there are three volumes in the Backup and Recovery documentation set.

- *Oracle Database Backup and Recovery Basics* explains the concepts of backup and recovery and the most common techniques for using RMAN for backup, recovery and reporting in more detail, as well as providing more information on how to plan a backup and recovery strategy.
- *Oracle Database Backup and Recovery Advanced User's Guide* presents in-depth information on RMAN architecture, backup and recovery concepts and mechanisms, advanced recovery techniques such as point-in-time recovery and database flashback features, and backup and recovery performance tuning. It also covers **user-managed backup and recovery**, using host operating system facilities instead of RMAN. This volume is essential for backup and recovery of more sophisticated database deployments, and for advanced recovery scenarios.
- *Oracle Database Backup and Recovery Reference* provides complete information on syntax and semantics for all Recovery Manager commands, and describes the database views available for reporting on backup and recovery activities.

Overview of the RMAN Environment

Installed with the database, Recovery Manager (RMAN) is an Oracle database client which performs backup and recovery tasks on your databases and automates administration of your backup strategies. It greatly simplifies backing up, restoring, and recovering database files.

The RMAN environment consists of the utilities and databases that play a role in backing up your data. At a minimum, the environment for RMAN must include the following:

- The **target database** to be backed up;
- The **RMAN client**, which interprets backup and recovery commands, directs server sessions to execute those commands, and records your backup and recovery activity in the target database control file.

Some environments will also use these optional components:

- A **flash recovery area**, a disk location in which the database can store and manage files related to backup and recovery;
- **Media management software**, required for RMAN to interface with backup devices such as tape drives;
- A **recovery catalog** database, a separate database schema used to record RMAN activity against one or more target databases.

About the Target Database

The target database is the database that you are backing up, restoring, or recovering with RMAN.

About the RMAN Client

RMAN is a command-line-oriented database client, much like SQL*Plus, with its own command syntax. From the RMAN client you can issue RMAN commands and SQL statements to perform and report on backup and recovery operations.

RMAN can take interactive input or read input from plain text files (called **command files**). RMAN then communicates with one or more server processes on the target database server which actually perform the work. You can also access RMAN through the Enterprise Manager; for details see *Oracle Enterprise Manager Administrator's Guide*.

The RMAN executable is typically installed in the same directory as the other database executables. On Unix systems, for example, the RMAN executable is located in `$ORACLE_HOME/bin`.

About the RMAN Repository

RMAN maintains metadata about the target database and its backup and recovery operations in the **RMAN repository**. Among other things, RMAN stores information about its own configuration settings, the target database schema, archived redo logs, and all backup files on disk or tape. RMAN's `LIST`, `REPORT`, and `SHOW` commands display RMAN repository information.

The primary store for RMAN repository data is always the control file of the target database. The `CONTROL_FILE_RECORD_KEEP_TIME` initialization parameter controls how long backup records are kept in the control file before those records are re-used to hold information about more recent backups.

Another copy of the RMAN repository data can also be saved in the recovery catalog.

About the Recovery Catalog

Using a recovery catalog preserves RMAN repository information if the control file is lost, making it much easier to restore and recover following the loss of the control file. (A backup control file may not contain complete information about recent available backups.) The recovery catalog can also store a much more extensive history of your backups than the control file, due to limits on the number of control file records.

In addition to RMAN repository records, the recovery catalog can also hold RMAN **stored scripts**, sequences of RMAN commands for common backup tasks. Centralized storage of scripts in the recovery catalog can be more convenient than working with command files.

Except for stored scripts, all of RMAN's features work equally well with or without a recovery catalog. For more information on the recovery catalog see *Oracle Database Backup and Recovery Advanced User's Guide*.

About the Flash Recovery Area

The Automatic Disk-Based Backup and Recovery feature simplifies managing disk space and files related to backup and recovery, by managing all backup and recovery related files in a **flash recovery area**. You set the flash recovery area

location and size on disk, using the DB_RECOVERY_FILE_DEST and DB_RECOVERY_FILE_DEST_SIZE initialization parameters. You also specify a retention policy that dictates when backups may be discarded. RMAN then manages your backup storage, deleting obsolete backups and backups already copied to tape when space is needed, but keeping as many backups on disk as space permits. This minimizes restores from tape during data recovery operations to shorten restore and recovery times.

About Media Managers

To access sequential media devices like tape libraries, RMAN uses third-party **media management software**. A media manager controls these devices during backup and recovery, managing the loading, labeling and unloading of media, among other functions. Media management devices are sometimes called SBT (system backup to tape) devices.

The Oracle Backup Solutions Program (BSP) works with vendors to help them produce media management software for their devices. For enterprises that already use media management software in their environment, many of those software products can be directly integrated with RMAN. Contact your media management software vendor for details about whether they participate in the BSP and have an RMAN-compatible media management layer.

About RMAN Channels

RMAN uses server sessions on the target database instance to perform all backup, restore and recovery operations. Each server session used by RMAN is known as an RMAN **channel**.

A channel can be either a **disk channel**, used for backup tasks that perform disk I/O, or an **sbt channel**, which is used to interact with media managers.

At the beginning of a series of RMAN commands, you can use the RMAN ALLOCATE CHANNEL command to **allocate channels**, specifying the number of server sessions to use for the tasks, and settings that affect the behavior of each server session. (Note that to use ALLOCATE CHANNEL you must group the commands for which the allocated channels apply using a RUN block. The RMAN RUN command is described in "Controlling Scripts: The RUN Command" on page 14.)

You can also use the RMAN CONFIGURE command to **configure channels**, specifying persistent settings for the channels RMAN should allocate by default if you do not explicitly allocate channels for a particular task. The channels allocated as a result of configured settings are sometimes referred to as **automatic channels**. If any channels are explicitly allocated then the configured channel settings are ignored.

The number of channels that are used for an RMAN job controls the amount of work that is run in parallel. When backing up or restoring files, RMAN automatically schedules the work to make the most use of all the available channels.

One disk channel is immediately started when RMAN first connects to the target database, and remains as long as RMAN is connected. This channel, known as the **default channel**, is not used for tasks involving bulk data transfer, such as backup or restore of database files.

Starting and Exiting RMAN

The RMAN client is started by issuing the `rman` command at the command prompt of your operating system.

RMAN must connect to a target database (with SYSDBA privileges) to perform backup and recovery tasks. RMAN can also connect to a recovery catalog database if you are using one. Specify target and recovery catalog databases using command line options or using the `CONNECT` command.

This command illustrates connecting RMAN to a target database and a recovery catalog at startup:

```
% rman TARGET / CATALOG cat_usr/pwd@cat_str
```

Connect to a target database without using a recovery catalog:

```
% rman TARGET SYS/pwd@target_str
```

Starting RMAN without connecting to a database:

```
% rman
```

Once started, RMAN displays an "`RMAN>`" prompt for your commands.

Syntax of Common RMAN Command-line Options

```
RMAN
[ TARGET [=] connectStringSpec
| { CATALOG [=] connectStringSpec }
| LOG [=] ['] filename ['] [ APPEND ]
.
.
.
]...

connectStringSpec::=
['] [userid] [/ [password]] [@net_service_name] ['] 
```

This example appends the output from an RMAN session to a text file at `$ORACLE_HOME/dbs/log/msglog.log`

```
% rman TARGET / LOG $ORACLE_HOME/dbs/log/msglog.log APPEND
```

To quit the RMAN client, type `EXIT` at the RMAN prompt:

```
RMAN> EXIT
```

Configuring Persistent Settings for the RMAN Environment

You can use the RMAN `CONFIGURE` command to create persistent settings in the RMAN environment, which apply to all subsequent operations, even if you exit and restart RMAN. These configured settings can specify disk and SBT channel behavior, backup destinations, policies affecting backup strategy, and others. Some frequently used configuration settings are described below. See ***Oracle Database Backup and Recovery Basics*** for more information about configuration settings.

Viewing Current Configured Settings

This command shows all configurable settings:

```
RMAN> SHOW ALL;
```

The output lists the CONFIGURE commands to re-create this configuration.

Configuring Disk Devices and Channels

By default, RMAN allocates one disk channel for all operations automatically, and directs all backups to disk if no destination is specified. If your database uses a flash recovery area, then backups to disk are stored there if no other location is specified in the BACKUP command. Otherwise, disk backups are assumed to be stored in a platform-specific default location.

You can also configure a **format** for a disk channel, to specify a different default location for backups, by using the FORMAT clause with CONFIGURE CHANNEL DEVICE TYPE DISK.

The following command configures RMAN to write disk backups to the /tmp directory:

```
RMAN> CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT '/tmp/%U';
```

The format specifier %U is replaced with unique filenames for the files when you take backups. Refer to *Oracle Database Backup and Recovery Advanced User's Guide* for more details on configuring destinations for your disk backups.

To undo the configuration of a default disk location for backups, use the following CONFIGURE command to clear the setting:

```
RMAN> CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT CLEAR;
```

RMAN will write backups to the default location.

Configuring Tape Devices and Channels

After configuring your media management software, you can make the media manager the default destination for RMAN backups:

```
RMAN> CONFIGURE DEFAULT DEVICE TYPE TO sbt;
```

Some media managers require a PARMS string to configure device settings:

```
RMAN> CONFIGURE CHANNEL DEVICE TYPE sbt PARMS='ENV=mml_env_settings';
```

Multiple channels can be configured to run backups in parallel. This command configures three sbt channels for use in RMAN jobs:

```
RMAN> CONFIGURE DEVICE TYPE sbt PARALLELISM 3;
```

Configuring a Retention Policy

Retention policy governs how long database backups are retained, and determines how far into the past you can recover your database. Retention policy can be set in terms of a recovery window (how far into the past you need to be able to recover your database), or a redundancy value (how many backups of each file must be retained). Choosing an effective retention policy is a vital part of your backup strategy.

This command ensures that RMAN retains all backups needed to recover the database to any point in time in the last 7 days:

```
RMAN> CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 7 DAYS;
```

This command ensures that RMAN retains three backups of each datafile:

```
RMAN> CONFIGURE RETENTION POLICY TO REDUNDANCY 3;
```

Use DELETE OBSOLETE to immediately delete backups no longer required by the retention policy. (For backups stored in a flash recovery area, you do not need to perform this step. The database automatically deletes obsolete backups in the flash recovery area when space is needed, as well as files that have been backed up to a media manager.)

You can also use the KEEP option of the BACKUP and CHANGE commands to override the configured retention policy for individual backups-- for example, to force the retention of a specific backup taken before a major database change.

Configuring Control File Autobackups

The control file can be automatically backed up after each RMAN backup as a way to protect the RMAN repository. The following command configures RMAN to create these **control file autobackups**:

```
RMAN> CONFIGURE CONTROLFILE AUTOBACKUP ON;
```

By default, RMAN automatically generates names for control file autobackups and stores them in the flash recovery area. The following command configures RMAN to write control file autobackups to the /mybackupdir directory:

```
RMAN> CONFIGURE CONTROLFILE AUTOBACKUP FORMAT
          FOR DEVICE TYPE DISK TO '/mybackupdir/cf%F';
```

The %F element of the format string combines the DBID, day, month, year, and sequence number to generate a unique filename. %F must be included in any control file autobackup format.

Restoring Default Values for Configured Settings

Reset any CONFIGURE setting to its default by running the command with the CLEAR option, as shown here:

```
RMAN> CONFIGURE CHANNEL DEVICE TYPE sbt CLEAR;
RMAN> CONFIGURE RETENTION POLICY CLEAR;
RMAN> CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK CLEAR;
```

Backing Up Database Files

Use the RMAN BACKUP command to back up files. You will usually configure default devices and channels in advance; BACKUP backs up your data to the configured default device and channels for the type of backup requested.

If you specify BACKUP AS COPY, then RMAN copies the files as **image copies**, bit-for-bit copies of database files created on disk. These are identical to copies of the same files that you can create with operating system commands like cp on Linux or COPY on Windows, but by using BACKUP AS COPY they are recorded in the RMAN repository and RMAN can use them in restore operations. Image copies cannot be created on tape.

This command creates image copy backups of all datafiles in the database:

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

If you specify BACKUP AS BACKUPSET, then RMAN stores its backups in **backup sets**. A backup set consists of one or more **backup pieces**, physical files containing the data, written in a format that only RMAN can access. Only RMAN can create and restore backup sets. Backup sets can be written to disk or tape, and they are the only that RMAN can use to write backups to tape.

The following command creates a backup of the database and archived logs on tape, in backup set format, using the configured channels:

```
RMAN> BACKUP DEVICE TYPE sbt DATABASE PLUS ARCHIVELOG;
```

> **Note:** Backing up datafiles as backup sets on disk can save disk space and time because RMAN can skip backing up some unused datafile blocks. Backup sets, once written on disk, can be moved to tape with the BACKUP BACKUPSET command. See the description of unused block compression in ***Oracle Database Backup and Recovery Reference*** for details.

Backing Up Individual Files

You can back up individual tablespaces, datafiles and control files, server parameter files, and backup sets with various options, as in these examples:

```
RMAN> BACKUP ARCHIVELOG COMPLETION TIME BETWEEN
        'SYSDATE-31' AND 'SYSDATE-7';
RMAN> BACKUP TABLESPACE system, users, tools;
RMAN> BACKUP AS BACKUPSET DATAFILE
        'ORACLE_HOME/oradata/trgt/users01.dbf',
        'ORACLE_HOME/oradata/trgt/tools01.dbf';
RMAN> BACKUP DATAFILE 1,3,5;
RMAN> BACKUP CURRENT CONTROLFILE TO '/backup/curr_cf.copy';
RMAN> BACKUP SPFILE;
RMAN> BACKUP BACKUPSET ALL;
```

Backup Options

Here are some often-used BACKUP command options:

Parameter	Example	Explanation
FORMAT	FORMAT '/tmp/%U'	Specifies a location and name for backup pieces and copies. You must use substitution variables to generate unique filenames.
TAG	TAG 'monday_bak'	Specifies a user-defined string as a label for the backup. If you do not specify a tag, then RMAN assigns a default tag with the date and time.

The following BACKUP commands illustrate these options:

```
RMAN> BACKUP FORMAT='AL_%d/%t/%s/%p' ARCHIVELOG LIKE '%arc_dest%';
RMAN> BACKUP TAG 'weekly_full_db_bkup' DATABASE MAXSETSIZE 10M;
RMAN> BACKUP COPIES 2 DEVICE TYPE sbt BACKUPSET ALL;
```

Incremental Backups

If you specify `BACKUP INCREMENTAL`, RMAN will create **incremental backups** of your database. Incremental backups capture on a block-by-block basis changes in your database since a previous incremental backup. The starting point for an incremental backup strategy is a **level 0 incremental backup**, which backs up all blocks in the database. **Level 1** incremental backups, taken at regular intervals, contain only changed blocks since a previous incremental backup. These can be **cumulative** (including all blocks changed since the most recent level 0 backup) or **differential** (including only blocks changed since the most recent incremental backup, whether it is level 0 or level 1).

Incremental backups are generally smaller and faster to create than full database backups. Recovery from an incremental backup is faster than recovery using redo logs alone. During a restore from incremental backup, the level 0 backup is used as the starting point, then changed blocks are updated based on level 1 backups where possible to avoid re-applying changes from redo one at a time. Recovering with incremental backups requires no additional effort on your part. If incremental backups are available, RMAN will use them during recovery.

Incrementally Updated Backups

RMAN's **incrementally updated backups** feature allows for a more efficient incremental backup routine. Changes from level 1 backups can be used to roll forward an image copy level 0 incremental backup, so that it includes all changes as of the SCN at which the level 1 incremental backup was created. Recovery using the updated level 0 incremental backup is faster, because all changes from the level 1 incremental backup have already been applied.

See *Oracle Database Backup and Recovery Basics* for more details and examples for incremental backups and incrementally updated backups.

Validating Backups

You can run a test RMAN backup that does not generate any output. Validation confirms that a backup could be run, by confirming that all database files exist, are in their correct location, and are free of physical and logical corruption. For example:

```
RMAN> BACKUP VALIDATE DATABASE ARCHIVELOG ALL;
```

Restoring and Recovering Database Files

Use the `RESTORE` and `RECOVER` commands for RMAN restore and recovery of physical database files. Restoring datafiles is retrieving them from backups as needed for a recovery operation. Recovery is the application of changes from redo logs and incremental backups to a restored datafile, to bring the datafile to a desired SCN or point in time.

Recovering the Whole Database

Use the `RESTORE DATABASE` and `RECOVER DATABASE` commands on the whole database. For example:

```
RMAN> STARTUP FORCE MOUNT;
RMAN> RESTORE DATABASE;
RMAN> RECOVER DATABASE;
RMAN> ALTER DATABASE OPEN;
```

Note that the database must not be open when restoring or recovering the entire database.

Recovering Current Tablespaces

Use the RESTORE TABLESPACE and RECOVER TABLESPACE commands on individual tablespaces when the database is open. Take the tablespace that needs recovery offline, restore and then recover the tablespace, and bring the recovered tablespace online. The following steps recover the users tablespace:

```
RMAN> SQL 'ALTER TABLESPACE users OFFLINE';
RMAN> RESTORE TABLESPACE users;
RMAN> RECOVER TABLESPACE users;
RMAN> SQL 'ALTER TABLESPACE users ONLINE';
```

Recovering Current Datafiles

Use the RESTORE DATAFILE and RECOVER DATAFILE commands on individual current datafiles when the database is open. Take the datafile that needs recovery offline, restore and recover the datafile, and bring the datafile online. For example, to restore and recover datafile 7:

```
RMAN>   SQL 'ALTER DATABASE DATAFILE 7 OFFLINE';
RMAN>   RESTORE DATAFILE 7;
RMAN>   RECOVER DATAFILE 7;
RMAN>   SQL 'ALTER DATABASE DATAFILE 7 ONLINE';
```

Recovering Individual Data Blocks

RMAN can recover individual corrupted datafile blocks. When RMAN performs a complete scan of a file for a backup, any corrupted blocks are listed in V$DATABASE_BLOCK_CORRUPTION. Corruption is usually reported in alert logs, trace files or results of SQL queries. Use BLOCKRECOVER to repair all corrupted blocks:

```
RMAN> BLOCKRECOVER CORRUPTION LIST;
```

You can also recover individual blocks, as shown in this example:

```
RMAN> BLOCKRECOVER DATAFILE 7 BLOCK 233, 235 DATAFILE 4 BLOCK 101;
```

Validating Restores

You can run a RESTORE... VALIDATE operation to confirm that a restore operation can be performed successfully. RMAN decides which backup sets, datafile copies, and archived logs are needed for the operation, and scans them to verify that they are usable. For example:

```
RMAN> RESTORE DATABASE VALIDATE;
```

Reporting on RMAN Operations

The RMAN LIST and REPORT commands, generate reports on backup activities based on the RMAN repository. Use SHOW ALL to display the current RMAN configuration. You can also query the views described in "Backup and Recovery Views" on page 23.

Listing Backups

Run the LIST BACKUP and LIST COPY commands to display information about backups and datafile copies listed in the repository. You can display specific objects, as in the following examples:

```
RMAN> LIST BACKUP OF DATABASE;
RMAN> LIST COPY OF DATAFILE 1, 2, 3;
RMAN> LIST BACKUP OF ARCHIVELOG FROM SEQUENCE 1437;
RMAN> LIST CONTROLFILECOPY "/tmp/cf.cpy";
RMAN> LIST BACKUPSET OF DATAFILE 1;
```

For backups, you can control the format of LIST output with these options:

Parameter	Example	Explanation
BY BACKUP	LIST BACKUP OF DATABASE BY BACKUP	Organizes the output by backup set. This is the default mode of presentation.
BY FILE	LIST BACKUP BY FILE	Lists the backups according to which file was backed up.
SUMMARY	LIST BACKUP SUMMARY	Displays summary output. By default, the output is VERBOSE.

For both backups and copies you have the following additional options:

Parameter	Example	Explanation
EXPIRED	LIST EXPIRED COPY	Lists backups that are recorded in the RMAN repository but that were not present at the expected location on disk or tape during the last CROSSCHECK command. Such backups may have been deleted outside of RMAN.
RECOVERABLE	LIST BACKUP RECOVERABLE	Specifies datafile backups or copies that are available and that can be restored and recovered in the current database incarnation.

Reporting on Database Files and Backups

The REPORT command performs more complex analysis than LIST. Some of the main options are:

Parameter	Example	Explanation
NEED BACKUP	REPORT NEED BACKUP DATABASE	Shows which files need backing up under current retention policy. Use optional REDUNDANCY and RECOVERY WINDOW parameters to specify different criteria.
OBSOLETE	REPORT OBSOLETE	Lists backups that are obsolete under the configured retention policy. Use optional REDUNDANCY and RECOVERY WINDOW parameters to specify criteria.

Parameter	Example	Explanation
UNRECOVERABLE	REPORT UNRECOVERABLE	Lists all datafiles for which an unrecoverable operation has been performed against an object in the datafile since the last backup of the datafile.
SCHEMA	REPORT SCHEMA	Reports the tablespaces and datafiles in the database at the current time (default) or a different time.

Monitoring RMAN Through V$ Views

Status information for jobs in progress and completed jobs is stored in V$RMAN_STATUS. V$RMAN_OUTPUT contains the text ouptut of all RMAN jobs.

To see status information on jobs in V$RMAN_STATUS use the following query:

```
SELECT OPERATION, STATUS, MBYTES_PROCESSED, START_TIME, END_TIME from
V$RMAN_STATUS;
```

To correlate a channel with a process, run the following query in SQL*Plus while the RMAN job is executing:

```
SQL> COLUMN CLIENT_INFO FORMAT a30
SQL> COLUMN SID FORMAT 999
SQL> COLUMN SPID FORMAT 9999

SQL> SELECT s.SID, p.SPID, s.CLIENT_INFO
      FROM V$PROCESS p, V$SESSION s
      WHERE p.ADDR = s.PADDR
      AND CLIENT_INFO LIKE 'rman%';
```

To calculate the progress of an RMAN job, run the following query in SQL*Plus while the RMAN job is executing:

```
SQL> SELECT SID, SERIAL#, CONTEXT, SOFAR, TOTALWORK,
       ROUND(SOFAR/TOTALWORK*100,2) "% COMPLETE"
       FROM V$SESSION_LONGOPS
       WHERE OPNAME LIKE 'RMAN%' AND OPNAME NOT LIKE '%aggregate%'
       AND TOTALWORK != 0 AND SOFAR <> TOTALWORK;
```

Managing the RMAN Repository

RMAN repository metadata is always stored in the control file of the target database. You can also create a recovery catalog in a separate database, and RMAN will record its metadata there as well.

Monitoring Control File Records

If you do not use a recovery catalog, then eventually RMAN control file records are overwritten. Set this initialization parameter in the parameter file of the target database to determine how long records are kept:

```
CONTROL_FILE_RECORD_KEEP_TIME = number_of_days_to_keep
```

Crosschecking Backups

The CROSSCHECK command checks whether RMAN backups and copies in the repository are still readable by RMAN. Assuming that you have configured automatic channels, you can run these commands:

```
RMAN> CROSSCHECK BACKUP; # checks RMAN backups on configured devices
RMAN> CROSSCHECK COPY; # checks RMAN image copies on configured devices
```

If backups are stored with a media manager and sbt channels are not configured, then you must allocate a maintenance channel before CROSSCHECK and DELETE commands on sbt devices:

```
RMAN>
    ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE sbt;
   RMAN> CROSSCHECK BACKUP;
```

Deleting Backups Created with RMAN

The DELETE command removes RMAN backups and copies from DISK and sbt devices, marks the objects as DELETED in the control file, and removes the records from the recovery catalog (if you use a catalog). For example:

```
RMAN> DELETE BACKUPSET 101, 102, 103;
RMAN> DELETE CONTROLFILECOPY '/tmp/cf.cpy';
RMAN> DELETE NOPROMPT ARCHIVELOG UNTIL SEQUENCE = 7300;
RMAN> DELETE BACKUP OF SPFILE TABLESPACE users DEVICE TYPE sbt;
RMAN> DELETE BACKUP OF DATABASE LIKE '/tmp%'; # pattern match
RMAN> DELETE ARCHIVELOG ALL BACKED UP 2 TIMES TO DEVICE TYPE sbt;
```

The following options of the DELETE command are also useful:

Parameter	Example	Explanation
EXPIRED	DELETE EXPIRED	Deletes the backups and copies marked as EXPIRED (that is, "not found") by the CROSSCHECK command.
OBSOLETE	DELETE OBSOLETE	Deletes the backups and copies that are obsolete under the retention policy. REDUNDANCY and RECOVERY WINDOW parameters override the configured policy.
NOPROMPT	DELETE NOPROMPT OBSOLETE	Specifies that you do not want to be prompted to confirm the files to be deleted.

Cataloging and Uncataloging Backups and Copies

The CATALOG command adds information about useable backups to the RMAN repository. Use this command to record backups created with tools other than RMAN, such as datafile copies created with operating system-level utilities. You can also use this command if you have backups which are created using RMAN but which are no longer listed in the RMAN repository. RMAN can use these backups in restore and recovery operations. For example:

```
RMAN> CATALOG DATAFILECOPY '/backup/users01.bak'; # copy made with operating
system copy cmd
RMAN> CATALOG LIKE '/backup'
```

Note that the second example adds all usable backups where the filepath begins with /backup to the RMAN repository, including files in the directory /backup/users01.bak, files in subdirectories such as

/backup/tuesday/users01.bak.old, and files in directories whose name starts with /backup, such as /backup-2001/users01.bak.old. Take care when choosing your argument for CATALOG LIKE.

The CHANGE . . . UNCATALOG syntax lets you remove information about backups and copies from the RMAN repository. If you manually delete a backup using operating system commands, CHANGE. . .UNCATALOG updates the repository to reflect that change. For example:

```
RMAN> CHANGE CONTROLFILECOPY '/tmp/cf.cpy' UNCATALOG;
RMAN> CHANGE BACKUPSET 121,122,127,203,300 UNCATALOG;
```

Repetitive Tasks: RMAN and Scripting

While the use of configured channels and other settings reduces many common RMAN operations to a single command, your backup routine may include frequently-used multi-step processes. RMAN supports the use of stored scripts (discussed in *Oracle Database Backup and Recovery Advanced User's Guide*) and command files to help manage these recurring tasks. The RMAN RUN command provides a degree of flow-of-control in your scripts.

Using Command Files

A **command file** is a client-side text file containing RMAN commands, exactly as you enter them at the RMAN prompt. Execute the contents of a command file using the RMAN @ command:

```
RMAN> @/my_dir/my_command_file.txt  # runs specified command file
```

Any file extension may be used. You can also launch RMAN with a command file to run, as shown here:

```
% rman @/my_dir/my_command_file.txt
```

Controlling Scripts: The RUN Command

The RUN command lets you issue a series of RMAN commands to be executed as a group. If one command fails, the remaining commands in the block will not be executed. Note, however, that RMAN will still try to execute as many tasks related to a failed command as possible.

Here is an example of a RUN command:

```
RUN {
    BACKUP ARCHIVELOG ALL DELETE ALL INPUT;
    BACKUP INCREMENTAL LEVEL 0 TAG mon_bkup DATABASE;
}
```

If backup of one or more of the archived logs fails, RMAN will still back up all archived logs that can be backed up, because those tasks are all caused by the one BACKUP command. However, the BACKUP INCREMENTAL command following the BACKUP ARCHIVELOG command is not executed.

The SET, SWITCH DATAFILE, and ALLOCATE CHANNEL commands, used within a RUN block, override channel configurations and other backup parameters set with the CONFIGURE command for the duration of the RUN block. See *Oracle Database Backup and Recovery Reference* for details.

Use RUN blocks in command files to stop execution if one command fails.

RMAN Syntax Quick Reference

This section gives an overview of the most common RMAN commands and their most commonly used options. Commands documented here include:

- @
- BACKUP
- CHANGE
- CONFIGURE
- CROSSCHECK
- DELETE
- LIST
- RECOVER
- REPORT
- RESTORE
- RUN
- SET
- SHOW

The following subclauses are used in the parameters to several commands:

- archivelogRecordSpecifier
- completedTimeSpec
- datafileSpec
- deviceSpecifier
- maintQualifier
- untilClause

Syntax descriptions use vertical ellipses to indicate less-frequently-used parameters and options. Refer to the *Oracle Database Backup and Recovery Reference* for complete documentation of syntax and semantics of RMAN commands.

@

Reads in a command file; executes each command in it in order.

```
@filename
```

archivelogRecordSpecifier

This subclause specifies a range of archived redo logs.

```
ARCHIVELOG
{ ALL
| LIKE 'string_pattern'
| archlogRange [LIKE 'string_pattern' [THREAD [=] integer]]
}

{ { { UNTIL TIME | FROM TIME } [=] 'date_string'
  | { TIME BETWEEN 'date_string' AND
    | FROM TIME [=] 'date_string' UNTIL TIME [=]
```

```
    }
    'date_string'
  | UNTIL SCN [=] integer
  | SCN BETWEEN integer AND integer
  | FROM SCN [=] integer [UNTIL SCN [=] integer]
  }
  [THREAD [=] integer]
| { UNTIL SEQUENCE [=] integer
  | FROM SEQUENCE [=] integer [UNTIL SEQUENCE [=] integer]
  | SEQUENCE [BETWEEN integer AND] integer
  }
  [THREAD [=] integer]
}
```

BACKUP

Backs up database files, archived logs, backups, and copies.

```
BACKUP
[ (
  (FULL | INCREMENTAL LEVEL [=] integer)
  | [ (FULL | INCREMENTAL LEVEL [=] integer) ]
  AS (COPY | BACKUPSET)
  | AS (COPY | BACKUPSET)
  (FULL | INCREMENTAL LEVEL [=] integer)
  )
]
[backupOperand [backupOperand]...] backupSpec [backupSpec]...
[PLUS ARCHIVELOG [backupSpecOperand [backupSpecOperand]...]];


backupOperand::=
{ FORMAT [=] 'format_string' [, 'format_string']...
| CHANNEL ['] channel_id ']
| CUMULATIVE
| MAXSETSIZE [=] integer [ K | M | G ]
| TAG [=] ['] tag_name ']
| keepOption
| SKIP { OFFLINE | READONLY | INACCESSIBLE }
| VALIDATE
| NOT BACKED UP [SINCE TIME [=] 'date_string']
| COPIES [=] integer
| DEVICE TYPE deviceSpecifier
.
.
.
}

backupSpec::=
[(]
{ BACKUPSET
  { {ALL | completedTimeSpec }
  | primary_key) [, primary_key]...
  }
| COPY OF { DATABASE
          | TABLESPACE ['] tablespace_name ['] [, ['] tablespace_name ']]...
          | DATAFILE datafileSpec [, datafileSpec]...
          }
| DATAFILE datafileSpec [, datafileSpec]...
| DATAFILECOPY 'filename' [, 'filename']...
```

```
| DATAFILECOPY FROM TAG [=] ['] tag_name ['] [, ['] tag_name ']]...
| DATAFILECOPY { ALL | LIKE 'string_pattern' }
| TABLESPACE ['] tablespace_name ['] [, ['] tablespace_name ']]...
| DATABASE
| archivelogRecordSpecifier
| CURRENT CONTROLFILE [FOR STANDBY]
| CONTROLFILECOPY 'filename'
| SPFILE
}
[backupSpecOperand [backupSpecOperand]...]

backupSpecOperand::=
{ FORMAT [=] 'format_string' [, 'format_string']...
| CHANNEL ['] channel_id ']
| CUMULATIVE
| MAXSETSIZE [=] integer [ K | M | G ]
| TAG [=] ['] tag_name ']
| keepOption
| SKIP { OFFLINE | READONLY | INACCESSIBLE }
| NOT BACKED UP [ SINCE TIME [=] 'date_string'
                | integer TIMES
                ]
| DELETE [ALL] INPUT
.
.
.
}
```

CHANGE

Updates status of a backup or copy in the RMAN repository.

```
CHANGE
{ { BACKUP | COPY } [OF listObjList] [ maintQualifier [maintQualifier]...]
| archivelogRecordSpecifier
| recordSpec [DEVICE TYPE deviceSpecifier [, deviceSpecifier]...]
}
{ AVAILABLE | UNAVAILABLE | UNCATALOG | keepOption }
[DEVICE TYPE deviceSpecifier [, deviceSpecifier]...];
```

completedTimeSpec

```
COMPLETED
{ AFTER [=]
| BETWEEN 'date_string' AND | BEFORE [=] } 'date_string'
```

CONFIGURE

Change persistent RMAN configuration settings.

```
CONFIGURE
{ deviceConf
| backupConf
| { AUXNAME FOR DATAFILE datafileSpec
  | SNAPSHOT CONTROLFILE NAME
  }
  { TO 'filename' | CLEAR }
| cfauConf
};
```

```
deviceCon::=
{ DEFAULT DEVICE TYPE { TO deviceSpecifier | CLEAR }
| DEVICE TYPE deviceSpecifier { PARALLELISM integer | CLEAR }
| [AUXILIARY] CHANNEL [integer] DEVICE TYPE deviceSpecifier
  { allocOperandList | CLEAR }
}

allocOperandList::=
{ PARMS [=] 'channel_parms'
| FORMAT [=] 'format_string' [, 'format_string']...
| { MAXPIECESIZE [=] integer | RATE [=] integer } [ K | M | G ]
.
.
.
}...
connectStringSpec::=
['] [userid] [/ [password]] [@net_service_name] [']

backupConf::=
{ RETENTION POLICY { TO { RECOVERY WINDOW OF integer DAYS
                          | REDUNDANCY [=] integer
                          | NONE
                        }
                     | CLEAR
                   }
| MAXSETSIZE { TO { integer [ K | M | G ]
                  | UNLIMITED
                  }
              | CLEAR
             }
| { ARCHIVELOG | DATAFILE }
  BACKUP COPIES FOR DEVICE TYPE deviceSpecifier
  { TO integer | CLEAR }
| BACKUP OPTIMIZATION { ON | OFF | CLEAR }
| EXCLUDE FOR TABLESPACE tablespace_name [CLEAR]
}

cfauConf::==
CONTROLFILE AUTOBACKUP
{ ON
| OFF
| CLEAR
| FORMAT FOR DEVICE TYPE deviceSpecifier { TO 'format string' | CLEAR }
}
```

CROSSCHECK

Checks whether backup pieces, proxy copies, and disk copies still exist.

```
CROSSCHECK
{
  { BACKUP [OF listObjList]
  | COPY [OF listObjList]
  | archivelogRecordSpecifier
  } [maintQualifier [maintQualifier]...]
| recordSpec [DEVICE TYPE deviceSpecifier [, deviceSpecifier]...]
};

listObjList::=
[ DATAFILE datafileSpec [, datafileSpec]...
```

```
| TABLESPACE ['] tablespace_name ['] [, ['] tablespace_name ']]...
| archivelogRecordSpecifier
| DATABASE [SKIP TABLESPACE ['] tablespace_name ['] [, ['] tablespace_name
[']]...]
| CONTROLFILE
| SPFILE
]...

recordSpec::=
{ { BACKUPPIECE | PROXY }
  { 'media_handle' [, 'media_handle']...
  | primary_key [, primary_key]...
  | TAG [=] ['] tag_name ']
  }
| BACKUPSET primary_key [, primary_key]...
| { CONTROLFILECOPY | DATAFILECOPY }
  { { primary_key [, primary_key]...
    | 'filename' [, 'filename']...
    }
  | TAG [=] ['] tag_name ['] [, ['] tag_name ']]...
  }
| ARCHIVELOG
  { primary_key [, primary_key]...
  | 'filename' [, 'filename']...
  }
}
```

datafileSpec

```
datafileSpec::=
{ 'filename'| integer }
```

DELETE

Deletes backups and copies from disk or tape media, and updates the RMAN repository accordingly.

```
DELETE [FORCE] [NOPROMPT]
{ [EXPIRED]
  {
    { BACKUP [OF listObjList]
    | COPY   [OF listObjList]
    | archivelogRecordSpecifier
    } [maintQualifier [maintQualifier]...]
  | recordSpec [DEVICE TYPE deviceSpecifier [, deviceSpecifier]...]
  }
| OBSOLETE [obsOperandList]
  [DEVICE TYPE (deviceSpecifier [, deviceSpecifier]...]
};

obsOperandList::=
[ REDUNDANCY [=] integer | RECOVERY WINDOW OF integer DAYS ]...
```

deviceSpecifier

```
deviceSpecifier::=
{ DISK | ['] media_device ['] }
```

LIST

Lists the backups and copies recorded in the repository.

```
LIST
{ INCARNATION [OF DATABASE [['] database_name [']]]
| [EXPIRED]
  { listObjectSpec
    [ maintQualifier | RECOVERABLE [untilClause] ]...
  | recordSpec
  }
};

listObjectSpec::=
{ BACKUP [OF listObjList] [listBackupOption]
| COPY [OF listObjList]
| archivelogRecordSpecifier
}

listObjectList::=
[ DATAFILE datafileSpec [, datafileSpec]...
| TABLESPACE ['] tablespace_name ['] [, ['] tablespace_name [']]...
| archivelogRecordSpecifier
| DATABASE [SKIP TABLESPACE ['] tablespace_name ['] 
  [, ['] tablespace_name [']]...]
| CONTROLFILE
| SPFILE
]...

listBackupOption::=
[ [BY BACKUP] [VERBOSE]
| SUMMARY
| BY { BACKUP SUMMARY | FILE }
]
```

maintQualifier

```
{ TAG [=] ['] tag_name ['] 
| completedTimeSpec
| LIKE 'string_pattern'
| DEVICE TYPE deviceSpecifier [, deviceSpecifier]...
| BACKED UP integer TIMES TO DEVICE TYPE deviceSpecifier
}
```

RECOVER

Performs media recovery from RMAN backups and copies.

```
RECOVER [DEVICE TYPE deviceSpecifier [, deviceSpecifier]...]
recoverObject [recoverOptionList];

recoverObject::=
{ DATABASE
  [ untilClause
  | [untilClause] SKIP [FOREVER] TABLESPACE
    ['] tablespace_name ['] [, ['] tablespace_name [']]...
  ]
| TABLESPACE ['] tablespace_name ['] [, ['] tablespace_name [']]...
| DATAFILE datafileSpec [, datafileSpec]...
}

recoverOptionList::=
{ DELETE ARCHIVELOG [MAXSIZE {integer [K | M | G]}]
| CHECK READONLY
```

```
| NOREDO
| { FROM TAG | ARCHIVELOG TAG } [=] ['] tag_name ['] 
.
.
.
}...
```

REPORT

Reports backup status of your database: which files are in the database, which files need backups, and which backups are obsolete or unrecoverable.

```
REPORT
{ { NEED BACKUP [ { INCREMENTAL | DAYS } [=] integer
                  | REDUNDANCY [=] integer
                  | RECOVERY WINDOW OF integer DAYS)
                 ]
  | UNRECOVERABLE
  }
  reportObject
| SCHEMA [atClause]
| OBSOLETE [obsOperandList]
}
[ DEVICE TYPE deviceSpecifier [,deviceSpecifier]... ]

reportObject::=
[ DATAFILE datafileSpec [, datafileSpec]...
| TABLESPACE ['] tablespace_name ['] [, ['] tablespace_name ']]...
| DATABASE [SKIP TABLESPACE ['] tablespace_name ['] [, ['] tablespace_name
[']]...]
]

atClause::=
{ AT TIME [=] 'date_string'
| AT SCN [=] integer
| AT SEQUENCE [=] integer THREAD [=] integer
}

obsOperandList::=
[ REDUNDANCY [=] integer | RECOVERY WINDOW OF integer DAYS ]...
```

RESTORE

Restores RMAN backups and copies.

```
RESTORE
[(] restoreObject [(restoreSpecOperand [restoreSpecOperand]...] [)]...
[ CHANNEL ['] channel_id ']
| PARMS [=] 'channel_parms'
| FROM { BACKUPSET | DATAFILECOPY }
| untilClause
| FROM TAG [=] ['] tag_name ']
| VALIDATE
| DEVICE TYPE deviceSpecifier [, deviceSpecifier]...
.
.
.
]...;

restoreObject::=
{ CONTROLFILE [TO 'filename']
```

```
| DATABASE
  [SKIP [FOREVER] TABLESPACE
   ['] tablespace_name ['] [, ['] tablespace_name ']]...
  ]
| DATAFILE datafileSpec [, datafileSpec]...
| TABLESPACE ['] tablespace_name ['] [, ['] tablespace_name ']]...
| archivelogRecordSpecifier
| SPFILE [TO [PFILE] 'filename']
}

restoreSpecOperand::=
{ CHANNEL ['] channel_id ']
| FROM TAG [=] ['] tag_name ']
| PARMS [=] 'channel_parms'
| FROM
  { AUTOBACKUP
    [{ MAXSEQ | MAXDAYS } [=] integer)]...
  | 'media_handle'
  }
}
```

RUN

Some RMAN commands are only valid inside a RUN block.

```
RUN {
  ...
}
```

SET

Creates settings that apply only to the current RMAN session.

```
SET { set_rman_option [;] | set_run_option; }

set_rman_option::=
{ ECHO { ON | OFF }
| DBID [=] integer
| CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE deviceSpecifier TO 'frmt_
string'

set_run_option::=
{ NEWNAME FOR DATAFILE datafileSpec TO { 'filename' | NEW }
| ARCHIVELOG DESTINATION TO 'log_archive_dest'
| untilClause
| COMMAND ID TO 'string'
| CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE deviceSpecifier TO 'frmt_
string'
.
.
.
}
```

SHOW

Displays the currently enabled `CONFIGURE` commands.

```
SHOW
{ RETENTION POLICY
| [DEFAULT] DEVICE TYPE
| [AUXILIARY] CHANNEL [FOR DEVICE TYPE deviceSpecifier]
```

```
| MAXSETSIZE
| { DATAFILE | ARCHIVELOG } BACKUP COPIES
| BACKUP OPTIMIZATION
| SNAPSHOT CONTROLFILE NAME
| AUXNAME
| EXCLUDE
| CONTROLFILE AUTOBACKUP [FORMAT]
| ALL
};
```

untilClause

```
{ UNTIL TIME [=] 'date_string'
| UNTIL SCN [=] integer
| UNTIL SEQUENCE [=] integer THREAD [=] integer
}
```

Backup and Recovery Views

The following table describes views you can use to query the RMAN repository. The V$ views reflect the RMAN repository as stored in the database control file, and the recovery catalog views reflect the RMAN repository as stored in the recovery catalog.

Refer to the *Oracle Database Reference* for details on V$ views, and *Oracle Database Backup and Recovery Reference* for details on recovery catalog views.

Control File V$ View	Recovery Catalog View	View Describes
V$ARCHIVED_LOG	RC_ARCHIVED_LOG	Archived and unarchived redo logs
V$BACKUP_DATAFILE	RC_BACKUP_ CONTROLFILE	Control files in backup sets
V$BACKUP_CORRUPTION	RC_BACKUP_ CORRUPTION	Corrupt block ranges in datafile backups
V$BACKUP_DATAFILE	RC_BACKUP_DATAFILE	Datafiles in backup sets
V$BACKUP_FILES	RC_BACKUP_FILES	RMAN backups and copies in the repository.
V$BACKUP_PIECE	RC_BACKUP_PIECE	Backup pieces
V$BACKUP_REDOLOG	RC_BACKUP_REDOLOG	Archived logs in backups
V$BACKUP_SET	RC_BACKUP_SET	Backup sets
V$BACKUP_SPFILE	RC_BACKUP_SPFILE	Server parameter files in backup sets
V$DATAFILE_COPY	RC_CONTROLFILE_ COPY	Control file copies on disk
V$COPY_CORRUPTION	RC_COPY_CORRUPTION	Information about datafile copy corruptions
V$DATABASE	RC_DATABASE	Databases registered in the recovery catalog (RC_ DATABASE) or information about the currently mounted database (V$DATABASE)

Control File V$ View	Recovery Catalog View	View Describes
V$DATABASE_BLOCK_CORRUPTION	RC_DATABASE_BLOCK_CORRUPTION	Database blocks marked as corrupt in the most recent RMAN backup or copy
V$DATABASE_INCARNATION	RC_DATABASE_INCARNATION	All database incarnations registered in the catalog
V$DATAFILE	RC_DATAFILE	All datafiles registered in the recovery catalog
V$DATAFILE_COPY	RC_DATAFILE_COPY	Datafile image copies
V$LOG_HISTORY	RC_LOG_HISTORY	Historical information about online redo logs
V$OFFLINE_RANGE	RC_OFFLINE_RANGE	Offline ranges for datafiles
V$PROXY_ARCHIVEDLOG	RC_PROXY_ARCHIVEDLOG	Archived log backups created by proxy copy
V$PROXY_CONTROLFILE	RC_PROXY_CONTROLFILE	Control file backups created by proxy copy
V$PROXY_DATAFILE	RC_PROXY_DATAFILE	Datafile backups created by proxy copy
V$LOG and V$LOGFILE	RC_REDO_LOG	Online redo logs for all incarnations of the database since the last catalog resynchronization
V$THREAD	RC_REDO_THREAD	All redo threads for all incarnations of the database since the last catalog resynchronization
n/a	RC_RESYNC	Recovery catalog resynchronizations
V$RMAN_CONFIGURATION	RC_RMAN_CONFIGURATION	RMAN persistent configuration settings
V$TABLESPACE	RC_TABLESPACE	All tablespaces registered in the recovery catalog, all dropped tablespaces, and tablespaces that belong to old incarnations

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at http://www.oracle.com/accessibility/.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

Oracle Database Backup and Recovery Quick Start Guide, 10*g* Release 2 (10.2)
B14193-02