

5

Creating Backups with RMAN

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Usman Qamar (usman.qamar@live.com) has a non-transferable license to use this Student Guide.

Objectives

After completing this lesson, you should be able to:

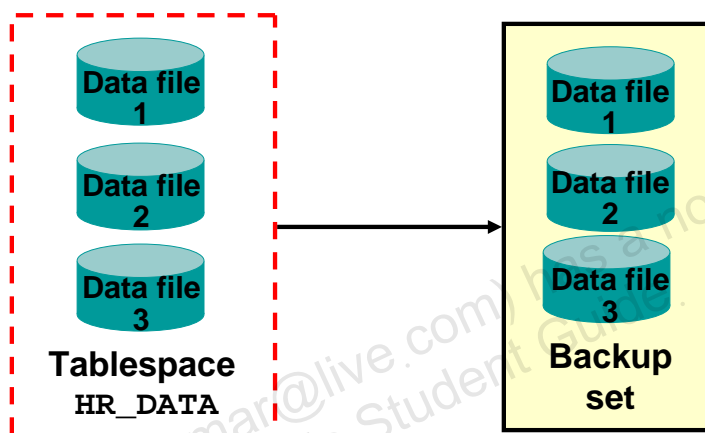
- Create image file backups
- Create a whole database backup
- Create a full database backup
- Enable fast incremental backup
- Create duplex backup sets
- Back up a backup set
- Create RMAN multi-section backup
- Create an archival backup for long-term retention
- Report on and maintain backups

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Creating Backup Sets

```
RMAN> BACKUP AS BACKUPSET  
2> FORMAT '/BACKUP/df_%d_%s_%p.bus'  
3> TABLESPACE hr_data;
```



ORACLE

Copyright © 2009, Oracle. All rights reserved.

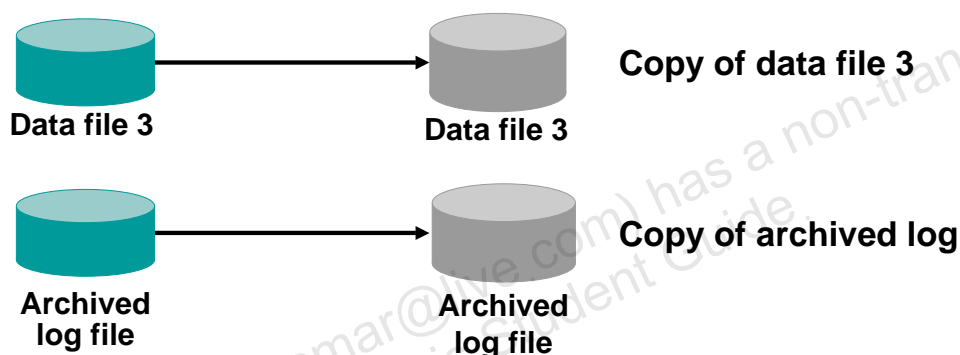
Creating Backup Sets

RMAN can store its backups in an RMAN-exclusive format called a backup set. A backup set is a collection of files called backup pieces, each of which may contain a backup of one or more database files.

Note: The `FORMAT` parameter specifies a pattern to use in creating a file name for the backup pieces created by this command. The `FORMAT` specification can also be provided through the `ALLOCATE CHANNEL` and `CONFIGURE` commands.

Creating Image Copies

```
RMAN> BACKUP AS COPY DATAFILE '/ORADATA/users_01_db01.dbf';  
RMAN> BACKUP AS COPY ARCHIVELOG LIKE '/arch%';
```



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Creating Image Copies

An image copy is a clone of a single data file, archived redo log, or control file. An image copy can be created with the `BACKUP AS COPY` command or with an operating system command. When you create the image copy with the `RMAN BACKUP AS COPY` command, the server session validates the blocks in the file and records the copy information in the control file.

An image copy has the following characteristics:

- An image copy can be written only to disk. When large files are being considered, copying may take a long time, but restoration time is reduced considerably because the copy is available on the disk.
- If files are stored on disk, they can be used immediately by using the `SWITCH` command in RMAN, which is equivalent to the `ALTER DATABASE RENAME FILE SQL` statement.
- In an image copy, all blocks are copied, whether they contain data or not, because an Oracle database process copies the file and performs additional actions such as checking for corrupt blocks and registering the copy in the control file. To speed up the process of copying, you can use the `NOCHECKSUM` parameter. By default, RMAN computes a checksum for each block backed up, and stores it with the backup. When the backup is restored, the checksum is verified. For more information about the `NOCHECKSUM` option of the `BACKUP` command, see the *Oracle Database Backup and Recovery Reference*.

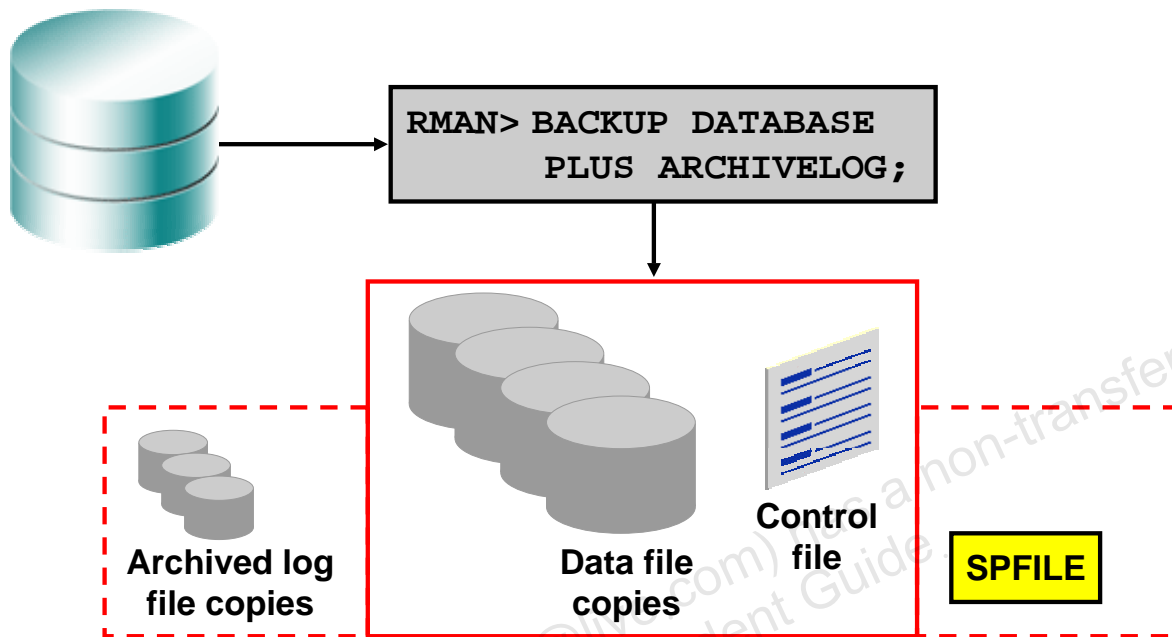
Creating Image Copies (continued)

- An image copy can be part of a full or incremental level 0 backup because a file copy always includes all blocks. You must use the level 0 option if the copy will be used in conjunction with an incremental backup set.

The example in the slide creates the following image copies:

- A copy of the /ORADATA/users01_db01.dbf data file
- A copy of the archived redo log files

Creating a Whole Database Backup



Creating a Whole Database Backup

A whole database backup can be either backup sets or image copies of the entire set of data files and must include the control file. You can optionally include the server parameter file (SPFILE) and archived redo log files. Using Recovery Manager (RMAN) to make an image copy of all the database files simply requires mounting or opening the database, starting RMAN, and entering the BACKUP command shown in the slide. Optionally, you can supply the DELETE INPUT option when backing up archivelog files. That causes RMAN to remove the archivelog files after backing them up. This is useful especially if you are not using a Fast Recovery Area, which would perform space management for you, deleting files when space pressure grows. In that case, the command in the slide would look like this:

```
RMAN> BACKUP DATABASE PLUS ARCHIVELOG DELETE INPUT;
```

You must have issued the following CONFIGURE commands to make the backup as described previously:

- CONFIGURE DEFAULT DEVICE TYPE TO disk;
- CONFIGURE DEVICE TYPE DISK BACKUP TYPE TO COPY;
- CONFIGURE CONTROLFILE AUTOBACKUP ON;

You can also create a backup (either a backup set or image copies) of previous image copies of all data files and control files in the database by using the following command:

```
RMAN> BACKUP COPY OF DATABASE;
```

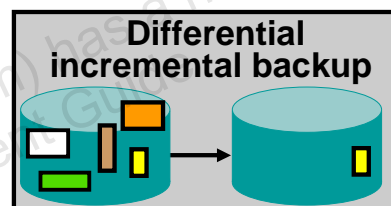
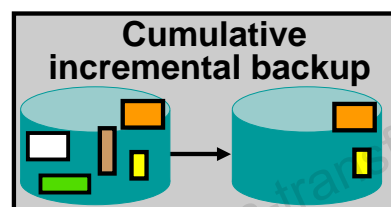
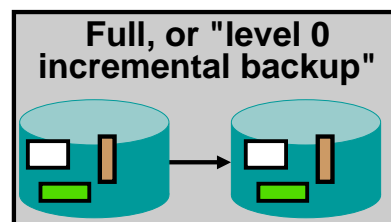
Creating a Whole Database Backup (continued)

By default, RMAN executes each BACKUP command serially. However, you can parallelize the copy operation by:

- Using the `CONFIGURE DEVICE TYPE DISK PARALLELISM n` command, where `n` is the desired degree of parallelism
- Allocating multiple channels
- Specifying one `BACKUP AS COPY` command and listing multiple files

RMAN Backup Types

- A full backup contains all used data file blocks.
- A level 0 incremental backup is equivalent to a full backup that has been marked as level 0.
- A cumulative level 1 incremental backup contains only blocks modified since the last level 0 incremental backup.
- A differential level 1 incremental backup contains only blocks modified since the last incremental backup.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

RMAN Backup Types

Full Backups

A full backup is different from a whole database backup. A full data file backup is a backup that includes every used data block in the file. RMAN copies all blocks into the backup set or image copy, skipping only those data file blocks that have never been used. For a full image copy, the entire file contents are reproduced exactly. A full backup cannot be part of an incremental backup strategy; it cannot be the parent for a subsequent incremental backup.

Incremental Backups

An incremental backup is either a level 0 backup, which includes every block in the data files except blocks that have never been used, or a level 1 backup, which includes only those blocks that have been changed since a previous backup was taken. A level 0 incremental backup is physically identical to a full backup. The only difference is that the level 0 backup (as well as an image copy) can be used as the base for a level 1 backup, but a full backup can never be used as the base for a level 1 backup.

Incremental backups are specified using the `INCREMENTAL` keyword of the `BACKUP` command. You specify `INCREMENTAL LEVEL [0 | 1]`.

RMAN Backup Types (continued)

RMAN can create multilevel incremental backups as follows:

- **Differential:** Is the default type of incremental backup that backs up all blocks changed after the most recent incremental backup at either level 1 or level 0
- **Cumulative:** Backs up all blocks changed after the most recent backup at level 0

Examples

- To perform an incremental backup at level 0, use the following command:
`RMAN> BACKUP INCREMENTAL LEVEL 0 DATABASE;`
- To perform a differential incremental backup, use the following command:
`RMAN> BACKUP INCREMENTAL LEVEL 1 DATABASE;`
- To perform a cumulative incremental backup, use the following command:
`RMAN> BACKUP INCREMENTAL LEVEL 1 CUMULATIVE DATABASE;`

RMAN makes full backups by default if neither FULL nor INCREMENTAL is specified. Unused block compression causes never-written blocks to be skipped when backing up data files to backup sets, even for full backups.

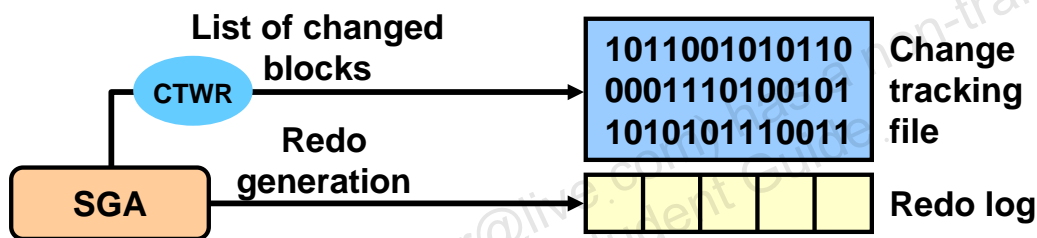
A full backup has no effect on subsequent incremental backups, and is not considered part of any incremental backup strategy, although a full image copy backup can be incrementally updated by applying incremental backups with the RECOVER command. This is covered in the lesson titled “Using RMAN to Perform Recovery.”

Note: It is possible to perform any type of backup (full or incremental) of a database that is in NOARCHIVELOG mode—if, of course, the database is not open. Note also that recovery is limited to the time of the last backup. The database can be recovered to the last committed transaction only when the database is in ARCHIVELOG mode.

Fast Incremental Backup

Implemented by block change tracking, which:

- Maintains a record of what blocks have changed since the last backup
- Writes this record to a file, as redo is generated
- Is automatically accessed when a backup is done, making the backup run faster



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Fast Incremental Backup

The goal of an incremental backup is to back up only those data blocks that have changed since a previous backup. You can use RMAN to create incremental backups of data files, tablespaces, or the whole database. When making an incremental backup, RMAN reads only those blocks referenced to locate the changed blocks since the last backup. That makes the backup smaller because only changed blocks are backed up. It also makes recovery faster because fewer blocks need to be restored.

You can perform fast incremental backup by enabling block change tracking. Block change tracking writes to a file the physical address of each block that is changed. When it is time to perform the incremental backup, RMAN can look at the block change tracking file, and back up only those blocks referenced there; it does not have to scan every block to see if it has changed since the last backup. This makes the incremental backup faster.

The maintenance of the tracking file is fully automatic and does not require your intervention. The size of the block change tracking file is proportional to the:

- Database size, in bytes
- Number of enabled threads in a RAC environment
- Number of old backups maintained by the block change tracking file

The minimum size for the block change tracking file is 10 MB, and any new space is allocated in 10 MB increments. The Oracle database does not record block change information by default.

Enabling Fast Incremental Backup

ORACLE Enterprise Manager 10g Database Control

Setup Preferences Help Logout Database

Database Instance: orcl > Backup Settings

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.

```
ALTER DATABASE
{ENABLE|DISABLE} BLOCK CHANGE TRACKING
[USING FILE '...']
```

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Enabling Fast Incremental Backup

You enable block change tracking from the Database Control home page. Navigate to Availability > Backup Settings > Policy. You do not need to set the block change tracking file destination if the DB_CREATE_FILE_DEST initialization parameter is set because the file is created as an Oracle Managed File (OMF) file in the DB_CREATE_FILE_DEST location. You can, however, specify the name of the block change tracking file, placing it in any location you choose.

You can also enable or disable this feature by using an ALTER DATABASE command. If the change tracking file is stored in the database area with your database files, then it is deleted when you disable change tracking. You can rename the block change tracking file by using the ALTER DATABASE RENAME command. Your database must be in the MOUNT state to rename the tracking file. The ALTER DATABASE RENAME FILE command updates the control file to refer to the new location. You can use the following syntax to change the location of the block change tracking file:

```
ALTER DATABASE RENAME FILE '...' TO '...';
```

Note: RMAN does not support backup and recovery of the block change tracking file. For this reason, you should not place it in the Fast Recovery Area.

Monitoring Block Change Tracking

```
SQL> SELECT filename, status, bytes
2 FROM v$block_change_tracking;
```

```
SQL> SELECT file#, avg(datafile_blocks),
2         avg(blocks_read),
3         avg(blocks_read/datafile_blocks)
4         * 100 AS PCT_READ_FOR_BACKUP,
5         avg(blocks)
5 FROM v$backup_datafile
6 WHERE used_change_tracking = 'YES'
7 AND incremental_level > 0
8 GROUP BY file#;
```

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Monitoring Block Change Tracking

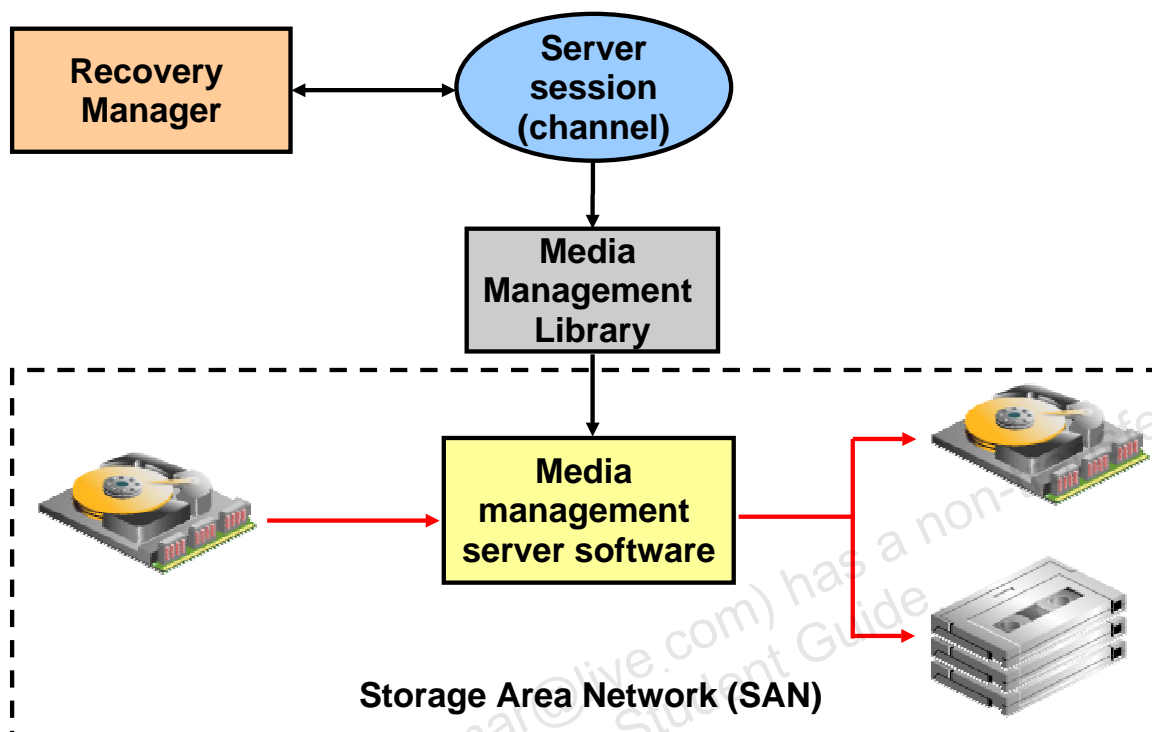
The output of the V\$BLOCK_CHANGE_TRACKING view shows where the block change tracking file is located, the status of block change tracking (ENABLED/DISABLED), and the size (in bytes) of the file.

The query on the V\$BACKUP_DATAFILE view shows how effective the block change tracking is in minimizing the incremental backup I/O (the PCT_READ_FOR_BACKUP column). A high value indicates that RMAN reads most blocks in the data file during an incremental backup. You can reduce this ratio by decreasing the time between the incremental backups.

A sample formatted output from the V\$BACKUP_DATAFILE query is shown below:

FILE#	BLOCKS_IN_FILE	BLOCKS_READ	PCT_READ_FOR_BACKUP	BLOCKS_BACKED_UP
1	56320	4480	7	462
2	3840	2688	70	2408
3	49920	16768	33	4457
4	640	64	10	1
5	19200	256	1	91

Performing Proxy Copies



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Performing Proxy Copies

Use the PROXY option of the RMAN BACKUP command to request a MML to perform the copy of the files.

Syntax:

```
BACKUP [AS BACKUPSET] ... PROXY [ONLY] DATABASE|TABLESPACE....
```

The PROXY ONLY option is useful for those media managers and storage networks where having the backup done by proxy may substantially reduce the storage net traffic.

Some media management products can completely manage all data movement between Oracle data files and the backup devices. Some products that use high-speed connections between storage and media subsystems can reduce much of the backup load from the primary database server. This is beneficial in that the copying takes place across the SAN instead of the LAN. At that point, RMAN is no longer involved in the operation, except for communicating status across the LAN to and from the MML.

Creating Duplexed Backup Sets Using BACKUP COPIES

```
RMAN> BACKUP AS BACKUPSET DEVICE TYPE sbt  
2> COPIES 2  
3> INCREMENTAL LEVEL 0  
4> DATABASE;
```

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Creating Duplexed Backup Sets Using BACKUP COPIES

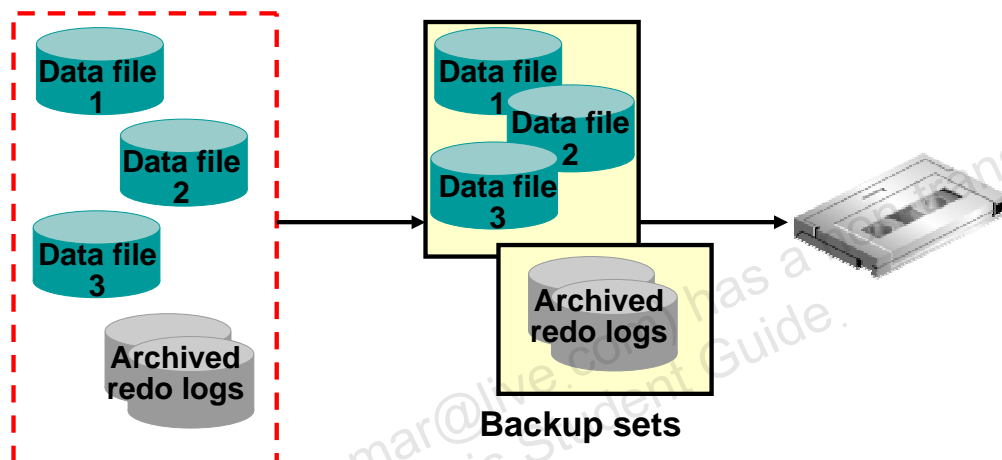
You can use the BACKUP command with the COPIES option to override other COPIES or DUPLEX settings to create duplexed backup sets.

To duplex a backup with BACKUP COPIES, perform the following steps:

1. Specify the number of identical copies with the COPIES option of the BACKUP command.
2. Issue a LIST BACKUP command to verify your backup.

Creating Backups of Backup Sets

```
RMAN> BACKUP DEVICE TYPE DISK AS BACKUPSET  
2> DATABASE PLUS ARCHIVELOG;  
RMAN> BACKUP DEVICE TYPE sbt BACKUPSET ALL;
```



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Creating Backups of Backup Sets

Use the RMAN BACKUP BACKUPSET command to back up previously created backup sets. Only backup sets that were created on device type DISK can be backed up using RMAN. The backup sets can be backed up to any available device type.

The BACKUP BACKUPSET command uses the default disk channel to copy backup sets from disk to disk. To back up from disk to tape, you must either configure or manually allocate a nondisk channel.

Backing Up Read-Only Tablespaces

Considerations for backing up read-only tablespaces:

- Backup optimization causes RMAN to back up read-only tablespaces only when there does not exist a backup that satisfies the retention policy.
- If you change the tablespace to read/write, back it up immediately.
- You can use the `SKIP READONLY` option of the RMAN `BACKUP` command to skip read-only tablespaces or data files.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

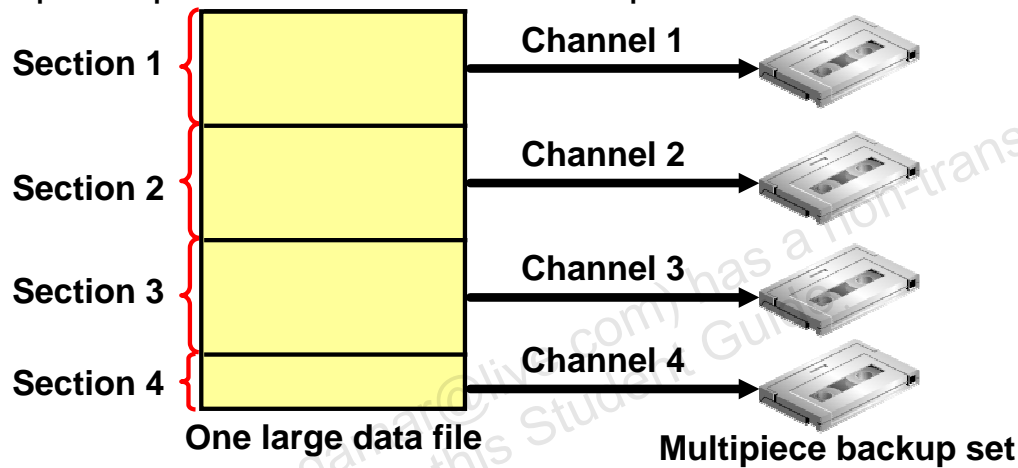
Backing Up Read-Only Tablespaces

Because read-only tablespaces are not being written to, there is no need to continually back them up as you do read/write tablespaces. You can use the `SKIP READONLY` option of the `BACKUP` command to let RMAN know to not back up read-only tablespaces.

Configuring Backup and Restore for Very Large Files

Multisection backups of a single file:

- Are created by RMAN, with your specified size value
- Are processed independently (serially or in parallel)
- Produce multipiece backup sets
- Improve performance of the backup



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Configuring Backup and Restore for Very Large Files

Oracle data files can be up to 128 TB in size. Normally, the smallest unit of an RMAN backup is an entire file. This is not practical with such large files. RMAN can optionally break up large files into sections and back up and restore these sections independently. You do this by creating multisection backups, which break up the files generated for the backup set into separate files. This can be done only with backup sets, not image copies.

Each file section is a contiguous range of blocks of a file. Each file section can be processed independently, either serially or in parallel. Backing up a file into separate sections can improve the performance of the backup operation, and it also allows large file backups to be restarted.

A multisection backup job produces a multipiece backup set. Each piece contains one section of the file. All sections of a multisection backup, except perhaps for the last section, are of the same size. There are a maximum of 256 sections per file.

Note: You should not apply large values of parallelism to back up a large file that resides on a small number of disks, as that would defeat the purpose of the parallel operation; multiple simultaneous accesses to the same disk device would be competing with each other.

This feature is built into RMAN. No installation is required beyond the normal installation of Oracle Database 11g. COMPATIBLE must be set to at least 11.0, because earlier releases cannot restore multisection backups.

Creating RMAN Multisection Backups

RMAN command syntax:

```
BACKUP <options> SECTION SIZE <integer> [K | M | G]
```

```
VALIDATE DATAFILE <options> SECTION SIZE <integer> [K | M | G]
```

Example:

```
RMAN> BACKUP DATAFILE 5 SECTION SIZE = 25M TAG 'section25mb';  
backing up blocks 1 through 3200  
piece handle=/u01/.../o1_mf_nnndf_SECTION25MB_382dryt4_.bkp  
tag=SECTION25MB comment=NONE  
...  
backing up blocks 9601 through 12800  
piece handle=/u01/.../o1_mf_nnndf_SECTION25MB_382dsto8_.bkp  
tag=SECTION25MB comment=NONE
```

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Creating RMAN Multisection Backups

The BACKUP and VALIDATE DATAFILE commands accept the following option:

```
SECTION SIZE <integer> [K | M | G]
```

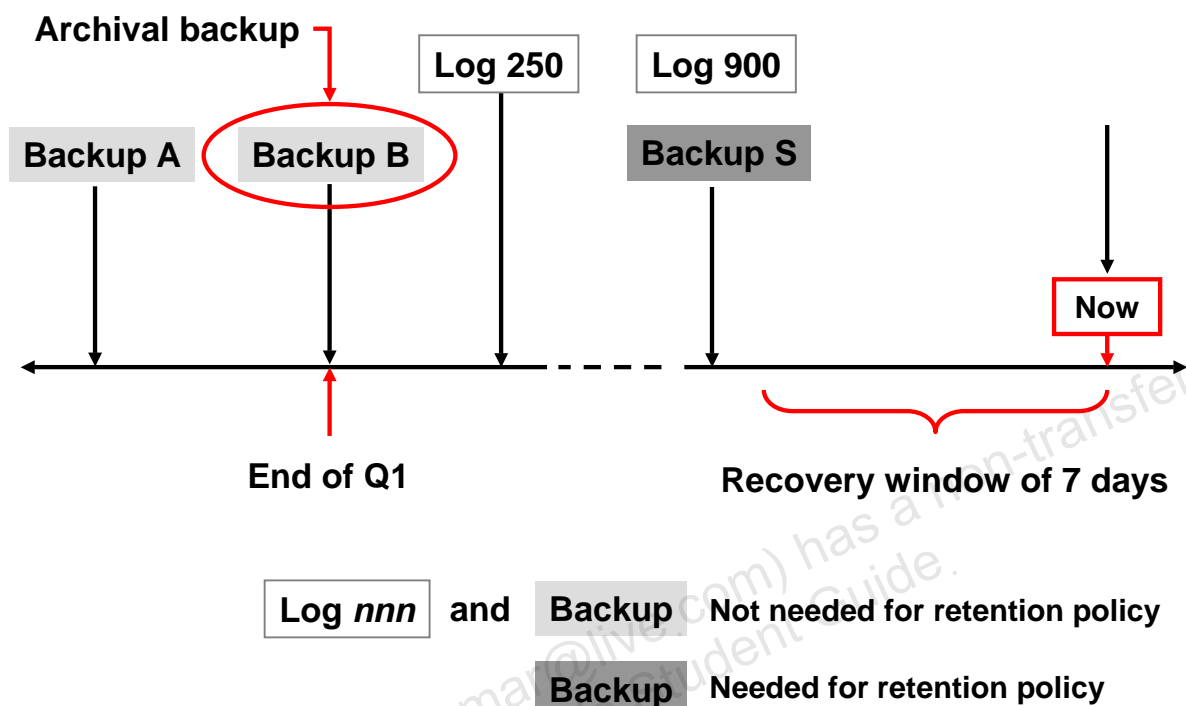
Use this to specify your planned size for each backup section. The option is both a backup command- and backup spec-level option, so that you can apply different section sizes to different files in the same backup job.

In the example in the slide, a backup of data file 5 is being taken, and the section size is specified as 25 MB. The data file is 100 MB in size, so four sections are created. Note that, as indicated by the block ranges, block contiguity is maintained as they are written to the section files.

Viewing Metadata About Your Multisection Backup

- The V\$BACKUP_SET and RC_BACKUP_SET views have a MULTI_SECTION column, which indicates whether this is a multisection backup or not.
- The V\$BACKUP_DATAFILE and RC_BACKUP_DATAFILE views have a SECTION_SIZE column, which specifies the number of blocks in each section of a multisection backup. Zero means a whole-file backup.

Archival Backups: Concepts



Copyright © 2009, Oracle. All rights reserved.

Archival Backups: Concepts

If you need to preserve an online backup for a specified amount of time, RMAN normally assumes you might want to perform point-in-time recovery for any time since that backup to the present. To satisfy this scenario, RMAN keeps the archived logs for that time period. However, you may have a requirement to simply keep the specific backup (and what is necessary to keep it consistent and recoverable) for a specified amount of time—for example, for two years. You do not have the intention of recovering to a point in time since that backup, but you just want to be able to recover to the exact time of the backup, and no later. You also want to maintain a retention policy that keeps your backup area free of clutter, so making it reach back two years is not acceptable. This is a common need, when meeting business or legal requirements for data retention.

An archival backup solves this problem. If you mark a backup as an archival backup, that attribute overrides any configured retention policy for the purpose of this backup. You can retain archival backups such that they are either considered obsolete only after a specific time that you specify, or never considered obsolete. If you want to specify the latter, you need to use a recovery catalog.

The `KEEP` clause creates an archival backup that is a snapshot of the database at a point in time. The only redo logs that are kept are those required to restore this backup to a consistent state. The `RESTORE POINT` clause issued after the backup is completed determines the number of redo logs that are kept (enough to restore the backup to the `RESTORE POINT` time).

Archival Backups: Concepts (continued)

An archival backup also guarantees that all of the files needed to restore the backup are included. RMAN includes the data files, `SPFILE`, archived log files (only those needed to recover an online backup), and the relevant autobackup files. All these files must go to the same media family (or group of tapes).

You can also specify a restore point to be created, which has the same SCN as the archival backup. That essentially gives a meaningful name to the point of time the backup was made.

After an archival backup is created, it is retained for as long as specified. Even if you have a much smaller retention window and run the `DELETE OBSOLETE` command, the archival backup remains.

This backup is a snapshot of the database at a point in time, and can be used to restore the database to another host for testing purposes, for example.

Note: Archival backups cannot be written to the Fast Recovery Area. So if you have one, you must provide a `FORMAT` clause to specify a different location.

Creating Archival Backups with EM

Schedule Customized Backup: Settings

Database **orcl.us.oracle.com** Cancel Back Step 3 of 5 Next

Backup Strategy **Customized Backup**

Object Type **Tablespaces**

These are the settings for your current backup job. You can select your backup destination directly from this page. You can also view the default settings or override the settings by clicking the buttons below.

☒ Disk

Disk Backup Location **/archive_bu**

☐ Tape

Media Management Vendor(MMV) Library Parameters **not specified**

View Default Settings Override Current Settings

Changed settings will only apply to the current backup job.

Override Retention Policy

☐ Do not override the retention policy

☒ Keep this backup for the specified number of days Days **365**

Device Backup Set Policy

Cancel OK

Creating Archival Backups with EM

To create an archival backup using Enterprise Manager, perform the following steps:

1. Select Availability > Schedule Backup > Schedule Customized Backup.
2. Follow the steps of the Schedule Customized Backup wizard until you are on the Settings page.
3. Click **Override Current Settings** and then the **Policy** tab. In the **Override Retention Policy** section, you can select to keep a backup for a specified number of days. A restore point is generated based on the backup job name. You probably also want to specify a different destination for the backup files; to do this, use the **Device** tab.

Backups created with the **KEEP** option include the SPFILE, control files, and archive redo log files required to restore this backup, and data files. This backup is a snapshot of the database at a point in time, and can be used to restore the database to another host.

Creating Archival Backups with RMAN

- Specifying the `KEEP` clause when the database is online includes both data file and archive log backup sets:

```
KEEP {FOREVER | UNTIL TIME [=] ' date_string '}
NOKEEP
[RESTORE POINT rsname]
```

- List all restore points known to the RMAN repository:

```
LIST RESTORE POINT ALL;
```

- Display a specific restore point:

```
LIST RESTORE POINT 'rsname';
```

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Creating Archival Backups with RMAN

Use the following syntax to create an archival backup using RMAN:

```
BACKUP ... KEEP {FOREVER|UNTIL TIME 'SYSDATE + <n>'} RESTORE POINT
<restore_point_name>
```

The `UNTIL TIME` clause enables you to specify when the archival backup is no longer immune to the retention policy. You can optionally specify `FOREVER`, meaning that the backup is an archival backup until you take some other action to change that.

Optionally, use the `RESTORE POINT` clause to specify the name of a restore point to be associated with this backup. The `RESTORE POINT` clause creates a “consistency” point in the control file. It assigns a name to a specific SCN. The SCN is captured just after the data file backup completes. The archival backup can be restored and recovered for this point in time, enabling the database to be opened. In contrast, the `UNTIL TIME` clause specifies the date until which the backup must be kept.

Managing Archival Database Backups

1 Archiving a database backup:

```
RMAN> CONNECT TARGET /  
RMAN> CONNECT CATALOG rman/rman@catdb  
RMAN> CHANGE BACKUP TAG 'consistent_db_bkup'  
2> KEEP FOREVER;
```

2 Changing the status of a database copy:

```
RMAN> CHANGE COPY OF DATABASE CONTROLFILE NOKEEP;
```

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Managing Archival Database Backups

The **CHANGE** command changes the exemption status of a backup or copy in relation to the configured retention policy. For example, you can specify **CHANGE . . . NOKEEP**, to make a backup that is currently exempt from the retention policy eligible for the **OBSOLETE** status.

The first example changes a consistent backup into an archival backup, which you plan to store offsite. Because the database is consistent and, therefore, requires no recovery, you do not need to save archived redo logs with the backup.

The second example specifies that any long-term image copies of data files and control files should lose their exempt status and so become eligible to be obsolete according to the existing retention policy. This statement essentially removes the archival attribute from those backup files. If you do not specify a tag, as in this case, then the **CHANGE** execution applies to all backups of the type specified. You should specify a tag to change only the backup files you intend to change.

Note: The **RESTORE POINT** option is not valid with the **CHANGE** command, because there is no way to create the restore point for a time that has already passed (when the backup was created).

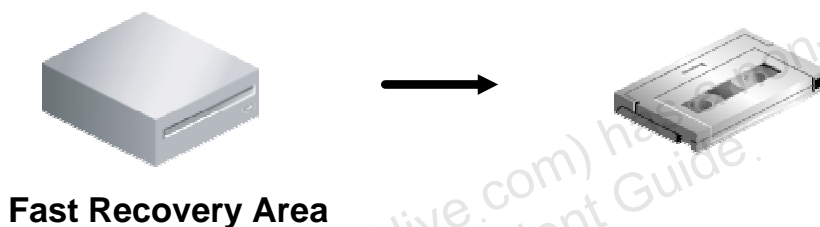
Backing Up Recovery Files

- Back up only the files in the Fast Recovery Area:

```
RMAN> BACKUP RECOVERY AREA
```

- Back up all recovery files:

```
RMAN> BACKUP RECOVERY FILES
```



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Backing Up Recovery Files

There are two ways to back up recovery data. The `BACKUP RECOVERY AREA` command backs up all files that are found in the current and any previous Fast Recovery Areas. The `BACKUP RECOVERY FILES` command backs up all recovery files, even if they are not in the FRA. You gain added protection from loss by using the latter, which would back up, for example, any copies of control files or data files that are not in the Fast Recovery Area.

By default, backup optimization is in effect for these two commands, even if you have disabled it using the `CONFIGURE` command. This means that the only recovery files that this command backs up are those that are not already backed up. You can force all files to be backed up by using the `FORCE` option.

You cannot specify `DEVICE TYPE DISK` for either of these commands.

Note: RMAN backs up only database files: data files, control files, SPFILES, archive log files, and backups of these files. Placing an operating system file in the Fast Recovery Area causes it to be included with a backup of the recovery area.

Managing Backups: Reporting

Use the following RMAN commands to obtain information about your backups:

- **LIST:** Displays information about backup sets, proxy copies, and image copies recorded in the repository
- **REPORT:** Produces a detailed analysis of the repository
- **REPORT NEED BACKUP:** Lists all data files that require a backup
- **REPORT OBSOLETE:** Identifies files that are no longer needed to satisfy backup retention policies

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Managing Backups: Reporting

Use the RMAN **LIST** command to display information about backup sets, proxy copies, and image copies recorded in the repository. Use this command to list:

- Backups and copies that do not have the **AVAILABLE** status in the RMAN repository
- Backups and copies of data files that are available and can possibly be used in a restore operation
- Backup sets and copies that contain a backup of a specified list of data files or specified tablespaces
- Backup sets and copies that contain a backup of any archived logs with a specified name or range
- Backup sets and copies restricted by tag, completion time, recoverability, or device
- Incarnations of a specified database or of all databases known to the repository
- Stored scripts in the recovery catalog

Use the RMAN **REPORT** command to analyze information in the RMAN repository in more detail.

The **REPORT NEED BACKUP** command is used to identify all data files that need a backup. The report assumes that the most recent backup would be used in the event of a restore.

Managing Backups: Reporting (continued)

Using the `REPORT OBSOLETE` command, you can identify files that are no longer needed to satisfy backup retention policies. By default, the `REPORT OBSOLETE` command reports which files are obsolete under the currently configured retention policy. You can generate reports of files that are obsolete according to different retention policies by using `REDUNDANCY` or `RECOVERY WINDOW` retention policy options with the `REPORT OBSOLETE` command.

Refer to the *Oracle Database Backup and Recovery Reference* for detailed syntax information.

Managing Backups: Dynamic Performance Views

Query the following dynamic performance views in the target database to obtain information about your backups:

- V\$BACKUP_SET: Backup sets created
- V\$BACKUP_PIECE: Backup pieces that exist
- V\$DATAFILE_COPY: Copies of data files on disk
- V\$BACKUP_FILES: Information about all files created when creating backups

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Managing Backups: Dynamic Performance Views

There are many views that provide backup-related information. The most commonly used ones are shown in the slide.

If you are using a recovery catalog, you can query corresponding views that contain the same information for each target database registered in the recovery catalog database. The corresponding views have the same name, except that the “V\$” is replaced with “RC_”. Also, they are in the schema owned by the recovery catalog owner. For example, the corresponding views in the recovery catalog, showing the information shown in the slide are: RC_BACKUP_SET, RC_BACKUP_PIECE, RC_DATAFILE_COPY, and RC_BACKUP_FILES.

In order to query the RC_BACKUP_FILES view, you must first execute the following in the recovery catalog database:

```
SQL> CALL DBMS_RCVMAN.SETDATABASE(null,null,null,<dbid>);
```

where <dbid> is the database ID of a target database.

Using Enterprise Manager to View Backup Reports

Results								
Input Summary								
Datafile			Control File			SPFile		
Files Backed Up	15		Files Backed Up	3		Files Backed Up	3	
Distinct Files	5		Distinct Files	1		Distinct Files	2	
Distinct Tablespaces	5		Total Size	28.92M		Total Size	0.01K	
Total Size	5.26G		Oldest Checkpoint Time	Aug 14, 2007 8:04:16 PM		Oldest Modification Time	Aug 14, 2007 7:54:20 PM	
Oldest Checkpoint Time	Aug 14, 2007 9:19:52 PM		Newest Checkpoint Time	Aug 14, 2007 9:21:59 PM		Newest Modification Time	Aug 14, 2007 9:01:45 PM	
Newest Checkpoint Time	Aug 14, 2007 9:21:52 PM							

View Backup Report

The following backup jobs are known to the database. The data is retrieved from the database control file.

Search

Status: Start Time: Type:

Results

Total 19 (Completed 15 Failed 4)

Backup Name	Status	Start Time	Time Taken	Type	Output Devices	Input Size	Output Size	Output Rate (Per Sec)
2007-08-15T00:18:04	COMPLETED	Aug 15, 2007 12:18:10 AM GMT+07:00	00:01:12	DATAFILE FULL	DISK	1.14M	304.00K	4.22K
2007-08-14T19:55:28	COMPLETED	Aug 14, 2007 7:56:20 PM GMT+07:00	01:25:50	DB INCR	DISK	1.86G	1.80G	367.41K
2007-08-14T02:52:20	COMPLETED	Aug 14, 2007 2:52:22 AM GMT+07:00	00:00:09	CONTROLFILE	SBT_TAPE	9.63M	0.00K	0.00K

Manage

- [Schedule Backup](#)
- [Manage Current Backups](#)
- [Backup Reports](#)
- [Manage Restore Points](#)
- [Perform Recovery](#)
- [View and Manage Transactions](#)

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Using Enterprise Manager to View Backup Reports

You can use the Backup Report page to display lists of backup jobs that are known to the database through the information recorded about them in the database control file.

You can customize the jobs that appear in the Results table by using the Search fields at the top of the page. The Results table lists basic information about each backup job, such as the Start Time, the Time Taken, and the Status of the backup job. You can also use the Results table to drill down to individual, detailed backup job reports by using the link in the Backup Name column.

You can drill down to a Summary of Job page of the backup job by clicking the Status of the job in the Results table, where you can view the contents of the output log.

Click the Backup Name link, and you can use the Backup Report page to display detailed information about that backup. The information displayed on this page is derived from the information recorded in the database control file.

The Backup Report page displays result information in the Result section in various categories, such as Input Summary, containing rollup information about the files that were backed up; Output Summary, containing rollup information about the Backup Sets and Image Copies; and then Inputs and Outputs sections that display tables containing detailed job information about the data files, control files, backup sets, backup pieces, and image copies.

Managing Backups: Cross-Checking and Deleting

Use the following RMAN commands to manage your backups:

- **CROSSCHECK:** Verifies the status of backups and copies recorded in the RMAN repository against media such as disk or tape
- **DELETE EXPIRED:** Removes only files whose status in the repository is `EXPIRED`
- **DELETE OBSOLETE:** Deletes backups that are no longer needed

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Managing Backups: Cross-Checking and Deleting

Use the `CROSSCHECK` command to ensure that data about backups in the recovery catalog or control file is synchronized with actual files on disk or in the media management catalog. The `CROSSCHECK` command operates only on files that are recorded in the RMAN repository.

The `CROSSCHECK` command checks only objects marked `AVAILABLE` or `EXPIRED` by examining the files on disk for `DISK` channels or by querying the media manager for `sbt` channels. The `CROSSCHECK` command updates the repository records for any files that it is unable to find to `EXPIRED`. It does not delete any files that it is unable to find.

The `DELETE` command can remove any file that the `LIST` and `CROSSCHECK` commands can operate on. For example, you can delete backup sets, archived redo logs, and data file copies. The `DELETE` command removes both the physical file and the catalog record for the file. The `DELETE OBSOLETE` command deletes backups that are no longer needed. It uses the same `REDUNDANCY` and `RECOVERY WINDOW` options as `REPORT OBSOLETE`.

If you delete backups without using RMAN, you can use the `UNCATALOG` command to remove the files from the recovery catalog, or you can use the `CROSSCHECK` and `DELETE EXPIRED` commands.

Refer to the *Oracle Database Backup and Recovery Reference* for detailed syntax information.

Quiz

A full database backup can be used as the basis for incremental backups.

1. True
2. False

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Answer: 2

Quiz

RMAN can always take a backup when the database is closed.

1. True
2. False

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Answer: 1

Summary

In this lesson, you should have learned how to:

- Create image file backups
- Create a whole database backup
- Create a full database backup
- Enable fast incremental backup
- Create duplex backup sets
- Back up a backup set
- Create RMAN multi-section backup
- Create an archival backup for long-term retention
- Report on and maintain backups

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Practice 5 Overview: Creating Backups

This practice covers the following topics:

- Taking an archival backup
- Enabling block change tracking
- Recovering from a damaged block
- Reporting on existing backups
- Backing up the control file

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Usman Qamar (usman.qamar@live.com) has a non-transferable
license to use this Student Guide.