



S K I L L B U I L D E R S

Oracle10g New Features By Example

V2.1

Oracle10g New Features By Example

SkillBuilders, Inc.
24 Salt Pond Road, Unit C-4
South Kingstown, RI 02880
www.skillbuilders.com
knowledge@skillbuilders.com

Author: David G. Anderson

Legal Notice

Copyrights

Copyright © 2005-2006 SkillBuilders, Inc. All rights reserved. Printed in the United States of America.

No part of this publication may be reproduced, distributed or displayed in any form or by any means, or stored in a database, retrieval system or other media without the prior written permission of SkillBuilders, Inc.

Trademarks

The product/service names mentioned herein are manufacturer/publisher trademarks and are used only for purposes of identification.

Microsoft, Microsoft Windows 95, Microsoft Windows 98, Microsoft Windows NT, and Microsoft Windows are registered trademarks of Microsoft Corporation. Netscape, Netscape Navigator, and the Netscape logo are registered trademarks of Netscape Communications Corporation in the United States and other countries. Oracle, Oracle8i, Oracle9i and Oracle10g are registered trademarks of Oracle Corporation.

All other brand, product and service names are the trademarks of their respective companies or owners as mentioned herein.

Disclaimer

Information within this publication is subject to change without notice. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. The author and SkillBuilders, Inc. make no representation or warranty with respect to the contents herein and specifically disclaims any implied warranties of fitness for any particular purpose. SkillBuilders, Inc. disclaims all liability for any direct, indirect, incidental or consequential, special or exemplary damages resulting from the use of the information in this document or from the use of any products described in this document.

1. Backup and Recovery Enhancements 1.1

Lesson Objectives	1.2
Flash Recovery Area: Concepts.....	1.3
Flash Recovery Area: Benefits	1.4
Flash Recovery Area: Configuration	1.6
Flash Recovery Area: Monitoring	1.7
Flashback Database.....	1.9
Flashback Logs	1.11
Configure Database.....	1.13
Flashback Database: Example	1.14
Flashback Database TO BEFORE RESETLOGS	1.15
Monitoring Flashback Database	1.16
Restore Points	1.18
Simplified Recovery Through RESETLOGS (SRR).....	1.21
SRR Architectural Changes	1.22
Compressed Backups.....	1.25
Change Tracking.....	1.26
Incrementally Updated Image Copies.....	1.28
SWITCH DATABASE Command	1.30
Encrypted Backups	1.32
RMAN DROP DATABASE	1.33
RMAN CATALOG Command.....	1.34
DURATION Parameter	1.35
New V\$ Views.....	1.36
More RMAN Enhancements	1.37
Additional R2 RMAN Enhancements	1.39
User-Managed Enhancements	1.41
Oracle Secure Backup.....	1.42
Lesson Summary.....	1.43

2. Flashback Features 2.1

Lesson Objectives	2.2
Flashback Evolution	2.3
Flashback Setup: Undo	2.4
Flashback Setup: LOBs	2.6
Flashback Version Query.....	2.7
Flashback Version Query Example	2.9
Flashback Version Query Privileges.....	2.11
FLASHBACK_TRANSACTION_QUERY View	2.12
Flashback Table Concepts	2.14
Flashback Table Prep.....	2.15
Flashback Table: Undo Application Changes	2.16

Flashback Table Notes.....	2.17
Recycle Bin.....	2.19
LIFO to the Recycle Bin.....	2.21
Purging Recycle Bin	2.22
Flashback Drop.....	2.23
Flashback Drop Implications.....	2.24
Flashback Drop Limitations.....	2.25
Lesson Summary.....	2.26

3. Automatic Storage Management 3.1

Lesson Objectives.....	3.2
Introducing ASM.....	3.3
Automated S.A.M.E.....	3.4
Highlights.....	3.5
ASM Instance Architecture	3.7
ASM Install Tips.....	3.9
Starting ASM.....	3.11
ASM Disks.....	3.12
ASM Diskgroups	3.13
ASM Failgroups.....	3.14
Creating Diskgroup	3.15
Altering Diskgroup.....	3.16
Referencing Disk Group.....	3.18
ASMCMD.....	3.20
More R2 Enhancements.....	3.21
Lesson Summary.....	3.22
Resources	3.24

4. SQL Enhancements 4.1

Lesson Objectives... ..	4.2
DML Error Logging.....	4.4
Error Logging Example	4.5
Regular Expressions... ..	4.6
REGEXP_LIKE.....	4.8
Common Metasymbols	4.9
Perl Expressions.....	4.10
Matching Sets and Repeaters.....	4.11
Backreferences	4.12
Summary: Regular Expressions.....	4.14
Case-Insensitive Sort... ..	4.16
Case-Insensitive Search... ..	4.20
Accent-Insensitive Sort and Search	4.22
Effect on Index Use... ..	4.23

Text Literal Quote Character	4.25
MERGE Enhancements (1)	4.26
MERGE Example (1).....	4.27
MERGE Enhancements (2)	4.28
MERGE Example (2).....	4.29
Partition Outer Join.....	4.30
MODEL Clause Concepts	4.33
MODEL Components.....	4.34
MODEL: Example 1.....	4.35
MODEL: Example 2.....	4.37
ORA_ROWSCN.....	4.38
ORA_ROWSCN Use.....	4.40
New Datatypes.....	4.41
LOB Enhancements	4.42
New Functions	4.43
RETURNING Aggregates	4.44
CONNECT_BY_ROOT	4.45
CONNECT BY: NOCYCLE.....	4.46
CONNECT BY: Pseudocolumns.....	4.47
Asynchronous COMMIT.....	4.48
Miscellaneous SQL Features	4.49
Introduction to XQuery.....	4.50
Oracle XQuery Support	4.51
XQuery Examples.....	4.52
Lesson Summary.....	4.53

5. Job Scheduler	5.1
-------------------------------	------------

Lesson Objectives.....	5.2
Introduction.....	5.3
Scheduler Components.....	5.5
Job Scheduling.....	5.7
Control Resources.....	5.8
Monitoring Jobs.....	5.9
More Features	5.11
Example: Create Schedule.....	5.12
Linux: Create Job.....	5.13
Windows: Create Job.....	5.14
Linux: OS Script.....	5.15
Windows: OS BAT File.....	5.16
DBA_SCHEDULER_JOBS	5.17
DBA_SCHEDULER_SCHEDULES	5.18
Other Dictionary Views.....	5.19
Miscellaneous	5.20
Scheduler Privileges	5.21

Lesson Summary.....	5.22
Resources	5.23

6. Performance and Tuning I: General Enhancements..... 6.1

Lesson Objectives	6.2
Automatic Shared Memory Management	6.3
Auto SGA Demo.....	6.6
V\$SESSION Enhancements.....	6.8
Finding Blockers.....	6.10
Automatic Optimizer Statistics.....	6.12
DBMS_STATS.....	6.14
Rule-Based Optimization.....	6.15
CBO: Hash Group By.....	6.16
CBO: Costed Transformations.....	6.18
New and Changed Hints	6.19
Index Skip Scan.....	6.20
SQL*Plus AUTOTRACE.....	6.22
PL/SQL Performance.....	6.23
End-to-End Tracing	6.24
Other Trace Enhancements.....	6.26
Self Tuning Checkpointing.....	6.28
Miscellaneous Features.....	6.29
Other R2 Enhancements	6.31
Lesson Summary.....	6.32

7. Performance and Tuning II New Performance Statistics..... 7.1

Lesson Objectives	7.2
Management Packs.....	7.3
... Management Packs	7.4
New Performance Data.....	7.5
Metrics	7.6
Metric Thresholds.....	7.7
Server-Generated Alerts	7.8
Time Model Statistics	7.9
V\$SESS_TIME_MODEL	7.10
Database Time	7.11
Active Session History: ASH	7.12
ASH Example: CPU	7.13
ASH Example: Blockers.....	7.14
Wait Classes.....	7.16
V\$SESSION_WAIT_CLASS	7.18
V\$SESSION_WAIT_HISTORY.....	7.19
V\$EVENT_HISTOGRAM.....	7.20

V\$FILE_HISTOGRAM	7.21
Lesson Summary.....	7.22

8. Performance and Tuning III: AWR & Advisors 8.1

Lesson Objectives	8.2
Management Packs	8.3
Introduction to AWR	8.5
AWR Snapshots	8.7
Controlling AWR.....	8.8
AWR Views.....	8.9
AWR Reporting	8.11
Tuning Advisors	8.12
Introduction to ADDM	8.14
ADDM Report Excerpt.....	8.16
ADDM Findings	8.17
Enabling ADDM.....	8.18
SQL Tuning Advisor	8.21
STA Interfaces	8.22
Input to SQL Tuning Advisor.....	8.23
Running STA	8.24
STA Demo	8.25
SQL Profiles	8.28
SQL Profile Demo (1).....	8.29
SQL Profile Demo (2).....	8.30
SQL Profile Demo (3).....	8.31
Other AWR-Based Advisors	8.32
Other Advisors.....	8.33
Lesson Summary.....	8.34

9. 10g PL/SQL Enhancements 9.1

Lesson Objectives	9.2
Performance: Compiler.....	9.3
Optimization Comparison.....	9.7
Compiler Warnings.....	9.9
Native Compilation Review.....	9.15
Native Compilation: New Stuff	9.16
Conditional Compilation.....	9.17
FORMAT_ERROR_BACKTRACE	9.19
GET_CPU_TIME	9.20
FORALL Improvements.....	9.21
Regular Expressions	9.24
Quoting Mechanism.....	9.26
Introducing UTL_MAIL.....	9.27

UTL_MAIL SET UP	9.28
UTL_MAIL EXAMPLE.....	9.29
Nested Table Enhancements.....	9.30
New and Revised Packages	9.33
Lesson Summary.....	9.34

10. Security Enhancements 10.1

Lesson Objectives	10.2
CONNECT Role	10.3
FGAC Review.....	10.5
VPD Policy Types Review	10.7
10g VPD Policy Types	10.8
Column-Level VPD	10.9
10g VPD Example	10.10
Fine Grained Auditing Review	10.12
Review: Create FGA Policy	10.13
Review: Display Audited Activity.....	10.14
10g Auditing Enhancements.....	10.15
DBMS_CRYPTO	10.16
Transparent Data Encryption	10.17
Lesson Summary.....	10.20

11. SQL*Plus Enhancements 11.1

Lesson Objectives	11.2
Predefined Variables.....	11.3
Recycle Bin	11.4
SPOOL Enhancements	11.5
SQL*Plus AUTOTRACE.....	11.6
Miscellaneous	11.7
Lesson Summary.....	11.8

12. Tablespace, Table and Segment Management..... 12.1

Lesson Objectives	12.2
SYSAUX	12.3
BIGFILE Tablespace	12.5
SMALLFILE Tablespace	12.6
Default Tablespace	12.7
Rename Tablespace	12.8
Temp Tablespace Groups	12.9
Transportable Tablespace	12.11
Predicting Object Growth.....	12.12
Online Table Redefinition	12.13

Online Segment Shrink	12.14
Shrink: 9i versus 10g	12.15
How to Shrink	12.16
Lesson Summary	12.17

13. Utility Enhancements 13.1

Lesson Objectives	13.2
Introducing Data Pump	13.3
Server-Based Utility	13.4
Major Differences	13.5
Data Pump Modes	13.7
Data Pump Access Methods	13.8
Export Example	13.9
Import Example	13.12
Monitoring Jobs	13.13
Filtering	13.14
Remap & Transformation	13.15
Flashback Support	13.16
expdp Features	13.17
impdp Features	13.18
Network Export / Import	13.19
Network Import Example	13.20
Transporting Tablespaces	13.21
TTS Example	13.22
DBMS_FILE_TRANSFER	13.23
DBMS_DATAPUMP	13.25
Data Pump Dictionary	13.26
Enterprise Manager	13.27
Datapump Resources	13.28
External Tables	13.29
External Tables	13.30
External Table Resources	13.33
Logminer Enhancements	13.34
EXP / IMP Transport Tablespace	13.35
Lesson Summary	13.36

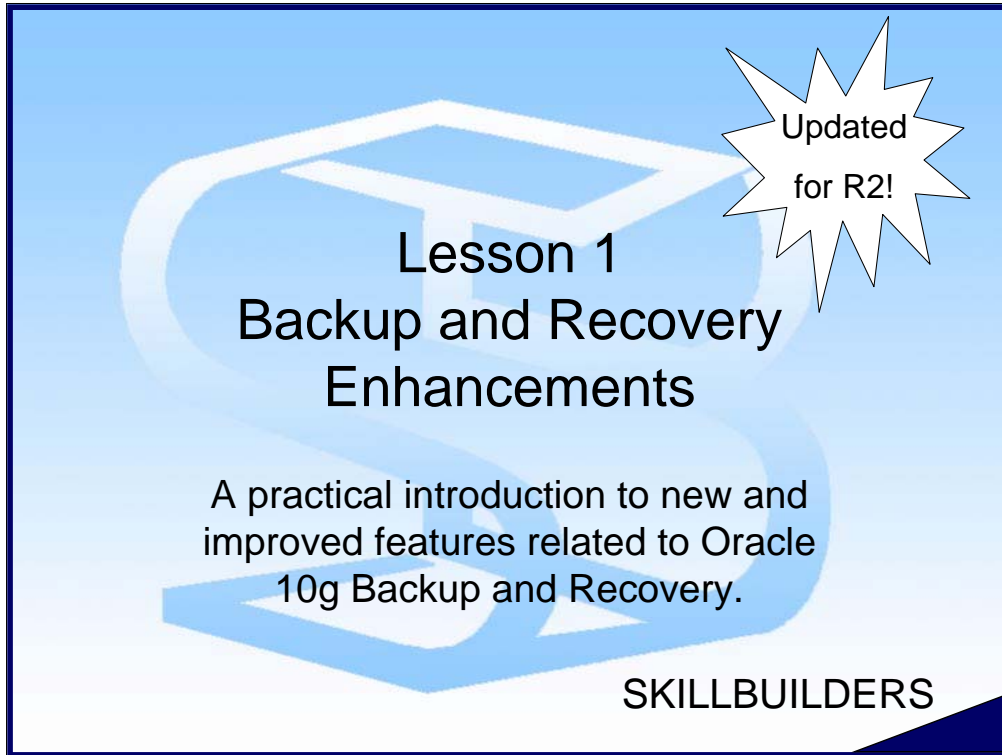
14. Enterprise Manager Database Control 14.1

Lesson Objectives	14.2
Introduction to EM DB Control	14.3
Database Control vs. Grid Control	14.5
Management Packs	14.7
Configuring Database Control	14.9
Starting dbconsole Process	14.12

Starting EM / EM Administrators.....	14.13
EM Privileges	14.14
Database Home Page	14.15
OEM Notification System	14.16
Metrics and Thresholds.....	14.18
Notification: Alerts	14.19
Notification: Methods	14.20
Notification: Email Address... ..	14.21
Notification System Bug.....	14.23
Notification: Rules	14.24
Rules: Properties	14.25
Rules: Availability	14.26
Rules: Metrics	14.27
Rules: Objects	14.28
Rules: Methods	14.29
Rules: Summary.....	14.30
Notification: Schedule	14.31
Notification: Email Received.....	14.32
R2 Features	14.33
Lesson Summary.....	14.34

15. Miscellaneous Enhancements 15.1

Lesson Objectives.....	15.2
Guaranteed Retention	15.3
Rollback Monitoring.....	15.4
Easy Connect Connections	15.5
High Watermark Statistics	15.6
Database Feature Usage.....	15.7
Deprecated Parameters	15.8
Release 1 Parameters	15.9
Release 2 Parameters	15.10
Lesson Summary.....	15.11

A blue-themed slide with a white graduation cap icon in the background. The text is centered and includes a starburst graphic in the top right corner. The starburst contains the text "Updated for R2!".

Lesson 1
Backup and Recovery
Enhancements

A practical introduction to new and improved features related to Oracle 10g Backup and Recovery.

SKILLBUILDERS

Author: Dave Anderson

Release Date: October, 2004

Last Revision: July, 2006




1.2

Lesson Objectives

- Flash Recovery Area
- Flashback Database
- Restore Points
- Simplified Recovery through RESETLOGS
- Compressed Backups
- Change Tracking
- Incrementally Updated Image Copies
- SWITCH DATABASE
- New V\$ Views
- Encrypted Backups
- CATALOG Command
- DROP DATABASE Command
- Miscellaneous New Features

© 2005-2006 SkillBuilders, Inc.



Flash Recovery Area: Concepts

1.3

- Storage area for all recovery-related files
 - Redo logs , Archive logs
 - Control file backups
 - RMAN backups
 - Flashback logs
- Optional, but Oracle recommends
 - Required for Flashback Database and Guaranteed Restore Points
- Use file system or ASM
- Put on separate disks from datafile and log disks

© 2005-2006 SkillBuilders, Inc.

The Flash Recovery Area, introduced with Oracle 10g Release 1, is an optional (but recommended) storage area for many types of recovery-related files. In addition to redo logs, archive logs, RMAN backupsets, this includes flashback logs, required for the FLASHBACK DATABASE statement. You will learn more about flashback logs and the FLASHBACK DATABASE statement later in this course.

The flash recovery area is optional. However, it is required for the 10g features Flashback Database and Guaranteed Restore Points, discussed in more detail later in this lesson.

The flash recovery area can be created on a OS file system or on Automated Storage Management (ASM) disks. ASM is the new integrated file and volume management system included in all editions of Oracle 10g. Refer to the *Automated Storage Management* lesson in this course for more information on ASM.

Whatever type of file system chosen, DBAs will want to create the flash recovery area on separate disks from the datafile and redo log disks. This will prevent the loss of both the primary database files and the backups (and flashback logs) if a disk fails.

The slide features a blue background with a white border. In the top left corner is a stylized 'S' logo. The title 'Flash Recovery Area: Benefits' is centered at the top. Below the title is a bulleted list of benefits. At the bottom of the slide, there are three RMAN command examples in a terminal window style, with callout boxes explaining their equivalence and scope. A copyright notice is at the very bottom.

Flash Recovery Area: Benefits

- Simplifies management
 - Auto-delete files when space needed
 - Obsolete backup sets, archive logs
 - Files backed up to tape
 - Issues warnings when space constrained
- Easy to backup all recovery-related files to tape

```
RMAN> BACKUP RECOVERY AREA;
```

Equivalent

```
RMAN> BACKUP DB_RECOVERY_FILE_DEST;
```

```
RMAN> BACKUP RECOVERY FILES;
```

Backup all recovery files to tape, even if not in flash recovery area

© 2005-2006 SkillBuilders, Inc.

The flash recovery area can simplify database management. For example, the database automatically deletes obsolete logs and backup sets when space is required, removing the need for the DBA to manage (e.g. clean-up) the disks containing these files. Recovery-related files become obsolete when they exceed the RMAN retention policy or have been backed up to tape. Refer to the next page to see the alert log error message issued if the flash recovery area begins to run out of space.

Another new RMAN command provides a convenient technique for backing up the recovery area to tape (tape is the mandatory destination for this command). This is accomplished with the "BACKUP RECOVERY AREA" command (BACKUP DB_RECOVERY_FILE_DEST is a synonym for BACKUP RECOVERY AREA). Files in the flash recovery area that are backed up to tape are retained on disk but are eligible for deletion if space becomes constrained. Refer to the BACKUP command in the **Oracle® Database Recovery Manager Reference 10g (Release 2)** for more information.

Oracle 10g RMAN also introduces the BACKUP RECOVERY FILES command. This command backs up all disk-based recovery files to tape, whether they are in the flash recovery area or not.

Notes for this slide continue on the next page...

This is the alert log entry if you run short of space in the flash recovery area:

```
Errors in file /u01/app/oracle/admin/orcl/udump/orcl_ora_2723.trc:
```


```
ORA-19815: WARNING: db_recovery_file_dest_size of 2147483648 bytes is  
92.12% used, and has 169270784 remaining bytes available.
```

```
*****
```

You have the following choices to free up space from
flash recovery area:

1. Consider changing your RMAN retention policy.
If you are using dataguard, then consider changing your RMAN archivelog
deletion policy.
2. Backup files to tertiary device such as tape using the
RMAN command BACKUP RECOVERY AREA.
3. Add disk space and increase the db_recovery_file_
dest_size parameter to reflect the new space.
4. Delete unnecessary files using the RMAN DELETE command. If an OS
command was used to delete files, then use RMAN CROSSCHECK and RMAN
DELETE EXPIRED commands.

```
*****
```



1.6

Flash Recovery Area: Configuration

- Set location
 - Separate from datafiles
- Set space limit
 - Maximum space dedicated to flashback area
 - database size + size of incremental backups + size of all archive logs not backed up to tape
- If Flashback Database desired, set retention limit
 - How far back can we flashback database?

NAME	VALUE
db_recovery_file_dest	/mnt/mickeymantle
db_recovery_file_dest_size	10G
db_flashback_retention_target	1440

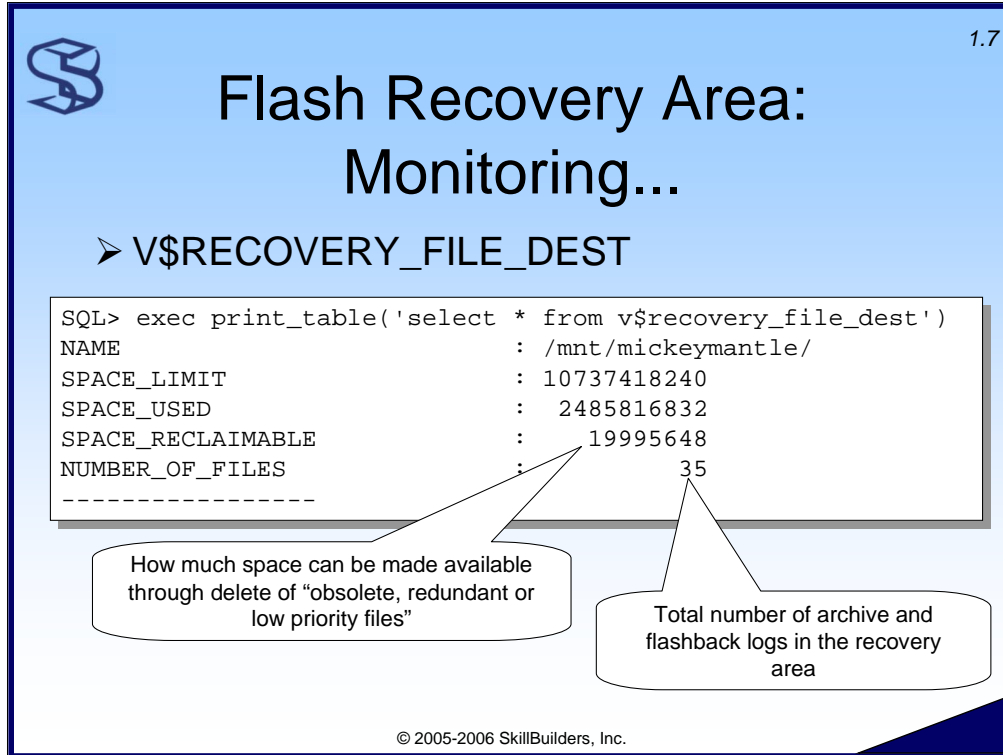
© 2005-2006 SkillBuilders, Inc.

Set the location, space limit and retention limit (in minutes) for flashback logs (if flashback database capability is desired) when establishing the flash recovery area. Since this is a recovery-related area, *keep this on separate disk devices from the database files.*

Refer to Section 3.2 of the **Oracle Database Backup and Recovery Basics 10g Release 2** manual and Chapter 5 of the **Oracle Database Backup and Recovery Advanced User's Guide 10g Release 2** for more information on configuring the Flash Recovery Area.

Supplemental Notes

1. The Database Configuration Assistant supports the configuration of the Flash Recovery Area.
2. The LOG_ARCHIVE_START parameter is deprecated in Oracle10g.



The slide features a blue background with a white border. In the top left corner is a stylized 'S' logo. The title 'Flash Recovery Area: Monitoring...' is centered at the top. Below the title is a right-pointing arrow followed by the text 'V\$RECOVERY_FILE_DEST'. A white box contains a SQL query and its output. Two callout boxes point to specific values in the output: one points to 'SPACE_RECLAIMABLE' and the other points to 'NUMBER_OF_FILES'. The copyright notice '© 2005-2006 SkillBuilders, Inc.' is at the bottom center.

1.7

Flash Recovery Area: Monitoring...

➤ V\$RECOVERY_FILE_DEST


```
SQL> exec print_table('select * from v$recovery_file_dest')
NAME                               : /mnt/mickeymantle/
SPACE_LIMIT                         : 10737418240
SPACE_USED                          : 2485816832
SPACE_RECLAIMABLE                   : 19995648
NUMBER_OF_FILES                     : 35
-----
```

How much space can be made available through delete of "obsolete, redundant or low priority files"


Total number of archive and flashback logs in the recovery area

© 2005-2006 SkillBuilders, Inc.

A new view, V\$RECOVERY_FILE_DEST, is available to describe the current flash recovery area. Here we can quickly find space allocated, used and "reclaimable". Reclaimable space can be made available via the database deleting "obsolete, redundant or low priority files." Files become obsolete or redundant when they exceed the RMAN retention policy or have been backed up to tape.



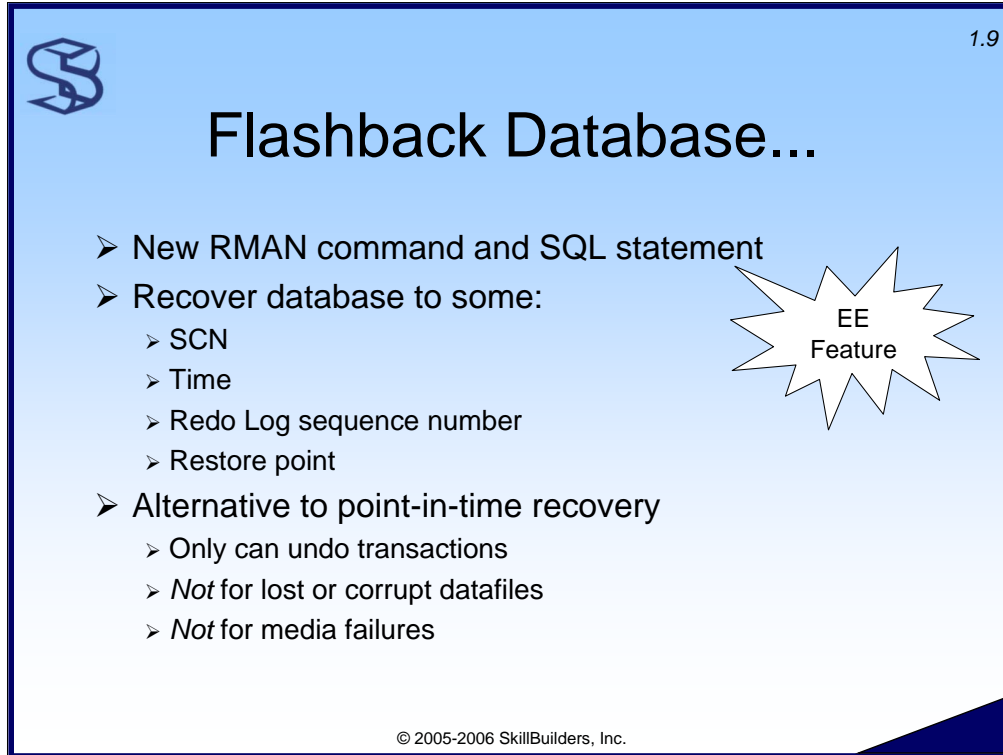
...Flash Recovery Area: Monitoring

➤ V\$FLASH_RECOVERY_AREA_USAGE 

```
SQL> exec system.print_table -
> ('select * from v$flash_recovery_area_usage')
FILE_TYPE                : ARCHIVELOG
PERCENT_SPACE_USED       : 6.02
PERCENT_SPACE_RECLAIMABLE : 0
NUMBER_OF_FILES          : 5
-----
FILE_TYPE                : BACKUPPIECE
PERCENT_SPACE_USED       : .21
PERCENT_SPACE_RECLAIMABLE : .1
NUMBER_OF_FILES          : 2
-----
```

© 2005-2006 SkillBuilders, Inc.

Oracle10g Release 2 provides another dynamic performance view, V\$FLASH_RECOVERY_AREA_USAGE. This view records space used by file type, where the file type can be CONTROLFILE, ONLINELOG, ARCHIVELOG, BACKUPPIECE, IMAGECOPY, FLASHBACKLOG. The PERCENT_SPACE_USED is the percentage of the total space as reported by V\$RECOVERY_FILE_DEST.SPACE_USED.



The slide features a blue background with a dark blue border. In the top-left corner is the SkillBuilders logo, a stylized 'S' with a 'B' inside. In the top-right corner is the number '1.9'. The main title 'Flashback Database...' is centered in a large, bold, black font. Below the title is a bulleted list of features. To the right of the list is a white starburst graphic containing the text 'EE Feature'. At the bottom center of the slide is the copyright notice '© 2005-2006 SkillBuilders, Inc.'.

Flashback Database...


- New RMAN command and SQL statement
- Recover database to some:
 - SCN
 - Time
 - Redo Log sequence number
 - Restore point
- Alternative to point-in-time recovery
 - Only can undo transactions
 - *Not* for lost or corrupt datafiles
 - *Not* for media failures

EE Feature

© 2005-2006 SkillBuilders, Inc.

Oracle 10g Release 1 (Enterprise Edition only) provides a new and easy alternative to using RMAN (or other methods) to perform a point-in-time (incomplete) recovery: Flashback Database. The FLASHBACK DATABASE statement is implemented as an SQL statement and an RMAN command. (The RMAN command provides more functionality, such as the ability to flashback database to a log sequence number.)

Flashback database is an alternative to point-in-time (incomplete) recovery. However, it cannot be used in cases of corrupt datafiles or media failure. The database must be in good working order for the FLASHBACK DATABASE statement to work.

1.10

...Flashback Database


- Versus traditional PIT recovery
 - Easy to use
 - Faster
 - Restoring files unnecessary
 - Can easily re-flashback more than once
- FLASHBACK DATABASE statement uses
 - Flashback logs
 - Archived redo logs

© 2005-2006 SkillBuilders, Inc.

The benefits (versus traditional point-in-time recovery) include:

- As I will demonstrate later in this lesson, it is easy to use.
- Restoring datafiles from backup sets or image copies is unnecessary. Therefore, recovery is faster than traditional point-in-time recovery.
- After you have flashed back the database you can query the database, confirm you like what you see, then use OPEN RESETLOGS to make the database generally available. If you do not like the point in time you have flashed back to, simply flash back again.

The flashback database operation uses a combination of flashback logs and archived redo logs to restore the database to the point requested. Next, we'll learn more about flashback logs.



1.11

Flashback Logs

- FLASHBACK DATABASE uses flashback logs
- Logs contain changed blocks
- Automatic and ongoing creation of new logs
 - Tune for best performance
 - Undocumented parameters help?
- Written to Flash Recovery Area
- Transactions dictate frequency / size of logs
- Automatic deletion of obsolete logs

© 2005-2006 SkillBuilders, Inc.

Flashback logs are written to the flash recovery area if the database is configured for flashback database support. (We will see how to do that next.) The logs contain changed blocks. The flashback logs are different from redo logs and undo – they are a separate, optional recording of database activity.

The transaction rate on your database will have an affect on the performance of creating the logs and the number and size of the logs. Oracle says the overhead of enabling flashback database is “comparatively limited”. Refer to chapter 5 of the **Oracle Database Backup and Recovery Basics** manual for tuning tips. I’ll summarize them here:

- Use fast disks for the flash recovery area
- Avoid file system caching. Use ASM if possible (refer to the *Automatic Storage Management* lesson of this course for more information on ASM).
- Use striped storage volumes, with a small (128k) stripe size (using ASM provides this).

Logs are automatically deleted when:

- The flash recovery area becomes constrained (i.e. runs out of free space) and:
- The log is not needed for a recovery within the window specified in the DB_FLASHBACK_RETENTION_TARGET parameter.

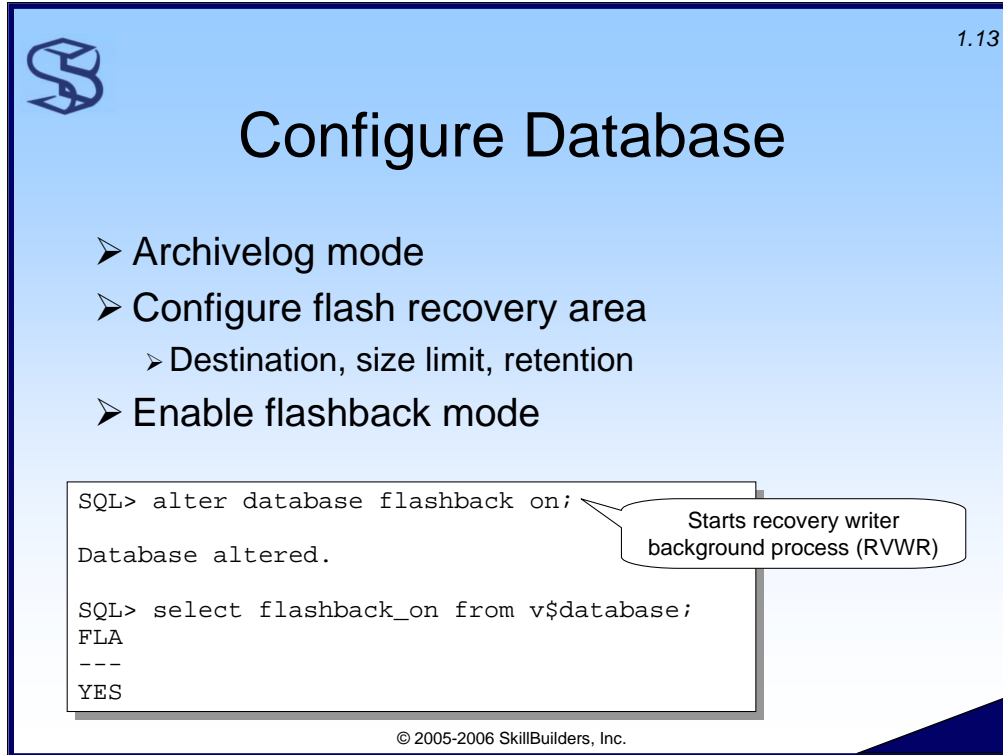
Notes for this slide continue on the next page...

There are a few undocumented flashback database-related parameters that could be investigated for additional control and tuning options. See the script `flashback_parameters_undocumented.sql` for the following code:

```
SYS@orcl> SELECT x$ksppi.kspinm, x$ksppi.kspdesc, x$ksppcv.kspstvl
 2 FROM x$ksppi,      x$ksppcv
 3 WHERE x$ksppi.indx = x$ksppcv.indx
 4 and x$ksppi.kspinm like '/_%flashback%' escape '/'
 5 order by 1;
```

KSPPINM	KSPDESC	KSPSTVL
<code>_flashback_allow_noarchivelog</code>	Allow enabling flashback on noarchivelog database	FALSE
<code>_flashback_barrier_interval</code>	Flashback barrier interval in seconds	1800
<code>_flashback_copy_latches</code>	Number of flashback copy latches	10
<code>_flashback_fuzzy_barrier</code>	Use flashback fuzzy barrier	TRUE
<code>_flashback_generation_buffer_size</code>	flashback generation buffer size	4194304
<code>_flashback_hint_barrier_percent</code>	Flashback hint barrier percent	20
<code>_flashback_log_io_error_behavior</code>	Specify Flashback log I/O error behavior	0
<code>_flashback_log_min_size</code>	Minimum flashback log size	100
<code>_flashback_log_size</code>	Flashback log size	1000
<code>_flashback_logfile_enqueue_timeout</code>	flashback logfile enqueue timeout for opens	600
<code>_flashback_max_log_size</code>	Maximum flashback log size in bytes (OS limit)	0
<code>_flashback_max_n_log_per_thread</code>	Maximum number of flashback logs per flashback thread	2048
<code>_flashback_n_log_per_thread</code>	Desired number of flashback logs per flashback thread	128
<code>_flashback_standby_barrier_interval</code>	Flashback standby barrier interval in seconds	1800
<code>_flashback_verbose_info</code>	Print verbose information about flashback database	FALSE
<code>_flashback_write_size_qm</code>	Desired flashback write size in quarter MB	4
<code>_percent_flashback_buf_partial_full</code>	Percent of flashback buffer filled to be considered partial full	50
<code>_validate_flashback_database</code>	Scan database to validate result of flashback database	FALSE
<code>_verify_flashback_redo</code>	Verify that the redo logs needed for flashback are available	TRUE

19 rows selected.



The slide features a blue background with a white border. In the top left corner is a stylized 'S' logo. In the top right corner is the number '1.13'. The main title 'Configure Database' is centered in a large, bold, black font. Below the title is a bulleted list of three items, each preceded by a right-pointing arrowhead. The first item is 'Archivelog mode'. The second item is 'Configure flash recovery area', which has a sub-bullet 'Destination, size limit, retention'. The third item is 'Enable flashback mode'. Below the list is a screenshot of a SQL terminal window with a white background and a grey border. The terminal shows two SQL commands and their outputs. The first command is 'alter database flashback on;', followed by the output 'Database altered.'. A callout bubble points to this output with the text 'Starts recovery writer background process (RVWR)'. The second command is 'select flashback_on from v\$database;', followed by the output 'FLA', '---', and 'YES'. At the bottom center of the slide is the copyright notice '© 2005-2006 SkillBuilders, Inc.'.

Configure Database

- Archivelog mode
- Configure flash recovery area
 - Destination, size limit, retention
- Enable flashback mode

```
SQL> alter database flashback on;
Database altered.

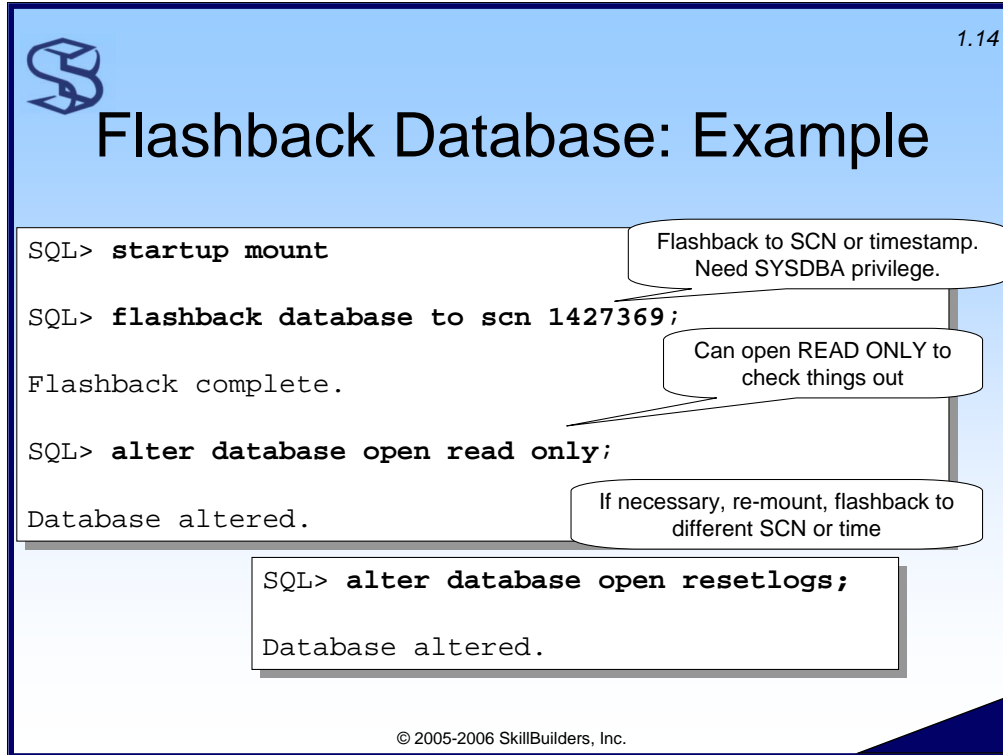
SQL> select flashback_on from v$database;
FLA
---
YES
```

Starts recovery writer background process (RVWR)

© 2005-2006 SkillBuilders, Inc.

The database must be configured for flashback database:

- The database must be in archivelog mode.
- Configure a flash recovery area. The flash recovery area is the repository for the flashback logs. Refer to the information presented earlier in this lesson and Chapter 5 of the Oracle Database Backup and Recovery Basics manual to learn how to configure the flash recovery area.
- Put the database in flashback mode. As shown above, this is done with the ALTER DATABASE FLASHBACK ON statement. This starts a new background process (RVWR) to write the logs.



The screenshot shows a terminal window with the following SQL commands and their outputs:

```
SQL> startup mount
Flashback to SCN or timestamp.
Need SYSDBA privilege.

SQL> flashback database to scn 1427369;
Flashback complete.

SQL> alter database open read only;
Database altered.

SQL> alter database open resetlogs;
Database altered.
```

Callouts provide additional information:

- "Flashback to SCN or timestamp. Need SYSDBA privilege." points to the first command.
- "Can open READ ONLY to check things out" points to the third command.
- "If necessary, re-mount, flashback to different SCN or time" points to the second command.

© 2005-2006 SkillBuilders, Inc.

This is a simple example of using flashback database. Here, I flashback to an SCN which I might have gotten from a flashback version query (or the logminer utility). The database is opened in read-only mode, at which time tables can be queried. If you decide you would like to flashback to a different point in time, shutdown and mount the database, then use flashback database again. When the database is flashed back to the desired point in time, open the database with the RESETLOGS option.


Supplemental Notes

Offline datafiles are *not* flashed back. However, if there are referential integrity issues because you have not flashed back one or more datafiles, you'll receive the error:

```
ORA-01152: file 4 was not restored from a sufficiently old backup
```

So, essentially, you cannot do this – you will have to either recover or drop the file.

Finally, note that I would normally recommend a full backup be taken after the OPEN RESETLOGS operation. However, Oracle 10g provides a seamless recovery across RESETLOGS so this is not as essential as it used to be. Refer to this lesson for more information.



1.15

Flashback Database TO BEFORE RESETLOGS

- R2 introduces “TO BEFORE RESETLOGS” option
- Restore to the previous incarnation
- Undo an OPEN RESETLOGS

```
SYS@orcl> startup mount
ORACLE instance started.

SYS@orcl> flashback database to before resetlogs;

Flashback complete.


SYS@orcl> alter database open resetlogs;

Database altered.
```

© 2005-2006 SkillBuilders, Inc.

Release 2 of Oracle 10g introduces the “TO BEFORE RESETLOGS” clause of the FLASHBACK DATABASE statement. It provides an easy way to flashback across a RESETLOGS operation – to restore to the previous incarnation of the database. (It is actually supported by the same new architecture that supports the “simplified recovery across resetlogs operations”, discussed later in this lesson.) The FLASHBACK DATABASE TO BEFORE RESETLOGS statement provides an easy way to undo an OPEN RESETLOGS operation.

See Chapter 7 of the **Backup and Recovery Basics** manual for more information and examples of this feature. Also refer to the **Oracle Backup and Recovery Reference 10g Release 2** and the **Oracle Database SQL Reference 10g Release 2** for syntax and more information on the FLASHBACK DATABASE statement.



1.16

Monitoring Flashback Database...

- V\$FLASHBACK_DATABASE_LOG can be used to
 - Determine how far back flashback possible
 - Estimate space requirements for your workload


```
SYSTEM@orcl> exec print_table-
      ('select * from v$flashback_database_log')
```

OLDEST_FLASHBACK_SCN	:	3581965	
OLDEST_FLASHBACK_TIME	:	30-may-2006 14:35:02	
RETENTION_TARGET	:	1440	
FLASHBACK_SIZE	:	701038592	
ESTIMATED_FLASHBACK_SIZE	:	112975872	

Lowest SCN in flashback logs

© 2005-2006 SkillBuilders, Inc.

The new V\$FLASHBACK_DATABASE_LOG dynamic performance view reveals some useful information about the state of flashback database and flashback log size requirements. For example, OLDEST_FLASHBACK_SCN and OLDEST_FLASHBACK_TIME reveal how far back the database can be flashed back. FLASHBACK_SIZE shows the amount of space currently being used by flashback data. ESTIMATED_FLASHBACK_SIZE shows how much space – given the current workload – is actually required to maintain 1440 minutes of retention.



...Monitoring Flashback Database

1.17

- V\$FLASHBACK_DATABASE_STAT can be used to
 - See bytes written during last flashback interval


```
SYSTEM@orcl> exec print_table-  
                ('select * from v$flashback_database_stat')
```

BEGIN_TIME	:	09-jun-2006 14:17:59
END_TIME	:	09-jun-2006 14:56:44
FLASHBACK_DATA	:	5103616
DB_DATA	:	26370048
REDO_DATA	:	1952768
ESTIMATED_FLASHBACK_SIZE	:	0

© 2005-2006 SkillBuilders, Inc.

The V\$FLASHBACK_DATABASE_STAT view can be used to see how many bytes of flashback data (database and redo too) were written during the last "interval". The length of an interval is undocumented (or at least I could not find it).

It also can be helpful in determining space requirements for the flashback logs.

1.18

Restore Points...

- Equate arbitrary name with an SCN
 - Flashback Database to restore point
- Normal restore point
 - Flashback target limited by DB_FLASHBACK_RETENTION_TARGET
- Guaranteed restore point
 - Alternative to duplicate database before major upgrade
 - Does not require flashback logging
 - Database can hang if no space for flashback logs

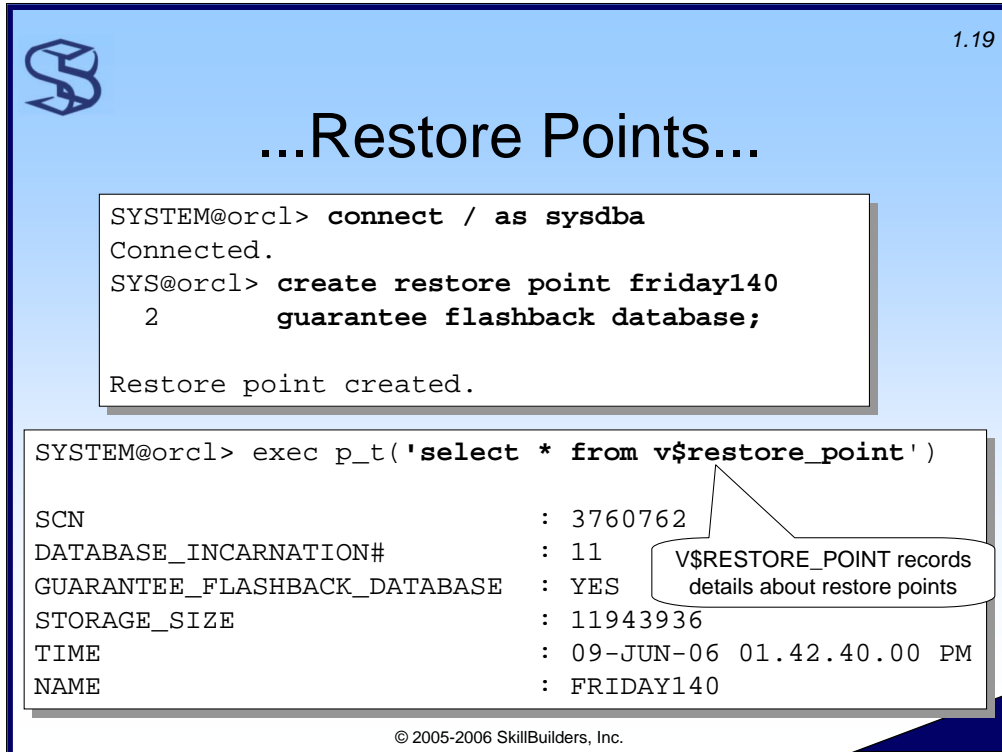
© 2005-2006 SkillBuilders, Inc.

Restore points give the DBA a convenient method of defining an arbitrary string that is equated to an SCN. The restore point (i.e. the “string”) can be used in a FLASHBACK DATABASE statement.

Normal restore points can be used to flashback database as far back as the DB_FLASHBACK_RETENTION_VALUE (flashback logs older than that can be purged). SELECT ANY DICTIONARY or FLASHBACK ANY TABLE privilege is required to create a normal restore point.

Guaranteed restore points prohibit the database from purging flashback logs needed to flashback the database to that point. CAUTION: If you run out of space in the flash recovery area, the database will hang. Oracle 10g issues an “alert” if reclaimable space is less than 15%. (“Reclaimable” means creating free/available space by deleting obsolete logs, RMAN backup pieces, etc.) Alerts are recorded in the DBA_OUTSTANDING_ALERTS view and, using Enterprise Manager, can easily notify the DBA via email or page. SYSDBA privilege is required to create a guaranteed restore point.

Next, let’s look at an example...



The screenshot shows a SQL*Plus session with a blue background. In the top left corner is a stylized 'S' logo. In the top right corner, the number '1.19' is displayed. The main title of the slide is '...Restore Points...'. The session transcript shows the following commands and output:

```
SYSTEM@orcl> connect / as sysdba
Connected.
SYS@orcl> create restore point friday140
2          guarantee flashback database;

Restore point created.
```

The next command and its output are shown in a separate box:

```
SYSTEM@orcl> exec p_t('select * from v$restore_point')
```

SCN	:	3760762
DATABASE_INCARNATION#	:	11
GUARANTEE_FLASHBACK_DATABASE	:	YES
STORAGE_SIZE	:	11943936
TIME	:	09-JUN-06 01.42.40.00 PM
NAME	:	FRIDAY140

A callout box points to the output, stating: "V\$RESTORE_POINT records details about restore points".

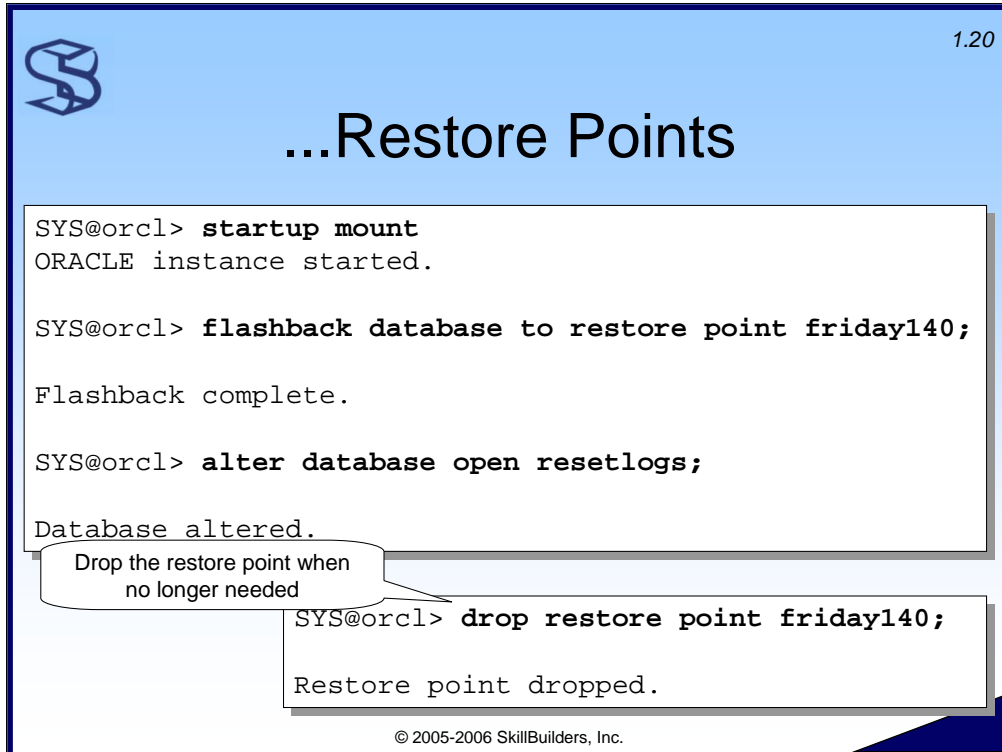
© 2005-2006 SkillBuilders, Inc.

Creating a restore point is done with the CREATE RESTORE POINT statement. In this example, a guaranteed restore point is created, requiring SYSDBA privilege.

Details concerning restore points are recorded in a new view V\$RESTORE_POINT.

Refer to Chapter 5 of the **Backup and Recovery Basics** manual for more information and examples of this feature. Also refer to the **Oracle10g SQL Reference** manual for the syntax of the CREATE RESTORE POINT statement.

Refer to the supplied script **v\$restore_point.sql** for a copy of this code.



The screenshot shows a terminal window with a blue header containing a logo and the text "...Restore Points" and "1.20". The terminal output is as follows:

```
SYS@orcl> startup mount
ORACLE instance started.

SYS@orcl> flashback database to restore point friday140;

Flashback complete.

SYS@orcl> alter database open resetlogs;

Database altered.

SYS@orcl> drop restore point friday140;

Restore point dropped.
```


A callout box points to the `drop restore point` command with the text: "Drop the restore point when no longer needed".

© 2005-2006 SkillBuilders, Inc.

In this example we see the `FLASHBACK DATABASE` statement being used to flashback to the restore point "FRIDAY140" (created in the previous example). The `OPEN RESETLOGS` is required to open the database for read / write activity.

Restore points are a persistent object. Therefore, you can flashback database to a restore point, retry the upgrade or application and, if another failure occurs, flashback again to the restore point.

Because restore points persist in the database, they should (must in the case of guaranteed restore points) be dropped; otherwise, the flash recovery area can fill with unnecessary flashback logs.



Simplified Recovery Through RESETLOGS (SRR)

1.21

- After PIT recovery or recovery with a backup controlfile, must OPEN RESETLOGS
- Now, backups taken *before* RESETLOGS can be used in recovery
- Benefits
 - No need to do full database backup after RESETLOGS
 - “Old” backups can be used to recover database
 - RMAN incrementals can be based on pre-RESETLOGS full backups
- Transparent
 - Existing RMAN RESTORE/RECOVERY scripts work


© 2005-2006 SkillBuilders, Inc.

After a point-in-time recovery (incomplete recovery) or a recovery using a backup control file, the Oracle database must be opened with the RESETLOGS operation. RESETLOGS creates a new incarnation of the database. An incarnation SCN and timestamp are recorded in the control file, data file headers, redo logs, archived logs and backup pieces.

In prior releases, backups and archive logs created before a RESETLOGS could not be used in the recovery of the current database incarnation unless an RMAN catalog was used to record all database incarnations. With Oracle10g, this is no longer true. This provides several benefits:

- There is no need to do a full database backup after OPEN RESETLOGS and before opening the database for general use.
- “Old” backups can be used in recovery. If newer backups are corrupted or lost, “old” backup can be used if available.
- RMAN incremental backups can be based on a level 0 backup taken before RESETLOGS.

Oracle has implemented this so as to be transparent to the DBA. i.e. Existing RMAN restore and recovery scripts will work, even if they access a backup created before a RESETLOGS operation.

1.22

SRR Architectural Changes...


- LOG_ARCHIVE_FORMAT
 - Name includes “%r”
 - RESETLOGS identifier
 - Keeps it unique from logs created by another incarnation
- V\$ARCHIVED_LOG new column
 - RESETLOGS_ID

© 2005-2006 SkillBuilders, Inc.

Obviously, there are some architecture changes required to support the simplified recovery through resetlogs. In essence, what Oracle 10g needs to do is keep track of each incarnation and be able to determine what recovery-related files belong to each incarnation. To do this, Oracle 10g implements the following changes:

- The LOG_ARCHIVE_FORMAT parameter (which controls the name of the archive log) must include a new variable, “%r”. The %r variable provides an incarnation identifier (Oracle calls it a “resetlogs identifier”). %r insures that the archive log name is unique if a new incarnation of the database is created. (Other items in the archive log name typically include %t (thread) and %s (sequence number) which can be redundant across incarnations because the sequence number is reset to 0 by RESETLOGS.)
- V\$ARCHIVED_LOG view contains a new column, RESETLOGS_ID, that contains the resetlogs identifier.


Architecture changes continued on the next slide...



1.23

...SRR Architectural Changes...


- V\$LOG_HISTORY and V\$OFFLINE_RANGE
 - Not cleared during RESETLOGS
 - New columns RESETLOGS_CHANGE# and RESETLOGS_TIME
- V\$DATABASE_INCARNATION
 - Records parent/child relationship of incarnations
 - RESETLOGS ID for each incarnation
 - New column FLASHBACK_DATABASE_ALLOWED
 - Possible to flashback to some point in that incarnation?



© 2005-2006 SkillBuilders, Inc.


Architecture changes (continued):

- Dynamic performance views V\$LOG_HISTORY and V\$OFFLINE_RANGE are not cleared by RESETLOGS operations.
- The V\$LOG_HISTORY and V\$OFFLINE_RANGE views contain new columns that hold the SCN and date of the database incarnation that the log belongs to.
- The dynamic performance view V\$DATABASE_INCARNATION contains new columns to support the tracking of the parent / child relationship between incarnations and the RESETLOGS identifier for each incarnation. Release 2 adds a new column, FLASHBACK_DATABASE_ALLOWED, which is demonstrated next in this lesson.

1.24

...SRR Architectural Changes

```
SYS@orcl> select prior_resetlogs_time,  
2         resetlogs_time,  
3         flashback_database_allowed  
4   from v$database_incarnation;
```




PRIOR_RES	RESETLOGS	FLASHBACK_DATABASE_ALLOWED
30-AUG-05	27-JAN-06	NO
27-JAN-06	31-MAR-06	NO
27-JAN-06	31-MAR-06	NO
31-MAR-06	25-APR-06	NO
31-MAR-06	26-APR-06	NO
31-MAR-06	26-APR-06	NO
26-APR-06	26-APR-06	NO
26-APR-06	27-APR-06	NO
26-APR-06	27-APR-06	YES
27-APR-06	02-JUN-06	YES

© 2005-2006 SkillBuilders, Inc.

Release 2 of Oracle 10g adds a new column to the V\$DATABASE_INCARNATION view called FLASHBACK_DATABASE_ALLOWED. This indicates whether or not the database will support a FLASHBACK DATABASE statement to some time or SCN within that incarnation. The ability to flashback into an incarnation would require flashback logs and archived redo logs old enough to support the operation.

See the supplied script `v$database_incarnation.sql` for a copy of this code.

1.25



Compressed Backups

➤ “Real” compressed backupset pieces

```

RMAN> backup as compressed backupset full database plus archivelog;

Starting backup at 08-SEP-04
current log archived

```

➤ If using ASM, get piece size from ASM instance

```

+ASM> select a.name, f.blocks, f.bytes
2 from v$asm_alias a, v$asm_file f
3 where a.file_number=f.file_number
4 and type = 'BACKUPSET'
5* order by f.type, a.name
+ASM> /

```

NAME	BLOCKS	BYTES
annnf0_TAG20040908T162645_0.284.5	12115	6202880
nnndf0_TAG20040908T161717_0.343.13	136003	557068288

Savings of 123888 blocks!

Your results will vary...

© 2005-2006 SkillBuilders, Inc.

RMAN now supports the creation of compressed backupset pieces. As you can see, my tests show significant reduction in piece size. Compression not only saves space but can save lots of time for network backups (due to the reduced number of bytes sent across the network).

Supplemental Notes

You can configure your RMAN environment so compression is the default. For example:

```

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

```

The V\$BACKUP_PIECE view has a new column, COMPRESSED, which will contain “YES” for compressed backup pieces.

1.26

Change Tracking...

➤ Change tracking eliminates need to read entire file during incremental backup

```

RMAN> sql 'alter database enable block change tracking';
sql statement: alter database enable block change tracking
RMAN> backup incremental level 0 database;
Starting backup at 08-SEP-04
channel ORA_DISK_1: backup set complete, elapsed time: 00:02:45

-- later, after normal database use. . .
RMAN> backup incremental level 3 database;
Starting backup at 08-SEP-04
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:03

```

OMF controls file location

Level 0 duration

Level 3 duration

Level 3 duration w/o change tracking time: 00:00:46


© 2005-2006 SkillBuilders, Inc.

Oracle10g RMAN incremental backups are potentially much faster because the entire datafile does not have to be read to determine which blocks have changed. Instead, the database records (i.e. tracks) changed blocks in a file. When an incremental backup is taken, RMAN accesses the “block change tracking file” to determine which blocks to backup.

In the test shown above, the elapsed time of the level 3 incremental backup is 3 seconds; the same level 3 incremental backup with change tracking disabled required 46 seconds.

This behavior must be configured with the ALTER DATABASE ENABLE BLOCK CHANGE TRACKING statement as shown above. The default location of the block change tracking file is the Oracle Managed File DB_CREATE_FILE_DEST parameter. Use the “FILE” clause if you are not using OMF or you want to put the file in a location other than DB_CREATE_FILE_DEST.

Refer to the **Oracle10g SQL Reference** for more information on the ALTER DATABASE statement and the **Oracle10g Database Backup and Recovery Basics** manual for more information on this feature in general.



1.27

...Change Tracking

- Block change recorded in a file

```
LINUX> select * from v$sqlblock_change_tracking;
```

STATUS	FILENAME	BYTES
ENABLED	+ASM_DISK_GROUP1/orcl/change_tracking/ctf.343.9	11599872

- File size estimate 1/30,000 of db size
- Caution
 - Oracle doc mentions performance hit
 - No specifics
 - Must Test

© 2005-2006 SkillBuilders, Inc.

Change tracking is implemented by the creation of a file that records which blocks are changed. The Oracle documentation says that the initial size of the file is 10MB and grows in 10MB increments. The space *required* is 1/30,000 of the blocks tracked. The file size shown in the example above is roughly 10MB.

While the file size should not be an issue, Oracle does mention a performance hit related to the use of this feature. Test before implementing.

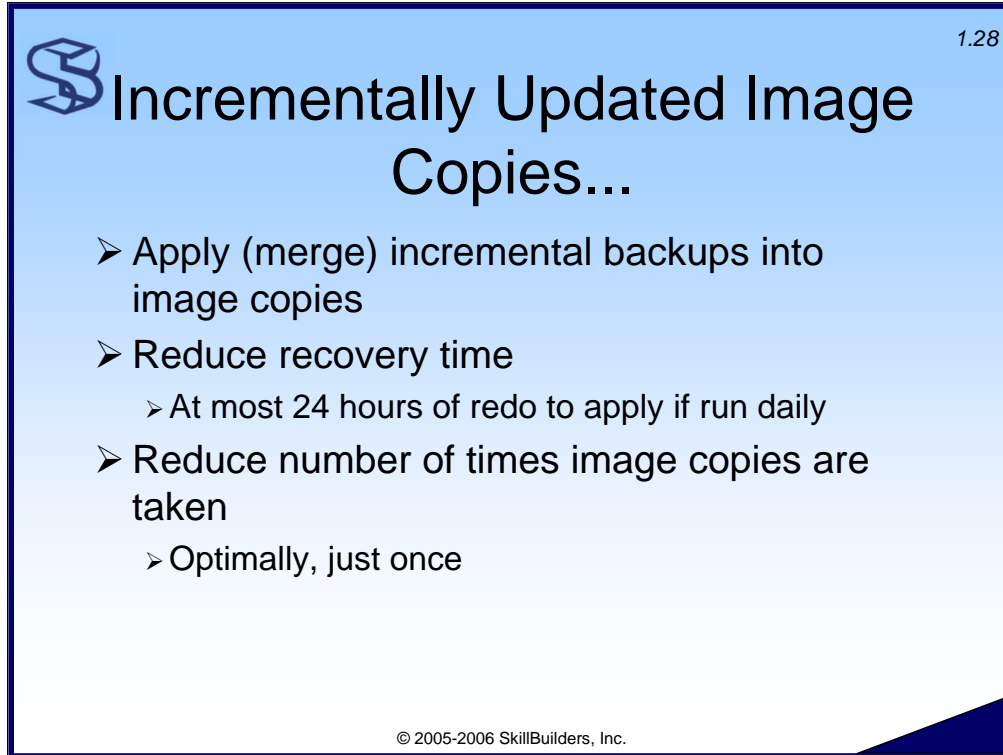
Refer to Chapter 4 of the **Oracle10g Database Backup and Recovery Basics** manual for more information on change tracking. Specifically, read the section entitled **“Improving Incremental Backup Performance: Change Tracking.”**

Supplemental Notes

Execute the following query to determine if change tracking is enabled for your database:

```
DAVE@linux3> select status from v$sqlblock_change_tracking;
```

```
STATUS
-----
ENABLED
```



1.28

Incrementally Updated Image Copies...

- Apply (merge) incremental backups into image copies
- Reduce recovery time
 - At most 24 hours of redo to apply if run daily
- Reduce number of times image copies are taken
 - Optimally, just once

© 2005-2006 SkillBuilders, Inc.

The RMAN Incrementally Updated Backup feature is designed to limit the amount of redo you will need to apply during recovery operations – *thus reduce recovery time*.

The basic idea is to create image copies of your datafiles, then subsequently update the image copies with incremental backups, i.e. merge level 1 incremental backups into an existing image copy.

1.29

...Incrementally Updated Image Copies

Run this script daily

Apply any available level 1 incrementals to the image copy with the tag "ic1"

Tags must match

```

RMAN> run {
2> recover copy of database with tag 'ic1';
3> backup
4>   incremental level 1 tag 'level_1_for_image_copy_update'
5>   for recover of copy with tag 'ic1'
6>   database;
7> }
Starting recover at 09-SEP-04

```

Create a level 1 incremental to be used to merge into image copy

Also can add:
"as compressed backupset"
"plus archivelog [delete input]"

© 2005-2006 SkillBuilders, Inc.

The script can be run daily. Each day this runs:

- The RECOVER COPY command updates all datafile image copies (with the tag "IC1") with the previous day's level 1 incremental. If it does not find an image copy to update or a level 1 to apply, the RECOVER command simply issues messages and successfully terminates. Note that an *identical* user-defined tag must be used on the RECOVER COPY command and the BACKUP FOR RECOVER OF COPY WITH TAG clause.
- The BACKUP command creates a new level 1 incremental. However, if a level 0 image copy does not exist (e.g. on the 1st run), the BACKUP command will create one.

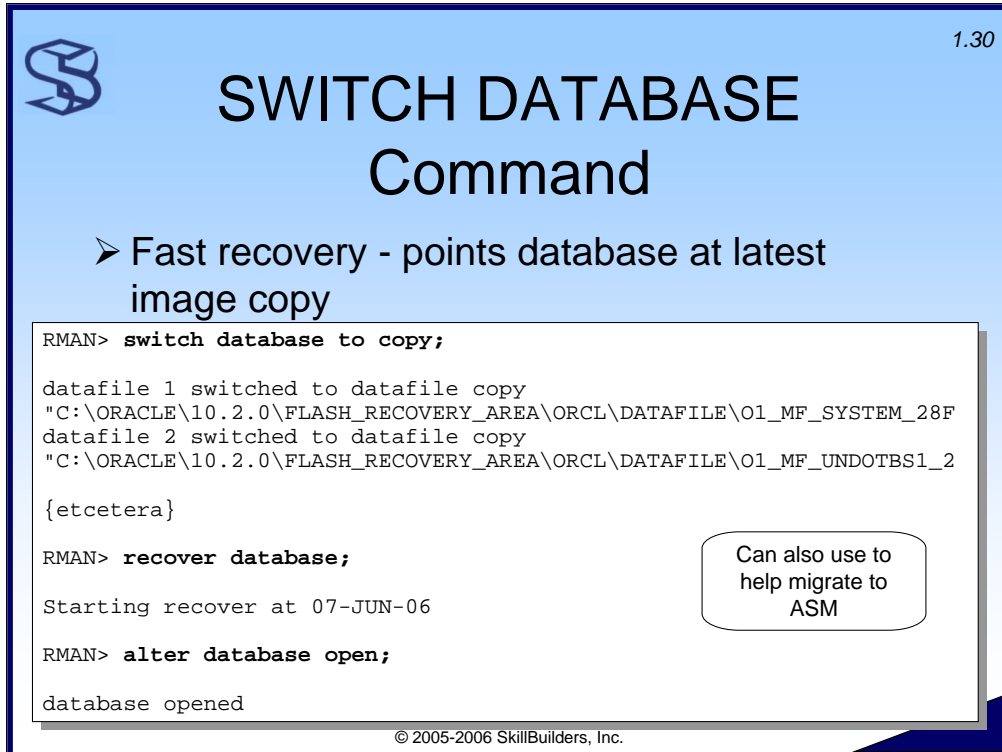
Thus, at all times, you have available for recovery:


- Image copy up to a maximum of 48 hours old
- Level 1 incremental up to a maximum of 24 hours old
- Online and archive logs to support point-in-time or complete recovery.

With this strategy, you will never have to apply more than 24 hours of redo (archive logs) to apply to perform a complete recovery.

Using a tag on the INCREMENTAL LEVEL 1 clause is optional, but I did not like the default tag RMAN supplied for the incrementals, so I supplied my own.

Refer to the article "**Oracle10g - RMAN Incrementally Updated Backups**" by Dave Anderson (find at www.skillbuilders.com) for a detailed discussion of this feature.



 **SWITCH DATABASE Command** 1.30

➤ Fast recovery - points database at latest image copy

```

RMAN> switch database to copy;

datafile 1 switched to datafile copy
"C:\ORACLE\10.2.0\FLASH_RECOVERY_AREA\ORCL\DATAFILE\01_MF_SYSTEM_28F
datafile 2 switched to datafile copy
"C:\ORACLE\10.2.0\FLASH_RECOVERY_AREA\ORCL\DATAFILE\01_MF_UNDOTBS1_2
{etcetera}

RMAN> recover database;

Starting recover at 07-JUN-06

RMAN> alter database open;

database opened

```

Can also use to help migrate to ASM

© 2005-2006 SkillBuilders, Inc.

A new 10g RMAN command – SWITCH DATABASE TO COPY – provides a convenient technique for quickly switching to the latest image copies (i.e. making the latest image copies the current datafiles). So, in the event of failure of the primary datafiles, you can very quickly switch to the image copies – without the overhead of manually changing the control file or restoring (i.e. copying) the image copies. You'll notice that after switching the image copies to the current datafiles, you need to recover the new datafiles (i.e. apply all changes made to the database since the image copy was created). If you use the incrementally update image copy technique demonstrated earlier in this lesson, your image copies will be quite current and the RECOVER operation will hopefully be quick (how long it takes is directly related to the amount of updates occurring on your database).

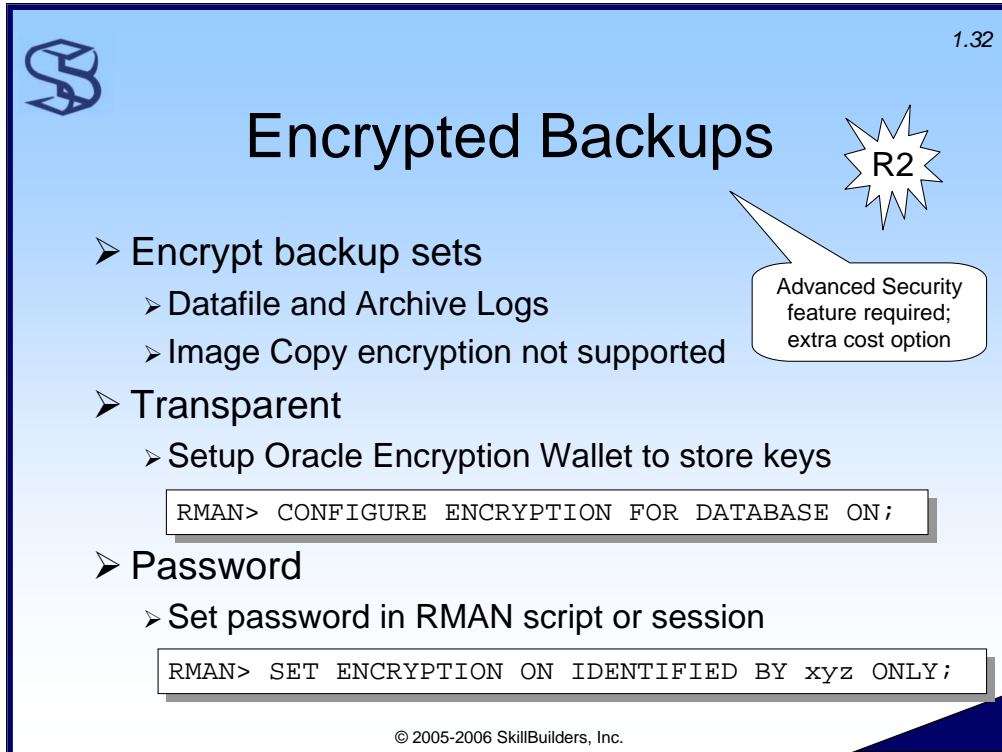
The SWITCH DATABASE command updates the control file(s) to point to the new datafiles, just like the ALTER DATABASE RENAME FILE statement. The SWITCH DATABASE command also updates the RMAN repository to reflect the change (i.e. it deletes the entries for the image copies since they are now the primary datafiles).

This might be a somewhat temporary solution since your database will now be operating using image copies in a *backup* location as the primary datafiles. In the example shown above, the SWITCH DATABASE command switched the database to use the image copies located in the Flash Recovery Area. However, you can easily take another image copy while the database is online – into the correct location for your datafiles – and then later, when the database can be brought down, use another SWITCH DATABASE TO COPY command to reset the database to use the datafiles in the proper production location. Refer to the supplied demonstration file **switch_database_to_copy_demo.txt** for an example of using this command.

Notes for this slide continue on the next page...

The SWITCH to COPY command also supports the DATAFILE and TABLESPACE level. Refer to the Oracle Database Backup and Recovery Reference 10g Release 2 for more information on the RMAN SWITCH command.

Finally, note that this command can be used as part of a procedure to migrate to Automatic Storage Management (ASM). Refer to the **Oracle Database Backup and Recovery Advanced User's Guide 10g Release 2** (see the section entitled "Disk-Based Migration of a Database to ASM").



1.32

Encrypted Backups

- Encrypt backup sets
 - Datafile and Archive Logs
 - Image Copy encryption not supported
- Transparent
 - Setup Oracle Encryption Wallet to store keys
- Password
 - Set password in RMAN script or session

```
RMAN> CONFIGURE ENCRYPTION FOR DATABASE ON;
```

```
RMAN> SET ENCRYPTION ON IDENTIFIED BY xyz ONLY;
```

Advanced Security feature required; extra cost option


© 2005-2006 SkillBuilders, Inc.

Another RMAN R2 feature (Enterprise Edition only) is support for encrypted backup sets (encrypting image copies is not supported).

Oracle supports three encryption techniques:

- Transparent – In this mode an encryption key is stored in the Oracle Encryption Wallet feature, which must be previously configured. I believe this (Oracle Encryption Wallet) also supports the R2 feature transparent column encryption. A password for encryption and decryption is *not* required. This is the default.
- Password – Password encryption requires that the DBA provide the same password on the backup and restore commands. Forgetting the password renders the backupset useless. This mode can be particularly useful when restoring on a remote system.
- Dual Mode – With this technique, either the password or the Wallet can be used to decrypt the backup.

Encryption of data requires CPU resources. I have not yet tested the overhead (June 2006).

1.33

RMAN DROP DATABASE

- New command removes
 - Controlfiles
 - Datafiles
 - Logfiles
 - Spfile
- Optionally drops
 - Archive logs
 - Backup pieces
 - Image copies
- Unregisters database
 - If connected to catalog
- Database in mount exclusive mode

© 2005-2006 SkillBuilders, Inc.

Oracle 10g introduces a DROP DATABASE command. It is offered as both a SQL statement and an RMAN command. The RMAN command offers more functionality than does the SQL statement. For example, the RMAN command offers the ability to optionally drop the archive logs and backup pieces as shown in this example:

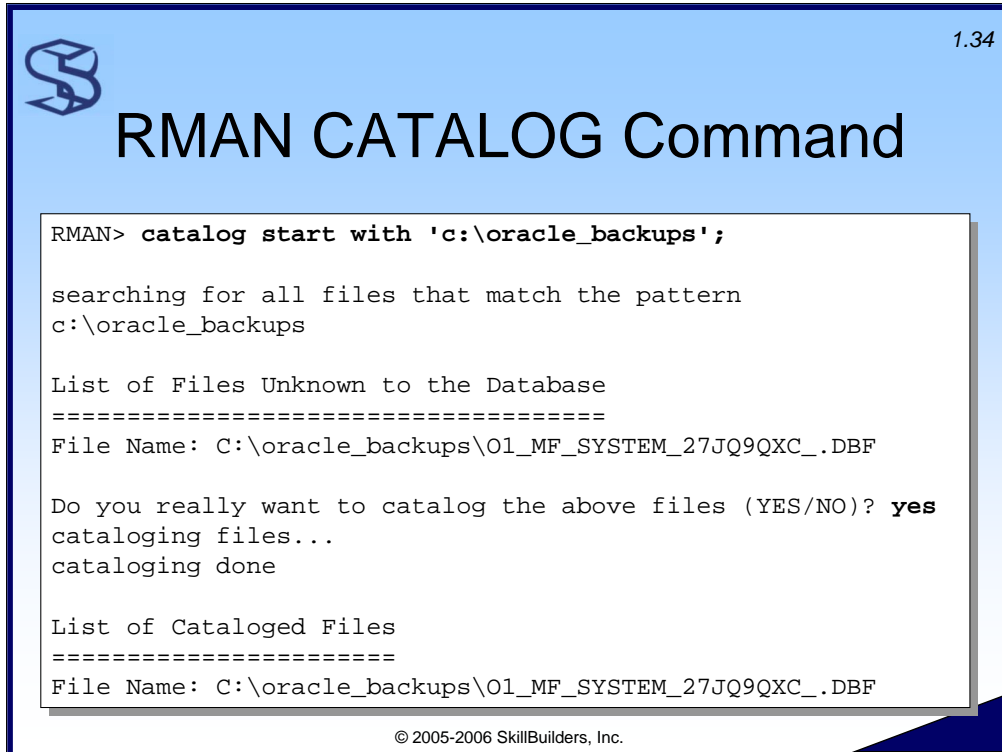
```
RMAN> DROP DATABASE INCLUDING BACKUPS;
```

Dropping the database drops the controlfiles, datafiles, logfiles and, if used, the SPFILE. If the database is connected to an RMAN catalog when the database is dropped, RMAN also unregisters the database from the catalog.

The database must not be open; it must be mounted in exclusive mode.

References:

- **Metalink Note:251412.1**
- **Oracle Database Recovery Manager Reference 10g Release 2 (10.2)**



The screenshot shows a terminal window with a blue header containing an Oracle logo and the text '1.34'. The main title is 'RMAN CATALOG Command'. The terminal output shows the execution of the 'catalog start with' command, a search for files, a list of unknown files, a confirmation prompt, and a list of cataloged files. The file name shown is 'C:\oracle_backups\01_MF_SYSTEM_27JQ9QXC_.DBF'. A copyright notice for SkillBuilders, Inc. is at the bottom.

```
RMAN> catalog start with 'c:\oracle_backups';

searching for all files that match the pattern
c:\oracle_backups

List of Files Unknown to the Database
=====
File Name: C:\oracle_backups\01_MF_SYSTEM_27JQ9QXC_.DBF

Do you really want to catalog the above files (YES/NO)? yes
cataloging files...
cataloging done

List of Cataloged Files
=====
File Name: C:\oracle_backups\01_MF_SYSTEM_27JQ9QXC_.DBF

© 2005-2006 SkillBuilders, Inc.
```

Oracle9i provided the CATALOG command to add information about datafile copies, control file copies, or archived logs to the target database control file and recovery catalog (if available). Oracle10g extends the capability of the CATALOG command to support:

- BACKUPPIECE – Use this option to catalog (in the control file and recovery catalog, if applicable) a backup piece that has been copied or moved to a new location.
- RECOVERY AREA or DB_RECOVERY_FILE_DEST – Use this option to catalog all backup sets, image copies and archive logs in the flash recovery area. A NOPROMPT option is available to eliminate prompting for each item. RECOVERY AREA and DB_RECOVERY_FILE_DEST are equivalent clauses.
- START WITH 'location' – Use this option to catalog all files in the specified location.

The example above demonstrates cataloging an image copy that was copied (using an operating system command) to a secondary location. If desired, after cataloging, use the LIST command to verify the cataloging of the image copy, backup piece or control file copy.

Refer to the **Oracle Database Recovery Manager Reference 10g Release 2** for more information and examples of the RMAN CATALOG command.

1.35

DURATION Parameter

- Control backup rate
- More flexible and intuitive than RATE parameter

```
RMAN> backup duration 0:30 minimize load tablespace users;  
Starting backup at 03-AUG-04
```


Maximum time allotted is 30 minutes

OPTIONAL:
"minimize load" says
"take the entire 30 minutes"

© 2005-2006 SkillBuilders, Inc.

The BACKUP DURATION parameter will do one of two things for you:

- If MINIMIZE LOAD is specified, RMAN will *reduce* resource consumption so that the job requires the amount of time specified. This is like the RATE parameter, but perhaps more intuitive.
- IF MINIMIZE LOAD is not specified, the time specified is the maximum time the BACKUP can execute. RMAN will *kill the job* when it reaches the specified time limit.



1.36


New V\$ Views

- R2 adds 18 new views
- **V\$RMAN_BACKUP_JOB_DETAILS**
 - Status, date/time, elapsed time, read rate, more
 - Helpful for tuning RMAN jobs
- **V\$BACKUP_SET_DETAILS**
 - Incremental level, number of pieces, size ,status
- **V\$BACKUP_PIECE_DETAILS**
 - File handles, status, size

© 2005-2006 SkillBuilders, Inc.

Oracle10g adds 18 new backup and recovery-related views. Refer to the Oracle Database Reference manual (Release 2) for details on these views:

V\$BACKUP_ARCHIVELOG_DETAILS	V\$BACKUP_SET_DETAILS
V\$BACKUP_ARCHIVELOG_SUMMARY	V\$BACKUP_SET_SUMMARY
V\$BACKUP_CONTROLFILE_DETAILS	V\$BACKUP_SPFILE_DETAILS
V\$BACKUP_CONTROLFILE_SUMMARY	V\$BACKUP_SPFILE_SUMMARY
V\$BACKUP_COPY_DETAILS	V\$RMAN_BACKUP_JOB_DETAILS
V\$BACKUP_COPY_SUMMARY	V\$RMAN_BACKUP_SUBJOB_DETAILS
V\$BACKUP_DATAFILE_DETAILS	V\$RMAN_BACKUP_TYPE
V\$BACKUP_DATAFILE_SUMMARY	V\$FLASH_RECOVERY_AREA_USAGE
V\$BACKUP_PIECE_DETAILS	V\$RMAN_ENCRYPTION_ALGORITHMS
V\$RMAN_OUTPUT	
V\$RMAN_STATUS	



1.37


More RMAN Enhancements...

- Archive log deletion policy
 - Automatic clean up of obsolete logs
- RESTORE DATABASE PREVIEW
 - Identify the backups that would be used
- Restore failover
 - If error reading backup, RMAN keeps trying different backup copies until all possible exhausted
- CONVERT Command
 - Convert datafiles to different platform

© 2005-2006 SkillBuilders, Inc.

Here is a brief description of additional Oracle10g RMAN enhancements:

- The CONFIGURE command can be used to configure an archivelog deletion policy. There are 2 main options: "NONE", which means RMAN can delete archive logs from the flash recovery area only after they've been backed up and are obsolete (according to the configured retention policy). "APPLIED ON STANDBY" means the archive logs can be deleted if they have been applied to a standby database.
- A new RESTORE DATABASE PREVIEW command is available which identifies which backups RMAN would use to perform the restore.
- RMAN will automatically failover to another backup piece (even from an older backup if necessary) if corruption is encountered. It will continue to failover until a non-corrupt backup is found or all possibilities are exhausted.
- RMAN 10g provides a CONVERT command to convert datafiles to a different platform. For example, a datafile from a Solaris server could be transported to a Windows server (consider using data pump transportable tablespace and DBMS_FILE_TRANSFER; refer to the *Utilities Enhancements* lesson in this course book for an example). Refer to the **Oracle Backup and Recovery Reference 10g Release 2** for more information on the CONVERT command.



1.38

...More RMAN Enhancements

- Channel failover for BACKUP
- COPY command deprecated
 - BACKUP command with COPY parameter creates image copies
- Default backup can be configured to create
 - Normal backupset
 - or compressed backupset
 - or disk-based image copy
- RMAN catalog tablespace created in SYSAUX

© 2005-2006 SkillBuilders, Inc.

Additional RMAN enhancements, continued:

- RMAN now has the ability to try another channel if a channel fails during backup.
- The COPY command is deprecated (will eventually be desupported) because the BACKUP command can be used to create image copies. For example:


```
RMAN> backup as copy tablespace users;
```
- The CONFIGURE DEVICE TYPE command now supports configuring for normal backupset, compressed backup or image copy. (Note that the image copy option only works for DEVICE TYPE DISK..) For example:

```
RMAN> show device type ;
```

RMAN configuration parameters are:

```
CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO BACKUPSET; #
default
```


```
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

- To alleviate clutter in the SYSTEM tablespace, the RMAN catalog is now by default created in the new SYSAUX tablespace.



1.39

Additional R2 RMAN Enhancements...

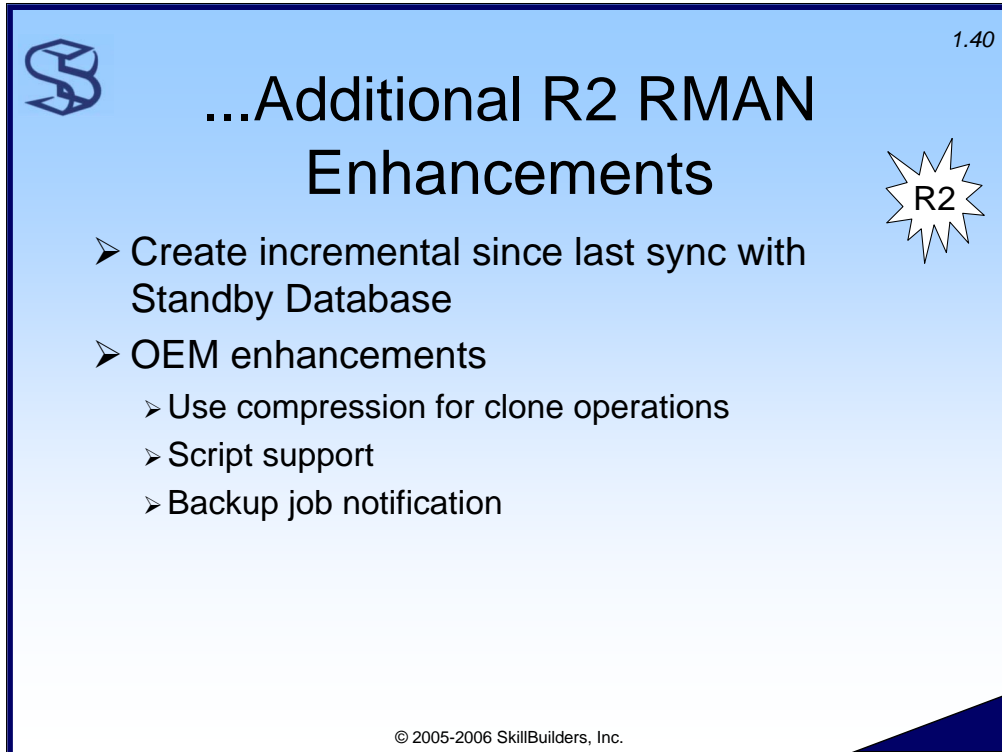


- Missing tempfiles automatically recreated during open
 - After whole database restore / recovery
- Unused block compression
 - Under the right conditions, unused blocks are not written to backupset
- RAC dynamic channel allocation

© 2005-2006 SkillBuilders, Inc.

Oracle10g Release 2 also provides these RMAN enhancements:

- After a whole (full) database restore and recovery, and at database open, Oracle will automatically create missing tempfiles. The original size, location, AUTOEXTEND and MAXSIZE attributes are used to recreate the tempfile. The tempfile must be locally managed (as opposed to dictionary managed). Refer to Chapter 6 of the **Oracle Database Backup and Recovery Basics 10g Release 2** manual for more information on this feature.
- If certain conditions are met, Release 2 of 10g RMAN will backup only in-use blocks (previous releases of RMAN backed up any block that ever contained data – even if it was not currently in-use (this was called “NULL compression”). Refer to the BACKUP command in the **Oracle 10g Release 2 Recovery Manager Reference** for more information. The conditions include:
 - Full or Level 0 backup
 - No guaranteed restore points in effect
 - Locally managed datafile
 - Disk-based backupset
- RAC backups no longer require manual channel allocation for each node – dynamic channel allocation is now supported. For more information, refer to Chapter 1 of the **Oracle 10g R2 New Features Guide** and Chapter 8 of the **Oracle 10g R2 Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide**.



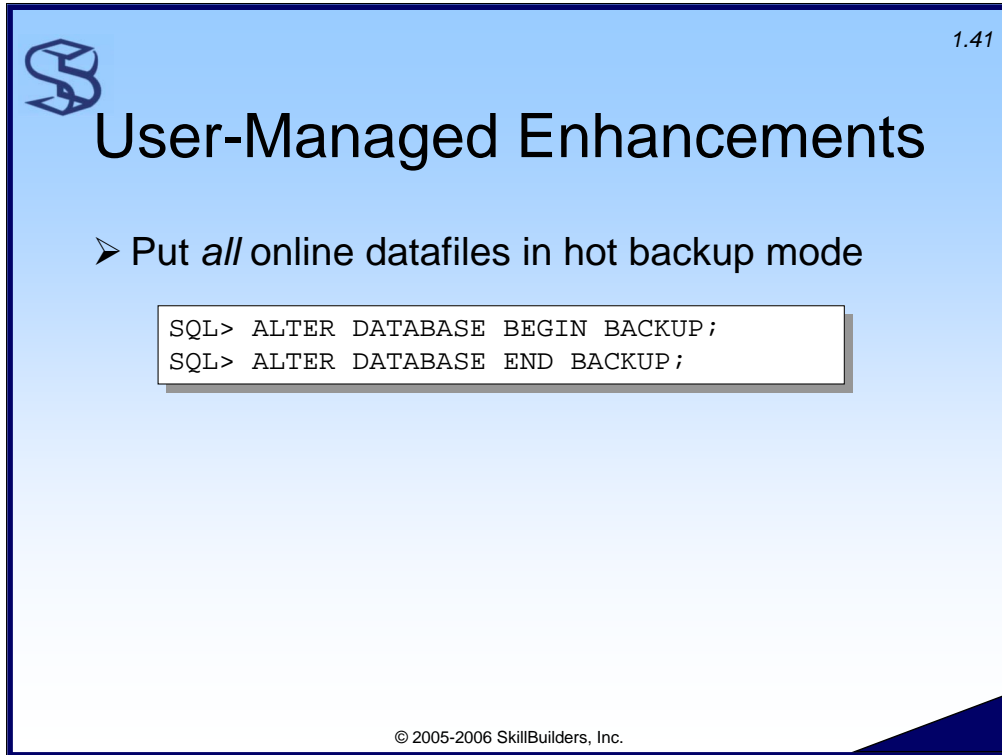
1.40

...Additional R2 RMAN Enhancements

- Create incremental since last sync with Standby Database
- OEM enhancements
 - Use compression for clone operations
 - Script support
 - Backup job notification

© 2005-2006 SkillBuilders, Inc.

- Oracle 10g R2 provides the ability to create an incremental backup as of some SCN. Only changes made at or after that SCN are backed up. This can be useful for updating a Standby Database with changes made to the primary database since the last synchronization with the standby. For more information refer to the **Oracle 10g R2 Backup and Recovery Advanced User's Guide** and the BACKUP INCREMENTAL FROM SCN [scn] DATABASE command.
- Oracle Enterprise Manager has been enhanced to support:
 - Compression during cloning operations, reducing data transmitted over a network.
 - RMAN scripts created by OEM can be edited. Refer to Chapter 9 of the **Oracle Database 2 Day DBA 10g Release 2** manual for more information.
 - Backup job notification



1.41


User-Managed Enhancements

- Put *all* online datafiles in hot backup mode

```
SQL> ALTER DATABASE BEGIN BACKUP;  
SQL> ALTER DATABASE END BACKUP;
```

© 2005-2006 SkillBuilders, Inc.

User-managed hot backups are made simpler with the new command `ALTER DATABASE BEGIN BACKUP`. This places all online and available datafiles in hot backup mode. This is in contrast to the `ALTER TABLESPACE BEGIN BACKUP` command, which needs to be repeated for each tablespace to be backed up.



1.42

Oracle Secure Backup


- Integrated tape backup management software
- Bundled media manager for RMAN
 - RMAN backup directly to tape
 - Replaces LSSV
- Supports
 - Encryption
 - Any number of any type devices
 - \$3000 per device
 - Backup Oracle Home
 - Networked environments
 - Enterprise Manager interface

© 2005-2006 SkillBuilders, Inc.

Oracle 10g Release 2 includes Oracle Secure Backup, an integrated tape management software solution. Oracle Secure Backup supports:

- RMAN. As an RMAN media manager, it enables RMAN to backup directly to tape devices. Oracle Secure Backup is positioned as a replacement for Legato Single Server Version, which was shipped at no cost with Release 1. LSSV supports just one tape device (among several other limitations).
- Encryption of backups.
- Any number of output devices, tape or otherwise, at a cost of \$3000 per device.
- Backup of non-database files. For example, you can backup the Oracle software in the Oracle Home.
- Networked environments. A single installation of Oracle Secure Backup can be used to

For more information, refer to the **Oracle Database Readme 10g Release 2**, section 10 Media Management Software. The Readme document can be accessed from the welcome.html message shipped with Oracle 10g Release 2.

1.43

Lesson Summary...

- Easier management of files with flash recovery area
- Flashback Database
 - Point-in-time recovery for entire database
 - Must configure database to create flashback logs
 - R2 supports
 - Restore points
 - "TO BEFORE RESETLOGS"
 - EE feature
- Restore points
 - Convenient technique for flashback database

© 2005-2006 SkillBuilders, Inc.

With the new features Oracle10g introduces, I think many DBAs will now start to view RMAN as a viable tool for backup and recovery of production databases.

While I have tried to highlight the "most important" RMAN enhancements in this lesson, there are more and undoubtedly you will find enhancements not covered here that are important to you. So, refer to the section "What's New in Backup and Recovery" in the **Backup and Recovery Advanced User's Guide** for more information.



1.44

...Lesson Summary...

- Transparent recovery through RESETLOGS
- Real compression for RMAN backup pieces
- Much faster incremental backups with Change Tracking
- Update image copies with incremental backups
 - Faster recovery
- SWITCH DATABASE to image copies

© 2005-2006 SkillBuilders, Inc.



1.45

...Lesson Summary

- Secure backups with encrypted RMAN backup sets
- Integrated media manager for tape backups
 - “Oracle Secure Backup”
- Enterprise Manager enhanced to support RMAN
- Lots of minor features!

© 2005-2006 SkillBuilders, Inc.