




# Lesson 16

## Managing Undo

A hands-on lesson on using Automatic Undo Management to manage undo in the Oracle database.

SKILLBUILDERS

16.2

## Lesson Objectives

- Define "undo"
- Identify methods of managing
- Learn to manage with automatic undo management (AUM)
- Implement AUM
  - Learn the new initialization parameters for AUM
  - Review Oracle8i *locally-managed tablespaces*
  - Create, modify & drop an undo tablespace
- Introduce new AUM-related data dictionary views
- Sizing the Undo Tablespace

© 2004 SkillBuilders, Inc.

We will start by defining the term undo and introducing the two methods of managing: manual and automatic. We will then learn the new method, Automatic Undo Management (AUM) in detail. Automatic Undo Management (AUM) is a new Oracle9i feature that deprecates the configuration and use of Rollback Segments.

Specifically, in this lesson we will:

- Provide a brief review of rollback segments (and remind ourselves what a pain in the neck they are!).
- Learn about the concepts and architecture of AUM. This will also uncover the benefits of AUM.
- Provide a brief review of locally managed tablespaces. Locally managed tablespaces were introduced in Oracle8i and are required by AUM.
- Provide procedures for implementing AUM.
- Look at the new data dictionary views available for monitoring and sizing the AUM tablespace.

Let's get started!

16.3

## What is Undo?

- Records contain “before image” of changed data
  - Prior to the start of the transaction
- Undo records are used for
  - ROLLBACK statement
    - Undo UPDATE, DELETE, INSERT statements
  - Statement-level read consistency
    - “Queries by other users against the data that is being changed will return the data as it existed before the change began” \*
  - Recovery
    - Rollback uncommitted transactions after redo logs applied

© 2004 SkillBuilders, Inc.

“Undo” is before-images of changed data. Undo records are automatically created by the database whenever a change is made. As the **Oracle Concepts** manual states "When an update occurs, the original data values changed by the update are recorded in the database's undo records."

Undo exists to:

- Support the ROLLBACK SQL statement.
- Support statement-level read consistency. A good definition of statement-level read consistency can be found in this quote from Kevin Loney: “Queries by other users against the data that is being changed will return the data as it existed before the change began.\*” Another way to explain it is to say that if a query starts at 12 Noon and executes for 15 minutes, all data retrieved by that query will be exactly as it was at 12 noon; updates made between 12 and 12:15 will not be included in the result set.
- Support database recovery operations. Recovery involves applying all changes in the redo logs (optionally up to a point-in-time or system change number), then undoing changes that have not been committed (i.e. a commit record is not found in the redo).

\* Quote from **Oracle8 DBA Handbook**, Kevin Loney, Oracle Press ISBN 0-07-882406



16.4

## Where is Undo Kept?

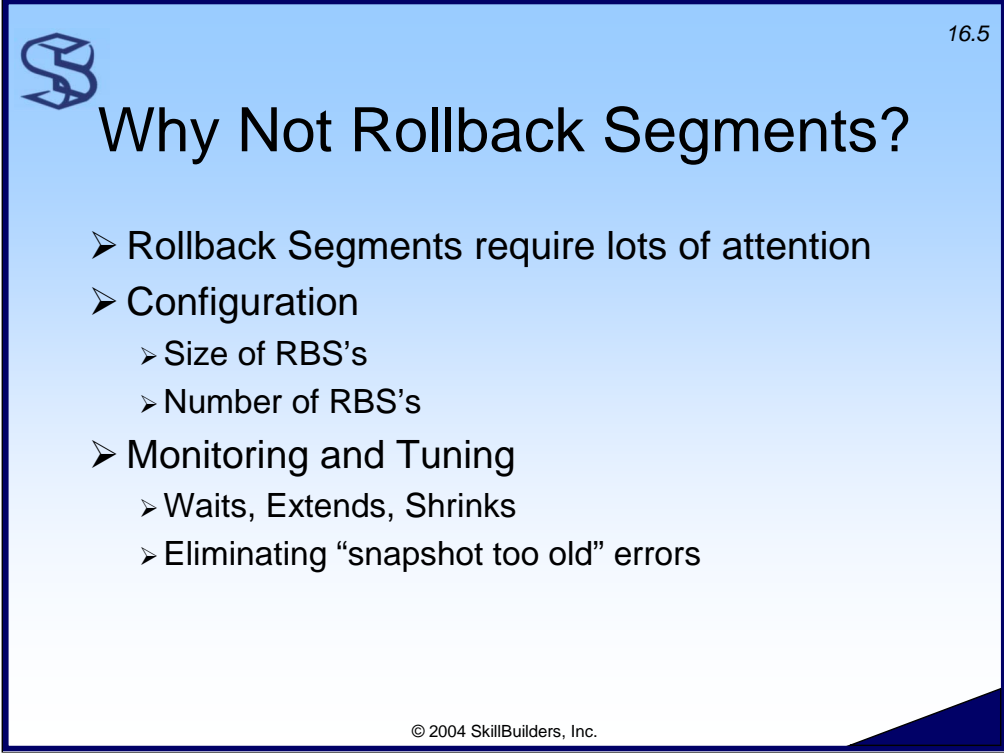
- Undo records require disk space
- Two management methods support
  - Automatic mode
    - DBA creates undo tablespace
    - Database creates undo segments in undo tablespace
  - Manual mode
    - Undo records written to rollback segments
    - DBA creates and manages rollback segments

© 2004 SkillBuilders, Inc.

Undo records need disk space. With Oracle9i, there are two methods of acquiring and managing the disk space, automatic and manual.

In Automatic Undo Management mode, we will create an undo tablespace and let the database create undo segments within the tablespace. In manual mode, we are responsible for creating and managing the rollback segments within the rollback tablespace.

According to **Oracle9i Database Concepts Release 2, Appendix B** Information on Deprecated Features, “Oracle strongly recommends the use” of AUM as well as locally managed tablespaces. The implication is that, at some point, in the future manually managed undo will not be supported. When that will happen is anyone’s guess. (It took many releases for Oracle to remove support for “connect internal.” As Mark Twain once remarked, “News of my death has been greatly exaggerated.”) However, you can still use rollback segments if you want to. Simply set `UNDO_MANAGEMENT=MANUAL`.



16.5

## Why Not Rollback Segments?

- Rollback Segments require lots of attention
- Configuration
  - Size of RBS's
  - Number of RBS's
- Monitoring and Tuning
  - Waits, Extends, Shrinks
  - Eliminating "snapshot too old" errors

© 2004 SkillBuilders, Inc.

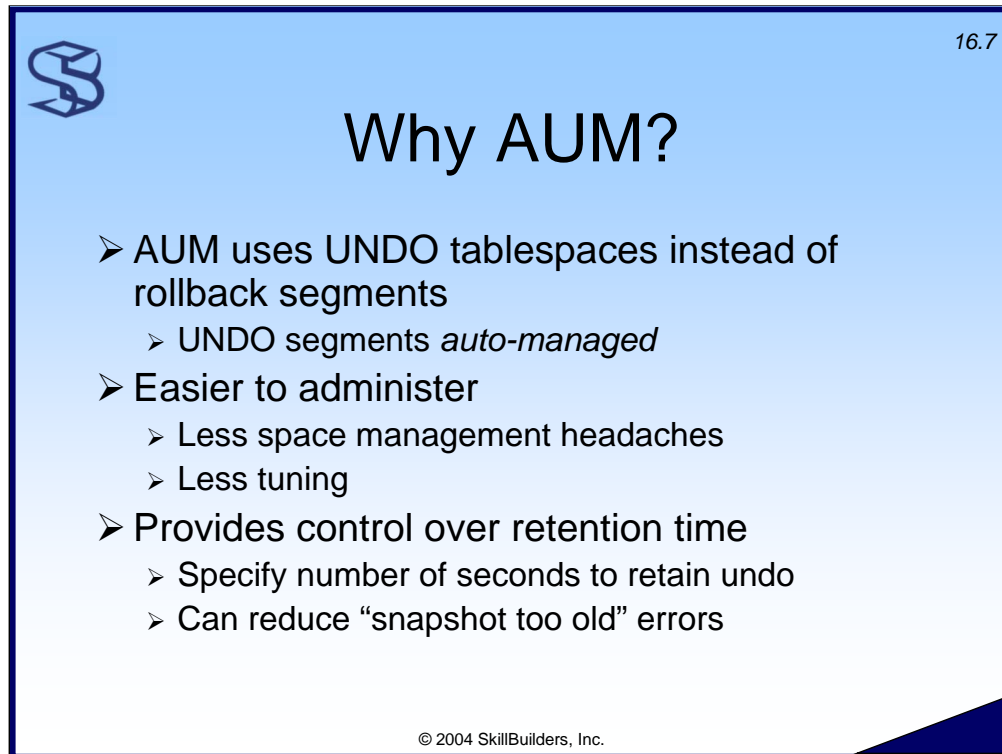
Rollback segments have always been a thorn in the side of Oracle DBAs. Planning for, creation, monitoring, and maintenance of rollback segments can sometimes be an onerous task even for experienced DBAs. Until precise statistics are available, configuring rollback segments is, at best, educated guesses based on anticipated number and size of transactions your database will support. The considerations involve:

- how big to make the tablespace(s) that will contain the rollback segments
- how many rollback segments to create inside that tablespace(s)
- what the segments' size should be
- max number of extents the rollback segments should be allowed
- what size to make the rollback segment parameter OPTIMAL

Monitoring rollback segments involves tracking waits for segments (a bad thing!), too many extends (rollback segments are too small), and excessive shrinks (OPTIMAL parameter needs to be increased).

And let's not forget about the dreaded "snapshot too old" errors. These typically occur when a long running query is accessing data that is being changed by another transaction; the before image of the data is overwritten in the rollback segment before the query gets to it.

In summary, tuning rollback segments is an ongoing task and a hassle. So, let's move on to AUM!



16.7

## Why AUM?


- AUM uses UNDO tablespaces instead of rollback segments
  - UNDO segments *auto-managed*
- Easier to administer
  - Less space management headaches
  - Less tuning
- Provides control over retention time
  - Specify number of seconds to retain undo
  - Can reduce “snapshot too old” errors

© 2004 SkillBuilders, Inc.

In automatic undo management mode, an undo segments are used instead of rollback segments. The benefit is that Oracle automatically allocates and maintains the undo segments within the undo tablespace.

Automatic undo management mode greatly simplifies undo management. Our greatest concern in this mode is the size of the undo tablespace. We no longer need to be concerned with number of rollback segments, size of the segments, transaction per segment, and so forth.

Automatic undo management mode also provides a simple way to specify, in seconds, a retention time. i.e. The period of time that undo records should be kept. This can reduce – possibly eliminate – “snapshot too old” errors.

16.8

## Implementing AUM


- 9i DBCA creates database with AUM
- Procedure for existing database:
  1. Planning
    - Search for apps that use SET TRANSACTION USE ROLLBACK statements
  2. Create an “undo tablespace”
    - Guesstimate initial size
    - UNDO per second \* desired retention \* blocksize
  3. Adjust initialization parameters
  4. Bounce database

© 2004 SkillBuilders, Inc.

OK, now that we know what AUM is, why we want to use it, let's get started with implementation. If creating a new database with the Oracle9i Database Configuration Assistant, it is easy; the DBCA automatically creates the database in AUM mode. Implementing AUM on an existing database involves the following steps:

1. Planning – Check for applications that depend on rollback segments. Code that includes the SET TRANSACTION USE ROLLBACK SEGMENT statement will fail unless errors are suppressed. See initialization parameters later in this lesson to see how to suppress errors.
2. Create the undo tablespace. We will need to guesstimate the initial size of our undo tablespace. A starting point can be calculated by multiplying UNDO per second (transaction rate - see V\$UNDOSTAT) \* blocksize \* your desired retention period. See chapter 13 of the **Oracle9i Administrators Guide** for a complete description of this calculation and Managing UNDO in general.
3. Adjust the initialization parameters.
4. Bounce the database.

Let's look at steps 2 through 4 in turn...



16.9

## Find Undo Tablespace...

➤ Query DBA\_TABLESPACES to find available undo tablespaces

```
1 select tablespace_name, status
2 from dba_tablespaces
3* where contents = 'UNDO'
SQL> /
```


TABLESPACE_NAME	STATUS
UNDOTBS1	ONLINE
UNDO_DATA_TS	ONLINE

Only one will be in use

© 2004 SkillBuilders, Inc.

The DBA\_TABLESPACES view contains a row for every tablespace. To determine what UNDO tablespaces are available, execute the query shown above.

Note that while more than one may be online, only one can be used at any given time. The exception to this is while switching to a new undo tablespace. See the section *Switching Tablespaces* later in this lesson for an example.



16.10

## ...Find Undo Tablespace


➤ DBA\_DATA\_FILES shows size

```
SQL> 1
      2 select tablespace_name, file_name, bytes
      3 from dba_data_files
      4 * where tablespace_name like 'UNDO%'
SQL> /
```

TABLESPACE_NAM	FILE_NAME	BYTES
UNDOTBS1	C:\ORACLE\ORADATA\DAVE\UNDOTBS	429916160
UNDO_DATA_TS	C:\ORACLE\ORADATA\DAVE\O1_MF_U	1048576

© 2004 SkillBuilders, Inc.

The DBA\_DATA\_FILES view shows the current size of the datafile(s) associated with the UNDO tablespace.

16.11

## Create Undo Tablespace...


- Create an “undo tablespace”
- Use the new UNDO clause on CREATE TABLESPACE

```
CREATE UNDO TABLESPACE skillbuilders_undo  
DATAFILE  
'c:\oracle9i\oradata\test\skillbuilders_undo01.dbf'  
SIZE 100M AUTOEXTEND ON;
```

© 2004 SkillBuilders, Inc.

The example above creates a tablespace named SKILLBUILDERS\_UNDO utilizing the new 9i keyword UNDO. Other notables:

- UNDO tablespaces must be locally managed tablespaces. The DEFAULT STORAGE clause is not allowed.
- The undo tablespaces cannot be used for anything but undo segments, else an ORA-30022 error is returned.
- Our example omits the EXTENT MANAGEMENT clause; therefore the default ALLOCATION TYPE of AUTOALLOCATE is used.



16.12

## ...Create Undo Tablespace

- Undo tablespace created with CREATE DATABASE statement

```

CREATE DATABASE prod
DATAFILE 'c:\oradata\tbs\system01.dbf' SIZE 60m
AUTOEXTEND ON NEXT 10m MAXSIZE unlimited
LOGFILE
. . .
UNDO TABLESPACE skillbuilders_undo
DATAFILE
'd:\oradata\tbs\ skillbuilders_undo.dbf'
SIZE 40m AUTOEXTEND OFF

```

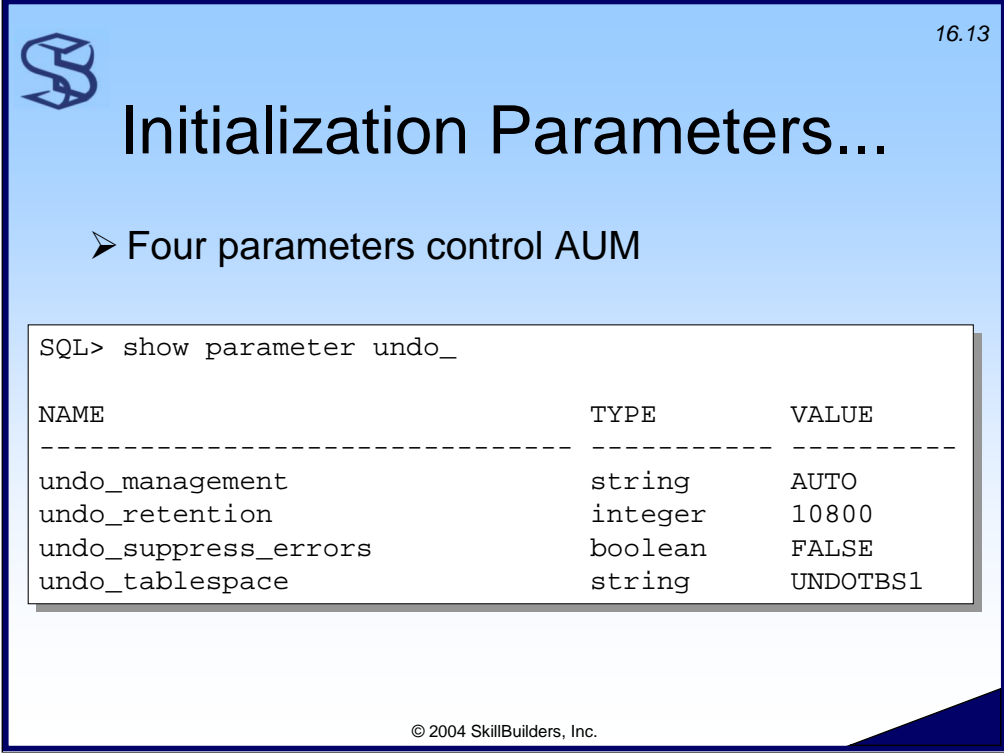
© 2004 SkillBuilders, Inc.

This excerpt above shows an undo tablespace being created as part of a CREATE DATABASE command. The CREATE DATABASE command is discussed in detail in the lesson *Creating a New Database*. A complete statement might look like this:

```

CREATE DATABASE prod
  DATAFILE 'c:\oradata\tbs\system01.dbf' SIZE 60m
  AUTOEXTEND ON NEXT 10m MAXSIZE unlimited
  LOGFILE
    GROUP 1
      ('c:\oradata\redo\log01a.log',
       'd:\oradata\redo\log01b.log') SIZE 500k,
    GROUP 2
      ('c:\oradata\redo\log02a.log',
       'd:\oradata\redo\log02b.log') SIZE 500k,
    GROUP 3
      ('c:\oradata\redo\log03a.log',
       'd:\oradata\redo\log03b.log') SIZE 500k
  MAXINSTANCES 1  MAXDATAFILES 300
  UNDO TABLESPACE skillbuilders_undo
    DATAFILE 'd:\oradata\tbs\ skillbuilders_undo.dbf' SIZE 40m
  AUTOEXTEND OFF

```



16.13

## Initialization Parameters...

➤ Four parameters control AUM


```
SQL> show parameter undo_
```

NAME	TYPE	VALUE
undo_management	string	AUTO
undo_retention	integer	10800
undo_suppress_errors	boolean	FALSE
undo_tablespace	string	UNDOTBS1

© 2004 SkillBuilders, Inc.

To enable AUM, parameter `UNDO_MANAGEMENT=AUTO` must be set. The parameter `UNDO_TABLESPACE` should also be set to a valid undo tablespace name; otherwise, Oracle will use a system-default undo tablespace, which, as a general rule, is not desirable.

Turn to the next page for details on these parameters.



16.14

## ...Initialization Parameters...


- UNDO\_MANAGEMENT
  - AUTO for AUM
  - MANUAL for rollback segments
  - Static parameter
    - Must bounce DB
- UNDO\_TABLESPACE
  - Undo tablespace name
  - Dynamic parameter
  - Can change with active transactions
    - New transactions use new
    - Original tablespace in use until all transactions complete

© 2004 SkillBuilders, Inc.

The four AUM-related parameters are described here:

- UNDO\_MANAGEMENT - Set equal to MANUAL or AUTO. This parameter is *not* dynamic. If AUTO is specified, Oracle will use an UNDO tablespace if available (details on the following slides). However, if an UNDO tablespace is not available, Oracle will use the SYSTEM rollback segment which is really NOT a good idea. In fact, Oracle gets so upset about having to use the SYSTEM rollback segment that it complains loudly by writing warnings to the alert log.
- UNDO\_TABLESPACE - Set to name of UNDO tablespace. This is a dynamic parameter. Note that if you change the UNDO tablespace with “alter system set UNDO\_TABLESPACE = TSNAME;” while there are active transactions, Oracle will continue to use the original tablespace until all active transactions have committed. Undo for new transactions is placed in the tablespace named in the ALTER SYSTEM command.

Notes continue on the next page...



16.15

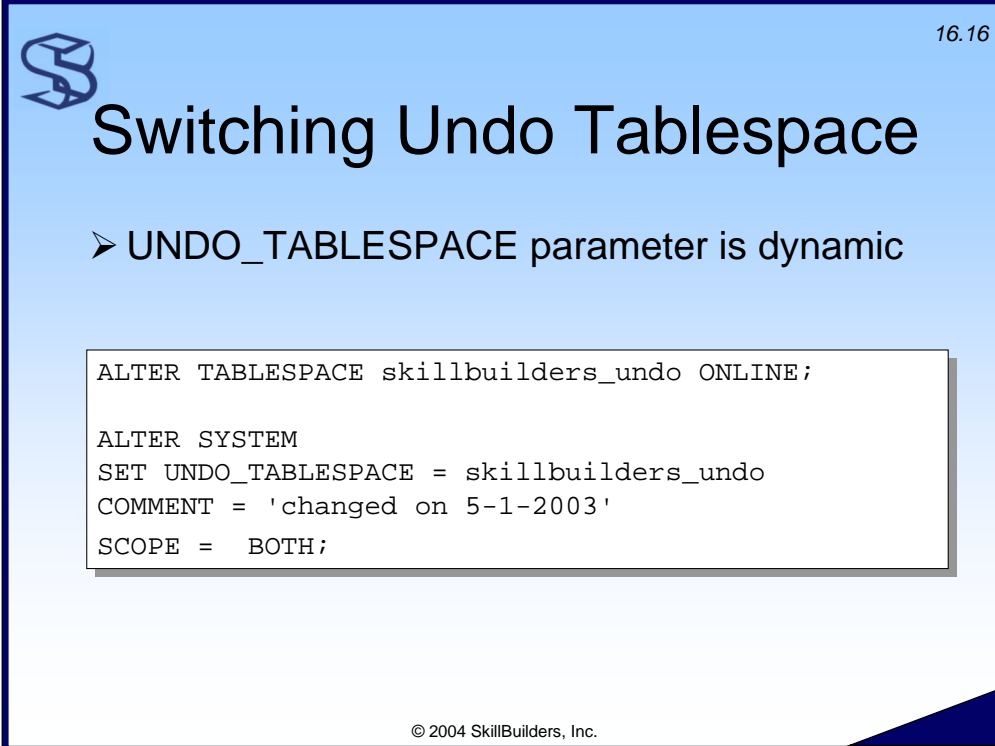
## ...Initialization Parameters

- UNDO\_RETENTION = 3600
  - Number of seconds to attempt to keep UNDO
  - Dynamic
  - Default = 900 seconds
- UNDO\_SUPPRESS\_ERRORS = TRUE
  - TRUE | FALSE
  - Support 8i apps with manual undo statements
  - e.g., SET TRANSACTION USE RBS
  - TRUE suppresses errors like ORA-30019
  - Can be set at SESSION level
- Bounce database

© 2004 SkillBuilders, Inc.

- UNDO\_RETENTION – Number of seconds Oracle should try to keep UNDO before being overwritten. Undo that is younger than the retention value is said to be “unexpired”. The default is 900 seconds. Set to a minimum of the longest running query in your database. Range is 0 to 2 Gb (approx 25,000 days!). Increasing provides a larger window for Flashback Query, but will cause Oracle to keep more undo, requiring a larger undo tablespace. If an out-of-space condition occurs in the tablespace, Oracle will start using unexpired undo, which can cause “snapshot too old” errors to occur. This is a dynamic parameter.
- UNDO\_SUPPRESS\_ERRORS - (TRUE|FALSE) – Provides support for applications written using manual UNDO commands such as SET TRANSACTION USE ROLLBACK SEGMENT. Suppresses errors such as ORA-30019. This parameter can be set at the system level (in the database parameter file or ALTER SYSTEM) or at the session level with ALTER SESSION.

After adding these parameters to your database parameter file, shutdown and restart your database to implement AUM.



16.16

## Switching Undo Tablespace


➤ UNDO\_TABLESPACE parameter is dynamic

```
ALTER TABLESPACE skillbuilders_undo ONLINE;  
  
ALTER SYSTEM  
SET UNDO_TABLESPACE = skillbuilders_undo  
COMMENT = 'changed on 5-1-2003'  
SCOPE = BOTH;
```

© 2004 SkillBuilders, Inc.

Assume that `UNDO_TABLESPACE = SKILLBUILDERS_OTHER_UNDO` has been specified and we would like to change the UNDO tablespace. To do so, we would issue the commands shown in this example.

Oracle places the old undo tablespace in `PENDING OFFLINE` mode which means that existing transactions continue but the tablespace will not accept new transaction undo. When all active transactions have committed, the tablespace is placed in `OFFLINE` mode.

16.17

## Dropping Undo Tablespace

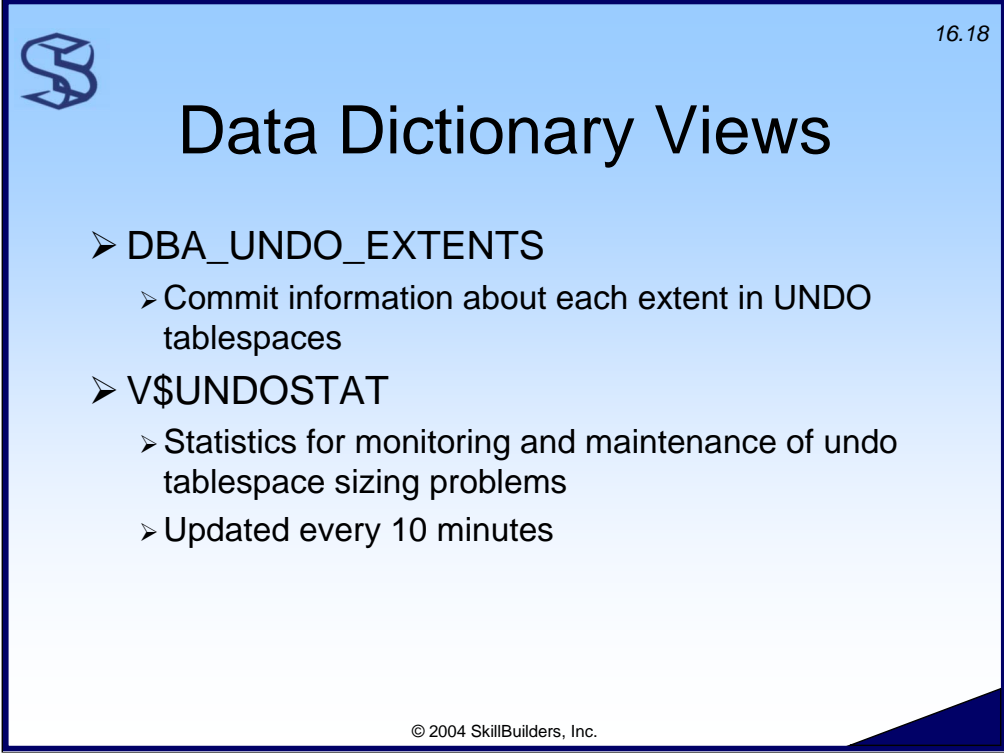
- Drop with DROP TABLESPACE command
- Must be inactive
- Undo retention time ignored

```
DROP TABLESPACE skillbuilders_undo ;
```

© 2004 SkillBuilders, Inc.

An UNDO tablespace may also be dropped provided that it is not in use by any instance and does not contain information about incomplete transactions.

Additionally, the UNDO\_RETENTION parameter is ignored during DROP. Oracle does not care that you may have specified a six month retention period. When the DROP is processed, the tablespace is history! It will, however, wait for active transactions to complete.



16.18

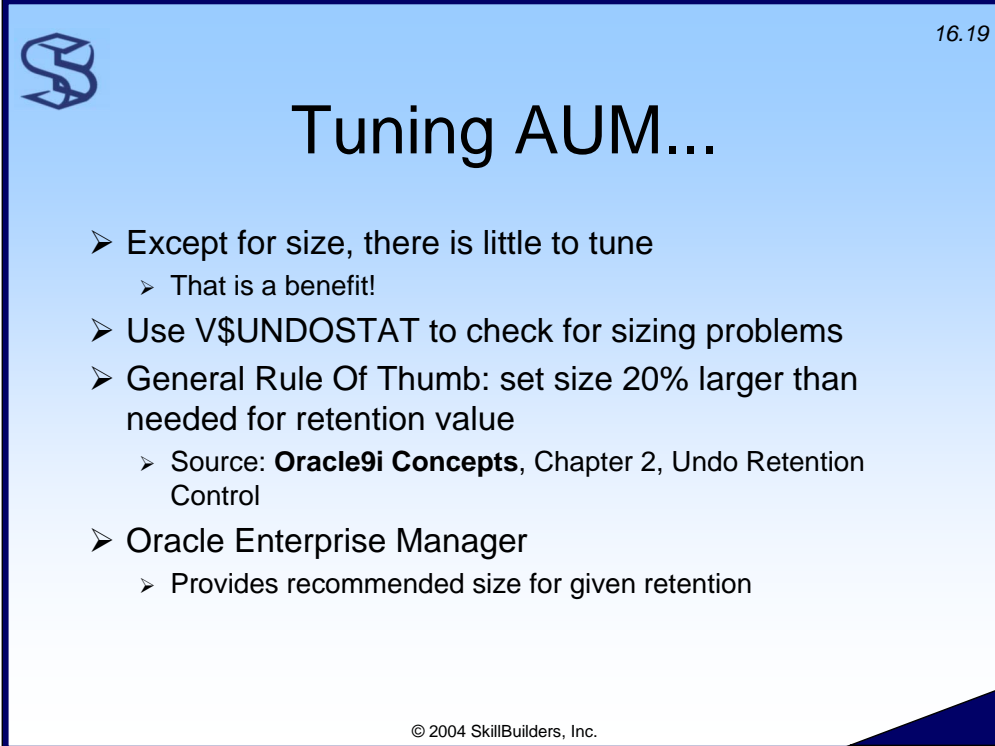
## Data Dictionary Views

- DBA\_UNDO\_EXTENTS
  - Commit information about each extent in UNDO tablespaces
- V\$UNDOSTAT
  - Statistics for monitoring and maintenance of undo tablespace sizing problems
  - Updated every 10 minutes

© 2004 SkillBuilders, Inc.

Data dictionary view DBA\_UNDO\_EXTENTS provides commit information about each extent in UNDO tablespaces. See the example in the *Tuning AUM* section of this lesson.

Dynamic performance view V\$UNDOSTAT provides Oracle-generated statistics for monitoring and maintenance of undo tablespace sizing problems. It works for both AUM and manual (rollback segment) undo.



16.19

## Tuning AUM...


- Except for size, there is little to tune
  - That is a benefit!
- Use V\$UNDOSTAT to check for sizing problems
- General Rule Of Thumb: set size 20% larger than needed for retention value
  - Source: **Oracle9i Concepts**, Chapter 2, Undo Retention Control
- Oracle Enterprise Manager
  - Provides recommended size for given retention

© 2004 SkillBuilders, Inc.

The whole point of AUM is the little maintenance required, especially compared to rollback segments. However, certain columns in the V\$UNDOSTAT dynamic view will help the DBA keep abreast, or better yet ahead, of any potential problems due to lack of space in the undo tablespace.

A general rule-of-thumb provided by the **Oracle Concepts** manual is to make the size of the undo tablespace 20% larger than what is needed for your specified retention value (see parameter UNDO\_RETENTION). Use the V\$UNDOSTAT.UNDOBLKS to determine the number of undo blocks used, and translate that into the undo tablespace size.

Oracle Enterprise Manager will recommend a size for the UNDO tablespace too.



16.20

## ...Tuning AUM...


- V\$UNDOSTAT provides a snapshot every 10 minutes
  - UNDOBLKS
    - Total number of undo blocks
  - UNXPSTEALCNT
    - Number of prematurely expired un-expired blocks
  - MAXQUERYLEN
    - Length, in seconds, of the longest-running transaction in the past 24 hours
  - NOSPACERRCNT
    - Number of space unavailable errors from writes of undo blocks

© 2004 SkillBuilders, Inc.

A description of helpful columns in V\$UNDOSTAT:

- UNDOBLKS – this column contains the total number of undo blocks generated during period of statistical collection. If this number is far larger than the UNDO tablespace's size, then the size of the undo tablespace needs to be increased.

Additional information is found on the next page...



16.21

## ...Tuning AUM...

➤ This shows not many blocks used by stealing occurring

```

1  select to_char(begin_time, 'hh:mi:ss' ) as time,
2         undoblks, unxpstealcnt
3  from v$undostat
4  where begin_time > sysdate - 1/24
5* order by 1 desc
SQL> /

```


TIME	UNDOBLKS	UNXPSTEALCNT
02:52:54	1	0
02:42:54	18	14
02:32:54	16	3

Limit output to last 60 minutes

Reduce stealing by increasing size of tspace

© 2004 SkillBuilders, Inc.

- UNXPSTEALCNT – Number of unexpired blocks (as set in parameter UNDO\_RETENTION) that are being expired prematurely; that is, other transactions have taken over those blocks because not enough UNDO space was available for those transactions. This column's value is particularly useful when a new application and database is brought online: the famous and frustrating “snapshot too old” error can appear as more and more users start to use the system, generating increasing numbers of undo blocks. The Oracle DBA can increase the value of parameter UNDO\_RETENTION\_TIME to get rid of such errors. Sometimes raising this value is not enough to get rid of all “snapshot too old” errors, and the DBA has to additionally add more space to the undo tablespace.
- MAXQUERYLEN – this column contains the value, in seconds, of the longest-running transaction in the past 24 hours. The undo management parameter UNDO\_RETENTION should be set to a value larger than the value in MAXQUERYLEN.
- NOSPACERRCNT – This column keep track of the number of any *space unavailable* errors generated when Oracle writes undo transaction blocks to an undo tablespace. If this value is not zero, increase the size of the undo tablespace.




16.22

## ...Tuning AUM

- V\$TRANSACTION and V\$ROLLSTAT are also useful for monitoring
  - V\$TRANSACTION contains information on active transactions
    - Start time
    - # of UNDO blocks used
    - # of physical I/Os
  - V\$ROLLSTAT contains information about undo/rollback segments
    - # of extents in rollback segment
    - Size of rollback segment
    - # of times the rollback segment size increased

© 2004 SkillBuilders, Inc.

There is also some useful transaction-related information in V\$TRANSACTION and V\$ROLLSTAT. V\$TRANSACTION contains information on active transactions. V\$ROLLSTAT contains information about undo segments if operating in AUM mode; rollback segment information if running in manual mode.

16.23


## Adding Space

- Add space to an undo tablespace by adding a datafile

```
ALTER TABLESPACE skillbuilders_undo  
ADD DATAFILE  
'c:\oracle9i\oradata\test\sb_undo02.dbf'  
SIZE 10M AUTOEXTEND ON;
```

© 2004 SkillBuilders, Inc.

If you determine that the undo tablespace is not large enough, adding space can be accomplished by adding a datafile to the tablespace as shown in this example.



16.24

## Resources

- **Oracle9i Administration** manual
  - lesson 13, Managing Undo Space
- **Oracle Concepts** Manual
  - lessons 2, 3 and 5
- Article by Jonathan Gennick
  - **Oracle8i Locally Managed Tablespaces**
  - <http://www.oracle.com/oramag/oracle/00-nov/index.html?o60o8i.html>
- Metalink.oracle.com
  - **Doc ID:** Note:135090.1

© 2004 SkillBuilders, Inc.



16.25

## Summary

- Automatic Undo Management can reduce rollback segment headaches
- Easy to setup and manage
  - Initialization parameters
  - V\$UNDOSTAT, DBA\_UNDO\_EXTENTS
- Provides explicit control over undo retention period
  - UNDO\_RETENTION parameter

© 2004 SkillBuilders, Inc.

A blue rectangular slide with a dark blue border. In the top right corner, the number '16.26' is displayed. On the left side, there is a logo consisting of a stylized 'S' and 'U' intertwined. To the right of the logo, the title 'Automatic Undo Management Workshop' is written in a large, bold, black font. Below the title, a bullet point with a right-pointing arrow is followed by the text 'Automatic Undo Management (AUM)'. At the bottom center of the slide, the copyright notice '© 2004 SkillBuilders, Inc.' is visible.

16.26

# Automatic Undo Management Workshop

➤ Automatic Undo Management (AUM)

© 2004 SkillBuilders, Inc.

## AUM Workshop

1. What are the current values of the four AUM parameters for your database?
2. Create a new undo tablespace called UNDO\_DATA\_TS. Use OMF for the location and filename if OMF is active. Create a 1M datafile with AUTOEXTEND OFF.
3. Make the tablespace UNDO\_DATA\_TS the undo tablespace for your database. Verify that the change worked.
4. Execute the supplied script AUM1.SQL. Ignore any “ORA-00942: table or view does not exist” error messages. Note that you *should* get an error: ORA-30036: unable to extend segment by 16 in undo tablespace ‘UNDO\_DATA\_TS’.
5. Query V\$UNDOSTAT to determine the number of undo blocks consumed in the last 60 minutes. Was there any attempted stealing of unexpired blocks? If you experienced any attempted stealing of unexpired blocks, what would you do to correct this?
6. Revert to the original undo tablespace. Rerun supplied script AUM1.SQL. Insure there are no errors.