

Oracle Recovery

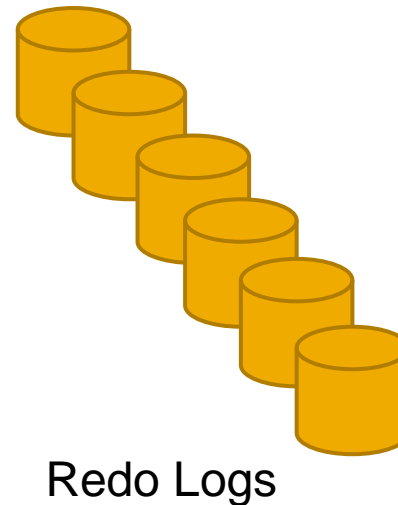
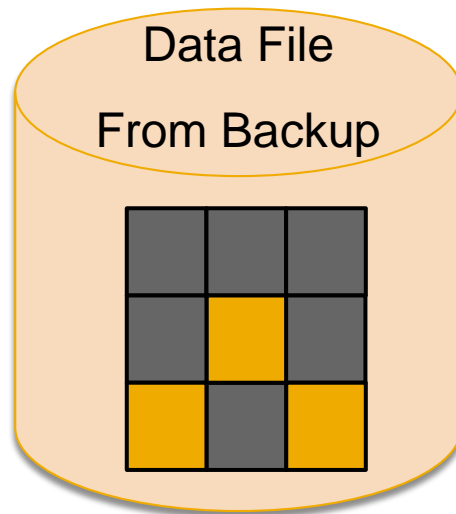
Mingqian (Tim) Zhong
June, 2014



What is Recovery?

What is recovery?

Oracle recovery is the process of **applying redo logs** to the **data files** so that finally the database can be opened in a **consistent state**.



When is Recovery required?

When is Recovery required?

Logical/Software Failure

- Oracle bug
- OS error
- Human error

Physical/Hardware Failure

- Power outage
- Storage crash

What happens
in real life?

Case 1

Company ABC has experienced a hardware failure and afterwards identified ABAP dumps due to error ORA-1578 (Block corruption).

They decided to take the following approaches

- Restore the last backup
- Execute a complete recovery

Case 1

```
SQL> startup
```

```
Database mounted.
```

```
ORA-01113: file 1 needs media recovery
```

```
ORA-01110: data file 1: '/oracle/ABC/system.dbf'
```

```
SQL> recover database
```

```
ORA-00283: recovery session canceled due to errors
```

```
ORA-01610: recovery using the BACKUP CONTROLFILE option  
must be done
```


Case 1

```
SQL> RECOVER DATABASE USING BACKUP CONTROLFILE UNTIL CANCEL
ORA-00279: change 21375038 generated at 09/21/2013 20:12:47 needed
for thread 1
ORA-00289: suggestion : /oracle/oraarch/arch1_4_633901491.dbf
ORA-00280: change 21375038 for thread 1 is in sequence #4
Specify log: {=suggested | filename | AUTO | CANCEL}
AUTO
ORA-01547: warning: RECOVER succeeded but OPEN RESETLOGS would get
error below
ORA-01195: online backup of file 1 needs more recovery to be
consistent
ORA-01110: data file 1: '/oracle/ABC/system.dbf'
```

What to do now???

Case 1

What should/could be done?

- Understand the situation!
 - What is BACKUP CONTROL FILE?
 - What is RESETLOGS?
 - What is BACKUP mode?
- Make a complete consistency check on the system!
 - 23345 Consistency check of ORACLE database
 - 365481 Block corruptions
 - 540463 FAQ Consistency Checks + Block Corruptions

Case 2

Company XYZ dropped a table mistakenly.

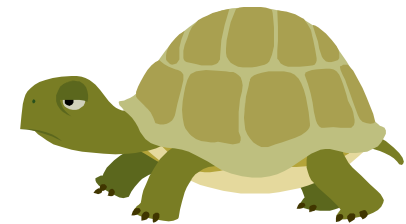
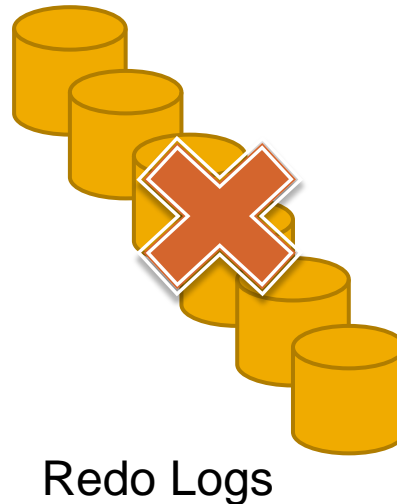
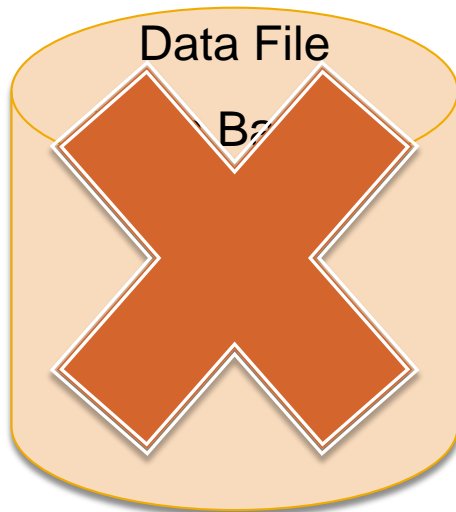
They decided to take the following approaches

- Restore the last backup to a test machine
- Recover the database until the timestamp before the table is dropped
- Export/Import the table to the current system

Case 2

What could go wrong? (Murphy's law)

- The backup itself is not readable ☹️
- Some archived redologs are not readable ☹️
- The recover procedure is too slow ☹️



Case 2

What should/could be done?

- Check the backup on a regular basis! (Apply redologs as a test)
- Use flashback technology so that recovery is not even required
- Recover only part of the database that contains the dropped table
- Active parallelism during recovery

Lessons Learned and Messages to Customers

- Unchecked backup is useless!
- Regular consistency check is necessary
- Test of recovery is important
 - Down time estimation
 - Get familiar with the procedure
- Do not make any rash decisions
- Understand all the technical details behind the commands, output, and information collected

What is Recovery Exactly?



What is recovery exactly?

Oracle recovery is all about

Synchronization!

- SCN
- RBA
- Data file
- Redo logs
- Control files

Goal

After this session you will be able to

- Understand that the essential of recovery is to synchronize
- Explain different types of SCN
- Explain checkpoint
- Explain relationships between SCN and data file status
- Explain different types of recovery
 - Complete
 - Incomplete
 - Using backup control file
 - Recreate control file (resetlogs / noresetlogs)

Outline

- 1. SCN**
2. Checkpoint
3. Database Status and Recovery
4. Tablespace and Datafile Status
5. Roadmap

SCN (System Change Number)

Overview

SCN

- System Change Number: Defines the committed version of the database.

RBA

- Redo Block Address: address of the redo entry

Control File

- Record all the related info for database recovery. (e.g. SCN, RBA)

SCN and RBA Format

SCN

- e.g.: 0x0000.0019237a
- SCN wrap (2 Bytes) and SCN base (4 Bytes)
- 48bits long, thus at a rate of 16384 SCN per second, SCN could be allocated for 534 years without running out. At this rate, an Oracle database will run out of SCN in June, 2522.

RBA

- e.g. rba:(0x35.2.10)
- Sequence# (4 Bytes), Block# (4 Bytes), Offset within the block (2 Bytes)

Control File Structure

Control file structure (only recover-related sections are listed):

- DATABASE ENTRY
- CHECKPOINT PROGRESS RECORDS
- REDO THREAD RECORDS
- LOG FILE RECORDS
- DATA FILE RECORDS
- TEMP FILE RECORDS
- TABLESPACE RECORDS
- LOG FILE HISTORY RECORDS
- ARCHIVED LOG RECORDS

Type of SCN (Location)

System checkpoint SCN(control file)

DATABASE ENTRY => v\$database.checkpoint_change#

Datafile checkpoint SCN(control file)

DATA FILE RECORDS => v\$datafile.checkpoint_change#

Stop SCN(control file and datafile header)

DATA FILE RECORDS => v\$datafile.last_change#

Start SCN (data file header)

v\$datafile_header.checkpoint_change#

Redo log SCN (control file)

LOG FILE RECORDS => v\$log.first_change#

LOG FILE HISTORY RECORDS =>

v\$log_history.first_change#/next_change#/resetlogs_change#

ARCHIVED LOG RECORDS => v\$archived_log.first_change#/next_change#

Current SCN :

```
SELECT DBMS_FLASHBACK.GET_SYSTEM_CHANGE_NUMBER,  
SCN_TO_TIMESTAMP(DBMS_FLASHBACK.GET_SYSTEM_CHANGE_NUMBER) FROM DUAL;  
SELECT TIMESTAMP_TO_SCN(SYSTIMESTAMP) AS SCN FROM DUAL;  
SELECT CURRENT_SCN FROM V$DATABASE;
```

System checkpoint SCN (control file – DATABASE ENTRY)

```
SELECT TO_CHAR(CHECKPOINT_CHANGE#, 'XXXXXXXX') SCN_HEX FROM V$DATABASE;  
SCN_HEX
```

```
-----
```

```
19237a
```

```
*****
```

DATABASE ENTRY

```
*****
```

```
(size = 316, compat size = 316, section max = 1, section in-use = 1,  
last-recid= 0, old-recno = 0, last-recno = 0)
```

```
(extent = 1, blkno = 1, numrecs = 1)
```

```
03/17/2011 15:28:50
```

```
DB Name "TIM"
```

```
Database flags = 0x00404001 0x00001200
```

```
Controlfile Creation Timestamp 03/17/2011 15:28:51
```

```
Incplmt recovery scn: 0x0000.00000000
```

```
Resetlogs scn: 0x0000.000bc19f Resetlogs Timestamp 03/17/2011 15:28:55
```

```
Prior resetlogs scn: 0x0000.00000001 Prior resetlogs Timestamp 09/05/2010 15:39:48
```

```
Redo Version: compatible=0xb200000
```

```
#Data files = 6, #Online files = 6
```

```
Database checkpoint: Thread=1 scn: 0x0000.0019237a
```

```
Threads: #Enabled=1, #Open=1, Head=1, Tail=1
```


Datafile checkpoint SCN and Stop SCN (control file – DATA FILE RECORDS)

```
SELECT FILE#,TO_CHAR(CHECKPOINT_CHANGE#, 'XXXXXXXX') DF_CKPT_SCN_HEX,TO_CHAR(LAST_CHANGE#, 'XXXXXXXX')  
STOP_SCN_HEX FROM V$DATAFILE;
```

```
FILE# DF_CKPT_S STOP_SCN_  
-----  
1      19237a  
2      19237a  
3      19237a  
4      19237a  
5      19237a  
6      19237a
```

DATA FILE RECORDS

```
(size = 520, compat size = 520, section max = 100, section in-use = 6,  
last-rcid= 76, old-recno = 0, last-recno = 0)  
(extent = 1, blkno = 11, numrecs = 100)
```

DATA FILE #1:

name #7: /oracle/TIM/oradata/tim/system01.dbf

creation size=0 block size=8192 status=0xe head=7 tail=7 dup=1

tablespace 0, index=1 krfil=1 prev_file=0

unrecoverable scn: 0x0000.00000000 01/01/1988 00:00:00

Checkpoint cnt:143 **scn: 0x0000.0019237a** 03/29/2011 10:45:20

Stop scn: 0xffff.ffffffff 03/29/2011 10:43:53

Creation Checkpointed at scn: 0x0000.00000007 09/05/2010 15:39:53

thread:0 rba:(0x0.0.0)

...

Online Checkpointed at scn: 0x0000.000bc19f 03/17/2011 15:28:55

thread:1 rba:(0x1.2.0)

Start SCN (data file header)

```
SELECT FILE#,TO_CHAR(CHECKPOINT_CHANGE#, 'XXXXXXXX') START_SCN_HEX FROM V$DATAFILE_HEADER;
```

```
FILE# START_SCN
-----
1      19237a
2      19237a
3      19237a
4      19237a
5      19237a
6      19237a
```

DATA FILE #1:

```
name #7: /oracle/TIM/oradata/tim/system01.dbf
creation size=0 block size=8192 status=0xe head=7 tail=7 dup=1
tablespace 0, index=1 krfil=1 prev_file=0
unrecoverable scn: 0x0000.00000000 01/01/1988 00:00:00
Checkpoint cnt:143 scn: 0x0000.0019237a 03/29/2011 10:45:20
Stop scn: 0xffff.ffffffff 03/29/2011 10:43:53
Creation Checkpointed at scn: 0x0000.00000007 09/05/2010 15:39:53
thread:0 rba:(0x0.0.0)
...
Offline scn: 0x0000.000bc19e prev_range: 0
Online Checkpointed at scn: 0x0000.000bc19f 03/17/2011 15:28:55
thread:1 rba:(0x1.2.0)
...
Checkpointed at scn: 0x0000.0019237a 03/29/2011 10:45:20
thread:1 rba:(0x35.2.10)
```

Redo log SCN (control file - LOG FILE RECORDS)

```
SELECT GROUP#, TO_CHAR(SEQUENCE#, 'XXXXXXX') SEQ_HEX, STATUS, TO_CHAR(FIRST_CHANGE#, 'XXXXXXX') FIRST_CHANGE_HEX,  
TO_CHAR(NEXT_CHANGE#, 'XXXXXXX') NEXT_CHANGE_HEX FROM V$LOG ORDER BY SEQUENCE#;
```

GROUP#	SEQ_HEX	STATUS	FIRST_CHA	NEXT_CHAN
3	33	INACTIVE	1855cf	18a21e
1	34	INACTIVE	18a21e	192379
2	35	CURRENT	192379	#####

LOG FILE RECORDS

```
(size = 72, compat size = 72, section max = 16, section in-use = 3,  
last-recid= 3, old-recno = 0, last-recno = 0)  
(extent = 1, blkno = 10, numrecs = 16)  
LOG FILE #1:  
  name #3: /oracle/TIM/oradata/tim/redo01.log  
  Thread 1 redo log links: forward: 2 backward: 0  
  siz: 0x19000 seq: 0x00000034 hws: 0x4 bsz: 512 nab: 0x6b25 flg: 0x1 dup: 1  
  Archive links: fwr: 0 back: 0 Prev scn: 0x0000.001855cf  
  Low scn: 0x0000.0018a21e 03/29/2011 05:00:13  
  Next scn: 0x0000.00192379 03/29/2011 10:45:20  
LOG FILE #2:  
  name #2: /oracle/TIM/oradata/tim/redo02.log  
  Thread 1 redo log links: forward: 3 backward: 1  
  siz: 0x19000 seq: 0x00000035 hws: 0x2 bsz: 512 nab: 0xffffffff flg: 0x8 dup: 1  
  Archive links: fwr: 0 back: 0 Prev scn: 0x0000.0018a21e  
  Low scn: 0x0000.00192379 03/29/2011 10:45:20  
  Next scn: 0xffff.ffffffff 01/01/1988 00:00:00  
LOG FILE #3:  
  name #1: /oracle/TIM/oradata/tim/redo03.log  
  Thread 1 redo log links: forward: 0 backward: 2  
  siz: 0x19000 seq: 0x00000033 hws: 0x3 bsz: 512 nab: 0x139f8 flg: 0x1 dup: 1  
  Archive links: fwr: 0 back: 0 Prev scn: 0x0000.001817fa  
  Low scn: 0x0000.001855cf 03/28/2011 22:01:31  
  Next scn: 0x0000.0018a21e 03/29/2011 05:00:13
```

Redo log SCN (control file - LOG FILE HISTORY RECORDS)

```
SELECT SEQUENCE#, TO_CHAR(FIRST_CHANGE#, 'XXXXXXXX') FIRST_CHANGE_HEX,
TO_CHAR(NEXT_CHANGE#, 'XXXXXXXX') NEXT_CHANGE_HEX FROM V$LOG_HISTORY ORDER BY SEQUENCE#;
OR
SELECT SEQUENCE#, TO_CHAR(FIRST_CHANGE#, 'XXXXXXXX') FIRST_CHANGE_HEX,
TO_CHAR(SWITCH_CHANGE#, 'XXXXXXXX') SWITCH_CHANGE_HEX FROM V$LOGHIST ORDER BY SEQUENCE#;
SEQUENCE# FIRST_CHA NEXT_CHAN
```

```
-----
      1      bc19f      c4bd9
      2      c4bd9      c59a1
      3      c59a1      c7bfc
      .....
     48     18176b     1817ba
     49     1817ba     1817fa
     50     1817fa     1855cf
     51     1855cf     18a21e
     52     18a21e     192379
```

LOG FILE HISTORY RECORDS

```
(size = 56, compat size = 56, section max = 292, section in-use = 52,
 last-recid= 52, old-recno = 1, last-recno = 52)
(extent = 1, blkno = 95, numrecs = 292)
```

Earliest record:

```
RECID #1 Recno 1 Record timestamp 03/17/11 15:45:50 Thread=1 Seq#=1 Link-Recid=0 kccic-Recid=2
 Low scn: 0x0000.000bc19f 03/17/11 15:28:55 Next scn: 0x0000.000c4bd9
```

Latest record:

```
RECID #52 Recno 52 Record timestamp 03/29/11 10:45:20 Thread=1 Seq#=52 Link-Recid=51 kccic-Recid=2
 Low scn: 0x0000.0018a21e 03/29/11 05:00:13 Next scn: 0x0000.00192379
RECID #51 Recno 51 Record timestamp 03/29/11 05:00:13 Thread=1 Seq#=51 Link-Recid=50 kccic-Recid=2
 Low scn: 0x0000.001855cf 03/28/11 22:01:31 Next scn: 0x0000.0018a21e
RECID #50 Recno 50 Record timestamp 03/28/11 22:01:31 Thread=1 Seq#=50 Link-Recid=49 kccic-Recid=2
 Low scn: 0x0000.001817fa 03/28/11 17:48:36 Next scn: 0x0000.001855cf
RECID #49 Recno 49 Record timestamp 03/28/11 17:48:36 Thread=1 Seq#=49 Link-Recid=48 kccic-Recid=2
 Low scn: 0x0000.001817ba 03/28/11 17:48:03 Next scn: 0x0000.001817fa
RECID #48 Recno 48 Record timestamp 03/28/11 17:48:03 Thread=1 Seq#=48 Link-Recid=47 kccic-Recid=2
 Low scn: 0x0000.0018176b 03/28/11 17:47:30 Next scn: 0x0000.001817ba
```

```
.....
RECID #3 Recno 3 Record timestamp 03/17/11 16:08:53 Thread=1 Seq#=3 Link-Recid=2 kccic-Recid=2
 Low scn: 0x0000.000c59a1 03/17/11 15:46:47 Next scn: 0x0000.000c7bfc
RECID #2 Recno 2 Record timestamp 03/17/11 15:46:47 Thread=1 Seq#=2 Link-Recid=1 kccic-Recid=2
 Low scn: 0x0000.000c4bd9 03/17/11 15:45:50 Next scn: 0x0000.000c59a1
RECID #1 Recno 1 Record timestamp 03/17/11 15:45:50 Thread=1 Seq#=1 Link-Recid=0 kccic-Recid=2
 Low scn: 0x0000.000bc19f 03/17/11 15:28:55 Next scn: 0x0000.000c4bd9
```

Redo log SCN (control file - ARCHIVED LOG RECORDS)

```

SELECT SUBSTR(NAME,58,24) NAME, SEQUENCE#, TO_CHAR(FIRST_CHANGE#,'XXXXXXXX') FIRST_CHANGE_HEX,
TO_CHAR(NEXT_CHANGE#,'XXXXXXXX') NEXT_CHANGE_HEX FROM V$ARCHIVED_LOG ORDER BY SEQUENCE#;
NAME SEQUENCE# FIRST_CHA NEXT_CHA
-----
o1_mf_1_42_6s0my2wy_.arc 42 17d38b 17d68f
o1_mf_1_43_6s15otq2_.arc 43 17d68f 1807c2
o1_mf_1_44_6s1c7rys_.arc 44 1807c2 18162f
o1_mf_1_45_6s1c9y46_.arc 45 18162f 18169d
o1_mf_1_46_6s1ccg8g_.arc 46 18169d 181706
o1_mf_1_47_6s1cdldp_.arc 47 181706 18176b
o1_mf_1_48_6s1cfmkk_.arc 48 18176b 1817ba
o1_mf_1_49_6s1cgnt0_.arc 49 1817ba 1817fa
o1_mf_1_50_6s1t8v50_.arc 50 1817fa 1855cf
o1_mf_1_51_6s2l5xb8_.arc 51 1855cf 18a21e
o1_mf_1_52_6s3712w2_.arc 52 18a21e 192379
  
```

ARCHIVED LOG RECORDS

```

(size = 584, compat size = 584, section max = 28, section in-use = 11,
 last-recid= 11, old-recno = 1, last-recno = 11)
(extent = 1, blkno = 98, numrecs = 28)
Earliest record:
RECID #1 Recno 1 Record timestamp 03/28/11 11:07:15 Thread=1 Seq#=42
Flags: <produced by archive operation> <created by the ARCH process>
Resetlogs scn and time scn: 0x0000.000bc19f 03/17/11 15:28:55
filename /oracle/TIM/fast_recovery_area/TIM/archivelog/2011_03_28/o1_mf_1_42_6s0my2wy_.arc
Low scn: 0x0000.0017d38b 03/28/11 10:51:37 Next scn: 0x0000.0017d68f 03/28/11 11:07:14
Block count=1146 Blocksize=512
Latest record:
RECID #11 Recno 11 Record timestamp 03/29/11 10:45:26 Thread=1 Seq#=52
Flags: <produced by archive operation> <created by the ARCH process>
Resetlogs scn and time scn: 0x0000.000bc19f 03/17/11 15:28:55
filename /oracle/TIM/fast_recovery_area/TIM/archivelog/2011_03_29/o1_mf_1_52_6s3712w2_.arc
Low scn: 0x0000.0018a21e 03/29/11 05:00:13 Next scn: 0x0000.00192379 03/29/11 10:45:20
Block count=27428 Blocksize=512
RECID #10 Recno 10 Record timestamp 03/29/11 05:00:17 Thread=1 Seq#=51
Flags: <produced by archive operation> <created by the ARCH process>
Resetlogs scn and time scn: 0x0000.000bc19f 03/17/11 15:28:55
filename /oracle/TIM/fast_recovery_area/TIM/archivelog/2011_03_29/o1_mf_1_51_6s2l5xb8_.arc
Low scn: 0x0000.001855cf 03/28/11 22:01:31 Next scn: 0x0000.0018a21e 03/29/11 05:00:13
Block count=80375 Blocksize=512
.....
RECID #2 Recno 2 Record timestamp 03/28/11 16:10:08 Thread=1 Seq#=43
Flags: <produced by archive operation> <created by the ARCH process>
Resetlogs scn and time scn: 0x0000.000bc19f 03/17/11 15:28:55
filename /oracle/TIM/fast_recovery_area/TIM/archivelog/2011_03_28/o1_mf_1_43_6s15otq2_.arc
Low scn: 0x0000.0017d68f 03/28/11 11:07:14 Next scn: 0x0000.001807c2 03/28/11 16:10:02
Block count=82569 Blocksize=512
RECID #1 Recno 1 Record timestamp 03/28/11 11:07:15 Thread=1 Seq#=42
Flags: <produced by archive operation> <created by the ARCH process>
Resetlogs scn and time scn: 0x0000.000bc19f 03/17/11 15:28:55
filename /oracle/TIM/fast_recovery_area/TIM/archivelog/2011_03_28/o1_mf_1_42_6s0my2wy_.arc
Low scn: 0x0000.0017d38b 03/28/11 10:51:37 Next scn: 0x0000.0017d68f 03/28/11 11:07:14
Block count=1146 Blocksize=512
  
```

Current SCN

```
SELECT
    'DBMS_FLASHBACK' SOURCE,
    DBMS_FLASHBACK.GET_SYSTEM_CHANGE_NUMBER SCN
FROM DUAL
UNION ALL
SELECT
    'SYSTIMESTAMP' SOURCE,
    TIMESTAMP_TO_SCN(SYSTIMESTAMP) SCN
FROM DUAL
UNION ALL
SELECT
    'V$DATABASE' SOURCE,
    CURRENT_SCN SCN
FROM V$DATABASE;
```

SOURCE	SCN
-----	-----
DBMS_FLASHBACK	16117156
SYSTIMESTAMP	16117156
V\$DATABASE	16117156

Outline

1. SCN
- 2. Checkpoint**
3. Database Status and Recovery
4. Tablespace and Datafile Status
5. Roadmap

Checkpoint

Checkpoint Overview

Definition of Checkpoint

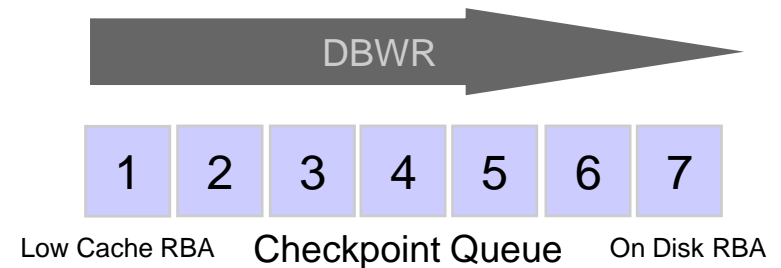
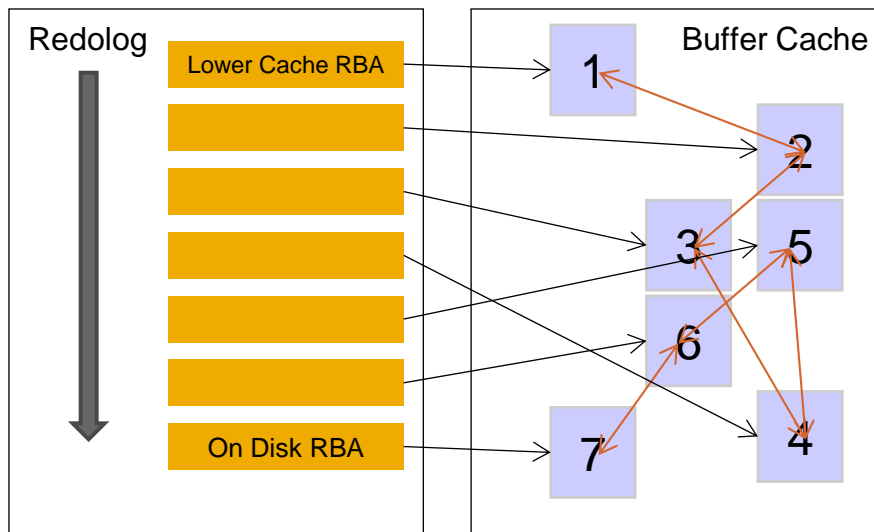
- A time point that guarantee all dirty blocks that were dirtied before an SCN are flushed to disk.
- Checkpoint is used to determine where the recovery should start.

Type of Checkpoint

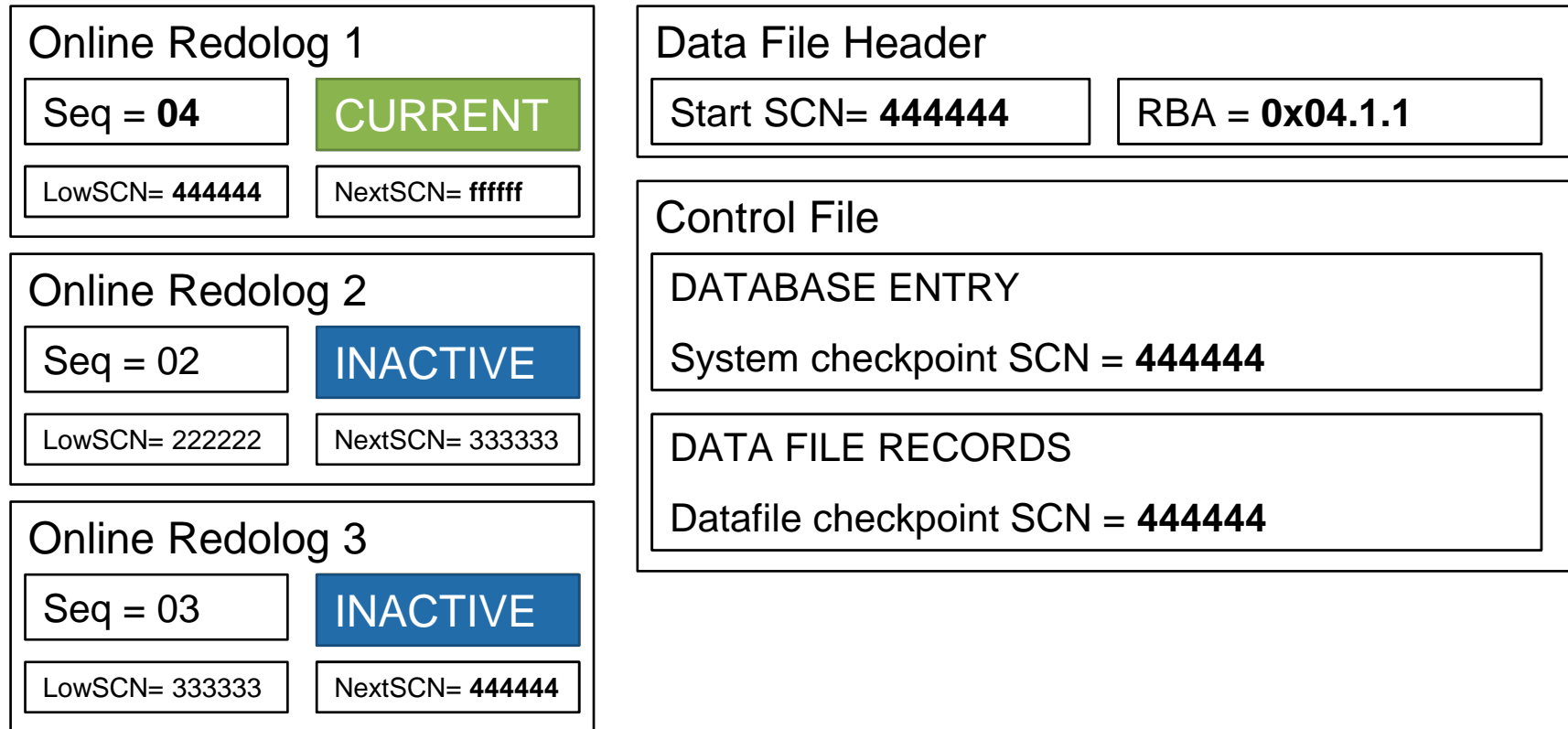
- 1. Log Switch Checkpoint
- 2. Full Checkpoint
- 3. Incremental Checkpoint
- 4. DDL/Object Checkpoint

Checkpoint Queue

- All the dirty blocks in buffer cache maintain a two-direction link in the block header
- Checkpoint queue is a list of dirty buffers in chronological order of when they were first dirtied
- The RBA of redolog entries are linked to the DBA of the blocks in the buffer cache.



1. Log Switch Checkpoint - Demo



CKPT reads the following:
Current SCN = 444444
Current RBA from Log buffer (0x04.1.1)

DBWR Finished!
All the related SCN/RBA are updated.

Log Switch Checkpoint Finished!

1. Log Switch Checkpoint – Detail Procedure 1

1. The status of next redo log becomes: CURRENT. The previous redo log becomes: ACTIVE
2. In control file: REDO THREAD RECORDS, LOG FILE RECORDS are updated to reflect the status change; LOG FILE RECORDS is updated with the current SCN

GROUP#	SEQ_HEX	STATUS	FIRST_CHA	NEXT_CHAN
1	3d	INACTIVE	1babea	1be6b7
2	3e	INACTIVE	1be6b7	1c094c
3	3f	CURRENT	1c094c	#####

alter system switch logfile;

GROUP#	SEQ_HEX	STATUS	FIRST_CHA	NEXT_CHAN
2	3e	INACTIVE	1be6b7	1c094c
3	3f	ACTIVE	1c094c	1c4304
1	40	CURRENT	1c4304	#####

3. CKPT set the target checkpoint to the first RBA in the CURRENT redo log file and triggers DBWR to do its job
Fri Apr 01 11:23:52 2011

Beginning log switch checkpoint up to RBA [0x40.2.10], SCN: 1852164 [0x1c4304]

Thread 1 advanced to log sequence 64 (LGWR switch)

Current log# 1 seq# 64 mem# 0: /oracle/TIM/oradata/tim/redo01.log

Fri Apr 01 11:23:56 2011

Archived Log entry 22 added for thread 1 sequence 63 ID 0x376ad4f2 dest 1:

Fri Apr 01 11:28:45 2011

Completed checkpoint up to RBA [0x40.2.10], SCN: 1852164

1. Log Switch Checkpoint – Detail Procedure 2

4. When DBWR finishes its job

- The previous redo log becomes INACTIVE and is updated in control file LOG FILE RECORDS

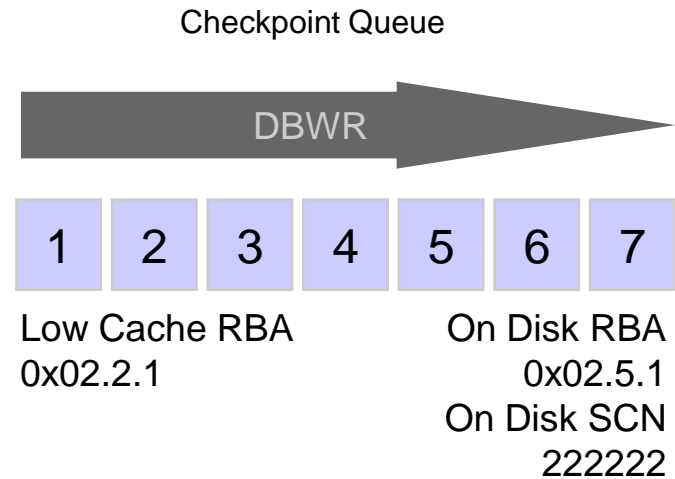
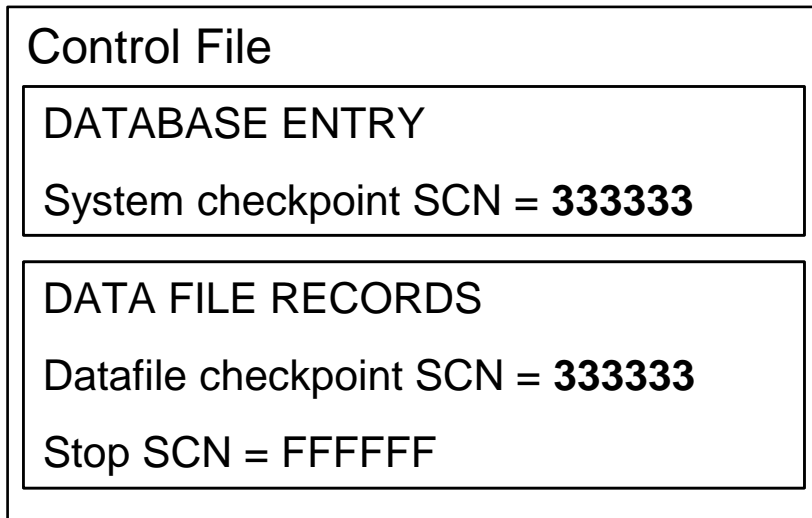
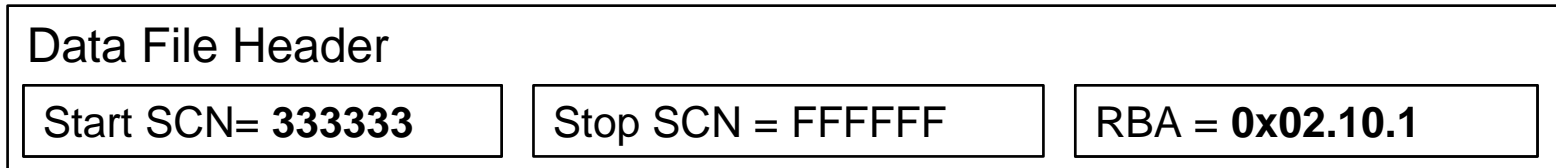
GROUP#	SEQ_HEX	STATUS	FIRST_CHA	NEXT_CHAN
2	3e	INACTIVE	1be6b7	1c094c
3	3f	INACTIVE	1c094c	1c4304
1	40	CURRENT	1c4304	#####

- CKPT read SCN recorded when log switch starts from control file LOG FILE RECORDS and update the following:
 - SCN in control file (REDO THREAD RECORDS)
 - System checkpoint SCN (control file DATABASE ENTRY) (v\$database.checkpoint_change#)
 - Datafile checkpoint SCN(control file DATA FILE RECORDS) (v\$datafile.checkpoint_change#)
 - Start SCN (data file header) (v\$datafile_header.checkpoint_change#)

1. Log Switch Checkpoint – Detail Procedure 3

- CKPT check the RBA that points to the first entry in the current redo log and update the following:
 - RBA in Control file (REDO THREAD RECORDS)
 - RBA in each data file header
 - RBA is NOT maintained in control file DATA FILE RECORDS

2. Full Checkpoint - Demo



DBWR Start to flush all the dirty blocks from the beginning of the checkpoint queue until the current tail of the checkpoint queue.

DBWR finished its job.
CKPT update all the related SCN and RBA info

Full Checkpoint Finished !

2. Full Checkpoint – Detail Procedure

1. CKPT check the **on disk RBA and its SCN (or current?)** in control file (CHECKPOINT PROGRESS RECORDS) and set the checkpoint target to this “on disk RBA”, so that all the blocks dirtied before this time point will be flushed to data file.
2. DBWR is immediately triggered and write all dirty blocks from checkpoint queue to the data file
 - No DML is allowed (no dirty block should be generated) before full checkpoint complete
3. After DBWR finishes its job:
 - CKPT update the target SCN into the following:
 - SCN in control file (REDO THREAD RECORDS)
 - System checkpoint SCN (control file DATABASE ENTRY)
(v\$database.checkpoint_change#)
 - Datafile checkpoint SCN(control file DATA FILE RECORDS)
(v\$datafile.checkpoint_change#)
 - Start SCN (data file header) (v\$datafile_header.checkpoint_change#)
 - *Stop SCN (control file DATA FILE RECORDS and data file header)
(v\$datafile.last_change#) [Only under some circumstances]*
 - CKPT update the target RBA to the following:
 - RBA in control file (REDO THREAD RECORDS)
 - RBA in each data file header
 - RBA in control file (CHECKPOINT PROGRESS RECORDS) will be found to be ffffffff.fffffff.ffff, which means no dirty block exist, it only happens after full checkpoint for a short while and will be updated soon again during the heartbeat

3. Incremental Checkpoint - Control File Heartbeat

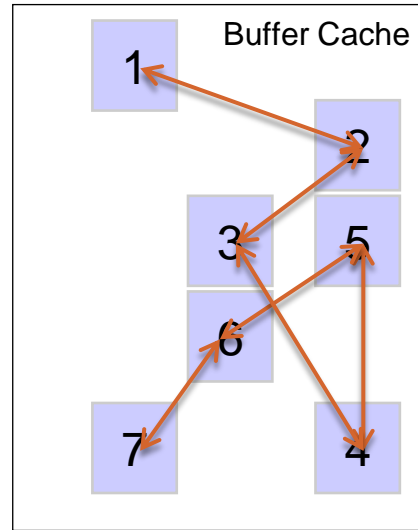
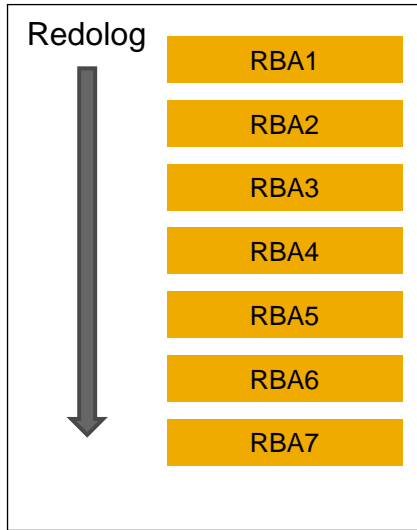
- Every 3 seconds, CKPT write the RBA that is corresponding to the first entry in checkpoint queue to Control file (CHECKPOINT PROGRESS RECORDS) (**low cache rba**)
- All the dirty block with respect to the redo before **low cache rba** are already written to the data file by DBWR.
- CKPT write the RBA of last entry in checkpoint queue (i.e. the last entry in current online redo log) to control file (CHECKPOINT PROGRESS RECORDS) (**on disk rba**)
- The SCN of the last dirty block in checkpoint queue is called **on disk scn** (linked to **on disk rba**)

```
*****
CHECKPOINT PROGRESS RECORDS
*****
(size = 8180, compat size = 8180, section max =
11, section in-use = 0,
last-recid= 0, old-recno = 0, last-recno = 0)
(extent = 1, blkno = 2, numrecs = 11)
THREAD #1 - status:0x2 flags:0x0 dirty:76
low cache rba: (0x35.a06c.0)
on disk rba: (0x35.a1c2.0)
on disk scn: 0x0000.0019637d 03/29/2011 16:09:30
resetlogs scn: 0x0000.000bc19f 03/17/2011 15:28:55
heartbeat: 747076922 mount id: 930769268
```



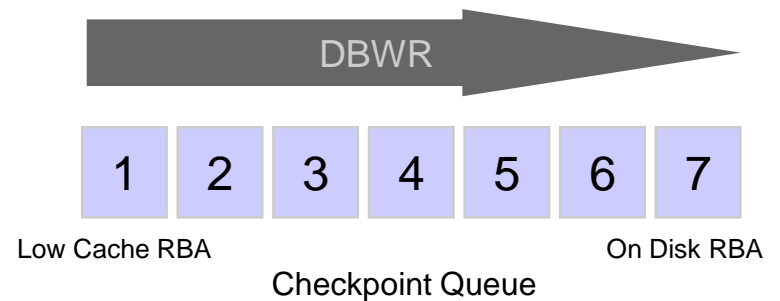
```
*****
CHECKPOINT PROGRESS RECORDS
*****
(size = 8180, compat size = 8180, section max =
11, section in-use = 0,
last-recid= 0, old-recno = 0, last-recno = 0)
(extent = 1, blkno = 2, numrecs = 11)
THREAD #1 - status:0x2 flags:0x0 dirty:30
low cache rba: (0x35.a1b8.0)
on disk rba: (0x35.a275.0)
on disk scn: 0x0000.00196461 03/29/2011 16:13:36
resetlogs scn: 0x0000.000bc19f 03/17/2011 15:28:55
heartbeat: 747077009 mount id: 930769268
```

3. Incremental Checkpoint - Demo



Incremental Checkpoint Started!

Incremental Checkpoint Finished!



Redo Entry



Redo Entry Not Required for Recovery

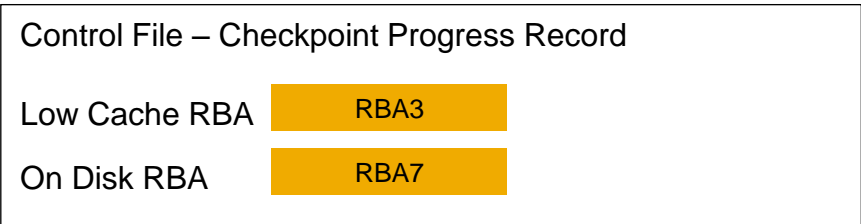


Dirty Block



Non dirty Block

Control File Heartbeat!



3. Incremental Checkpoint - Summary

- The purpose of the incremental checkpoint is to reduce the instance recovery time.
- Incremental checkpoint is running continuously all the time.
- CKPT regularly triggers DBWR to write from the beginning of the checkpoint queue until a target RBA.
- CKPT will NOT wait until DBWR finish its job, instead, CKPT just monitor the progress and update the info every 3 seconds in the control file (heartbeat).
- CKPT will trigger the incremental checkpoint based on the following 4 parameters plus one internal factor. These parameters basically determine the **maximum checkpoint queue length** that is tolerable. So if many parameters are active, then the factor that determines shortest tolerable checkpoint queue length will override others.
 - **FAST_START_IO_TARGET (0)**
Maximum number of I/O operations on the Data Files to be processed during recovery (not used anymore from 9i). Replaced by the second parameter
 - **FAST_START_MTTR_TARGET (0)**
Number of seconds the database takes to perform crash recovery
 - **LOG_CHECKPOINT_TIMEOUT (1800)**
This parameter signifies that no buffer will remain dirty (in the cache) for more than 1800 seconds.
 - **LOG_CHECKPOINT_INTERVAL (0)**
Number of redo log file blocks required for recovery.
 - **90% OF SMALLEST REDO LOG**

4. DDL/Object Checkpoint

1. Occurred during command like
'DROP/TRUNCATE TABLE/INDEX/PARTITION'.
2. All the dirty blocks corresponding to the involved objects will be flushed to disk.

Checkpoint - Comparison

	1. Incremental Checkpoint	2. Log Switch Checkpoint	3. Full Checkpoint
Trigger Condition	Controlled by 4 parameters + 1 factor	<ul style="list-style-type: none"> Alter system switch log Current redo log is full 	<ul style="list-style-type: none"> Alter system checkpoint Shutdown (except abort) Alter tablespace begin backup Alter tablespace offline (*) Alter tablespace read only Alter tablespace read write (back from read only)
Where is SCN updated	Do NOT update SCN	<ul style="list-style-type: none"> SCN in control file (REDO THREAD RECORDS) System checkpoint SCN (control file DATABASE ENTRY) (v\$database.checkpoint_change#) Datafile checkpoint SCN(control file DATA FILE RECORDS) (v\$datafile.checkpoint_change#) Start SCN (data file header) (v\$datafile_header.checkpoint_change#) 	<ul style="list-style-type: none"> SCN in control file (REDO THREAD RECORDS) System checkpoint SCN (control file DATABASE ENTRY) (v\$database.checkpoint_change#) Datafile checkpoint SCN(control file DATA FILE RECORDS) (v\$datafile.checkpoint_change#) Start SCN (data file header) (v\$datafile_header.checkpoint_change#) Stop SCN (control file DATA FILE RECORDS and data file header) (v\$datafile.last_change#) (**)
Where is RBA updated	Update RBA to Control file (CHECKPOINT PROGRESS RECORDS) as low cache rba	<ul style="list-style-type: none"> Update RBA in Control file (REDO THREAD RECORDS) Update RBA in each data file header 	<ul style="list-style-type: none"> Update RBA in Control file (REDO THREAD RECORDS) Update RBA in each data file header

Outline

1. SCN
2. Checkpoint
- 3. Database Status and Recovery**
4. Tablespace and Datafile Status
5. Roadmap

Database Status and Recovery

Database Status and Recovery

Shutdown

- SHUTDOWN NORMAL / IMMEDIATE

SCN Check before Database Open and During Recovery

- Old Data File -> Media Recovery
- Instance Crash -> Instance Recovery
- Datafile in offline mode
- Tablespace in offline mode
- Old Control File -> Recover using backup control file

Recovery

- Complete Media Recovery with Latest Control File
- Instance Recovery (Automatically done after Database open)
- Incomplete Recovery:
 - Redo log is lost -> Resetlog
 - Control file is old / Database Structure change -> Rebuild control file

Database Status and Recovery

Shutdown

- SHUTDOWN NORMAL / IMMEDIATE

SCN Check before Database Open and During Recovery

- Old Data File -> Media Recovery
- Instance Crash -> Instance Recovery
- Datafile in offline mode
- Tablespace in offline mode
- Old Control File -> Recover using backup control file

Recovery

- Complete Media Recovery with Latest Control File
- Instance Recovery (Automatically done after Database open)
- Incomplete Recovery:
 - Redo log is lost -> Resetlog
 - Control file is old / Database Structure change -> Rebuild control file

SHUTDOWN NORMAL / IMMEDIATE - Demo

Data File 1 Header

Start SCN= **222222**

Stop SCN = **222222**

RBA = **0x02.2.1**

Data File 2 Header

Start SCN= **222222**

Stop SCN = **222222**

RBA = **0x02.2.1**

Control File

DATABASE ENTRY
System checkpoint SCN = **222222**

DATA FILE 1 RECORDS
Datafile checkpoint SCN = **222222**
Stop SCN = **222222**
DATA FILE 2 RECORDS
Datafile checkpoint SCN = **222222**
Stop SCN = **222222**

A full checkpoint is triggered on all data files.
CKPT update SCN and RBA accordingly

If a database is shutdown cleanly, the STOP
SCN is synchronized with all other SCN, it is no
longer FFFFFFFF

SHUTDOWN NORMAL / IMMEDIATE; @ 222222

Shutdown - Description

SHUTDOWN NORMAL / IMMEDIATE;

- A full checkpoint is triggered on all the data files.
- Current SCN are updated in the following area:
 - System checkpoint SCN (control file DATABASE ENTRY) (v\$database.checkpoint_change#)
 - Datafile checkpoint SCN(control file DATA FILE RECORDS) (v\$datafile.checkpoint_change#)
 - Start SCN (data file header) (v\$datafile_header.checkpoint_change#)
 - Stop SCN (control file DATA FILE RECORDS and data file header) (v\$datafile.last_change#), it is no longer FFFFFFFF.
- RBA are updated in all the datafile headers.

SHUTDOWN ABORT; / Instance failure

- No full checkpoint is triggered.
- Stop SCN is still FFFFFFFF.

Database Status and Recovery

Shutdown

- SHUTDOWN NORMAL / IMMEDIATE

SCN Check before Database Open and During Recovery

- Old Data File -> Media Recovery
- Instance Crash -> Instance Recovery
- Datafile in offline mode
- Tablespace in offline mode
- Old Control File -> Recover using backup control file

Recovery

- Complete Media Recovery with Latest Control File
- Instance Recovery (Automatically done after Database open)
- Incomplete Recovery:
 - Redo log is lost -> Resetlog
 - Control file is old / Database Structure change -> Rebuild control file

SCN Check before Database Open – Demo 1

Data File 1 Header

Start SCN= 333333

Stop SCN = FFFFFFFF

RBA = 0x03.3.1

Data File 2 Header

Start SCN= 333333

Stop SCN = FFFFFFFF

RBA = 0x03.3.1

Control File

DATABASE ENTRY

System checkpoint SCN = 333333

DATA FILE 1 RECORDS

Datafile checkpoint SCN = 333333

Stop SCN = 333333

DATA FILE 2 RECORDS

Datafile checkpoint SCN = 333333

Stop SCN = 333333

Reason:

Data file is restored from an old backup

Media recovery is required:

Redolog starting from RBA=0x01.1.1 is needed for Datafile 1

Redolog starting from RBA=0x02.2.1 is needed for Datafile 2

Case 1: Datafile checkpoint SCN > Start SCN

SCN Check before Database Open – Demo 2

Data File 1 Header

Start SCN= 222222

Stop SCN = FFFFFFFF

RBA = 0x02.1.1

Data File 2 Header

Start SCN= 222222

Stop SCN = FFFFFFFF

RBA = 0x02.1.1

Control File

DATABASE ENTRY

System checkpoint SCN = 222222

DATA FILE 1 RECORDS

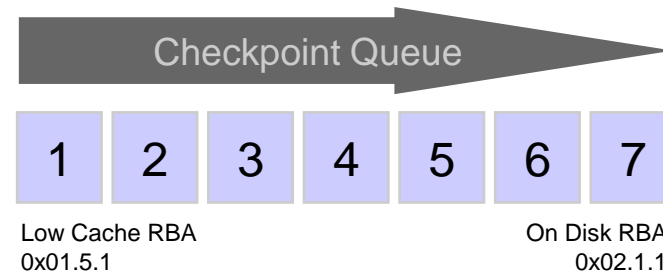
Datafile checkpoint SCN = 222222

Stop SCN = FFFFFFFF

CKPT PROGRESS RECORDS

Low Cache RBA = 0x02.1.1

On Disk RBA = 0x02.8.1



Reason:

Power failure / Instance crash

Instance recovery is required:

Redolog starting from low cache RBA until on disk RBA will be applied.

Due to incremental checkpoint, it is not required to apply the redo entry from the beginning of a redo log -> save time

Case 2: Start SCN <> Stop SCN and Stop SCN = FFFFFFFF

SCN Check before Database Open – Demo 3

Data File 1 Header

Start SCN= 333333

Stop SCN = FFFFFFFF

RBA = 0x03.3.1

Data File 2 Header

Start SCN= 333333

Stop SCN = FFFFFFFF

RBA = 0x03.3.1

Control File

DATABASE ENTRY
System checkpoint SCN = 333333

DATA FILE 1 RECORDS
Datafile checkpoint SCN = 333333
Stop SCN = FFFFFFFF
DATA FILE 2 RECORDS
Datafile checkpoint SCN = 333333
Stop SCN = FFFFFFFF

Reason:

Datafile is in offline mode

Media recovery not required to open the database.

Media recovery required to bring the datafile online.

Case 3: Start SCN <> Stop SCN and Start SCN <> 0 and Stop SCN <> FFFFFFFF

SCN Check before Database Open – Demo 4

Data File 1 Header

Start SCN= 333333

Stop SCN = FFFFFFFF

RBA = 0x03.3.1

Data File 2 Header

Start SCN= 333333

Stop SCN = FFFFFFFF

RBA = 0x03.3.1

Control File

DATABASE ENTRY
System checkpoint SCN = 333333

DATA FILE 1 RECORDS
Datafile checkpoint SCN = 333333
Stop SCN = FFFFFFFF
DATA FILE 2 RECORDS
Datafile checkpoint SCN = 333333
Stop SCN = FFFFFFFF

Reason:

Tablespace is in offline mode

Media recovery not required to open the database.

Media recovery not required to bring the tablespace online:

Only the SCN and RBA will be updated.

Case 4: Start SCN = 0

SCN Check before Database Open – Demo 5

Data File 1 Header

Start SCN= 222222

Stop SCN = 222222

RBA = 0x02.2.1

Data File 2 Header

Start SCN= 222222

Stop SCN = 222222

RBA = 0x02.2.1

Control File

DATABASE ENTRY

System checkpoint SCN = 111111

DATA FILE 1 RECORDS

Datafile checkpoint SCN = 111111

Stop SCN = 111111

DATA FILE 2 RECORDS

Datafile checkpoint SCN = 111111

Stop SCN = 111111

Reason:

Control File is not up to date

```
SQL> recover database using backup  
controlfile;
```

Rebuild control file

Rebuild control file resetlog

Rebuild control file noresetlog

Case 5: System checkpoint SCN < Start SCN

SCN Check before Database Open

Datafile Checkpoint SCN (ctrl file DATA FILE RECORDS) > Start SCN (data file header)

- Reason => Data file is restored from a previous backup
- Media recovery is required:
 - Check RBA in each data file header as a start point to apply all the redo logs

Start SCN (data file header) <> Stop SCN (ctrl file DATA FILE RECORDS and data file header) and Stop SCN = FFFFFFFF

- Reason => Power failure / Instance crash
- Instance recovery required:
 - Check low cache RBA in ctrl file (CHECKPOINT PROGRESS RECORDS), start to apply the corresponding redo log until on disk RBA from ctrl file (CHECKPOINT PROGRESS RECORDS)
 - Because of the incremental checkpoint, we do not need to apply all the redo log entries from the beginning of a redo log.

Start SCN (data file header) <> Stop SCN (ctrl file DATA FILE RECORDS and data file header) and Start SCN <> 0 and Stop SCN <> FFFFFFFF

- Reason => Datafile in offline mode
- Datafile remains offline and database could be opened.
- The datafile need recovery to be brought back to online.

Start SCN (data file header) = 0

- Reason => Tablespace in offline mode
- No recovery required, the tablespace remains offline and database could be opened.

System Checkpoint SCN (ctrl file DATABASE ENTRY) < any of Start SCN (data file header)

- Reason => Control file not up to date; We have two choices:
 - Recover database using backup controlfile;
 - Rebuild control file, also with two choices:
 - Rebuild control file resetlog
 - Rebuild control file noresetlog

After all the SCN are synchronized, set the Stop SCN (both in ctrl file DATA FILE RECORDS and each data file header) of all the online datafiles to FFFFFFFF and then the database is open.

Database Status and Recovery

Shutdown

- SHUTDOWN NORMAL / IMMEDIATE

SCN Check before Database Open and During Recovery

- Old Data File -> Media Recovery
- Instance Crash -> Instance Recovery
- Datafile in offline mode
- Tablespace in offline mode
- Old Control File -> Recover using backup control file

Recovery

- Complete Media Recovery with Latest Control File
- Instance Recovery (Automatically done after Database open)
- Incomplete Recovery:
 - Redo log is lost -> Resetlog
 - Control file is old / Database Structure change -> Rebuild control file

Recovery – Complete Media Recovery with Latest Control File - Demo

Data File Header #1

Start SCN= 200000

Stop SCN = FFFFFFFF

RBA = 0x20.3.10

Data File Header #2

Start SCN= 200000

Stop SCN = FFFFFFFF

RBA = 0x20.3.10

Control File

DATABASE ENTRY
System checkpoint SCN = 200000

DATA FILE RECORDS #1
Datafile checkpoint SCN = 200000
Stop SCN = FFFFFFFF
DATA FILE RECORDS #2
Datafile checkpoint SCN = 200000
Stop SCN = FFFFFFFF

Archived Redolog 10

10.1

10.2

10.3

10.4

Archived Redolog 11

11.1

11.2

11.3

11.4

Online Redolog 20

...

20.1

20.2

20.3

Now the Database can be opened.

Recovery – Complete Media Recovery with Latest Control File - Description

Procedure

- Get RBA in each data file header (so that the sequence of the first required archived redo log is known)
- Read control file (ARCHIVED LOG RECORDS) to get the location of the archived log file
- Start to apply all the redo logs (both archived redo logs and online redo logs following the sequence)
- Complete recovery done

Recovery – Complete Media Recovery with Latest Control File - Description

Procedure

- Restore
- STARTUP MOUNT (Latest Control file info is read)

===System Checkpoint SCN from Control File===

SystemCKPTSCN SystemCKPTSCNHEX

819814

C8266

===Datafile Checkpoint SCN from Control File===

FileNumber DataFileCKPTSCN DataFileCKPTSCNHEX DataFileStopSCN DataFileStopSCNHEX

1

819814

C8266

819814

C8266

...

6

819814

C8266

819814

C8266

===Start SCN from Data File Header===

FileNumber StartSCNDFH StartSCNDFHHEX

1

782909

BF23D

...

6

782909

BF23D

Recovery – Complete Media Recovery with Latest Control File - Description

- ALTER DATABASE OPEN;

ERROR at line 1:

ORA-01113: file 1 needs media recovery

ORA-01110: data file 1: '/oracle/TIM/oradata/TIM/system01.dbf'

- RECOVER DATAFILE 6;

ORA-00279: change 782909 generated at 04/18/2011 18:00:38 needed for thread 1

ORA-00289: suggestion :

[/oracle/TIM/fast_recovery_area/TIM/archivelog/2011_04_18/o1_mf_1_11_6ts9m98y_.arc](#)

ORA-00280: change 782909 for thread 1 is in sequence #11

Specify log: {<RET>=suggested | filename | AUTO | CANCEL}

ORA-00279: change 809960 generated at 04/19/2011 09:00:40 needed for thread 1

ORA-00289: suggestion :

[/oracle/TIM/fast_recovery_area/TIM/archivelog/2011_04_19/o1_mf_1_14_6tv2hxy7_.arc](#)

ORA-00280: change 809960 for thread 1 is in sequence #14

Specify log: {<RET>=suggested | filename | AUTO | CANCEL}

[#15,#16 and #17 are online redologs]

Log applied.

Media recovery complete.

Recovery – Complete Media Recovery with Latest Control File - Description

- Compare the Start SCN of datafile 6 with other datafiles:

===System Checkpoint SCN from Control File===

SystemCKPTSCN	SystemCKPTSCNHEX
---------------	------------------

819814	C8266
--------	-------

===Start SCN from Data File Header===

FileNumber	StartSCNDFH	StartSCNDFHHEX
------------	-------------	----------------

1	782909	BF23D
---	--------	-------

2	782909	BF23D
---	--------	-------

3	782909	BF23D
---	--------	-------

4	782909	BF23D
---	--------	-------

6	819812	C8264
---	--------	-------

- We could see that datafile 6 is now synchronized with the system checkpoint SCN.
- When all the datafiles are synchronized then the database is considered to be complete recovered and then can be opened.

Recovery – Instance Recovery - Demo

Data File Header #1

Start SCN= 200000

Stop SCN = FFFFFFFF

RBA = 0x20.3.10

Data File Header #2

Start SCN= 200000

Stop SCN = FFFFFFFF

RBA = 0x20.3.10

Control File

DATABASE ENTRY

System checkpoint SCN = 200000

CHECKPOINT PROGRESS RECORDS

Low cache RBA = 19.3 On disk SCN = 200000

DATA FILE RECORDS #1

Datafile checkpoint SCN = 200000

Stop SCN = FFFFFFFF

DATA FILE RECORDS #2

Datafile checkpoint SCN = 200000

Stop SCN = FFFFFFFF

Online Redolog 19

19.1

19.2

19.3

19.4

Online Redolog 20

20.1

20.2

20.3

**Stop SCN are set to FFFFFFFF
and Database could be
opened.**

Recovery – Instance Recovery - Description

Procedure

- Check low cache RBA in ctrl file (CHECKPOINT PROGRESS RECORDS), start to apply the corresponding redo log until on disk RBA from ctrl file (CHECKPOINT PROGRESS RECORDS)
- Because of the incremental checkpoint, we do not need to apply all the redo log entries from the beginning of a redo log.

Recovery – Incomplete Recovery

Reason

- **Redo log is lost:** database can only be recovered until the end of the first non-broken consecutive redologs starting from the backup. Afterwards database has to be opened with Resetlogs.
- **Control file is from old backup:** oracle then lose the information of what is 'Current' so it will always consider the recovery to be an incomplete recovery even though in fact after the recovery the database could be in a complete status given all the redologs are available.

Solution

- **Resetlog:** this is to make sure that after an incomplete recovery, redo logs after this point of time will not be able to apply to the backups before this point of time.
- **Rebuild control file:** required if the current control file is lost and/or the database structure has been changed since last backup.

Incomplete Recovery – Resetlogs

When do we need to resetlogs

- After point-in-time/incomplete media recovery
- Recover using backup control files

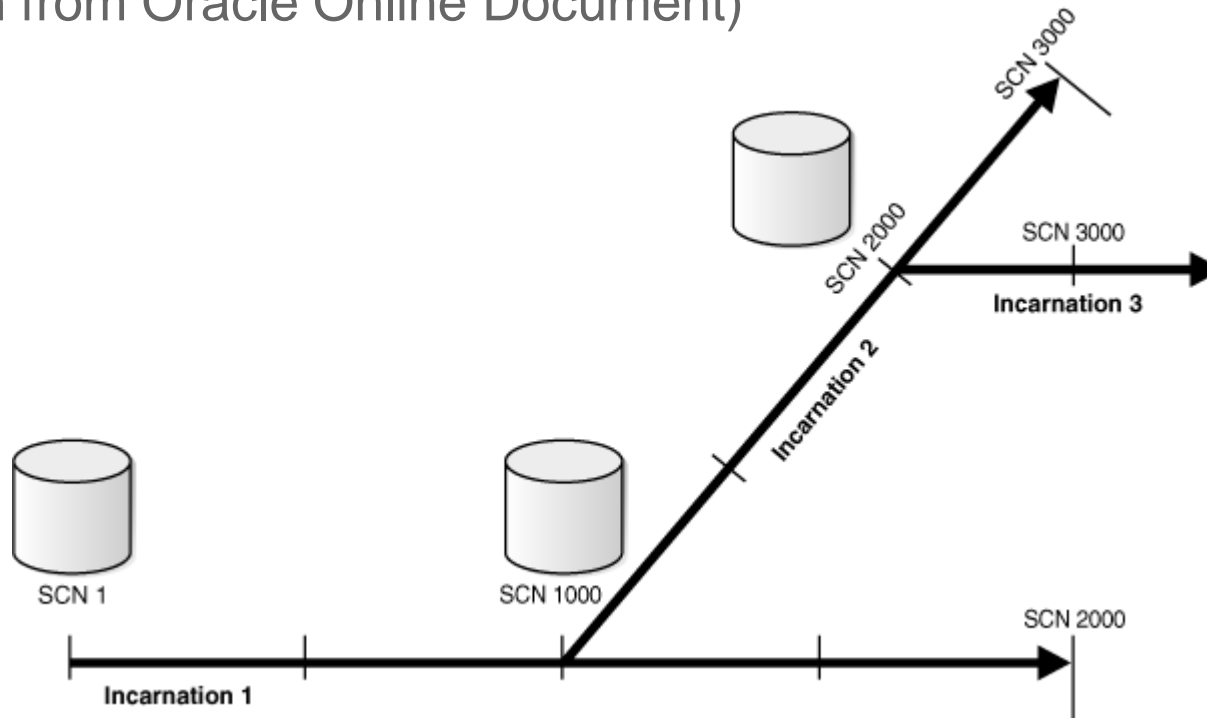
What is done during resetlogs

- All the online redo logs are archived and the content within are deleted
- Sequence of online redo log is reset to 1
- Update following:
 - control file (REDO THREAD RECORDS, LOG FILE RECORDS, LOG FILE HISTORY RECORDS, ARCHIVED LOG RECORDS) with the log sequence change info
 - data file header, redo log header, subsequent archived redo log header and control file (DATABASE ENTRY, CHECKPOINT PROGRESS RECORDS, ARCHIVED LOG RECORDS, INCARNATION RECORDS) with **RESETLOG SCN** and time stamp
- Oracle will never apply an archived logs to a data file if the RESETLOG SCN does NOT match
- Since now the RESETLOG SCN in the data file header has been changed, all the previous archived redo log are ORPHANED/USELESS

Incomplete Recovery – Resetlogs

Why do we need resetlogs

(figure taken from Oracle Online Document)



Incomplete Recovery – Recover using backup control file - Demo

Data File Header #1

Start SCN= 200000

Stop SCN = FFFFFFFF

RBA = 0x20.1.10

Data File Header #2

Start SCN= 200000

Stop SCN = FFFFFFFF

RBA = 0x20.1.10

Control File

DATABASE ENTRY
System checkpoint SCN = 200000

DATA FILE RECORDS #1
Datafile checkpoint SCN = 200000
Stop SCN = FFFFFFFF
DATA FILE RECORDS #2
Datafile checkpoint SCN = 200000
Stop SCN = FFFFFFFF

Archived Redolog 10

10.1

10.2

10.3

10.4

Archived Redolog 11

11.1

11.2

11.3

11.4

Online Redolog 20

...

20.1

20.2

20.3

So we need to open the database resetlogs as it is considered as an incomplete recovery

Incomplete Recovery – Recover using backup control file - Description

Procedure

- Check Start SCN (data file header) and RBA to guess for the archived redo log required for recovery
- Got the correct location of the redo log and apply
- Even if the current online redo log is applied, Oracle still considers it as an incomplete recovery
- OPEN DATABASE RESETLOGS is obliged

Media recovery complete.

```
SQL> alter database open;
```

```
alter database open
```

```
*
```

```
ERROR at line 1:
```

```
ORA-01589: must use RESETLOGS or NORESETLOGS option for database open
```

```
SQL> alter database open noresetlogs;
```

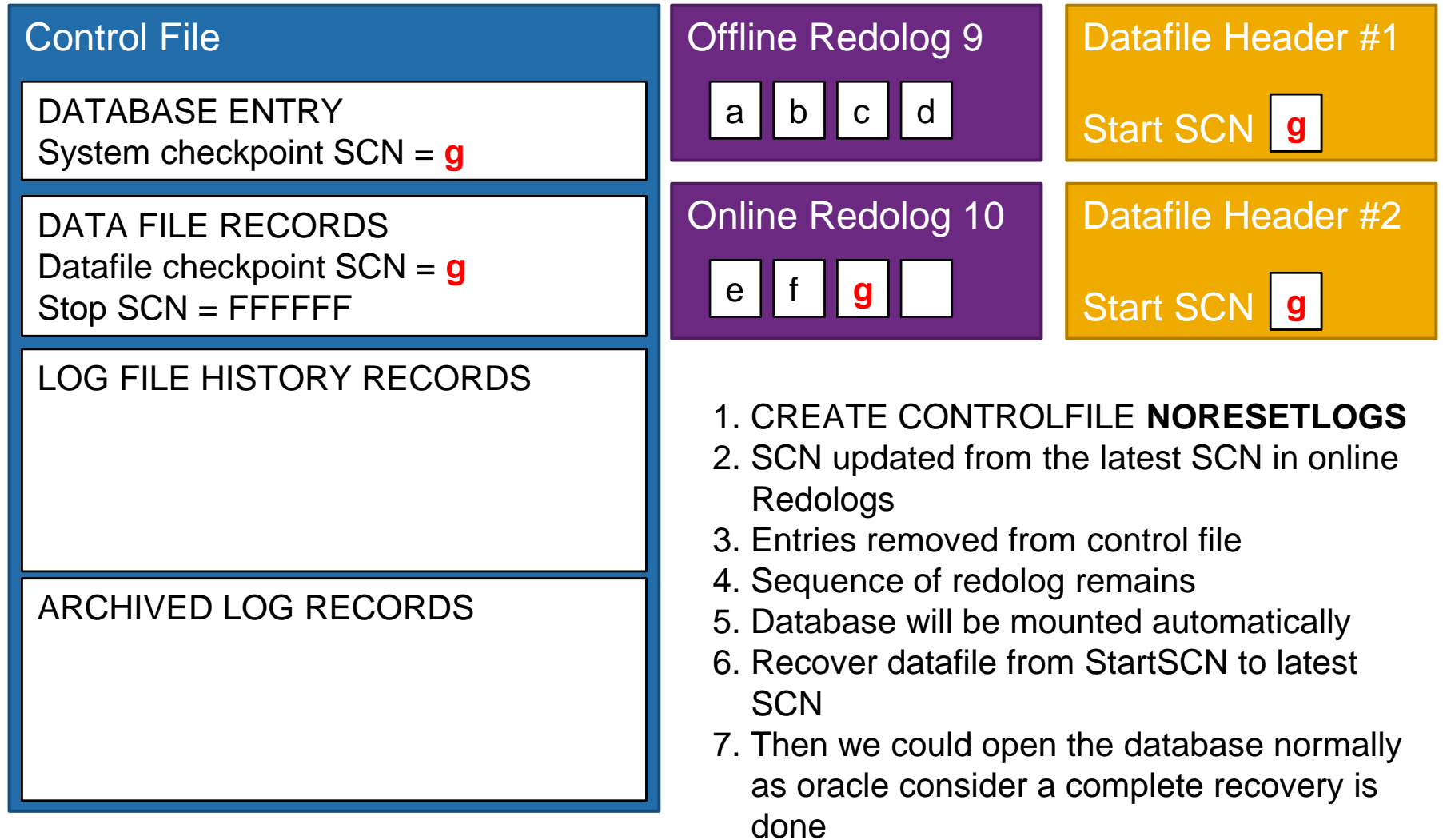
```
alter database open noresetlogs
```

```
*
```

```
ERROR at line 1:
```

```
ORA-01588: must use RESETLOGS option for database open
```

Incomplete Recovery - Recover using rebuilt control file (NORESETLOGS) - Demo



Incomplete Recovery - Recover using rebuilt control file (NORESETLOGS) - Description

Procedure

- Get any control file and STARTUP MOUNT
- ALTER DATABASE BACKUP CONTROLFILE TO TRACE;
- Two versions of control file will be found in the trace file:
- NORESETLOGS version: used when all online redo logs are available.

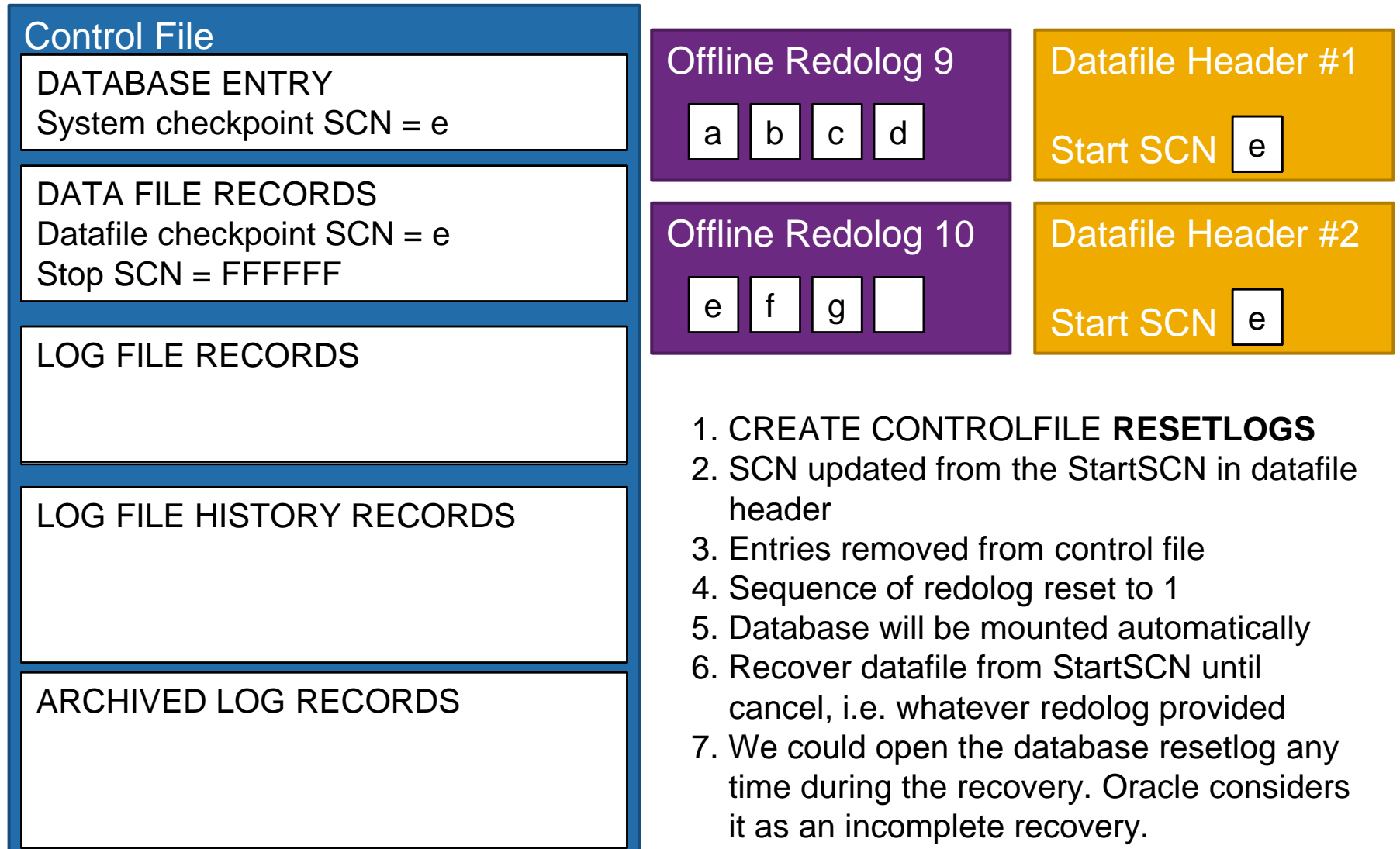
```
STARTUP NOMOUNT
CREATE CONTROLFILE REUSE DATABASE "TIM" NORESETLOGS ARCHIVELOG
  MAXLOGFILES 16
  MAXLOGMEMBERS 3
  MAXDATAFILES 100
  MAXINSTANCES 8
  MAXLOGHISTORY 292
LOGFILE
  GROUP 1 '/oracle/TIM/oradata/TIM/redo01.log' SIZE 10M BLOCKSIZE 512,
  GROUP 2 '/oracle/TIM/oradata/TIM/redo02.log' SIZE 10M BLOCKSIZE 512,
  GROUP 3 '/oracle/TIM/oradata/TIM/redo03.log' SIZE 10M BLOCKSIZE 512
DATAFILE
  '/oracle/TIM/oradata/TIM/system01.dbf',
  '/oracle/TIM/oradata/TIM/sysaux01.dbf',
  '/oracle/TIM/oradata/TIM/undotbs01.dbf',
  '/oracle/TIM/oradata/TIM/users01.dbf',
  '/oracle/TIM/oradata/TIM/users02.dbf'
CHARACTER SET WE8MSWIN1252;
RECOVER DATABASE
ALTER SYSTEM ARCHIVE LOG ALL;
ALTER DATABASE OPEN;
ALTER TABLESPACE TEMP ADD TEMPFILE '/oracle/TIM/oradata/TIM/temp01.dbf' REUSE;
```

Create Control File NORESETLOGS - Description

Procedure

- Start database NOMOUNT
- Create new control file (NORESETLOGS)
 - System checkpoint SCN and Datafile checkpoint SCN in control file will be updated by the latest SCN from Online Redolog.
 - Previous archived log history are deleted. (LOG FILE HISTORY RECORDS and ARCHIVED LOG RECORDS removed from control file)
- Database will be automatically mounted after the new control file is created
- All datafiles need to be fully recovered (from StartSCN to Latest SCN)
- Open database with the new control file
 - Previous Online redologs will be archived and log history info will be updated.
 - Online redologs are emptied, but the sequence number remains.

Incomplete Recovery - Recover using rebuilt control file (RESETLOGS) - Demo



Incomplete Recovery - Recover using rebuilt control file (RESETLOGS) - Description

Procedure

- RESETLOGS version

```
STARTUP NOMOUNT
CREATE CONTROLFILE REUSE DATABASE "TIM" RESETLOGS ARCHIVELOG
  MAXLOGFILES 16
  MAXLOGMEMBERS 3
  MAXDATAFILES 100
  MAXINSTANCES 8
  MAXLOGHISTORY 292
LOGFILE
  GROUP 1 '/oracle/TIM/oradata/TIM/redo01.log' SIZE 10M BLOCKSIZE 512,
  GROUP 2 '/oracle/TIM/oradata/TIM/redo02.log' SIZE 10M BLOCKSIZE 512,
  GROUP 3 '/oracle/TIM/oradata/TIM/redo03.log' SIZE 10M BLOCKSIZE 512
DATAFILE
  '/oracle/TIM/oradata/TIM/system01.dbf',
  '/oracle/TIM/oradata/TIM/sysaux01.dbf',
  '/oracle/TIM/oradata/TIM/undotbs01.dbf',
  '/oracle/TIM/oradata/TIM/users01.dbf',
  '/oracle/TIM/oradata/TIM/users02.dbf'
CHARACTER SET WE8MSWIN1252;
RECOVER DATABASE USING BACKUP CONTROLFILE
ALTER DATABASE OPEN RESETLOGS;
ALTER TABLESPACE TEMP ADD TEMPFILE '/oracle/TIM/oradata/TIM/temp01.dbf' REUSE;
```

Create Control File RESETLOGS - Description

Procedure

- Start database NOMOUNT
- Create new control file (RESETLOGS)
 - Current online redolog info are deleted. (LOG FILE RECORDS removed from control file)
 - Previous archived log history are deleted. (LOG FILE HISTORY RECORDS and ARCHIVED LOG RECORDS removed from control file)
- Database will be automatically mounted after the new control file is created
- Datafiles could be recovered until cancel
- Open database with the new control file (RESETLOGS)
 - Previous Online redologs will be archived and log history info will be updated.
 - Online redologs are emptied, the sequence is reset to 1.

Recovery - Comparison

	Recover using current control file(normal complete recovery)	Recover using backup control files	Create control file noresetlogs	Create control file resetlogs
What is updated from where	Not Relevant	Not Relevant	<ul style="list-style-type: none"> System checkpoint SCN (control file DATABASE ENTRY) from most recent SCN in current online redo log Datafile checkpoint SCN (control file DATA FILE RECORDS) from most recent SCN in current online redo log Control file REDO THREAD RECORDS and LOG FILE RECORDS from online redo log info 	<ul style="list-style-type: none"> Datafile checkpoint SCN (control file DATA FILE RECORDS) from the Start SCN (data filer header)
What is reset / not reset	Not Relevant	Not Relevant	<ul style="list-style-type: none"> Control file (LOG FILE HISTORY RECORDS, ARCHIVED LOG RECORDS) Archived log information are lost!! Sequence of log is not reset 	<ul style="list-style-type: none"> System checkpoint SCN (control file DATABASE ENTRY) Control file REDO THREAD RECORDS, LOG FILE RECORDS, LOG FILE HISTORY RECORDS, ARCHIVED LOG RECORDS) All log information are lost!! Sequence of log is reset
Recover starting from when/ SCN to when/SCN	Recover from Start SCN in each data file header until Database SCN in control file automatically.	Recover from StartSCN in each data file header to whatever is provided (Redo log)	<ul style="list-style-type: none"> Recover from start SCN in each data file header until the most recent SCN from online redo log Must do full recovery before open the database 	<ul style="list-style-type: none"> Recover from start SCN in each data file header until what ever is provided Could open the database immediately (incomplete recovery) Could also apply redo log as needed
Open database afterwards		<ul style="list-style-type: none"> Must open database resetlogs as oracle consider it as a incomplete recovery Need to provide redo log info manually 	<ul style="list-style-type: none"> Open database normal and oracle think it is a complete recovery. Recovery is done automatically 	<ul style="list-style-type: none"> Open database resetlogs and oracle think it is an incomplete recovery Recovery is done automatically
Why and when to use	Normal recovery	Control files are from backup	<ul style="list-style-type: none"> Will check if the Online redo log exist. Current online redologs are good Current control file is missing / Structure change to database after last backup 	<ul style="list-style-type: none"> Will not check if the online redo log exist or not Useful when online redologs are missing Current control file is missing / Structure change to database after last backup

Recovery - Comparison

	Open database noresetlogs	Open database resetlogs
Create control file noresetlogs	<ul style="list-style-type: none">▪ Must do full recovery before open the database	<ul style="list-style-type: none">▪ Possible after an incomplete recovery
Create control file resetlogs	<ul style="list-style-type: none">▪ Not possible	<ul style="list-style-type: none">▪ Could open immediately but will be an incomplete recovery▪ Could also apply logs until current, but still considered by Oracle as incomplete recovery and must open with reset logs

Outline

1. SCN
2. Checkpoint
3. Database Status and Recovery
- 4. Tablespace and Datafile Status**
5. Roadmap

Tablespace and Datafile Status

Tablespace and Datafile Status

Tablespace Status

- Read Only – Read Write
- Begin Backup Mode – End Backup Mode
- Offline – Online

Datafile Status

- Offline – Online

Tablespace – Read Only and Read Write - Demo

Data File Header - Involved

Start SCN= **333333**

Stop SCN = **FFFFFF**

RBA = **0x03.3.1**

Data File Header - Other

Start SCN= 111111

Stop SCN = FFFFFFFF

RBA = 0x01.1.1

Control File

DATABASE ENTRY
System checkpoint SCN = 111111

DATA FILE RECORDS - Involved
Datafile checkpoint SCN = **333333**
Stop SCN = **FFFFFF**
DATA FILE RECORDS - Other
Datafile checkpoint SCN = 111111
Stop SCN = FFFFFFFF

No checkpoint is required.
CKPT just update all the related SCN and RBA info

ALTER TABLESPACE READ WRITE; @ 333333

Tablespace – Read Only and Read Write - Description

ALTER TABLESPACE <NAME> READ ONLY;

- Have to wait until all the active transactions started before to commit or rollback.
- A full checkpoint is triggered on all the data files belong to the tablespace.
- SCN are updated in the following area:
 - Datafile checkpoint SCN(control file DATA FILE RECORDS) (v\$datafile.checkpoint_change#)
 - Start SCN (data file header) (v\$datafile_header.checkpoint_change#)
 - Stop SCN (control file DATA FILE RECORDS and data file header) (v\$datafile.last_change#), it is no longer FFFFFFFF.
- RBA are updated in all the datafile headers belong to the read only tablespace
- The procedure is the same as SHUTDOWN DATABASE

ALTER TABLESPACE <NAME> READ WRITE;

- SCN and RBA are updated into the same place as above.
- Stop SCN are set to FFFFFFFF

Tablespace – Begin Backup Mode and End Backup Mode - Demo

Data File Header - Involved

Start SCN= **333333**

Stop SCN = FFFFFFFF

RBA = **0x03.3.1**

Data File Header - Other

Start SCN= **333333**

Stop SCN = FFFFFFFF

RBA = **0x03.3.1**

Control File

DATABASE ENTRY

System checkpoint SCN = **333333**

DATA FILE RECORDS - Involved

Datafile checkpoint SCN = **333333**

Stop SCN = FFFFFFFF

DATA FILE RECORDS - Other

Datafile checkpoint SCN = **333333**

Stop SCN = FFFFFFFF

A checkpoint up to the System SCN (or the Datafile checkpoint SCN of other datafiles) is triggered on all the data files belong to the tablespace.

ALTER TABLESPACE END BACKUP;

Tablespace – Begin Backup Mode and End Backup Mode - Description

ALTER TABLESPACE <NAME> BEGIN BACKUP;

- Will not wait for active transactions to commit or rollback.
- A full checkpoint is triggered on all the data files belong to the tablespace.
- SCN are updated in the following area:
 - Datafile checkpoint SCN(control file DATA FILE RECORDS) (v\$datafile.checkpoint_change#)
 - Start SCN (data file header) (v\$datafile_header.checkpoint_change#)
- RBA are updated in all the datafile headers belong to backup mode tablespace

ALTER TABLESPACE <NAME> END BACKUP;

- A checkpoint up to the System SCN (or the Datafile checkpoint SCN of other datafiles) is triggered on all the data files belong to the tablespace.
- Related SCN and RBA are updated.

Tablespace – Offline and Online - Demo

Data File Header - Involved

Start SCN= **333333**

Stop SCN = **FFFFFF**

RBA = **0x03.3.1**

Data File Header - Other

Start SCN= 111111

Stop SCN = FFFFFFFF

RBA = 0x01.1.1

Control File

DATABASE ENTRY

System checkpoint SCN = 111111

DATA FILE RECORDS - Involved

Datafile checkpoint SCN = **333333**

Stop SCN = **FFFFFF**

DATA FILE RECORDS - Other

Datafile checkpoint SCN = 111111

Stop SCN = FFFFFFFF

No checkpoint is required.

CKPT just update all the related SCN and RBA info

ALTER TABLESPACE ONLINE; @ 333333

Tablespace – Offline and Online - Description

ALTER TABLESPACE <NAME> OFFLINE NORMAL;

- Will not wait for active transactions to commit or rollback.
- A full checkpoint is triggered on all the data files belong to the tablespace.
- SCN are updated in the following area:
 - Datafile checkpoint SCN(control file DATA FILE RECORDS) (v\$datafile.checkpoint_change#)
 - Stop SCN (control file DATA FILE RECORDS and data file header) (v\$datafile.last_change#), it is no longer FFFFFFFF.
- Start SCN (data file header) (v\$datafile_header.checkpoint_change#) of the involved datafiles are set to 0.
- RBA are updated in all the datafile headers belong to the read only tablespace.
- Media recovery is not required to bring the tablespace back to ONLINE.

ALTER TABLESPACE <NAME> ONLINE;

- Datafile checkpoint SCN and Start SCN are updated
- Stop SCN are set to FFFFFFFF
- RBA are updated in related datafile headers.
- OFFLINE RANGE RECORDS in control file are updated

Datafile – Offline and Online - Demo

Data File Header

Start SCN= **333333**

Stop SCN = **FFFFFF**

RBA = **0x03.3.1**

Control File

DATABASE ENTRY

System checkpoint SCN = 111111

DATA FILE RECORDS

Datafile checkpoint SCN = **333333**

Stop SCN = **FFFFFF**

No Checkpoint is triggered.
CKPT only updates the Stop SCN.

Media recovery is REQUIRED to bring the datafile back to ONLINE.

Recovery start from Start SCN/RBA until Stop SCN.

Recovery finished so that Start SCN is equal to Stop SCN, RBA is 0 because no more recovery is required.

Now we could bring the datafile online.

No Checkpoint is triggered.
CKPT only updates the relative SCN and RBA

ALTER DATABASE DATAFILE ONLINE; @ 333333

Datafile – Offline and Online - Description

ALTER DATABASE DATAFILE <NAME> OFFLINE;

- Does NOT trigger full checkpoint on the datafile.
- Current SCN are updated in the following area:
 - Stop SCN (control file and data file header) (v\$datafile.last_change#), it is no longer FFFFFFFF.
- Datafile checkpoint SCN(control file DATA FILE RECORDS) (v\$datafile.checkpoint_change#) and Start SCN (data file header) (v\$datafile_header.checkpoint_change#) are NOT touched.
- RBA are NOT updated.
- Media recovery is REQUIRED to bring the datafile back to ONLINE.

ALTER DATABASE DATAFILE <NAME> ONLINE;

- Prerequisite is to recover the datafile:
 - Apply the redo from the RBA info in the datafile header until the Stop SCN got from datafile header.
 - After the media recovery, Start SCN of the datafile header is equal to the Stop SCN.
 - RBA in datafile header is set to 0 because no more recovery is required.
- Now that the Start SCN and Stop SCN are the same, the datafile could be brought to ONLINE.
- Current SCN are updated in the following area:
 - Datafile checkpoint SCN(control file DATA FILE RECORDS) (v\$datafile.checkpoint_change#)
 - Start SCN (data file header) (v\$datafile_header.checkpoint_change#)
 - Stop SCN (control file DATA FILE RECORDS and data file header) (v\$datafile.last_change#) is set to FFFFFFFF again
- RBA are updated in the involved datafile header.

Tablespace/DataFile Status - Comparison

Operation	Tablespace	Datafile
Read Only	<ul style="list-style-type: none"> • Have to wait until all the active transactions started before to commit or rollback. • Full checkpoint is triggered on datafiles belong to the tablespaces. • SCN Status <ul style="list-style-type: none"> • Datafile checkpoint SCN (updated to current SCN) • Start SCN (updated to current SCN) • Stop SCN (updated to current SCN, no longer FFFFFFF) • RBA Updated • The above procedure is the same as SHUTDOWN DATABASE • SCN and RBA will then be frozen for these datafiles. 	<ul style="list-style-type: none"> • N/A
Back to Read Write	<ul style="list-style-type: none"> • No Recovery is required. No checkpoint is required. • SCN Status <ul style="list-style-type: none"> • Datafile checkpoint SCN (updated to current SCN) • Start SCN (updated to current SCN) • Stop SCN (updated back to FFFFFFF) • RBA Updated 	<ul style="list-style-type: none"> • N/A
Begin Backup	<ul style="list-style-type: none"> • Will NOT wait for active transactions to commit or rollback. • Full checkpoint is triggered on datafiles belong to the tablespaces. • SCN Status <ul style="list-style-type: none"> • Datafile checkpoint SCN (updated to current SCN) • Start SCN (updated to current SCN) • Stop SCN (Not touched, still FFFFFFF) • RBA Updated • SCN and RBA will then be frozen for these datafiles. But any further checkpoint info will still be updated in the datafile header. 	<ul style="list-style-type: none"> • N/A
End Backup	<ul style="list-style-type: none"> • A checkpoint up to the System SCN (or the Datafile checkpoint SCN of other datafiles) is triggered on all the data files belong to the tablespace. • SCN Status <ul style="list-style-type: none"> • Datafile checkpoint SCN (updated to System SCN, i.e. synchronize to other datafile) • Start SCN (updated to System SCN) • Stop SCN (Not touched, still FFFFFFF) • RBA Updated 	<ul style="list-style-type: none"> • N/A

Tablespace/DataFile Status - Comparison

Operation	Tablespace	Datafile
Offline	<ul style="list-style-type: none"> • Can't issue if database is read-only • Will NOT wait for active transactions to commit or rollback. • Full checkpoint is triggered on datafiles belong to the tablespaces. • SCN Status <ul style="list-style-type: none"> • Datafile checkpoint SCN (updated to current SCN) • Start SCN (updated to 0) • Stop SCN (updated to current SCN, no longer FFFFFFF) • RBA Updated • SCN and RBA will then be frozen for these datafiles. 	<ul style="list-style-type: none"> • Can't issue if database is in NOARCHIVELOG mode • No Full checkpoint triggered • SCN Status <ul style="list-style-type: none"> • Datafile checkpoint SCN (NOT touched) • Start SCN (NOT touched) • Stop SCN (updated to current SCN, no longer FFFFFFF) • RBA NOT updated • SCN and RBA will then be frozen for the datafile.
Back to Online	<ul style="list-style-type: none"> • Media Recovery NOT required • Checkpoint NOT required • SCN Status <ul style="list-style-type: none"> • Datafile checkpoint SCN (updated to current SCN) • Start SCN (updated to current SCN) • Stop SCN (updated back to FFFFFFF) • RBA Updated 	<ul style="list-style-type: none"> • Media Recovery is required • SCN Status after recovery <ul style="list-style-type: none"> • Datafile checkpoint SCN (updated to the Stop SCN) • Start SCN (updated to the Stop SCN) • Stop SCN (Not touched, used for the recovery) • RBA Updated after recovery (updated to 0) • SCN status after datafile Online <ul style="list-style-type: none"> • Datafile checkpoint SCN (updated to current SCN) • Start SCN (updated to current SCN) • Stop SCN (updated back to FFFFFFF) • RBA Updated

Tablespace and Datafile Status - Summary

Tablespace Status

- Read Only – Read Write
 - Wait for active transaction to commit (to guarantee consistent read)
 - Full checkpoint triggered
 - Stop SCN is updated (equal to Start SCN)
 - When change back, checkpoint to current SCN
- Begin Backup Mode – End Backup Mode
 - Do not wait for active transaction to finish
 - Full checkpoint triggered
 - In backup mode, further checkpoint info is recorded in data file header
 - When end backup, checkpoint synchronized to System SCN (other datafile)
- Offline – Online
 - Do not wait for active transaction to commit
 - Full checkpoint triggered
 - Start SCN set to 0, Stop SCN is updated
 - No recovery and checkpoint required when bring back to Online

Datafile Status

- Offline – Online
 - No checkpoint triggered
 - Stop SCN is updated (Not equal to Start SCN)
 - Need recovery when bring back to online so that Start SCN is equal to Stop SCN

Outline

1. SCN
2. Checkpoint
3. Database Status and Recovery
4. Tablespace and Datafile Status
- 5. Roadmap**

Roadmap



Roadmap – Be cool

Before manual restore/recovery:

- Study what has happened using alertlog, v\$ and DDIC views, dumping control file, file header, etc
- Draw pictures about the strategy and discuss with colleagues
- Backup the current situation in case the situation becomes worse

Roadmap – Collect information

Compare SCN

- System checkpoint SCN(control file) => `v$database.checkpoint_change#`
- Datafile checkpoint SCN(control file) => `v$datafile.checkpoint_change#`
- Stop SCN(control file and datafile header) => `v$datafile.last_change#`
- Start SCN (data file header) => `v$datafile_header.checkpoint_change#`
- Redo log SCN (control file) => `v$log.first_change#`
- Log file history SCN => `$log_history.first_change# / next_change# / resetlogs_change#`
- Archived log file history SCN => `v$archived_log.first_change# / next_change#`

Check File Status

- `v$datafile.status` => online, offline, system
- `v$datafile.enabled` => read only, read write
- `v$recover_file` => Media recovery information
- `v$backup.status` => active, not active

Roadmap – Proceed

Ask questions and proceed accordingly (SAP Note)

- Any tablespace in backup mode? (4162)
- Any tablespace in read-only mode?
- Any data file in offline mode?
- Which file is lost? (491160)
- Is the latest control file available?
- Is it possible to perform a complete recovery? (4161, 4164)
 - Latest online redologs available?
 - Any gap in the archived redologs?
- If recreate control file is necessary, any structure change since the last backup? (4165)

Summary

Summary

Now you should be able to

- Understand that the essential of recovery is to synchronize
- Explain different types of SCN
- Explain checkpoint
- Explain relationships between SCN and data file status
- Explain different types of recovery
 - Complete
 - Incomplete
 - Using backup control file
 - Recreate control file (resetlogs / noresetlogs)



Mingqian Zhong
AGS – SAP AG
mingqian.zhong@sap.com

Appendix

Commands for Dump

```
ALTER SESSION SET EVENTS 'IMMEDIATE TRACE NAME CONTROLF LEVEL 8';  
ALTER SESSION SET EVENTS 'IMMEDIATE TRACE NAME FILE_HDRS LEVEL 10';  
ALTER SESSION SET EVENTS 'IMMEDIATE TRACE NAME REDOHDR LEVEL 10';
```

```
ALTER SYSTEM DUMP LOGFILE 'FILENAME'  
  DBA MIN FILENO . BLOCKNO  
  DBA MAX FILENO . BLOCKNO;
```

```
ALTER SYSTEM DUMP LOGFILE 'FILENAME'  
  RBA MIN SEQNO . BLOCKNO  
  RBA MAX SEQNO . BLOCKNO;
```

```
ALTER SYSTEM DUMP LOGFILE 'FILENAME'  
  SCN MIN MINSN  
  SCN MAX MAXSCN;
```

```
ALTER SYSTEM DUMP LOGFILE 'FILENAME'  
  TIME MIN VALUE  
  TIME MAX VALUE;
```

Related SAP Note

4157	General flowchart for ORACLE recovery
4161	Complete Recovery
4162	Missing end backup
4163	Defective redo logs
4164	Incomplete Recovery ("...RECOVER UNTIL... ").
4165	Recovery after structure change
4190	Defective rollback segments
5701	RESTORE of the database (partial or complete)
19914	Subsequent processing after incomplete recovery
23070	Backup and Recovery Basic Concepts
491160	Restore scenarios for lost files of oracle databases
594601	Restore table in minimal copy of current database.
842240	FAQ Backup strategy of large and highly availability
941517	Recovery over resetlogs operation
966073	Oracle Database 10g: Flash Recovery Area
966117	Oracle Database 10g: Flashback Database
937492	FAQ: Oracle Flashback

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP BusinessObjects Explorer, StreamWork, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects Software Ltd. Business Objects is an SAP company.

Sybase and Adaptive Server, iAnywhere, Sybase 365, SQL Anywhere, and other Sybase products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Sybase, Inc. Sybase is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

The information in this document is proprietary to SAP. No part of this document may be reproduced, copied, or transmitted in any form or for any purpose without the express prior written permission of SAP AG.

This document is a preliminary version and not subject to your license agreement or any other agreement with SAP. This document contains only intended strategies, developments, and functionalities of the SAP® product and is not intended to be binding upon SAP to any particular course of business, product strategy, and/or development. Please note that this document is subject to change and may be changed by SAP at any time without notice.

SAP assumes no responsibility for errors or omissions in this document. SAP does not warrant the accuracy or completeness of the information, text, graphics, links, or other items contained within this material. This document is provided without a warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials. This limitation shall not apply in cases of intent or gross negligence.

The statutory liability for personal injury and defective products is not affected. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third-party Web pages nor provide any warranty whatsoever relating to third-party Web pages.