数据库常见的块错误和处理

Lunar

QQ: 5163721

www.lunar2013.com

以下内容仅限于在非生产环境测试，不能用于生产环境数据库，所有Internal的方法（BBED, DUL等等），请咨询Oracle Support部门，这些方法未经Oracle Support建议和许可，不能使用，否则后果自负！！

常见错误

ORA-1578	ORA-1578 is reported when a block is thought to be corrupt on read.
ORA-1410	This error is raised when an operation refers to a ROWID in a table for which there is no such row. The reference to a ROWID may be implicit from a WHERE CURRENT OF clause or directly from a WHERE ROWID=... clause. ORA 1410 indicates the ROWID is for a BLOCK that is not part of this table.
ORA-8103	The object has been deleted by another user since the operation began. If the error is reproducible, following may be the reasons:- a.) The header block has an invalid block type. b.) The data_object_id (seg/obj) stored in the block is different than the data_object_id stored in the segment header. See dba_objects.data_object_id and compare it to the decimal value stored in the block (field seg/obj).
ORA-8102	An ORA-08102 indicates that there is a mismatch between the key(s) stored in the index and the values stored in the table. What typically happens is the index is built and at some future time, some type of corruption occurs, either in the table or index, to cause the mismatch.
ORA-1498	Generally this is a result of an ANALYZE ... VALIDATE ... command. This error generally manifests itself when there is inconsistency in the data/Index block. Some of the block check errors that may be found:- a.) Row locked by a non-existent transaction b.) The amount of space used is not equal to block size c.) Transaction header lock count mismatch. While support are processing the tracefile it may be worth the re-running the ANALYZE after restarting the database to help show if the corruption is consistent or if it 'moves'. Send the tracefile to support for analysis. If the ANALYZE was against an index you should check the whole object. Eg: Find the tablename and execute: ANALYZE TABLE xxx VALIDATE STRUCTURE CASCADE;
ORA-1499	An error occurred when validating an index or a table using the ANALYZE command. One or more entries does not point to the appropriate cross-reference.
ORA-26040	Trying to access data in block that was loaded without redo generation using the NOLOGGING/UNRECOVERABLE option. This Error raises always together with ORA-1578
ORA-600 [12700]	Oracle is trying to access a row using its ROWID, which has been obtained from an index. A mismatch was found between the index rowid and the data block it is pointing to. The rowid points to a non-existent row in the data block The corruption can be in data and/or index blocks. ORA-600 [12700] can also be reported due to a consistent read (CR) problem.
ORA-600 [3020]	This is called a 'STUCK RECOVERY'. There is an inconsistency between the information stored in the redo and the information stored in a database block being recovered.

ORA-600 [4194]	A mismatch has been detected between Redo records and rollback (Undo) records. We are validating the Undo record number relating to the change being applied against the maximum undo record number recorded in the undo block. This error is reported when the validation fails.
ORA-600 [4193]	A mismatch has been detected between Redo records and Rollback (Undo) records. We are validating the Undo block sequence number in the undo block against the Redo block sequence number relating to the change being applied. This error is reported when this validation fails.
ORA-600 [4137]	While backing out an undo record (i.e. at the time of rollback) we found a transaction id mis-match indicating either a corruption in the rollback segment or corruption in an object which the rollback segment is trying to apply undo records on. This would indicate a corrupted rollback segment.
ORA-600 [6101]	Not enough free space was found when inserting a row into an index leaf block during the application of undo.
ORA-600 [2103]	Oracle is attempting to read or update a generic entry in the control file. If the entry number is invalid, ORA-600 [2130] is logged.
ORA-600 [4512]	Oracle is checking the status of transaction locks within a block. If the lock number is greater than the number of lock entries, ORA-600 [4512] is reported followed by a stack trace, process state and block dump. This error possibly indicates a block corruption.
ORA-600 [2662]	A data block SCN is ahead of the current SCN. The ORA-600 [2662] occurs when an SCN is compared to the dependent SCN stored in a UGA variable. If the SCN is less than the dependent SCN then we signal the ORA-600 [2662] internal error.
ORA-600 [4097]	We are accessing a rollback segment header to see if a transaction has been committed. However, the xid given is in the future of the transaction table. This could be due to a rollback segment corruption issue OR you might be hitting the following known problem.
ORA-600 [4000]	It means that Oracle has tried to find an undo segment number in the dictionary cache and failed.
ORA-600 [6006]	Oracle is undoing an index leaf key operation. If the key is not found, ORA-00600 [6006] is logged. ORA-600[6006] is usually caused by a media corruption problem related to either a lost write to disk or a corruption on disk.

ORA-600 [4552]	This assertion is raised because we are trying to unlock the rows in a block, but receive an incorrect block type. The second argument is the block type received.
ORA-600[6856]	Oracle is checking that the row slot we are about to free is not already on the free list. This internal error is raised when this check fails.
ORA-600[13011]	During a delete operation we are deleting from a view via an instead-of trigger or an Index organized table and have exceeded a 5000 pass count when we raise this exception.
ORA-600[13013]	During the execution of an UPDATE statement, after several attempts (Arg [a] passcount) we are unable to get a stable set of rows that conform to the WHERE clause.
ORA-600[13030]	
ORA-600[25012]	We are trying to generate the absolute file number given a tablespace number and relative file number and cannot find a matching file number or the file number is zero.
ORA-600[25026]	Looking up/checking a tablespace invalid tablespace ID and/or rdba found
ORA-600[25027]	Invalid tsn and/or rfn found
ORA-600 [kcbz_check_objd_typ_3]	An object block buffer in memory is checked and is found to have the wrong object id. This is most likely due to corruption.
ORA-600[kddummy_blkchk] & ORA-600[kdblkcheckerror]	ORA-600[kddummy_blkchk] is for 10.1/10.2 and ORA-600[kdblkcheckerror] for 11 onwards.
ORA-600[ktadrprc-1]	
ORA-600[ktsircinfo_num1]	This exception occurs when there are problems obtaining the row cache information correctly from sys.seg$. In most cases there is no information in sys.seg$.

Physical and Logical Block Corruptions

Physical Block Corruptions

This kind of block corruptions are normally reported by Oracle with error ORA-1578 and the detailed corruption description is printed in the alert log.

Corruption Examples are:

- Bad header - the beginning of the block (cache header) is corrupt with invalid values
- The block is Fractured/Incomplete - header and footer of the block do not match
- The block checksum is invalid
- The block is misplaced
- **Zeroed out blocks**

A block that is PHYSICAL corrupt is also called Media Corrupt Block. A Media Corrupt block is not a Soft Corrupt block.

OS Block Header

- The OS Block Header is in the first datafile block. It is used by Oracle to store Operating System information.
- It is the Block Zero. It is not the datafile header which is in Oracle Block 1.
- Corruption in Block Zero will not cause damage to the data and in versions lower than 11g it is not detected by dbverify/rman. In 11g Dbverify has been enhanced to detect it.
- dbfsize can be used to check the consistency of Block 0.
- When this block is corrupted the database may be opened with no errors. Block 0 is checked by some specific database operations like 'CREATE CONTROLFILE' or during upgrade to 11g.
- ORA-1503: CREATE CONTROLFILE failed
  ORA-1565: error in identifying file '/oradata/lunar.dbf'
  ORA-27047: unable to read the header block of file
  HP-UX Error: 22: Invalid argument
  Additional information: 2
- In 11g DBVerify reports error "DBV-00107: Unknown header format
- **$ dbfsize /oradata/lunar.dbf**

  Header block magic number is bad

  Header block file size is bad; trying raw file format...

  Header block magic number is bad

  alter database datafile '/oradata/users.dbf' resize <new size>;

全零块

DBV : Completely zero block found during dbv
RMAN trace： Completely zero block found during validation
V$DATABASE_BLOCK_CORRUPTION.CORRUPTION_TYPE： ALL ZERO
ORA-1578 : Corrupt block relative dba: <rdba> (file <file#>, block <block#>)
Completely zero block found during buffer read

Deploy Primary and Standby on Oracle Exadata Database Machine and Exadata Storage Servers

[illegible] by Oracle, the Exadata Database Machine and Sparc Supercluster also implements comprehensive Oracle Hardware Assisted Resilient Data (HARD) specifications, providing a unique level of validation for Oracle block data structures. The Exadata HARD checks include support for spfiles, controlfiles, log files, Oracle data files and Data Guard broker file when residing on Exadata Storage and works during an ASM rebalance or during ASM resync operations. Oracle Exadata Storage Server Software detects corruptions introduced into the I/O path between the database and storage. It stops corrupted data from being written to disk when a HARD check fails. This eliminates a large class of failures that the database industry had previously been unable to prevent. Examples of the Exadata HARD checks include: 1) redo and block checksum, 2) correct log sequence, 3) block type validation, 4) block number validation, 5) Oracle data structures such as block magic number, block size, sequence#, and block header and tail data structures. Exadata HARD checks are the most comprehensive list of Oracle data block checks initiating from the storage software (cellsrv) and works transparently after enabling database's DB_BLOCK_CHECKSUM parameter. Except for the case of Exadata storage, the Oracle HARD initiative has ended. Most past storage HARD implementations only provided checksums and very simple data block checks.

Starting with Oracle Exadata Software 11.2.3.3.0 and Oracle Database 11.2.0.4, Oracle Exadata Storage Server Software provides **Automatic Hard Disk Scrub and Repair.** This feature automatically inspects and repairs hard disks periodically when hard disks are idle. If bad sectors are detected on a hard disk, then Oracle Exadata Storage Server Software automatically sends a request to Oracle ASM to repair the bad sectors by reading the data from another mirror copy. By default, the hard disk scrub runs every two weeks. It's very lightweight and has enormous value add by fixing physical block corruptions even with infrequently access data.

Deploy Oracle Data Integrity eXtensions (DIX) with T10 Data Integrity Field (DIF) when not on Exadata

Oracle Linux team has collaborated with hardware vendors and Oracle database development to extend Oracle data integrity extensions from Oracle's operating system (Linux) to various vendor's host adapter down to the storage device. With these extensions, DIX provides end to end data integrity for reads and writes through a checksum validation. The prerequisite is to leverage certified storage, HBA and disk firmware. An example of this partnership is DIX integration with Oracle Linux, Emulex or QLogic Host Bus Adapters and any T10 DIF capable storage arrays such as EMC VMAX. Refer to the following documentation for more information.

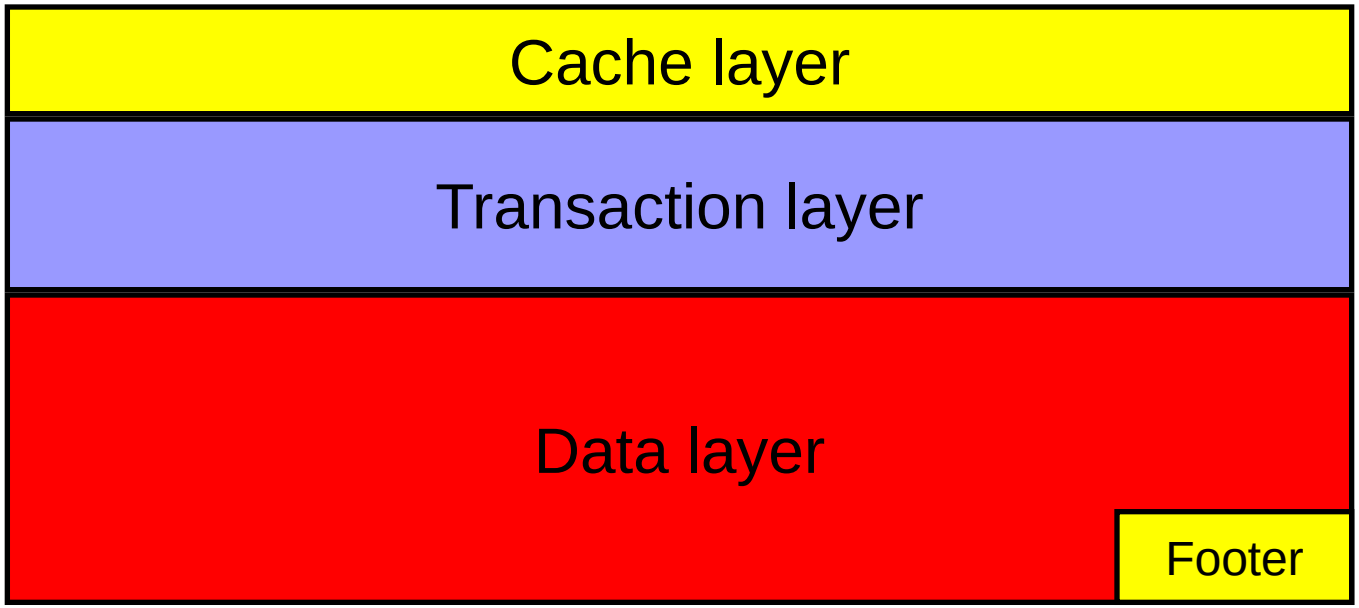
第三方软件或者硬件造成的物理损坏

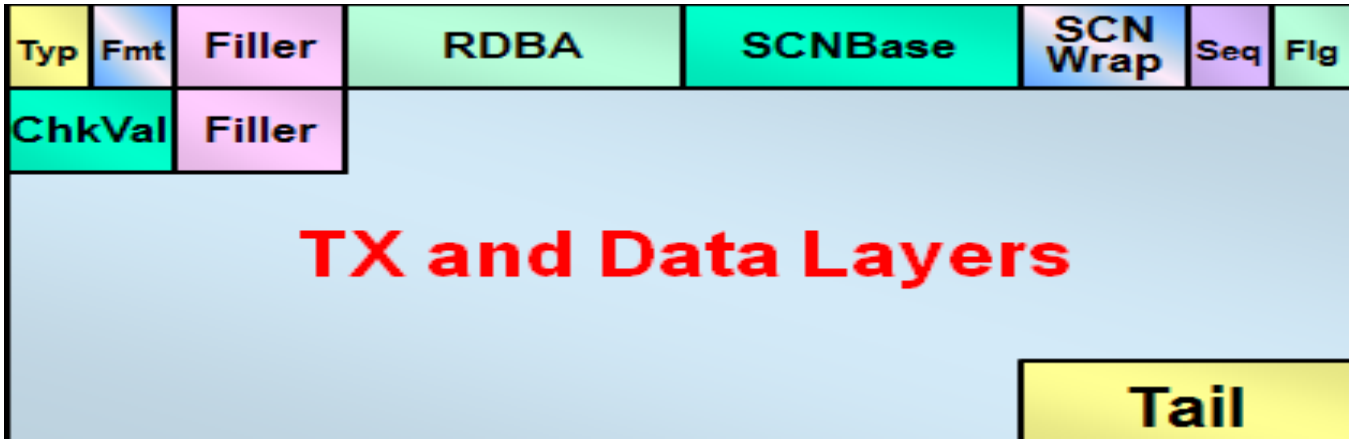
- VMWARE ESXI 5.0上使用Xsigo driver（HBA）造成redo 损坏（ORA-00354: corrupt redo log block header）
- Hitachi storage的微码可能导致写丢失，造成 ORA-600 [kdsgrp1] ORA-1499 ORA-1410 ORA-600 [3020]
- EXT4文件系统上设置filesystemio_options=SETALL可能造成 ORA-1578 ORA-353 ORA-19599（这是kernel-uek-2.6.39-200.29.3.el6uek内核的BUG）
- EMC SRDF进行灾备切换时可能造成ORA-600 [25027] ORA-600 [kdsgrp1] ORA-8103 ORA-1410
- EMC DATAMOVER文件系统（类似NAS)报ORA-1499 ORA-8103 ORA-1410 ORA-600 [4137] ORA-600 [4193] ORA-600 [4194]等等
- NetBackup做RMAN备份导致ORA-1578 ORA-27047 ORA-27048
- 部分NAS的NFS（比如Netapp的）可能导致ORA-1578，块写错位置了
- EMC DMX4阵列存储可能导致临时性的ORA-1578错误
- Veritas /Symantec 的卷管理器 当使用 PERSISTENCE=NO 时存在BUG ,导致 IO丢失
- EMC Networker Module for Oracle (NMO)的 BUG可能导致数据块被EMC覆盖掉

```
ORA-1578  "ORACLE data block corrupted (file # %s, block # %s)"
ORA-1565  "error in identifying file '%s'"
ORA-353   "log corruption near block %s change %s time %s"
ORA-354   "corrupt redo log block header"
ORA-27047 "unable to read the header block of file"
ORA-27048 "skgfifi: file header information is invalid"
ORA-752 "recovery detected a lost write of a data block"
ORA-600 [3020] "Stuck Recovery"
```

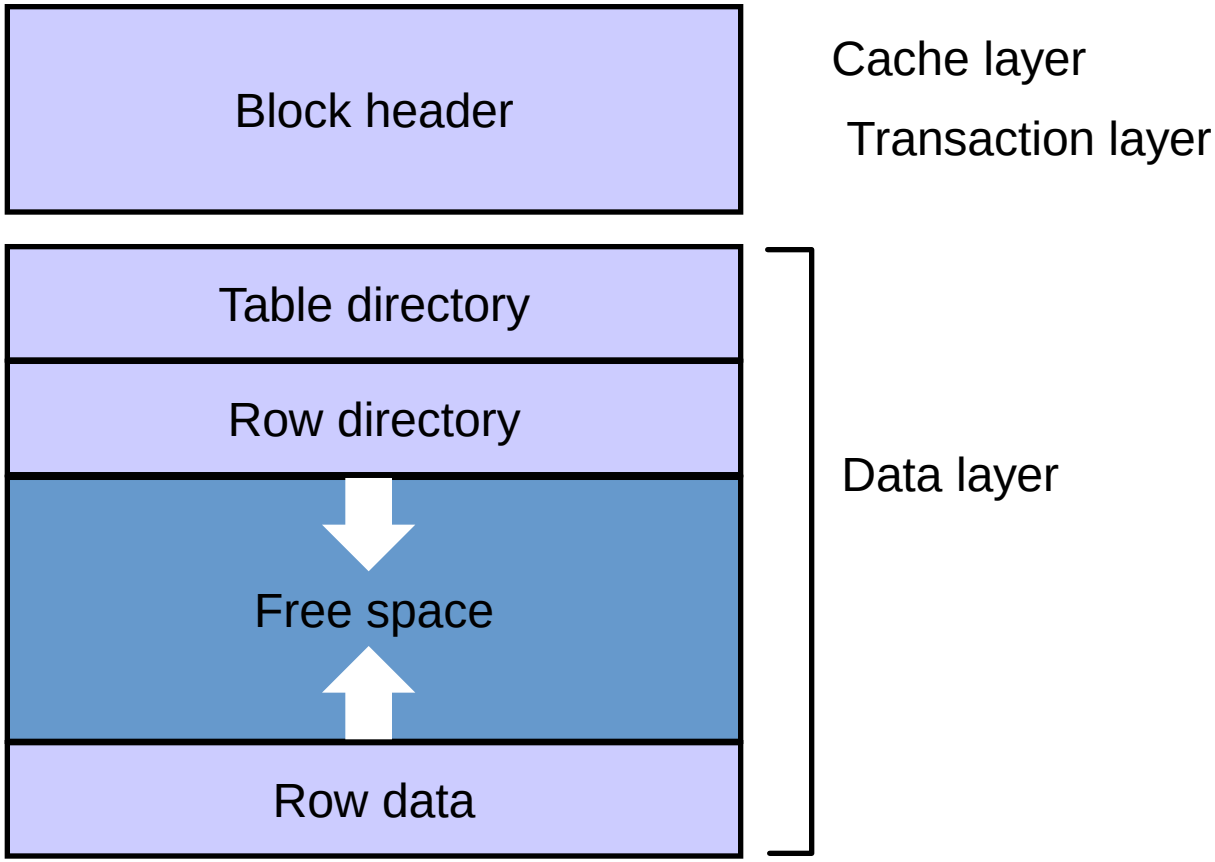
General Block Structure

- **Data and metadata**





Data Block Layout



```
SYS@lunar>select * from v$type_size where type like 'KC%';

COMPONENT        TYPE            DESCRIPTION                                                      TYPE_SIZE
---------------- --------------- ---------------------------------------------------------------- ----------
KCB              KCBH            BLOCK COMMON HEADER                                                      20

Elapsed: 00:00:00.00
SYS@lunar>select * from v$type_size where type like 'KTS%';

COMPONENT        TYPE            DESCRIPTION                                                      TYPE_SIZE
---------------- --------------- ---------------------------------------------------------------- ----------
KTS              KTSHC           SEGMENT HEADER                                                            8
KTS              KTSFS           SEGMENT FREE SPACE LIST                                                  20
KTS              KTSPHW          PAGE TABLE SEGMENT HWM                                                   60
KTS              KTSPHC          PAGE TABLE SEGMENT HEADER                                               112
KTS              KTSPFHC         LEVEL 1 BITMAP BLOCK HEADER                                             184
KTS              KTSPSHC         LEVEL 2 BITMAP BLOCK HEADER                                              96
KTS              KTSPTHC         LEVEL 3 BITMAP BLOCK HEADER                                              88

7 rows selected.

Elapsed: 00:00:00.00
SYS@lunar>select * from v$type_size where type like 'KTU%';

COMPONENT        TYPE            DESCRIPTION                                                      TYPE_SIZE
---------------- --------------- ---------------------------------------------------------------- ----------
KTU              KTUBH           UNDO HEADER                                                              16
KTU              KTUXE           UNDO TRANSACTION ENTRY                                                   40
KTU              KTUXC           UNDO TRANSACTION CONTROL                                                104

Elapsed: 00:00:00.01
SYS@lunar>
```

```
LUNAR@orcl>SELECT ROWID, SUBSTR(ROWID,15,4) "FILE",
SUBSTR(ROWID,1,8) "BLOCK",
  2    3  SUBSTR(ROWID,10,4) "ROW"
  4  from lunar.T
  5  where username='FF';

ROWID              FILE     BLOCK            ROW
------------------ -------- ---------------- --------
AAANcrAAEAAAAU0AAT 0AAT     AAANcrAA         AAAA

LUNAR@orcl>
LUNAR@orcl>select dbms_rowid.rowid_object(rowid) object_id,
  2  dbms_rowid.rowid_relative_fno(rowid) file_id,
  3  dbms_rowid.rowid_block_number(rowid) block_id ,
  4  dbms_rowid.rowid_row_number(rowid) num
  5  from lunar.t
  6  where username='FF';

 OBJECT_ID    FILE_ID   BLOCK_ID        NUM
---------- ---------- ---------- ----------
     55083          4       1332         19

LUNAR@orcl>
```

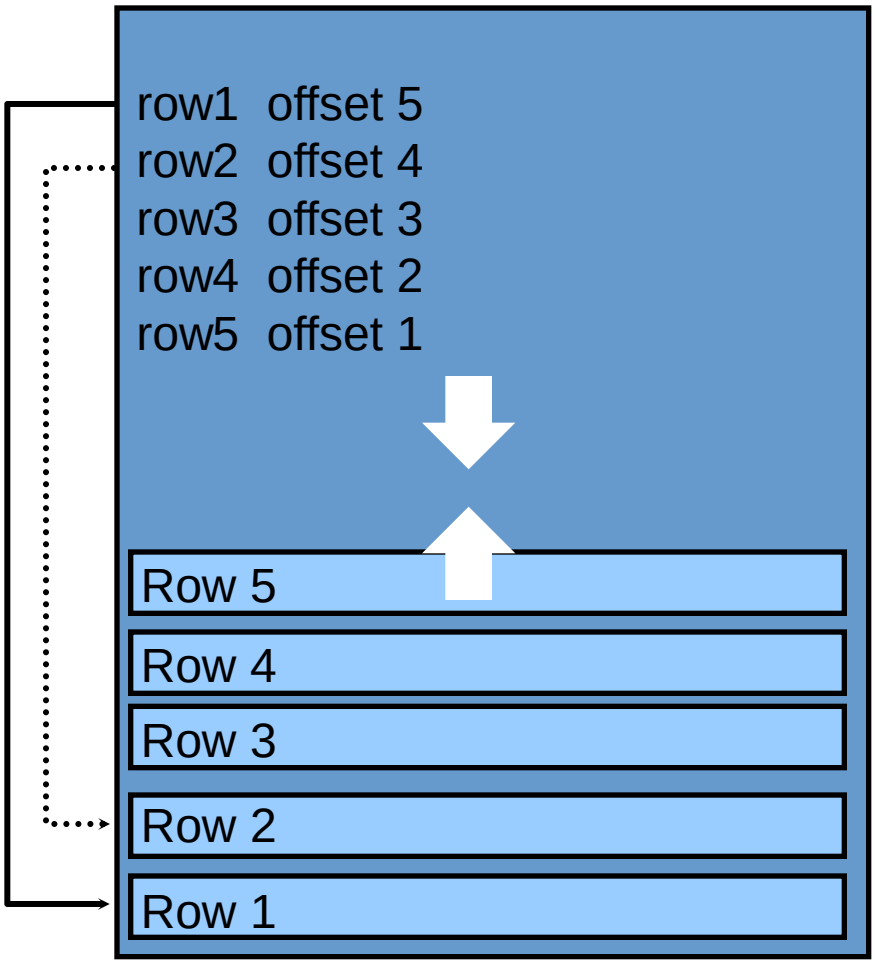
```
SYS@orcl>oradebug close_trace
Statement processed.
SYS@orcl>oradebug setmypid
Statement processed.
SYS@orcl>alter system dump datafile 4 block 1332;

System altered.

SYS@orcl>oradebug close_trace
Statement processed.
SYS@orcl>oradebug tracefile_name
/home/oracle/oracle/product/admin/orcl/udump/orcl_ora_13134.trc
SYS@orcl>
```

```
Block header dump:  0x01000534
 Object id on Block? Y
 seg/obj: 0xd72b  csc: 0x00.1bbc07  itc: 3  flg: E  typ: 1 - DATA
     brn: 0  bdba: 0x1000531 ver: 0x01 opc: 0
     inc: 0  exflg: 0

 Itl           Xid                  Uba         Flag  Lck        Scn/Fsc
0x01   0xffff.000.00000000  0x00000000.0000.00  C---    0  scn 0x0000.001ba2a9
0x02   0x0005.01a.00000378  0x008004e0.021c.05  ----    2  fsc 0x0055.00000000
0x03   0x0000.000.00000000  0x00000000.0000.00  ----    0  fsc 0x0000.00000000

data_block_dump,data header at 0xcc2b67c
===============
tsiz: 0x1f80
hsiz: 0x72
pbl: 0x0cc2b67c
bdba: 0x01000534
     76543210
flag=--------
ntab=1
nrow=48
frre=-1
fsbo=0x72
fseo=0xcf8
avsp=0xcd9
tosp=0xd30
0xe:pti[0]      nrow=48 offs=0
0x12:pri[0]     offs=0x1f2c
```

```
tab 0, row 19, @0xcf8
tl: 83 fb: --H-FL-- lb: 0x2  cc: 11
col  0: [ 2]  46 46
col  1: [ 2]  c1 59
col  2: [16]  30 42 37 43 31 44 35 41 44 39 46 33 43 41 4
col  3: [ 4]  4f 50 45 4e
col  4: *NULL*
col  5: *NULL*
col  6: [ 5]  55 53 45 52 53
col  7: [ 4]  54 45 4d 50
col  8: [ 7]  78 6e 02 04 01 1a 3b
col  9: [ 7]  44 45 46 41 55 4c 54
col 10: [22]
 44 45 46 41 55 4c 54 5f 43 4f 4e 53 55 4d 45 52 5f 47 52
```

```
0x64:pri[41]    offs=0xf72
0x66:pri[42]    offs=0xeff
0x68:pri[43]    offs=0xea9
0x6a:pri[44]    offs=0xe4e
0x6c:pri[45]    offs=0xe07
0x6e:pri[46]    offs=0xdbc
0x70:pri[47]    offs=0xd4b
block_row_dump:
tab 0, row 0, @0x1f2c
tl: 84 fb: --H-FL-- lb: 0x0  cc: 11
col  0: [ 4]  52 4d 41 4e
col  1: [ 2]  c1 4f
col  2: [16]  45 37 42 35 44 39 32 39 31 31 43 38 33 31 45 31
col  3: [ 4]  4f 50 45 4e
col  4: *NULL*
col  5: *NULL*
col  6: [ 4]  52 4d 41 4e
col  7: [ 4]  54 45 4d 50
col  8: [ 7]  78 6d 08 12 18 29 05
col  9: [ 7]  44 45 46 41 55 4c 54
col 10: [22]
 44 45 46 41 55 4c 54 5f 43 4f 4e 53 55 4d 45 52 5f 47 52 4f 55 50
tab 0, row 1, @0x1ed6
tl: 86 fb: --H-FL-- lb: 0x0  cc: 11
col  0: [ 4]  53 4e 50 57
col  1: [ 2]  c1 49
col  2: [16]  44 36 46 43 42 36 32 45 43 46 30 43 45 32 36 36
col  3: [ 4]  4f 50 45 4e
col  4: *NULL*
```

Block Usage: An Example



1. Insert from the bottom of the block and move up.
2. Try to reuse first free slot in row index.
3. Attempt to get space from free space.
4. If not enough space, compress block.

Fractured Block

A Fractured block means that the block is incomplete. Information from the block header does not match the block tail.

Corrupt block relative dba: 0x0380e573 (file 14, block 58739)
Fractured block found during buffer read
Data in bad block -
type: 6 format: 2 rdba: 0x0380e573
last change scn: 0x0288.8e5a**2f78 seq: 0x1** flg: 0x04
consistency value in **tail: 0x00780601**
check value in block header: 0x8739, computed block checksum: 0x2f00
spare1: 0x0, spare2: 0x0, spare3: 0x0
***
Reread of rdba: 0x0380e573 (file 14, block 58739) found same corrupted data

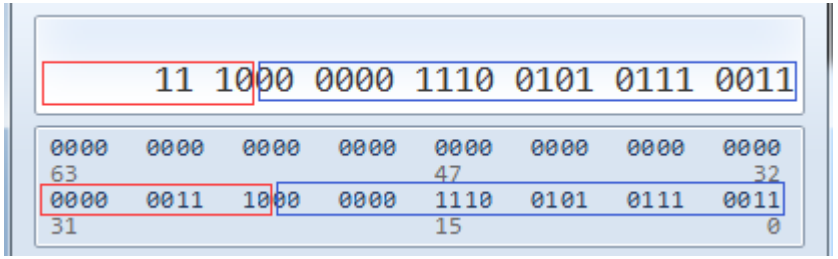
Tail = 2 lower bytes of SCN Base + type + seq

2f780601--→0106782f

地址转换

```
SYS@lunar>select dbms_utility.data_block_address_file(to_number(ltrim('0x0380e573','0x'),'xxxxxxxx')) as file_no,
  2  dbms_utility.data_block_address_block(to_number(ltrim('0x0380e573','0x'),'xxxxxxxx')) as block_no
  3  from dual;

   FILE_NO    BLOCK_NO
---------- ----------
        14       58739

Elapsed: 00:00:00.02
SYS@lunar>
```

```
SYS@lunar>select to_number('0380e573','xxxxxxxxxxxxxxxx') from dual;

TO_NUMBER('0380E573','XXXXXXXXXXXXXXXX')
----------------------------------------
                                58778995

Elapsed: 00:00:00.01
SYS@lunar>select dbms_utility.data_block_address_block(58778995) "BLOCK",
  2     dbms_utility.data_block_address_file(58778995) "FILE"
  3  from dual;

     BLOCK       FILE
---------- ----------
     58739         14

Elapsed: 00:00:00.02
SYS@lunar>
```



1110--→14　1110010101110011-→58739

```
SYS@lunar>select to_char(dbms_utility.make_data_block_address(14,58739),'xxxxxxxxxxxxxxxx') from dual;

TO_CHAR(DBMS_UTILITY.MAKE_DATA_BLOCK
------------------------------------
            380e573

Elapsed: 00:00:00.01
SYS@lunar>
```

Bad Checksum

Checksum是oracle写入后，其他外部因素导致块checksum改变的情况。

Checksum只有DBWR进程写入，或者直接从磁盘读取

Corrupt block relative dba: 0x0380a58f (file 14, block 42383)
Bad check value found during buffer read
Data in bad block -
type: 6 format: 2 rdba: 0x0380a58f
last change scn: 0x0288.7784c5ee **seq: 0x1** flg: 0x06
consistency value in tail: 0xc5ee0601
check value in block header: 0x68a7, **computed block checksum: 0x2f00**
spare1: 0x0, spare2: 0x0, spare3: 0x0
***
Reread of rdba: 0x0380a58f (file 14, block 42383) found same corrupted data

如果a、b两个值不相同，则异或结果为1。如果a、b两个值相同，异或结果为0

CHECKSUM=0x68a7 XOR 0x2f00= 0x47a7

:

DB_BLOCK_CHECKSUM determines whether **DBW*n*** and the direct loader will calculate a checksum (a number calculated from all the bytes stored in the block) and store it in the cache header of every data block when writing it to disk. Checksums are verified when a block is read - only if this parameter is **TYPICAL** or **FULL**and the last write of the block stored a checksum.

```
BBED> verify
DBVERIFY - Verification starting
FILE = /stage/lunar/system01.dbf
BLOCK = 1

Block 1 is corrupt
Corrupt block relative dba: 0x00400001 (file 0, block 1)
Bad check value found during verification
Data in bad block:
 type: 11 format: 2 rdba: 0x00400001
 last change scn: 0x0000.00000000 seq: 0x1 flg: 0x04
 spare1: 0x0 spare2: 0x0 spare3: 0x0
 consistency value in tail: 0x00000b01
 check value in block header: 0x7677
 computed block checksum: 0x1

BBED> p kcvfhbfh
struct kcvfhbfh, 20 bytes                   @0
   ub1 type_kcbh                    @0       0x0b
   ub1 frmt_kcbh                    @1       0xa2
   ub1 spare1_kcbh                  @2       0x00
   ub1 spare2_kcbh                  @3       0x00
   ub4 rdba_kcbh                    @4       0x00400001
   ub4 bas_kcbh                     @8       0x00000000
   ub2 wrp_kcbh                     @12      0x0000
   ub1 seq_kcbh                     @14      0x01
   ub1 flg_kcbh                     @15      0x04 (KCBHFCKV)
   ub2 chkval_kcbh                  @16      0x7677
   ub2 spare3_kcbh                  @18      0x0000

BBED> sum
Check value for File 1, Block 1:
current = 0x7677, required = 0x7676
BBED>
```

db_block_checksum

Oracle 6： db_block_compute_checksums

Oracle 7.2：

DB_BLOCK_CHECKSUM

LOG_BLOCK_CHECKSUM

8.1.5 - 8.1.7 ： DB_BLOCK_CHECKSUM

If set to TRUE -->> Processes all tablespaces and writes checksumming for redo log blocks as well.

Default is FALSE -->> Processes only SYSTEM tablespace

9.0.1 - 10g DB_BLOCK_CHECKSUM = TRUE

Default is TRUE -->> Processes all tablespaces and writes checksumming for redo log blocks as well.

If set to FALSE -->> Processes only SYSTEM tablespace

Dynamic: alter system

db_block_checksum

10gR2+ DB_BLOCK_CHECKSUM = OFF | TYPICAL | FULL

Default is TYPICAL and behaves just as TRUE in previous releases.

In FULL mode, Oracle also verifies the checksum before a change application from update/delete statements and recomputes it after the change is applied.

In addition, Oracle gives every log block a checksum before writing it to the current log.

11gR2：

DB_BLOCK_CHECKSUM

Property	Description
Parameter type	String
Syntax	DB_BLOCK_CHECKSUM = { OFF | FALSE | TYPICAL | TRUE | FULL }
Default value	TYPICAL
Modifiable	ALTER SESSION, ALTER SYSTEM
Basic	No

db_block_checking

10gR2+ DB_BLOCK_CHECKING = OFF | LOW | MEDIUM | FULL

Default is OFF

LOW - basic block header checks are performed after block contents change in memory (for example, after UPDATE or INSERT statements, on-disk reads, or inter-instance block transfers in RAC)

MEDIUM - all LOW checks are performed, as well as semantic block checking for all non-index-organized table blocks

FULL - all LOW and MEDIUM checks are performed, as well as semantic checks for index blocks (that is, blocks of subordinate objects that can actually be dropped and reconstructed when faced with corruption)

9.0.1 - 10.1 DB_BLOCK_CHECKING = TRUE

If set to True -->> enables block checking on ALL tablespaces.

Default is FALSE -->> sets block checking on the SYSTEM tablespace only.

Dynamic : ALTER SYSTEM

8.1.6 - 8.1.7 DB_BLOCK_CHECKING = TRUE

If set to True - enables block checking on ALL tablespaces.

Default is FALSE - sets block checking on the SYSTEM tablespace only.

db_block_checking

11.2：

DB_BLOCK_CHECKING

Property	Description
Parameter type	String
Syntax	DB_BLOCK_CHECKING = { FALSE | OFF | LOW | MEDIUM | TRUE | FULL }
Default value	FALSE
Modifiable	ALTER SYSTEM
Basic	No

11.2.0.3;：

ALTER SESSION set “_DISABLE_BLOCK_CHECKING”=true;

11g 和10.2：

DB_BLOCK_CHECKING的推荐值是 LOW or **OFF**（不推荐设置为MEDIUM or FULL ）

10.1 和9i ：

DB_BLOCK_CHECKING的推荐值是 FALSE （不推荐设置为TRUE ）

+ Database Version is 10.2.0.1.

+ If db_block_checking=TRUE and there are zero length columns found in a data block ORA-600[510] is reported along with ORA-1578 error:

```
Errors in file /opt/oracle/product/10.2/admin/udump/nldb76_ora_11979.trc:
ORA-01578: ORACLE data block corrupted (file # 78, block # 197993)
ORA-01110: data file 78: '/opt/oracle/oradata/local/nldb76_nlcompany_data60.dat'
ORA-01578: ORACLE data block corrupted (file # 78, block # 197993)
ORA-01110: data file 78: '/opt/oracle/oradata/local/nldb76_nlcompany_data60.dat'
ORA-00600: internal error code, arguments: [510], [0x2DBAE6530], [redo copy], [], [], [], [], []
```

db_block_checksum和db_block_checking的性能影响

alter system set db_block_checking=FULL;
alter system set db_block_checksum=FULL;

EST RESULTS (11.2.0.3 db)
==========

100,000 Rows	Compression		No Compression	
	DEFAULT	FULL	DEFAULT	FULL
Insert	2.41	2.72	2.07	2.06
Update	3.85	42.43	.83	7.43
Delete	2.23	28.13	.87	6.60

db_always_check_system_ts

8.1.6以前： db_always_check_system_ts

8.1.6中 ： _DB_ALWAYS_CHECK_SYSTEM_TS（缺省值为 TRUE）
Always perform block check and checksum for System tablespace

```
@% ./bbed datafile=/[redacted]system01.dbf
@BBED> set block 118
@BBED> set offset 102  <--- 0x66
@BBED> dump
@      07260727 ............    <--- avsp, tosp
@BBED> set mode edit
@BBED> modify /x 0727
@BBED> dump
@      07270727 ............
@BBED> exit
```

Block Misplaced

Corrupt block relative dba: **0x0d805a89** (file 54, block 23177)
Bad header found during buffer read
Data in bad block -
type: 6 format: 2 **rdba: 0x0d805b08**
last change scn: 0x0692.86dc08e3 seq: 0x1 flg: 0x04
consistency value in tail: 0x08e30601
check value in block header: 0x2a6e, **computed block checksum: 0x0**
spare1: 0x0, spare2: 0x0, spare3: 0x0
***

DBV-200: Block, dba <rdba>, already marked corrupted

db_block_checking=enable
ORA-600 [kddummy_blkchk] or ORA-600 [kdBlkCheckError]

- row locked by non-existent transaction
- Avsp tosp等等

行锁错误的例子

```
BBED> x /rccccccccccc
rowdata[127]                          @6031
------------
flag@6031: 0x2c (KDRHFL, KDRHFF, KDRHFH)
lock@6032: 0x00
cols@6033:    3

col   0[14] @6034: GLOBAL_DB_NAME
col    1[5] @6049: LUNAR
col   2[20] @6055: Global database name

BBED> verify
DBVERIFY - Verification starting
FILE = /stage/lunar/system01.dbf
BLOCK = 801

Block Checking: DBA = 4195105, Block Type = KTB-managed data block
data header at 0xf3465c
kdbchk: xaction header lock count mismatch
        trans=1 ilk=1 nlo=0
Block 801 failed with check code 6108

DBVERIFY - Verification complete
```

块内空间计算错误的例子

```
BBED> verify
DBVERIFY - Verification starting
FILE = /stage/lunar/system01.dbf
BLOCK = 801

Block Checking: DBA = 4195105, Block Type = KTB-managed data block
data header at 0xf3465c
kdbchk: the amount of space used is not equal to block size
        used=2054 fsc=7 avsp=6040 dtl=8096
Block 801 failed with check code 6110

DBVERIFY - Verification complete

Total Blocks Examined         : 1
Total Blocks Processed (Data) : 1
Total Blocks Failing   (Data) : 1
Total Blocks Processed (Index): 0
Total Blocks Failing   (Index): 0
Total Blocks Empty            : 0
Total Blocks Marked Corrupt   : 0
Total Blocks Influx           : 0
Message 531 not found;  product=RDBMS; facility=BBED


BBED>
```

Soft Corrupt Block

- A Software Corrupt Block is a former LOGICAL corrupt block marked as formally corrupt.
- A block that is PHYSICAL corrupt is also called Media Corrupt Block. A Media Corrupt block is not a Soft Corrupt block.
- **seq=0xff scn=0x0000.00000000 tail= 0x0000<block type>ff**
- **DBV-00200: Block, dba <rdba>, already marked corrupted**

- dbms_repair.FIX_CORRUPT_BLOCKS（ Software Corrupt Block ）
- blockrecover datafile <f> block <n> CLEAR （ PHYSICAL corruption ）

- Soft Corrupt的行为与ORA-1578、event 10231/10233 和 dbms_repair.SKIP_CORRUPT_BLOCKS的关系

检测 Soft Corruption

- RMAN备份会忽略Soft Corruption
- Soft Corrupt的块不计入MAXCORRUPT
- Media Recovery 会忽略Soft Corrupt
- RMAN validate命令不会在alert中记录Soft Corrupt的信息，但是会在v$database_block_corruption中记录
- DBV可以检测Soft Corruption

如果不设置event 10231或者类似事件，那么软损坏的块再次访问时报ORA-1578

```
SQL> select * from V$DATABASE_BLOCK_CORRUPTION;
         FILE#          BLOCK#         BLOCKS CORRUPTION_CHANGE# CORRUPTIO
-------------- -------------- -------------- ------------------ ---------
             6             10              1      8183236781662 LOGICAL
             6             42              1                  0 FRACTURED
             6             34              2                  0 CHECKSUM
             6             50              1      8183236781952 LOGICAL
             6             26              4                  0 FRACTURED
             1             66              1                  0 CORRUPT
```

V$DATABASE_BLOCK_CORRUPTION

V$DATABASE_BLOCK_CORRUPTION

V$DATABASE_BLOCK_CORRUPTION displays information about database blocks that were corrupted after the last backup.

Column	Datatype	Description
FILE#	NUMBER	Absolute file number of the datafile that contains the corrupt blocks
BLOCK#	NUMBER	Block number of the first corrupt block in the range of corrupted blocks
BLOCKS	NUMBER	Number of corrupted blocks found starting with BLOCK#
CORRUPTION_CHANGE#	NUMBER	Change number at which the logical corruption was detected. Set to 0 to indicate media corruption.
CORRUPTION_TYPE	VARCHAR2(9)	Type of block corruption in the datafile: • ALL ZERO - Block header on disk contained only zeros. The block may be valid if it was never filled and if it is in an Oracle7 file. The buffer will be reformatted to the Oracle8 standard for an empty block. • FRACTURED - Block header looks reasonable, but the front and back of the block are different versions. • CHECKSUM - optional check value shows that the block is not self-consistent. It is impossible to determine exactly why the check value fails, but it probably fails because sectors in the middle of the block are from different versions. • CORRUPT - Block is wrongly identified or is not a data block (for example, the data block address is missing) • LOGICAL - Specifies the range is for logically corrupt blocks. CORRUPTION_CHANGE# will have a nonzero value.

Block Recover

--块恢复（9i的NF）

blockrecover datafile 6 block 15;

blockrecover corruption list;

。。。

--11g

--恢复全部损坏的块

recover corruption list ;

Recover datafile <fileno> block <block number reported corrupt> ;

。。。

--恢复单个块

Recover datafile <fileno> block <block number > to <block number>

。。。

--根据dba地址恢复块

SELECT DBMS_UTILITY.MAKE_DATA_BLOCK_ADDRESS(7,201) FROM DUAL;

Recover tablespace <name> dba <integer value> ;

---11g

DRA（Data Recovery Advisor）：

•List Failure - 列出先前执行过的故障评估结果。

•Advise Failure – 给出手动和自动修复建议。

•Repair Failure - 通过运行由ADVISE FAILURE建议的最佳修复方案来自动修复故障。完成之后会重新验证现有故障

•Change Failure - 改变故障的状态。

Oracle9i Recovery Manager Reference Release 2 (9.2) ：The datafile header block (block 1) cannot be recovered.

Oracle Database Backup and Recovery Reference 11g Release 2 (11.2)： The data file header block (block 1) cannot be

Setting Multiple Events

In the parameter file, you have two methods:

1. **Use multiple, *consecutive* event lines:**

```
event = "10015 trace name context forever"
event = "10046 trace name context forever, level 12"
```

2. **Concatenate the events with a colon (:) as the separator:**

```
event = "10015 trace name context forever:
         10046 trace name context forever, level 12"
```

相关 event

Event:10210 Block check DATA blocks

Event:10211 Block check INDEX blocks

Event:10212 Block check CLUSTER blocks

Event:10231 Skip soft corrupted blocks on FTS

Event:10232 Dumps corrupted blocks to trace

Event:10233 Skip corrupt block on Index Range scans

。 。 。 。 。 。

查看当前版本的event

```
DECLARE
err_msg VARCHAR2(120);
BEGIN
dbms_output.enable (1000000);
FOR err_num IN 10000..10999
LOOP
err_msg := SQLERRM (-err_num);
IF err_msg NOT LIKE '%Message '||err_num||' not found%' THEN
dbms_output.put_line (err_msg);
END IF;
END LOOP;
END;
```

Event 10046

Session level:

```
alter session set events '10046 trace name context forever';
alter session set events '10046 trace name context forever, level 8';
alter session set events '10046 trace name context off';
```

11g alternatives:

```
alter session set events 'sql_trace';
alter session set events 'sql_trace wait=true';
alter session set events 'sql_trace off'';
```

Pre 7.3 oardbx:

event 10046 trace name context forever, level 4

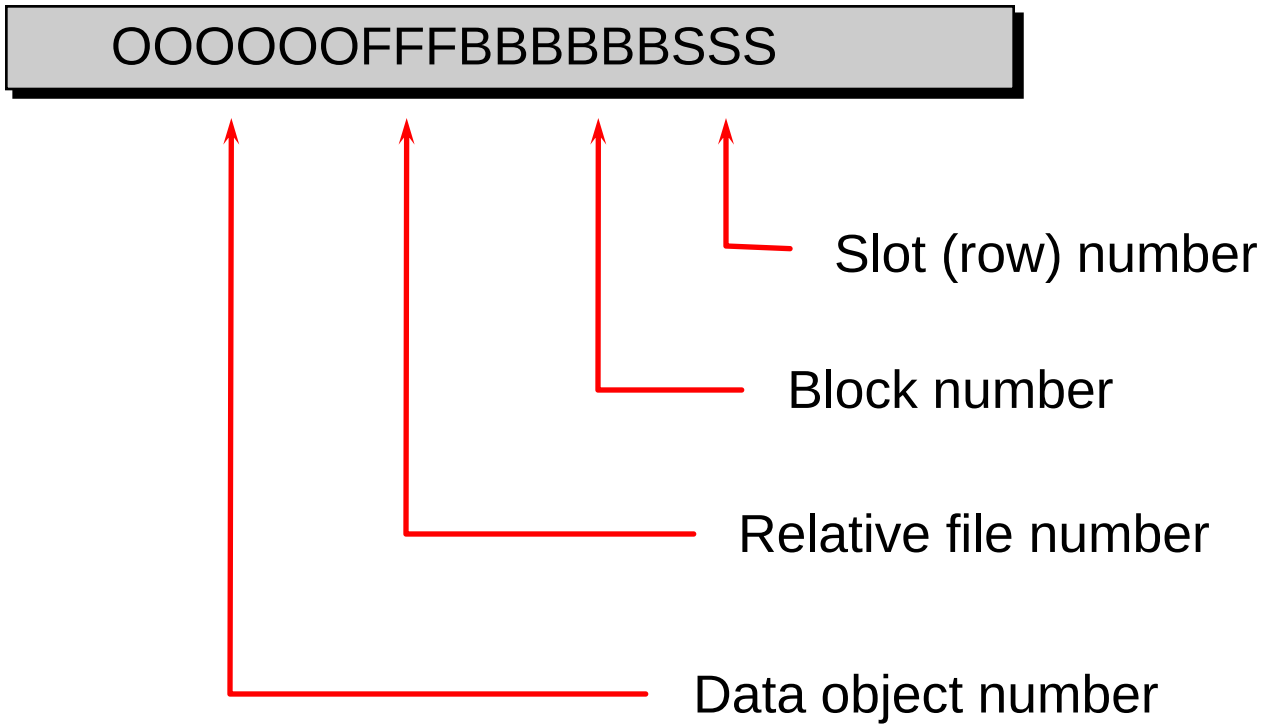
7.3+:

oradebug event 10046 trace name context forever, level 4

11g alternative:

oradebug event sql_trace bind=true

Oracle ROWID Format



ROWID的计算

根据rowid手工推算他的file#,block#,obj#,row# ： http://www.lunar2013.com/?p=1621

ROWID 和 Physical Database Limits

Physical Database Limits

Item	Type of Limit	Limit Value
Database Block Size	Minimum	2048 bytes; must be a multiple of operating system physical block size
Database Block Size	Maximum	Operating system dependent; never more than 32 KB
Database Blocks	Minimum in initial extent of a segment	2 blocks
Database Blocks	Maximum per datafile	Platform dependent; typically 2^{22} - 1 blocks
Controlfiles	Number of control files	1 minimum; 2 or more (on separate devices) strongly recommended
Controlfiles	Size of a control file	Maximum of 201031680 logical blocks
Database files	Maximum per tablespace	Operating system dependent; usually 1022
Database files	Maximum per database	65533 May be less on some operating systems Limited also by size of database blocks and by the DB_FILES initialization parameter for a particular instance
Database extents	Maximum per dictionary managed tablespace	4 GB * physical block size (with K/M modifier); 4 GB (without K/M modifier)
Database extents	Maximum per locally managed (uniform) tablespace	2 GB * physical block size (with K/M modifier); 2 GB (without K/M modifier)
Database file size	Maximum	Operating system dependent. Limited by maximum operating system file size; typically 2^{22} or approximately 4 million blocks

基本概念和名词解释

Dba: Data block address

Rdba: Root dba

RBA: Redo block address

UBA: Undo block address

Xid: Transaction ID

ITL: Interested Transaction List

XID=Undo.Segment.Number+Transaction.Table.Slot.Number+Wrap

UBA=Undo.Block. Address+Sequence+Last.Entry.in.UNDO.Record.Map

Transaction Identifiers

- **Transaction identifiers (XID) uniquely identify a transaction within the system; they are used within the Interested Transaction List (ITL) of the data block.**
- **A transaction identifier consists of:**
  - **Undo segment number**
  - **Transaction table slot number**
  - **Sequence number or wrap#**

XID = usn# . slot# . wrap#

Undo Block Address

- **The undo block address (UBA) uniquely identifies the undo block for a given transaction; it is found within the ITL of the data block.**
- **A UBA consists of:**
  - **Data block address (DBA) of the block**
  - **The sequence number of the block**
  - **The record number within the block**

UBA = DBA. seq#. rec#

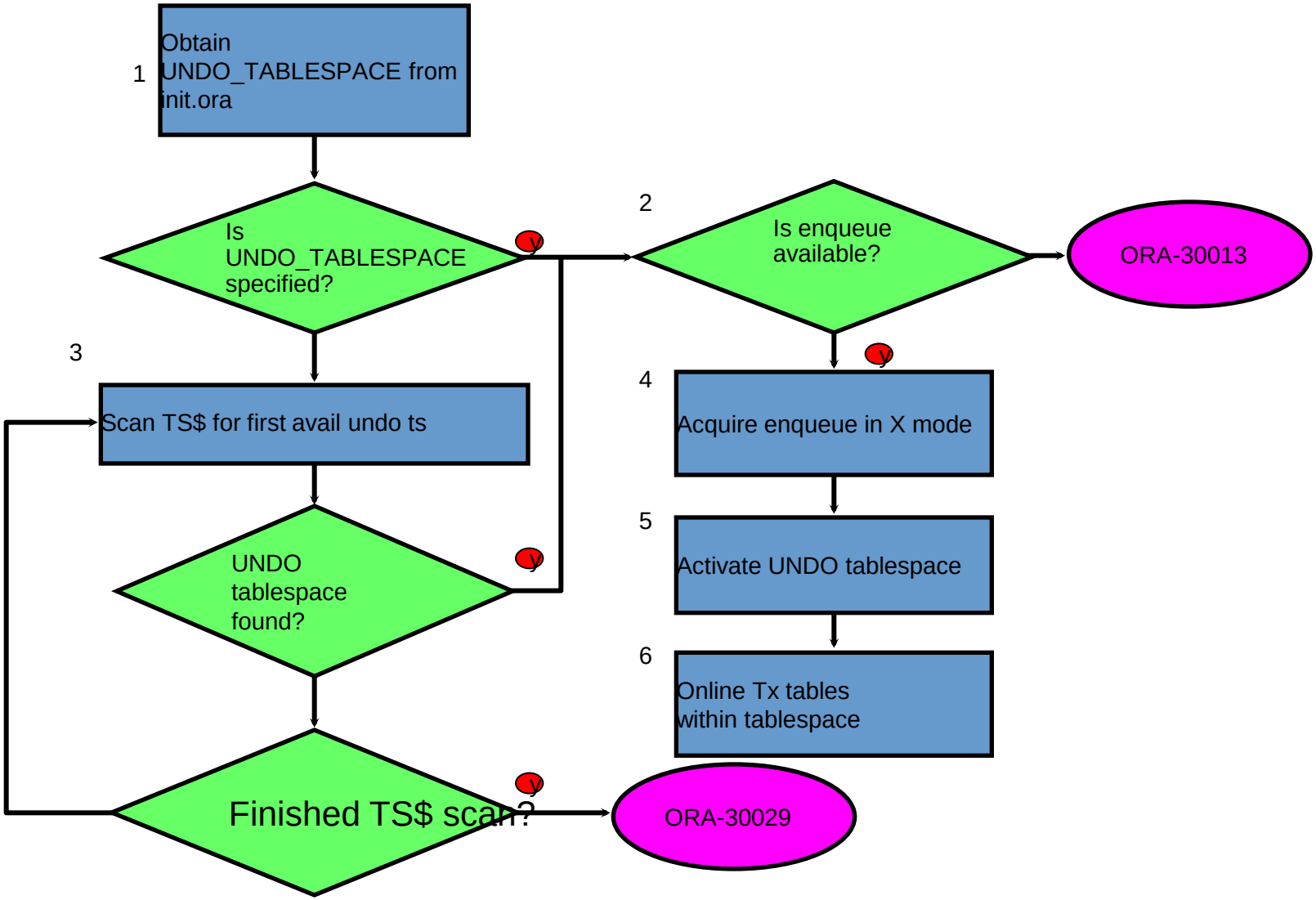
ORA-600 [4xxx]

.
ORA-600 [4000] -> bootstrap$ be locked

ORA-600 [4137] -> the two xid's involved
.
ORA-600 [4193] -> usn and seq#'s
.
ORA-600 [4194] -> usn and rec#'s
.
ORA-600 [4097] -> xid involved and seq# from undo header

ORA-600[4193],[42143],[58021]
.
 undo block sequence number = 42143
 redo block sequence number =58021

Instance Startup Using AUM



UBA和XID

V$TRANSACTION

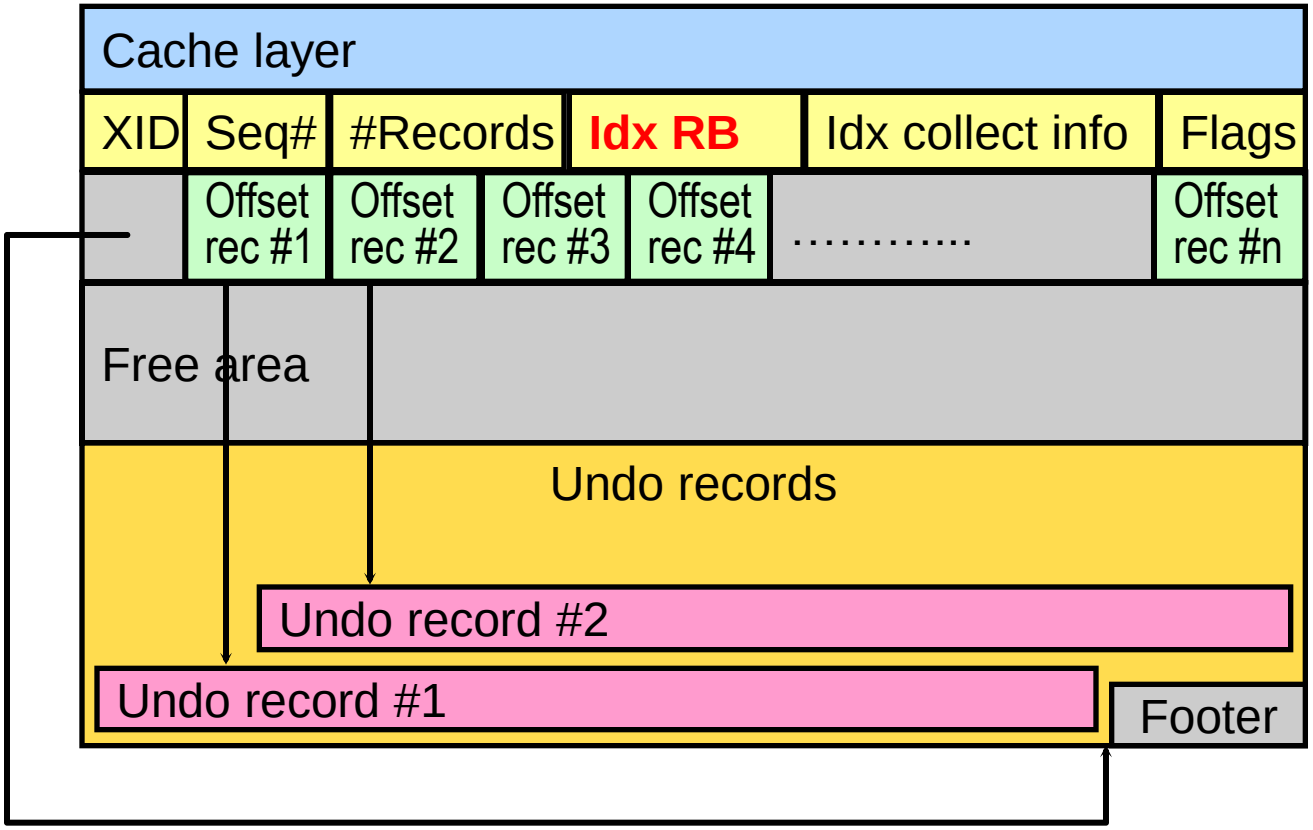
V$TRANSACTION lists the active transactions in the system.

Column	Datatype	Description
ADDR	RAW(4 | 8)	Address of the transaction state object
XIDUSN	NUMBER	Undo segment number
XIDSLOT	NUMBER	Slot number
XIDSQN	NUMBER	Sequence number
UBAFIL	NUMBER	Undo block address (UBA) filenum
UBABLK	NUMBER	UBA block number
UBASQN	NUMBER	UBA sequence number
UBAREC	NUMBER	UBA record number
STATUS	VARCHAR2(16)	Status

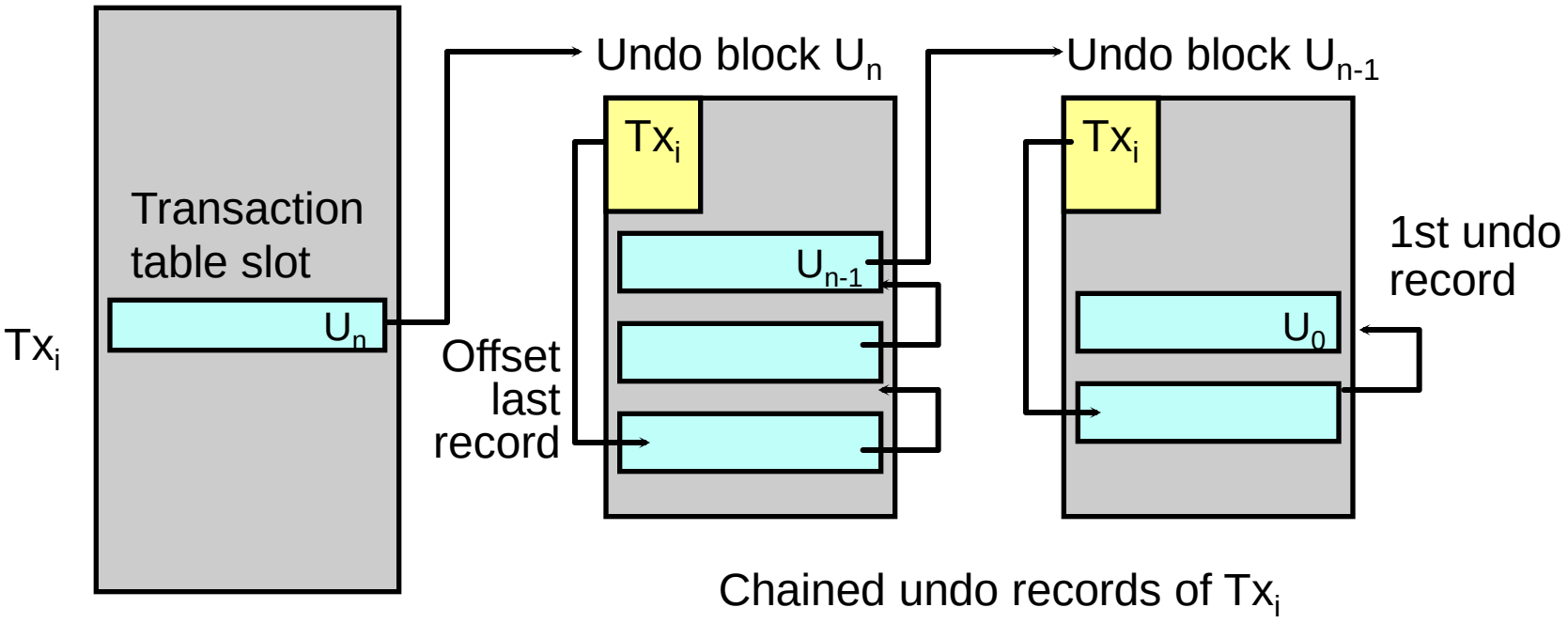
```
SYS@orcl>conn lunar/lunar as sysdba
Connected.
SYS@orcl>select xidusn,xidslot,xidsqn,ubablk,ubafil,ubarec,start_scn from v$transaction;

    XIDUSN    XIDSLOT     XIDSQN     UBABLK     UBAFIL     UBAREC  START_SCN
---------- ---------- ---------- ---------- ---------- ---------- ----------
         5         26        888       1248          2          4    1811149

SYS@orcl>oradebug setmypid
Statement processed.
SYS@orcl>alter system dump undo header '_SYSSMU5$';

System altered.

SYS@orcl>oradebug close_trace
Statement processed.
SYS@orcl>oradebug tracefile_name
/home/oracle/oracle/product/admin/orcl/udump/orcl_ora_13128.tr
SYS@orcl>
```

```
SYS@orcl>oradebug setmypid
Statement processed.
SYS@orcl>alter system dump undo block '_SYSSMU5$' xid 5 26 888;

System altered.

SYS@orcl>oradebug close_trace
Statement processed.
SYS@orcl>oradebug tracefile_name
/home/oracle/oracle/product/admin/orcl/udump/orcl_ora_13134.trc
SYS@orcl>
```

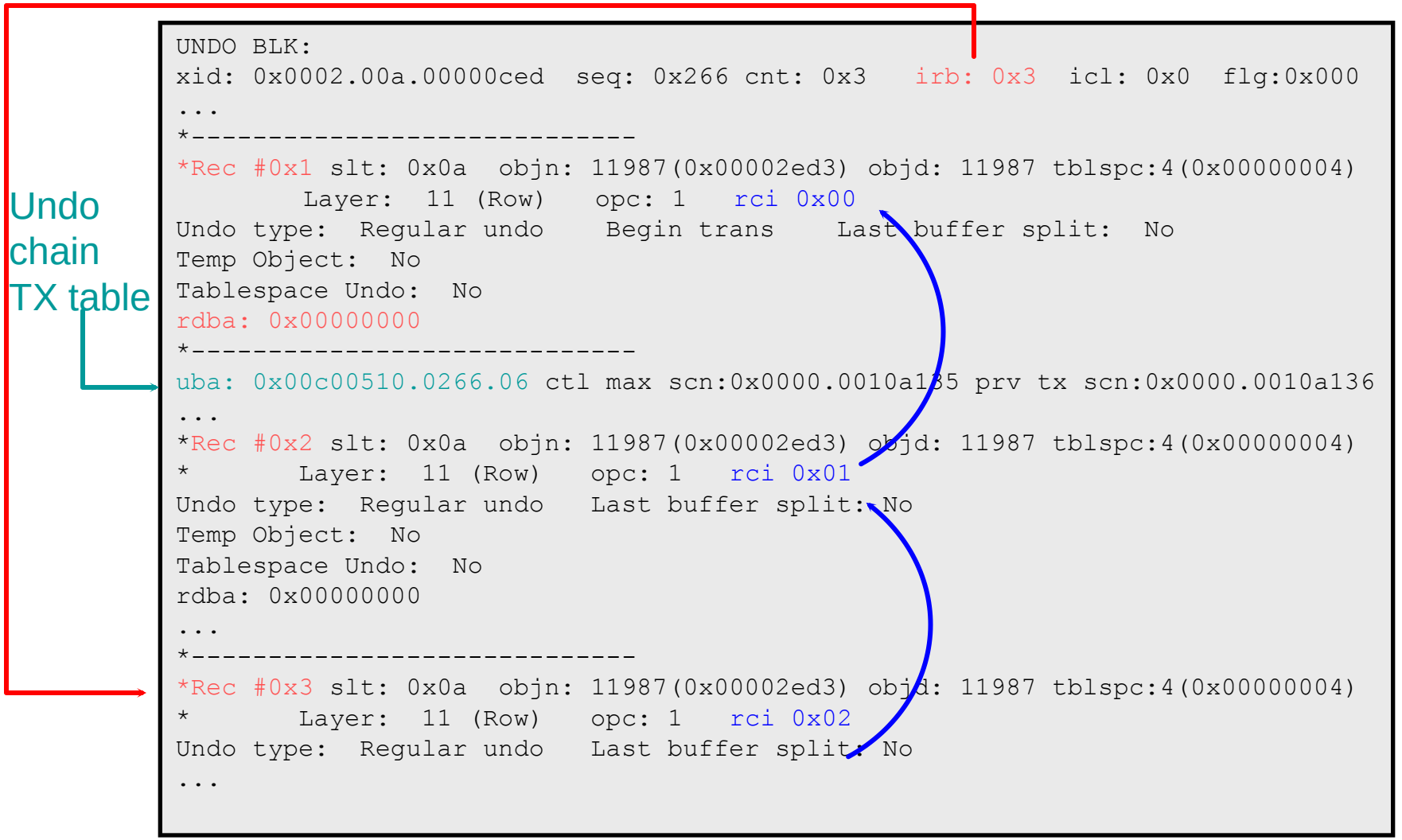
Undo Block


Cache layer
XID
Seq#
#Records
Idx RB
Idx collect info
Flags
Offset rec #1
Offset rec #2
Offset rec #3
Offset rec #4
.....
Offset rec #n
Free area
Undo records
Undo record #2
Undo record #1
Footer

Transaction Undo Chain



Example: Undo Record Chain Dump



相关知识

控制文件相关scn

v$database. checkpoint_change#

v$datafile. checkpoint_change#

数据文件相关scn

v$datafile_header. checkpoint_change#

文件头模糊

v$datafile_header. FUZZY

v$datafile. last_change#

```
  TRN CTL:: seq: 0x021c chd: 0x0011 ctl: 0x000d inc: 0x00000000 nfb: 0x0000
            mgc: 0x8201 xts: 0x0068 flg: 0x0001 opt: 2147483646 (0x7ffffffe)
           uba: 0x008004e0.021c.04 scn: 0x0000.001b9b00
Version: 0x01
  FREE BLOCK POOL::
    uba: 0x00000000.021c.03 ext: 0x1  spc: 0x1d3c
    uba: 0x00000000.021c.16 ext: 0x1  spc: 0x9fa
    uba: 0x00000000.0218.19 ext: 0x1  spc: 0x1244
    uba: 0x00000000.0159.01 ext: 0x2  spc: 0x1f88
    uba: 0x00000000.0000.00 ext: 0x0  spc: 0x0
  TRN TBL::

  index  state cflags  wrap#    uel         scn            dba            parent-xid    nub     stmt_num    cmt
  ------ ----- ------ -------- ----------  -------------- -------------- ------------- ------- ----------- ----------
   0x00     9    0x00  0x0378  0x0005  0x0000.001b9b13  0x00800046  0x0000.000.00000000  0x00000001  0x00000000  1384042814
   0x01     9    0x00  0x0378  0x0024  0x0000.001b9b3c  0x00800046  0x0000.000.00000000  0x00000001  0x00000000  1384042814
   0x02     9    0x00  0x0378  0x0000  0x0000.001b9b09  0x00800046  0x0000.000.00000000  0x00000001  0x00000000  1384042814
   0x03     9    0x00  0x0378  0x0028  0x0000.001b9b23  0x00800046  0x0000.000.00000000  0x00000001  0x00000000  1384042814
   0x04     9    0x00  0x0378  0x0009  0x0000.001b9b94  0x00800047  0x0000.000.00000000  0x00000001  0x00000000  1384042938
   0x05     9    0x00  0x0378  0x0003  0x0000.001b9b18  0x00800046  0x0000.000.00000000  0x00000001  0x00000000  1384042814
   0x06     9    0x00  0x0378  0x0004  0x0000.001b9b5f  0x00800047  0x0000.000.00000000  0x00000001  0x00000000  1384042814
   0x07     9    0x00  0x0378  0x0010  0x0000.001ba11f  0x008004de  0x0000.000.00000000  0x00000001  0x00000000  1384046414
   0x08     9    0x00  0x0378  0x0018  0x0000.001ba19b  0x008004e0  0x0000.000.00000000  0x00000001  0x00000000  1384046414
   0x09     9    0x00  0x0378  0x0021  0x0000.001b9d04  0x00800047  0x0000.000.00000000  0x00000001  0x00000000  1384044011
   0x0a     9    0x00  0x0378  0x0017  0x0000.001ba173  0x008004df  0x0000.000.00000000  0x00000001  0x00000000  1384046414
   0x0b     9    0x00  0x0378  0x002d  0x0000.001b9ebb  0x00800047  0x0000.000.00000000  0x00000001  0x00000000  1384045214
   0x0c     9    0x00  0x0378  0x0023  0x0000.001b9eee  0x008004de  0x0000.000.00000000  0x00000003  0x00000000  1384045222
   0x0d     9    0x00  0x0378  0xffff  0x0000.001ba1a6  0x008004e0  0x0000.000.00000000  0x00000001  0x00000000  1384046414
   0x0e     9    0x00  0x0377  0x002b  0x0000.001ba185  0x008004df  0x0000.000.00000000  0x00000001  0x00000000  1384046414
```

```
SYS@orcl>select dbms_utility.data_block_address_file(to_number(substr(uba, 3, 8),
  2                                                   'xxxxxxxxxxxx')) as undo_file#,
  3         dbms_utility.data_block_address_block(to_number(substr(uba, 3, 8),
  4                                                 'xxxxxxxxxxxx')) as undo_block,
  5         to_number(substr(uba, 12, 4), 'xxxxxxxxxxxx') undo_sequence,
  6         to_number(substr(uba, 17, 2), 'xxxxxxxxxxxx') undo_record
  7    from (select '0x008004e0.021c.04' uba from dual);

UNDO_FILE# UNDO_BLOCK UNDO_SEQUENCE UNDO_RECORD
---------- ---------- ------------- -----------
         2       1248           540           4

SYS@orcl>
```

index	state	cflags	wrap#	uel	scn	dba	parent-xid	nub	stmt_num	cmt
0x00	9	0x00	0x0378	0x0005	0x0000.001b9b13	0x00800046	0x0000.000.00000000	0x00000001	0x00000000	1384042814
0x01	9	0x00	0x0378	0x0024	0x0000.001b9b3c	0x00800046	0x0000.000.00000000	0x00000001	0x00000000	1384042814
0x02	9	0x00	0x0378	0x0000	0x0000.001b9b09	0x00800046	0x0000.000.00000000	0x00000001	0x00000000	1384042814
0x03	9	0x00	0x0378	0x0028	0x0000.001b9b23	0x00800046	0x0000.000.00000000	0x00000001	0x00000000	1384042814
0x04	9	0x00	0x0378	0x0009	0x0000.001b9b94	0x00800047	0x0000.000.00000000	0x00000001	0x00000000	1384042938
0x05	9	0x00	0x0378	0x0003	0x0000.001b9b18	0x00800046	0x0000.000.00000000	0x00000001	0x00000000	1384042814
0x06	9	0x00	0x0378	0x0004	0x0000.001b9b5f	0x00800047	0x0000.000.00000000	0x00000001	0x00000000	1384042814
0x07	9	0x00	0x0378	0x0010	0x0000.001ba11f	0x008004de	0x0000.000.00000000	0x00000001	0x00000000	1384046414
0x08	9	0x00	0x0378	0x0018	0x0000.001ba19b	0x008004e0	0x0000.000.00000000	0x00000001	0x00000000	1384046414
0x09	9	0x00	0x0378	0x0021	0x0000.001b9d04	0x00800047	0x0000.000.00000000	0x00000001	0x00000000	1384044011
0x0a	9	0x00	0x0378	0x0017	0x0000.001ba173	0x008004df	0x0000.000.00000000	0x00000001	0x00000000	1384046414
0x0b	9	0x00	0x0378	0x002d	0x0000.001b9ebb	0x00800047	0x0000.000.00000000	0x00000001	0x00000000	1384045214
0x0c	9	0x00	0x0378	0x0023	0x0000.001b9eee	0x008004de	0x0000.000.00000000	0x00000003	0x00000000	1384045222
0x0d	9	0x00	0x0378	0xffff	0x0000.001ba1a6	0x008004e0	0x0000.000.00000000	0x00000001	0x00000000	1384046414
0x0e	9	0x00	0x0377	0x002b	0x0000.001ba185	0x008004df	0x0000.000.00000000	0x00000001	0x00000000	1384046414
0x0f	9	0x00	0x0378	0x0012	0x0000.001b9ee7	0x0080004b	0x0000.000.00000000	0x00000003	0x00000000	1384045222
0x10	9	0x00	0x0378	0x0025	0x0000.001ba123	0x008004de	0x0000.000.00000000	0x00000001	0x00000000	1384046414
0x11	9	0x00	0x0378	0x0002	0x0000.001b9b05	0x00800046	0x0000.000.00000000	0x00000001	0x00000000	1384042814
0x12	9	0x00	0x0378	0x002e	0x0000.001b9eea	0x0080004b	0x0000.000.00000000	0x00000001	0x00000000	1384045222
0x13	9	0x00	0x0378	0x0027	0x0000.001ba032	0x008004de	0x0000.000.00000000	0x00000001	0x00000000	1384046002
0x14	9	0x00	0x0378	0x0029	0x0000.001ba158	0x008004df	0x0000.000.00000000	0x00000001	0x00000000	1384046414
0x15	9	0x00	0x0378	0x0014	0x0000.001ba153	0x008004de	0x0000.000.00000000	0x00000001	0x00000000	1384046414
0x16	9	0x00	0x0378	0x0026	0x0000.001ba0dc	0x008004de	0x0000.000.00000000	0x00000001	0x00000000	1384046414
0x17	9	0x00	0x0378	0x000e	0x0000.001ba179	0x008004df	0x0000.000.00000000	0x00000001	0x00000000	1384046414
0x18	9	0x00	0x0378	0x000d	0x0000.001ba1a0	0x008004e0	0x0000.000.00000000	0x00000001	0x00000000	1384046414
0x19	9	0x00	0x0378	0x0016	0x0000.001ba0ce	0x008004de	0x0000.000.00000000	0x00000001	0x00000000	1384046414
0x1a	10	0x80	0x0378	0x0001	0x0000.001ba2cd	0x008004e0	0x0000.000.00000000	0x00000001	0x00000000	0
0x1b	9	0x00	0x0377	0x002a	0x0000.001b9b52	0x00800046	0x0000.000.00000000	0x00000001	0x00000000	1384042814

1a------》slot 26

State----》10 未提交事物

Dba-----》uba---- 0x008004e0

```
SYS@orcl>select dbms_utility.data_block_address_file(to_number('008004e0','xxxxxxxxxxxxxx')) as undo_file# from dual;

UNDO_FILE#
----------
         2

SYS@orcl>select dbms_utility.data_block_address_block(to_number('008004e0','xxxxxxxxxxxxxx')) as undo_block# from dua

UNDO_BLOCK#
-----------
      1248

SYS@orcl>
```

oradebug setmypid

alter system dump undo block '_SYSSMU5$' xid 5 26 888;

oradebug close_trace

oradebug tracefile_name

Undo header：

```
  TRN CTL:: seq: 0x021c chd: 0x0011 ctl: 0x000d inc: 0x00000000 nfb: 0x0000
            mgc: 0x8201 xts: 0x0068 flg: 0x0001 opt: 2147483646 (0x7ffffffe)
            uba: 0x008004e0.021c.04 scn: 0x0000.001b9b00
```

```
SYS@orcl>select xidusn,xidslot,xidsqn,ubablk,ubafil,ubarec,start_scn from v$transaction;

    XIDUSN    XIDSLOT     XIDSQN    UBABLK     UBAFIL     UBAREC  START_SCN
---------- ---------- ---------- ---------- ---------- ---------- ----------
         5         26        888       1248          2          4    1811149
```

```
*******************************************************************************
UNDO BLK:  Extent: 1   Block: 7   dba (file#, block#): 2,0x000004e0
xid: 0x0005.01a.00000378  seq: 0x21c cnt: 0x4   irb: 0x4   icl: 0x0   flg: 0x0000

 Rec Offset      Rec Offset      Rec Offset      Rec Offset      Rec Offset
----------------------------------------------------------------------------
0x01 0x1f0c     0x02 0x1e30     0x03 0x1d54     0x04 0x1c48

*-----------------------------
* Rec #0x4  slt: 0x1a  objn: 55083(0x0000d72b)  objd: 55083  tblspc: 4(0x00000004)
*       Layer:  11 (Row)   opc: 1   rci 0x00
Undo type:  Regular undo    Begin trans    Last buffer split:  No
Temp Object:  No
Tablespace Undo:  No
rdba: 0x00000000
*-----------------------------
uba: 0x008004e0.021c.03 ctl max scn: 0x0000.001b9af9 prv tx scn: 0x0000.001b9b00
txn start scn: scn: 0x0000.001ba2cd logon user: 77
 prev brb: 8388678 prev bcl: 0
KDO undo record:
KTB Redo
op: 0x03  ver: 0x01
op: Z
KDO Op code: IRP row dependencies Disabled
  xtype: XA flags: 0x00000000  bdba: 0x01000534  hdba: 0x01000533
itli: 2  ispac: 0  maxfr: 4858
```

04e0--→block 1248

xid: 0x0005.01a.00000378
Undo Segment Number : 5
Transaction Table Slot Number : 26
Wrap : 888

```
uba: 0x008004e0.021c.03 ctl max scn: 0x0000.001b9af9 prv tx scn: 0x0000.001b9b00
txn start scn: scn: 0x0000.001ba2cd logon user: 77
 prev brb: 8388678 prev bcl: 0
KDO undo record:
KTB Redo
op: 0x03  ver: 0x01
op: Z
KDO Op code: IRP row dependencies Disabled
  xtype: XA flags: 0x00000000  bdba: 0x01000534  hdba: 0x01000533
itli: 2  ispac: 0  maxfr: 4858
tabn: 0 slot: 11(0xb) size/delt: 86
fb: --H-FL-- lb: 0x0  cc: 11
null:
01234567890123456789012345678901234567890123456789012345678901234567890123456789
----NN-----
tol  0: [ 5]  4c 55 4e 41 52
tol  1: [ 2]  c1 4e
tol  2: [16]  38 35 39 31 38 37 32 35 37 34 38 38 30 44 37 41
tol  3: [ 4]  4f 50 45 4e
tol  4: *NULL*
tol  5: *NULL*
tol  6: [ 5]  55 53 45 52 53
tol  7: [ 4]  54 45 4d 50
tol  8: [ 7]  78 6d 08 12 11 22 03
tol  9: [ 7]  44 45 46 41 55 4c 54
tol 10: [22]
 44 45 46 41 55 4c 54 5f 43 4f 4e 53 55 4d 45 52 5f 47 52 4f 55 50
```

```
SYS@orcl>select utl_raw.cast_to_varchar2('4c554e4152') from dual;

UTL_RAW.CAST_TO_VARCHAR2('4C554E4152')
--------------------------------------------------------------------------------
LUNAR

SYS@orcl>
```

```
LUNAR@orcl>update t set username='FF' where username='ZDP';

1 row updated.

LUNAR@orcl>
```

```
SYS@orcl>select xidusn,xidslot,xidsqn,ubablk,ubafil,ubarec,start_scn from v$transaction;

    XIDUSN    XIDSLOT     XIDSQN     UBABLK     UBAFIL     UBAREC  START_SCN
---------- ---------- ---------- ---------- ---------- ---------- ----------
         5         26        888       1248          2          5    1811149

SYS@orcl>
```

```
SYS@orcl>oradebug close_trace
Statement processed.
SYS@orcl>oradebug setmypid
Statement processed
SYS@orcl>alter system dump undo block '_SYSSMU5$' xid 5 26 888;

System altered.

SYS@orcl>oradebug close_trace
Statement processed.
SYS@orcl>oradebug tracefile_name
/home/oracle/oracle/product/admin/orcl/udump/orcl_ora_13134.trc
SYS@orcl>
```

```
SYS@orcl>oradebug setmypid
Statement processed.
SYS@orcl>alter system dump undo header '_SYSSMU5$';

System altered.

SYS@orcl>oradebug close_trace
Statement processed.
SYS@orcl>oradebug tracefile_name
/home/oracle/oracle/product/admin/orcl/udump/orcl_ora_13134.trc
SYS@orcl>
```

```
**********************************************************************
UNDO BLK:  Extent: 1    Block: 7    dba (file#, block#): 2,0x000004e0
xid: 0x0005.01a.00000378  seq: 0x21c cnt: 0x5   irb: 0x5   icl: 0x0   flg: 0x0000

 Rec Offset      Rec Offset      Rec Offset      Rec Offset      Rec Offset
----------------------------------------------------------------------------
0x01 0x1f0c     0x02 0x1e30     0x03 0x1d54     0x04 0x1c48     0x05 0x1bd4

*-----------------------
* Rec #0x5  slt: 0x1a  objn: 55083(0x0000d72b)  objd: 55083  tblspc: 4(0x00000004)
*       Layer:  11 (Row)    opc: 1    rci 0x04
Undo type:  Regular undo    Last buffer split:  No
Temp Object:  No
Tablespace Undo:  No
rdba: 0x00000000
*-----------------------------
KDO undo record:
KTB Redo
op: 0x02  ver: 0x01
op: C  uba: 0x008004e0.021c.04
KDO Op code: URP row dependencies Disabled
  xtype: XA flags: 0x00000000  bdba: 0x01000534  hdba: 0x01000533
itli: 2  ispac: 0  maxfr: 4858
tabn: 0 slot: 19(0x13) flag: 0x2c lock: 0 ckix: 183
ncol: 11 nnew: 1 size: 1
col  0: [ 3]  5a 44 50

*-----------------------
* Rec #0x4  slt: 0x1a  objn: 55083(0x0000d72b)  objd: 55083  tblspc: 4(0x00000004)
*       Layer:  11 (Row)    opc: 1    rci 0x00
```

1, 事物恢复的起点 irb: recoder 0x5 ， cnt 对应V$TRANSACTION.UBAREC

2, 从recoder 5 找到前一个需要恢复的记录 0x04

3, recoder 5中记录了before image：

```
SYS@orcl>select utl_raw.cast_to_varchar2('5a4450') from dual;

UTL_RAW.CAST_TO_VARCHAR2('5A4450')
--------------------------------------------------------------------------------
ZDP

SYS@orcl>
```

BBED初认识

- 从Oracle 7.3.2的部分平台和Oracle 8开始随产品一起发布
- 从Oracle 9i以后，BBED不再随产品发布，Linux和Unix下需要手工编译
- **BBED is a SUPPORT ONLY tool and should NOT be discussed with customers.**

```
BBED (Block Browser/Editor)
---------------------------
1. Getting access to BBED:

   BBED is shipped with Oracle8 releases, and with some Oracle7.3 releases.
   BBED is a SUPPORT ONLY tool and should NOT be discussed with customers.

   On UNIX the tool needs to be built using a command of the form:

        cd $ORACLE_HOME/rdbms/lib
        make -f ins_rdbms.mk $ORACLE_HOME/rdbms/lib/bbed

   OR to build in the bn directory:

        make -f ins_rdbms.mk BBED=$ORACLE_HOME/bin/bbed $ORACLE_HOME/bin/bbed

   Starting on 11g, BBED is no longer shipped and cannot be linked at customer's sit
   A diag patch needs to be requested to have to it deployed internally.

   On Windows NT a BBED.EXE is shipped as an executable and so is password
   protected.  This password protection is present on Unix from 8.1.6 onwards.
```

BBED初认识

WINDOWS

In Windows,You can find BBED executable in %ORACLE_HOME%/bin folder.

For Windows, BBED is no longer shipped as of version 10g

UNIX

In Unix, BBED executable needs to be built using the following steps:

```
cd $ORACLE_HOME/rdbms/lib
make -f ins_rdbms.mk $ORACLE_HOME/rdbms/lib/bbed
(OR)
To build in the bin directory:
make -f ins_rdbms.mk BBED=$ORACLE_HOME/bin/bbed $ORACLE_HOME/bin/bbed
```

For Unix, BBED is no longer shipped as of version 11g.

BBED安装和初使用---Oracle 8~8i

UNIX 平台：

从Oracle 8开始随产品附带，需要单独编译：

```
[oracle@lunar ~]$ env|grep ORA
ORACLE_SID=ora816
ORACLE_BASE=/ora816
ORACLE_HOME=/ora816
[oracle@lunar ~]$ cd $ORACLE_HOME/rdbms/lib
[oracle@lunar lib]$ make -f ins_rdbms.mk $ORACLE_HOME/rdbms/lib/bbed
```

```
[oracle@lunar lib]$
[oracle@lunar lib]$ make -f ins_rdbms.mk BBED=$ORACLE_HOME/bin/bbed $ORACLE_HOME/bin/bbed
/ora816/rdbms/lib/env_rdbms.mk:1954: warning: overriding commands for target `libclntsh.so'
/ora816/rdbms/lib/env_rdbms.mk:1877: warning: ignoring old commands for target `libclntsh.so'

Linking BBED utility (bbed)
rm -f /ora816/bin/bbed
gcc -o /ora816/bin/bbed -L/ora816/rdbms/lib/ -L/ora816/lib/  /ora816/lib/s0main.o /ora816/rdbms/lib/ssbbded.o /ora816/rdbms/lib/sbbdpt.o `sed -e 's/
-ljava//g' /ora816/lib/ldflags`     -lnsgr8 -lnzjs8 -ln8 -lnl8 /ora816/rdbms/lib/defopt.o /ora816/rdbms/lib/libdbtools8.a -lclntsh  `sed -e 's/-ljav
a//g' /ora816/lib/ldflags`     -lnsgr8 -lnzjs8 -ln8 -lnl8 -lnro8 `sed -e 's/-ljava//g' /ora816/lib/ldflags`     -lnsgr8 -lnzjs8 -ln8 -lnl8 -lclient8
  -lvsn8 -lwtc8 -lcommon8 -lgeneric8 -lwtc8 -lmm -lnls8  -lcore8 -lnls8 -lcore8 -lnls8  `sed -e 's/-ljava//g' /ora816/lib/ldflags`     -lnsgr8 -lnzj
s8 -ln8 -lnl8 -lnro8 `sed -e 's/-ljava//g' /ora816/lib/ldflags`     -lnsgr8 -lnzjs8 -ln8 -lnl8 -lclient8  -lvsn8 -lwtc8 -lcommon8 -lgeneric8  -ltrac
e8 -lnls8  -lcore8 -lnls8 -lcore8 -lnls8  -lclient8  -lvsn8 -lwtc8 -lcommon8 -lgeneric8 -lnls8  -lcore8 -lnls8 -lcore8 -lnls8    `cat /ora816/lib/sy
sliblist` `if [ -f /usr/lib/libsched.so ] ; then echo -lsched ; else true; fi`  -Wl,-rpath,/ora816/lib:/lib:/usr/lib -lm    `cat /ora816/lib/sysliblist` `if [ -f /usr/lib/libsched.so ] ; then echo -lsched ; else true; fi`  -ldl -lm  -ldl `sed -e 's/-ljava//g' /ora816/lib/ldflags`     -lnsgr8 -ln
zjs8 -ln8 -lnl8 -lnro8 `sed -e 's/-ljava//g' /ora816/lib/ldflags`     -lnsgr8 -lnzjs8 -ln8 -lnl8 `sed -e 's/-ljava//g' /ora816/lib/ldflags`     -lns
gr8 -lnzjs8 -ln8 -lnl8 -lnro8 `sed -e 's/-ljava//g' /ora816/lib/ldflags`     -lnsgr8 -lnzjs8 -ln8 -lnl8 `sed -e 's/-ljava//g' /ora816/lib/ldflags`
   -lnsgr8 -lnzjs8 -ln8 -lnl8 -lnro8 `sed -e 's/-ljava//g' /ora816/lib/ldflags`     -lnsgr8 -lnzjs8 -ln8 -lnl8 `sed -e 's/-ljava//g' /ora816/lib/ldf
lags`     -lnsgr8 -lnzjs8 -ln8 -lnl8 -lnro8 `sed -e 's/-ljava//g' /ora816/lib/ldflags`     -lnsgr8 -lnzjs8 -ln8 -lnl8 `sed -e 's/-ljava//g' /ora816/
lib/ldflags`     -lnsgr8 -lnzjs8 -ln8 -lnl8 -lnro8 `sed -e 's/-ljava//g' /ora816/lib/ldflags`     -lnsgr8 -lnzjs8 -ln8 -lnl8 -lsql8 -lsql8
[oracle@lunar lib]$ bbed
Password:

BBED: Release 2.0.0.0.0 - Limited Production on Sun Nov 10 06:58:04 2013

(c) Copyright 1999 Oracle Corporation.  All rights reserved.

************ !!! For Oracle Internal Use only !!! ***************

BBED>
```

BBED安装和初使用---9i~10g

在9i/10g中连接生成bbed:

cd $ORACLE_HOME/rdbms/lib

make -f ins_rdbms.mk $ORACLE_HOME/rdbms/lib/bbed

mv bbed $ORACLE_HOME/bin

```
[oracle@lunar ~]$ env|grep ORA
ORACLE_SID=orcl
ORACLE_BASE=/home/oracle/oracle/product
ORACLE_HOME=/home/oracle/oracle/product/10.2.0/db_1
[oracle@lunar ~]$ which bbed
~/oracle/product/10.2.0/db_1/bin/bbed
[oracle@lunar ~]$ cd bbed
[oracle@lunar bbed]$ cat bbed.par
blocksize=8192
listfile=/home/oracle/bbed/filelist.txt
mode=edit
password=blockedit
[oracle@lunar bbed]$ cat filelist.txt
         1 /home/oracle/oracle/product/oradata/ORCL/datafile/o1_mf_system_3sf9zrsf_.dbf                513802240
         2 /home/oracle/oracle/product/oradata/ORCL/datafile/o1_mf_undotbs1_3sf9zs19_.dbf               41943040
         3 /home/oracle/oracle/product/oradata/ORCL/datafile/o1_mf_sysaux_3sf9zrtg_.dbf                293601280
         4 /home/oracle/oracle/product/oradata/ORCL/datafile/o1_mf_users_3sf9zs2c_.dbf                 284426240
         5 /home/oracle/oracle/product/oradata/ORCL/datafile/o1_mf_example_3sfb3c6c_.dbf               72032256
         6 /home/oracle/oracle/product/oradata/ORCL/datafile/alex01.dbf                                10485760
         7 /home/oracle/oracle/product/oradata/ORCL/lunar.dbf                                                 0
         9 /home/oracle/oracle/product/oradata/system1.dbf                                             15728640
[oracle@lunar bbed]$ bbed parfile=bbed.par

BBED: Release 2.0.0.0.0 - Limited Production on Sun Nov 10 07:15:16 2013

Copyright (c) 1982, 2005, Oracle.  All rights reserved.

********** !!! For Oracle Internal Use only !!! ***************

BBED>
```

BBED安装和初使用---11g~12c

在11g和12c中生成bbed:

cd $ORACLE_HOME/rdbms/lib

make -f ins_rdbms.mk $ORACLE_HOME/rdbms/lib/bbed

```
[oracle@lunar ~]$ . ora112.env
[oracle@lunar ~]$ env|grep ORA
ORACLE_SID=bb
ORACLE_BASE=/u01/app/oracle
ORACLE_HOME=/u01/app/oracle/product/11.2.0.3/dbhome_1
[oracle@lunar ~]$ which bbed
/u01/app/oracle/product/11.2.0.3/dbhome_1/bin/bbed
[oracle@lunar ~]$
[oracle@lunar ~]$ ll $ORACLE_HOME/rdbms/lib/*bb*
-rw-r--r-- 1 oracle oinstall 3306 Feb 10  2013 /u01/app/oracle/product/11.2.0.3/dbhome_1/rdbms/lib/ssbbded.o
-rw-r--r-- 1 oracle oinstall 3976 Feb 10  2013 /u01/app/oracle/product/11.2.0.3/dbhome_1/rdbms/lib/sbbdpt.o
[oracle@lunar ~]$ ll $ORACLE_HOME/rdbms/mesg/*bb*
-rw-r--r-- 1 oracle oinstall 8704 Feb 10  2013 /u01/app/oracle/product/11.2.0.3/dbhome_1/rdbms/mesg/bbedus.msb
[oracle@lunar ~]$
```

BBED和ASM文件

```
ASMCMD> cp USERS.268.818251547 /tmp/datafile.sample.12c.dbf
copying +data/lunarbb/datafile/USERS.268.818251547 -> /tmp/datafile.sample.12c.dbf
ASMCMD> exit
[grid@lunar ~]$ env|grep ORA
ORACLE_SID=+ASM
ORACLE_BASE=/u01/app/grid
ORACLE_HOME=/u01/app/12.1/grid
[grid@lunar ~]$
```

```
[oracle@lunar ~]$ . ora12.env
[oracle@lunar ~]$ env|grep ORA
ORACLE_SID=lunarbb
ORACLE_BASE=/u01/app/oracle
ORACLE_HOME=/u01/app/oracle/product/12.1/dbhome_1
[oracle@lunar ~]$ which bbed
/u01/app/oracle/product/12.1/dbhome_1/bin/bbed
[oracle@lunar ~]$ cd bbed
[oracle@lunar bbed]$ cat filelist.lst.12c.datafile
1 /tmp/datafile.sample.12c.dbf
[oracle@lunar bbed]$
[oracle@lunar bbed]$ cat bbed.par
blocksize=8192
#listfile=/home/oracle/bbed/filelist.txt.11.2.redo
#listfile=/home/oracle/bbed/filelist.txt.11.2.datafile
listfile=/home/oracle/bbed/filelist.lst.12c.datafile
mode=edit
password=blockedit

[oracle@lunar bbed]$ bbed parfile=bbed.par

BBED: Release 2.0.0.0.0 - Limited Production on Thu Nov 28 00:12:04 2013

Copyright (c) 1982, 2013, Oracle and/or its affiliates.  All rights reserved.

************** !!! For Oracle Internal Use only !!! ***************

BBED>
```

BBED常用命令

```
[oracle@lunar bbed]$ bbed parfile=bbed.par

BBED: Release 2.0.0.0.0 - Limited Production on Sun Nov 10 07:13:39 2013

Copyright (c) 1982, 2005, Oracle.  All rights reserved.

********** !!! For Oracle Internal Use only !!! ***************

BBED> help
HELP [ <bbed command> | ALL ]

BBED> help all
SET DBA [ dba | file#, block# ]
SET FILENAME 'filename'
SET FILE file#
SET BLOCK [+/-]block#
SET OFFSET [ [+/-]byte offset | symbol | *symbol ]
SET BLOCKSIZE bytes
SET LIST[FILE] 'filename'
SET WIDTH character_count
SET COUNT bytes_to_display
SET IBASE [ HEX | OCT | DEC ]
SET OBASE [ HEX | OCT | DEC ]
SET MODE  [ BROWSE | EDIT ]
SET SPOOL [ Y | N ]
SHOW [ <SET parameter> | ALL ]
INFO
MAP[/v] [ DBA | FILENAME | FILE | BLOCK ]
DUMP[/v] [ DBA | FILENAME | FILE | BLOCK | OFFSET | COUNT ]
PRINT[/x|d|u|o|c] [ DBA | FILE | FILENAME | BLOCK | OFFSET | symbol | *symbol ]
EXAMINE[/Nuf] [ DBA | FILE | FILENAME | BLOCK | OFFSET | symbol | *symbol ]
```

案例分析1

```
Tue Oct 15 10:20:03 2013
Errors in file /oracle/app/product/diag/rdbms/zydb/zydb01/trace/zydb01_smon_43712808.trc  (incident=286036):
ORA-00600: internal error code, arguments: [4097], [3909], [23], [121212], [], [], [], [], [], [], [], []
Incident details in: /oracle/app/product/diag/rdbms/zydb/zydb01/incident/incdir_286036/zydb01_smon_43712808_i286036.trc
Tue Oct 15 10:20:08 2013
Trace dumping is performing id=[cdmp_20131015102008]
Tue Oct 15 10:20:11 2013
Sweep [inc][286036]: completed
Sweep [inc2][286036]: completed
Tue Oct 15 10:20:36 2013
Errors in file /oracle/app/product/diag/rdbms/zydb/zydb01/trace/zydb01_smon_43712808.trc:
ORA-00308: cannot open archived log '/arch3/zydb03/3_28984_766859529.dbf'
ORA-17503: ksfdopn:4 Failed to open file /arch3/zydb03/3_28984_766859529.dbf
ORA-17500: ODM err:File does not exist
ORA-00600: internal error code, arguments: [4097], [3909], [23], [121212], [], [], [], [], [], [], [], []
Fatal internal error happened while SMON was doing instance transaction recovery.
Errors in file /oracle/app/product/diag/rdbms/zydb/zydb01/trace/zydb01_smon_43712808.trc:
ORA-00600: internal error code, arguments: [4097], [3909], [23], [121212], [], [], [], [], [], [], [], []
SMON (ospid: 43712808): terminating the instance due to error 474
Tue Oct 15 10:20:36 2013
opiodr aborting process unknown ospid (36831730) as a result of ORA-1092
Tue Oct 15 10:20:36 2013
System state dump is made for local instance
Tue Oct 15 10:20:36 2013
opiodr aborting process unknown ospid (29229546) as a result of ORA-1092
System State dumped to trace file /oracle/app/product/diag/rdbms/zydb/zydb01/trace/zydb01_diag_36110646.trc
Tue Oct 15 10:20:37 2013
ORA-1092 : opitsk aborting process
Tue Oct 15 10:20:37 2013
License high water mark = 348
Instance terminated by SMON, pid = 43712808
USER (ospid: 44040620): terminating the instance
Instance terminated by USER, pid = 44040620
```

Oracle进行延迟块清除时,会查询回滚段头以确认事务状态,一些异常情况或者smon_scn_time信息紊乱等可能造成回滚段信息损坏，使得系统查询到的回滚段的信息超前于当前数据库的状态，类似情况就会造成ORA-600 [4097] 错误

编辑1 | zydb01_ora_15204648.trc

```
1  Trace file /oracle/app/product/diag/rdbms/zydb/zydb01/trace/zydb01_ora_15204648.trc
2  Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
3  With the Partitioning, Real Application Clusters, Oracle Label Security, OLAP,
4  Data Mining, Oracle Database Vault and Real Application Testing options
5  ORACLE_HOME = /oracle/app/product/db11gr2
6  System name:  AIX
7  Node name:  zdzydb01
8  Release:  1
9  Version:  6
10 Machine:  00F6E5DE4C00
11 Instance name: zydb01
12 Redo thread mounted by this instance: 1
13 Oracle process number: 30
14 Unix process pid: 15204648, image: oracle@zdzydb01 (TNS V1-V3)
```

```
75 Recovery sets nab of thread 1 seq 28859 to 3 with 8 zeroblks
76
77 *** 2013-10-16 12:44:52.808
78 2013-10-16 12:44:52.808043 : Validate domain 0
79 2013-10-16 12:44:52.808833 : Validated domain 0, flags = 0x0
80
81 *** 2013-10-16 12:44:55.093
82 Incident 414187 created, dump file: /oracle/app/product/diag/rdbms/zydb/zydb01/incident/incdir_414187/zydb01_ora_15204648_i414187.trc
83 ORA-00600: internal error code, arguments: [4097], [3909], [23], [121212], [], [], [], [], [], [], [], []
84
85 ORA-00600: internal error code, arguments: [4097], [3909], [23], [121212], [], [], [], [], [], [], [], []
86 ORA-00600: internal error code, arguments: [4097], [3909], [23], [121212], [], [], [], [], [], [], [], []
87
88 *** 2013-10-16 12:45:00.155
89 USER (ospid: 15204648): terminating the instance due to error 600
90
```

案例分析2

```
ORA-01092: ORACLE instance terminated. Disconnection forced
ORA-00704: bootstrap process failure
ORA-00600: internal error code, arguments: [4000], [3], [], [], [], [], [], [], [], [], []
Process ID: 15101
Session ID: 125 Serial number: 5


SYS@bb>

查看alert:
SMON: enabling cache recovery
Errors in file /u01/app/oracle/diag/rdbms/bb/bb/trace/bb_ora_15101.trc  (incident=14537):
ORA-00600: internal error code, arguments: [4000], [3], [], [], [], [], [], [], [], [], []
Incident details in: /u01/app/oracle/diag/rdbms/bb/bb/incident/incdir_14537/bb_ora_15101_i14537.trc
Use ADRCI or Support Workbench to package the incident.
See Note 411.1 at My Oracle Support for error and packaging details.
Errors in file /u01/app/oracle/diag/rdbms/bb/bb/trace/bb_ora_15101.trc:
ORA-00704: bootstrap process failure
ORA-00600: internal error code, arguments: [4000], [3], [], [], [], [], [], [], [], [], []
Errors in file /u01/app/oracle/diag/rdbms/bb/bb/trace/bb_ora_15101.trc:
ORA-00704: bootstrap process failure
ORA-00600: internal error code, arguments: [4000], [3], [], [], [], [], [], [], [], [], []
Error 704 happened during db open, shutting down database
USER (ospid: 15101): terminating the instance due to error 704
Instance terminated by USER, pid = 15101
ORA-1092 signalled during: ALTER DATABASE OPEN...
opiodr aborting process unknown ospid (15101) as a result of ORA-1092
```

```
*** 2013-10-27 21:55:54.685
Incident 14537 created, dump file: /u01/app/oracle/diag/rdbms/bb/bb/incident/incdir_14537/bb_ora_15101_i14537.trc
ORA-00600: internal error code, arguments: [4000], [3], [], [], [], [], [], [], [], [], [], []

ORA-00704: bootstrap process failure
ORA-00600: internal error code, arguments: [4000], [3], [], [], [], [], [], [], [], [], [], []
ORA-00704: bootstrap process failure
ORA-00600: internal error code, arguments: [4000], [3], [], [], [], [], [], [], [], [], [], []

*** 2013-10-27 21:55:56.922
USER (ospid: 15101): terminating the instance due to error 704
[root@lunar ~]#


dbkedDefDump(): Starting incident default dumps (flags=0x2, level=3, mask=0x0)
----- Current SQL Statement for this session (sql_id=6apq2rjyxmxpj) -----
select line#, sql_text from bootstrap$ where obj# != :1  ============注意这个是当前报错的语句，注意到是bootstrap$
```

摘要上面的信息，有用的如下：

```
rdba: 0x00400208 (1/520)  =============》说明root dba是file 1 block 520
Extent Header:: spare1: 0      spare2: 0      #extents: 1      #blocks: 7
Disk Lock:: Locked by xid:  0x0003.00c.00000180      ==============》被锁的事物的xid，转换成10进制是：3.12.384 ，这里跟报错信息匹配了，是回滚段3，事物槽是12，wrap是384
obj#: 59
  scn: 0x0000.0008d28a seq: 0x01 flg: 0x04 tail: 0xd28a1001

scn: 0x0000.0008d28a============》转化成10进制：  578186，说明是在bootstrap$在 这个scn被 lock了，实际上如果有备份，可以使用基于时间点的恢复，这个就是依据，如果有rman，可以进行测试
obj# 59是 bootstrap$表
```

uba的组成：

```
uba=Address Of Last Undo Block Used.Sequence.Last Entry in UNDO.Record.Map
```

例如：uba=0x0080006d.040d.1c（本例中不是这个数字）

xid的组成：

```
xid=Undo Segment Number.Transaction Table Slot Number. Wrap
xid=0x0003.00c.00000180
```

转换成10进制是：3.12.384 ，这里跟报错信息匹配了，是回滚段3，事物槽是12，wrap是384

```
[oracle@lunar bbed]$ bbed parfile=bbed.par

BBED: Release 2.0.0.0.0 - Limited Production on Sun Oct 27 23:17:33 2013

Copyright (c) 1982, 2011, Oracle and/or its affiliates.  All rights reserved.

************** !!! For Oracle Internal Use only !!! ***************

BBED> set file 1 block 520
        FILE#           1
        BLOCK#          520

BBED>
BBED> p ktech
struct ktech, 72 bytes                  @20
   ub4 spare1_ktech                     @20       0x00000000
   sword tsn_ktech                      @24       0
   ub4 lastmap_ktech                    @28       0x00000000
   ub4 mapcount_ktech                   @32       0x00000000
   ub4 extents_ktech                    @36       0x00000001
   ub4 blocks_ktech                     @40       0x00000007
   ub2 mapend_ktech                     @44       0x1020
   struct hwmark_ktech, 32 bytes        @48
      ub4 extno_ktehw                   @48       0x00000000
      ub4 blkno_ktehw                   @52       0x00000003
      ub4 extsize_ktehw                 @56       0x00000007
      ub4 blkaddr_ktehw                 @60       0x0040020c
      ub4 mapblk_ktehw                  @64       0x00000000
      ub4 offset_ktehw                  @68       0x00000000
      ub4 flblks_ktehw                  @72       0x00000001
      ub4 blkcnt_ktehw                  @76       0x00000003
   struct locker_ktech, 8 bytes         @80
      ub2 kxidusn                       @80       0x0003        看到这里就是我们的xid
      ub2 kxidslt                       @82       0x000c
      ub4 kxidsqn                       @84       0x00000180
   ub4 flag_ktech                       @88       0x00000001 (NONE)

BBED>
```

How to identify which rollback segment (Undo Segment) is corrupted?

Usually all the above errors will dump a trace file under udump directory. In most of the trace files you will see the following lines.

UNDO BLK:
xid: 0x0008.01c.00021ef1 seq: 0x4ad4 cnt: 0x1 irb: 0x1 icl: 0x0 flg: 0x0000

xid: 0x0008.01c.00021ef1 =Undo segment no + Slot no + Sequence no
Undo segment no =0x0008 =8

In the above example the undo segment number 8 is bad.

You can get the name of the undo segment from dba_rollback_segs view:
select segment_name from dba_rollback_segs where segment_id=8;

5. How to find if there are any active transactions in the undo segments?

If the undo segment number and name are known from the trace file then you can use the following command to dump the undo segment header.

alter system dump undo header "<undo segment name>";

eg: alter system dump undo header "_SYSSMU1$";

If the undo segment is unknown then set the following event in the init.ora file and bounce the instance. This event will dump all the undo segment headers.

event="10015 trace name context forever, level 10"

The trace file will be created under udump directory in both the cases.

In the trace file search for the Keyword "TRN TBL"

TRN TBL::

index state cflags wrap# uel scn dba parent-xid nub stmt_num

0x00 9 0x00 0x21eb1 0x0023 0x0000.d28c43e9 0x00000000 0x0000.000.00000000 0x00000000 0x00000000
.
.
0x14 10 0x90 0x21e6d 0x0000 0x0000.d28c4437 0x0080007e 0x0000.000.00000000 0x00000001 0x00000000
0x15 9 0x00 0x21e30 0x0020 0x0000.d28c4064 0x00800484 0x0000.000.00000000 0x00000003 0x00000000

Note the second column "state" specifies the status of the rollback segment .

If 10 then there is an active transaction.
If 9 then the transaction is committed.

摘要上面的信息，有用的如下：

```
rdba: 0x00400208 (1/520)  =============》说明root dba是file 1 block 520
Extent Header:: spare1: 0      spare2: 0      #extents: 1      #blocks: 7
Disk Lock:: Locked by xid:  0x0003.00c.00000180      ==============》被锁的事物的xid，转换成10进制是：3.12.384 ，这里跟报错信息匹配了，是回滚段3，事物槽是12，wrap是384
obj#: 59
  scn: 0x0000.0008d28a seq: 0x01 flg: 0x04 tail: 0xd28a1001

scn: 0x0000.0008d28a===========》转化成10进制：  578186，说明是在bootstrap$在 这个scn被 lock了，实际上如果有备份，可以使用基于时间点的恢复，这个就是依据，如果有rman，可以进行测试
obj# 59是 bootstrap$表

uba的组成：
uba=Address Of Last Undo Block Used.Sequence.Last Entry in UNDO.Record.Map
例如：uba=0x0080006d.040d.1c（本例中不是这个数字）

xid的组成：
xid=Undo Segment Number.Transaction Table Slot Number. Wrap
xid=0x0003.00c.00000180
转换成10进制是：3.12.384 ，这里跟报错信息匹配了，是回滚段3，事物槽是12，wrap是384
```

UNDO BLK:

xid: 0x0008.01c.00021ef1 seq: 0x4ad4 cnt: 0x1 irb: 0x1 icl: 0x0 flg: 0x0000

xid: 0x0008.01c.00021ef1 =Undo segment no + Slot no + Sequence no
Undo segment no =0x0008 =8

In the above example the undo segment number 8 is bad.

You can get the name of the undo segment from dba_rollback_segs view:
select segment_name from dba_rollback_segs where segment_id=8;

案例分析3

```
SQL> startup
ORACLE 例程已经启动。

Total System Global Area 1581916160 bytes
Fixed Size                  1336060 bytes
Variable Size             964693252 bytes
Database Buffers          603979776 bytes
Redo Buffers               11907072 bytes
数据库装载完毕。
ORA-00368: 重做日志块中的校验和错误
ORA-00353: 日志损坏接近块 12014 更改 9743799889 时间 12/05/2011 09:21:11
ORA-00312: 联机日志 3 线程 1: 'R:\ORADATA\HZYL\REDO03.LOG‘

SQL> col member for a35
SQL> select a.group#,a.status,b.member from v$log a,v$logfile b
  2 where a.group#=b.group#;

    GROUP# STATUS          MEMBER
---------- ---------------- -----------------------------------
         3 CURRENT          R:\ORADATA\HZYL\REDO03.LOG
         2 INACTIVE         R:\ORADATA\HZYL\REDO02.LOG
         1 INACTIVE         R:\ORADATA\HZYL\REDO01.LOG
```

SQL> recover database until cancel;

SQL>alter database open resetlogs

ORA-01547: 警告: RECOVER 成功但 OPEN RESETLOGS 将出现如下错误
ORA-01194: 文件 1 需要更多的恢复来保持一致性
ORA-01110: 数据文件 1: 'R:\ORADATA\HZYL\SYSTEM01.DBF‘

设置
_allow_resetlogs_corruption=TRUE
_allow_error_simulation=TRUE
recover database until cancel;
alter database open resetlogs;

导出导入重建数据库

案例分析4

数据库启动alert日志报如下错误
Tue Feb 14 09:34:11 2012

Errors in file
d:\oracle\product\10.2.0\admin\interlib\bdump\interlib_smon_2784.trc:

ORA-01595: error freeing extent (2) of rollback segment (3))

ORA-00607: Internal error occurred while making a change to a data block

ORA-00600: internal error code, arguments: [4194], [6], [30], [], [], [], [], []

Tue Feb 14 09:35:34 2012

Errors in file d:\oracle\product\10.2.0\admin\interlib\udump\interlib_ora_2824.trc:

ORA-00603: ORACLE server session terminated by fatal error

ORA-00600: internal error code, arguments: [4193], [2005], [2008], [], [], [], [], []

ORA-00600: internal error code, arguments: [4193], [2005], [2008], [], [], [], [], []

EVENT 38003

```
The objects affected are defined by kqlrtbso:
  hist_head$
  histgrm$
  i_hh_obj#_col#
  i_hh_obj#_intcol#
  i_obj#_intcol#
  i_h_obj#_col#
  c_obj#_intcol#

From 10.1 the following objects have been added:
  fixed_obj$
  tab_stats$
  ind_stats$
  i_fixed_obj$_obj#
  i_tab_stats$_obj#
  i_ind_stats$_obj#
  object_usage

From 11.1 the following object has been added:
  partobj$
```

案例分析 5
SYS.I_DEPENDENCY1 or SYS.I_DEPENDENCY2

```
1、11.2以后不能重现该问题，如果测试可以使用11.2以前的数据库版本
2、I_DEPENDENCY1
connect / as sysdba
startup upgrade
select * from bootstrap$ where sql_text like '%I_DEPENDENCY1%';
select obj# from obj$ where name='I_DEPENDENCY1';
update ind$ set flags=1024 where obj#=&obj_of_index;
REM ==>> ensure only 1 and 1 row is updated
commit
shutdown abort
startup restrict
alter index i_dependency1 rebuild;
analyze table dependency$ validate structure cascade;
alter system disable restricted session;

3、I_DEPENDENCY2
安装11.2数据库软件，然后在11.2环境下修复基表，再在原始数据库软件中打开：
 (1) 在11.2中操作按如下步骤操作：
       startup mount
       create pfile='/tmp/init$ORACLE_SID.ora' from spfile;
       startup upgrade pfile=/tmp/init$ORACLE_SID.ora
       select obj# from obj$ where name='I_DEPENDENCY2';
       update ind$ set flags=1024 where obj#=&obj_of_index;
       REM ==>> ensure only 1 and 1 row is updated
       commit
       shutdown immediate
 (2) 在原始数据库软件中操作如下步骤：
       startup upgrade
       alter index i_dependency2 rebuild;
       analyze table dependency$ validate structure cascade;
 (3) 建议重建数据库（expdp/impdp或者TTS等等）
```

ORA-600 [4194]

```
Diagnosis:

   This error is raised in kturdb which handles the adding of undo records
   by the application of redo.

   When we try to apply redo to an undo block (forward changes are made by
   the application of redo to a block), we check that the number of undo
   records in the undo block +1 matches the record number in the redo
   record.  Because we are adding a new undo record, we know that the record
   number in that undo block must be one greater than the maximum number in
   that block.

     So for UBA=0x08000592.00a0.0b

     0x08000592 is the dba of the undo block.
     0x00a0     is the seq# number that is in the block that THIS UNDO IS TO
                BE APPLIED TO.
     0x0b       is the number of undo records in the undo block.

     In the header this looks like:

       UNDO BLK::
     xid: 0x0004.00e.0000017f seq: 0x00a0 cnt: 0x0b ........

   Since we are adding a new undo record to our undo block, we would expect
   that the new record number is equal to the maximum record number in the
   undo block +1.  If this is not the case, we get ORA 600 [4194].

   This implies some kind of block corruption in either the redo or the
   undo block.  Look for other errors that would imply that a block is
   corrupted.
```

ORA-600 [4193]

Format: ORA-600 [4193] [a] [b]

VERSIONS:
  versions 6.0 to 10.1

DESCRIPTION:

  A mismatch has been detected between Redo records and Rollback (Undo)
  records.

  We are validating the Undo block sequence number in the undo block against
  the Redo block sequence number relating to the change being applied.

  This error is reported when this validation fails.

ARGUMENTS:
  Arg [a] Undo record seq number
  Arg [b] Redo record seq number

FUNCTIONALITY:
  KERNEL TRANSACTION UNDO

IMPACT:
  PROCESS FAILURE
  POSSIBLE ROLLBACK SEGMENT CORRUPTION

ORA-600 [4193] [a] [b] [ ] [ ] [ ]
Versions: 7.2.2 - 9.2.0 Source: ktuc.c

Meaning: seq# mismatch while adding an undo record to an undo block. This is done by the application of redo.

Argument Description:

a. (ktubhseq): undo record seq# - this is the seq# of the block that this undo record WILL BE APPLIED TO. This is from the Undo Block. It is NOT the seq# of the undo block itself.

b. (ktudbseq): redo RECORD seq# - this is the seq# number in the block that this redo WILL BE APPLIED TO. This is from the Redo Record.

ORA-600 [4097]

ORA-600 [4097]

The database crashed and system rollback segment has problems with transaction table.

Problem Explanation
==========

When accessing a rollback segment header to see if a transaction has been committed you observe the XID given is in the future of the transaction table.

Ie: the WRAP of the XID is higher than the current WRAP number on the RBS header.

Event 10015

数据库在mount状态下：

```
alter session set events '10015 trace name adjust_scn level n';
```

或

```
oradebug event 10015 trace name adjust_scn level n;
```

另外，在很多时候，可能我们还需要使用隐含参数*._minimum_giga_scn=n

注意，该参数在11.2.0.2.5和11.2.0.3.1中被取消掉。

以下版本不能使用了：

10.2.0.4.11

10.2.0.5.6

11.1.0.7.10

11.2.0.2.5

11.2.0.3.1

常用隐含参数

_allow_resetlogs_corruption

Actvie/Current redo log 坏块，IO错误，丢失等因为redo log异常导致数据库不能启动

主要是屏蔽redo前滚，强制打开数据库，可能导致redo中数据丢失，使用需要慎重

_offline_rollback_segments

强制把异常undo 设置为offline状态，主要处理回滚段存在但是异常情况

_corrupted_rollback_segments

强制直接标记回滚段不正常，主要用于处理回滚段损坏严重(比如丢失),比_offline_rollback_segments对数据库的破坏性更加严重

Undo段出现异常无法正常回滚回滚事务,导致数据库无法打开，例如含回滚事务的回滚段block出现坏块，回滚段和redo前滚信息不一致等

通过设置该参数屏蔽回滚段(该回滚段未提交事务自动提交)，将导致数据不一致，使用需要慎重

BBED的用途

bbed在数据库非open情况下修改block内容

主要用于一些场景恢复

1）缺少归档情况下数据文件online

2）系统基表事务未提交数据库不能open

3）部分坏块修复

4）……

DUL

dul是在数据库不正常open情况下，直接读取数据文件恢复数据内容

Dul主要用于以下场景恢复

1）数据库使用各种方法无法open

2）无删除表恢复

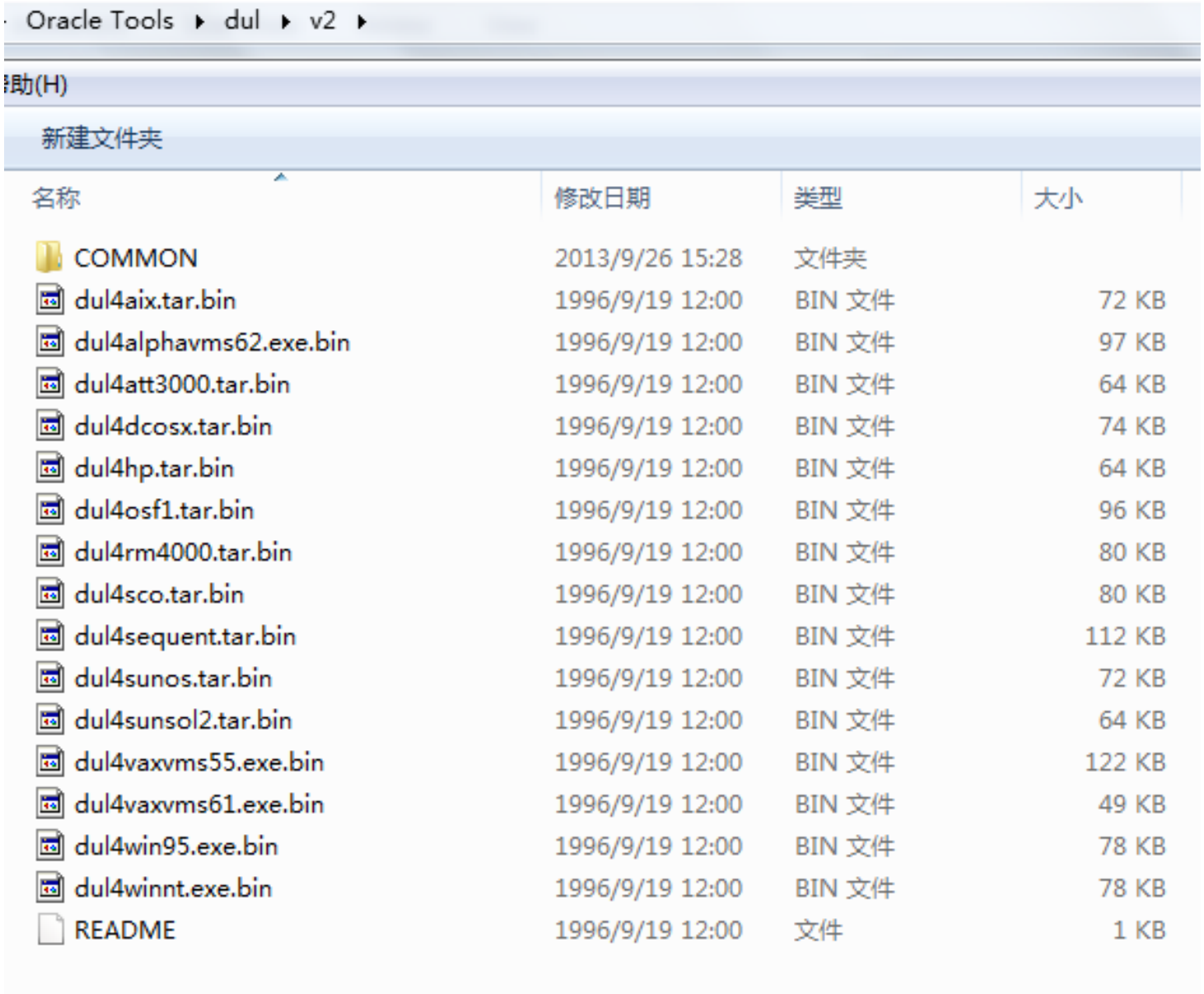
3）truncate table 删除

4）丢失system恢复

5）执行一些类似SQL的操作

6）抽取dmp文件

。。。。。


Oracle Tools ▸ dul ▸ v2 ▸
助(H)
新建文件夹
名称
修改日期
类型
大小
COMMON 2013/9/26 15:28 文件夹
dul4aix.tar.bin 1996/9/19 12:00 BIN 文件 72 KB
dul4alphavms62.exe.bin 1996/9/19 12:00 BIN 文件 97 KB
dul4att3000.tar.bin 1996/9/19 12:00 BIN 文件 64 KB
dul4dcosx.tar.bin 1996/9/19 12:00 BIN 文件 74 KB
dul4hp.tar.bin 1996/9/19 12:00 BIN 文件 64 KB
dul4osf1.tar.bin 1996/9/19 12:00 BIN 文件 96 KB
dul4rm4000.tar.bin 1996/9/19 12:00 BIN 文件 80 KB
dul4sco.tar.bin 1996/9/19 12:00 BIN 文件 80 KB
dul4sequent.tar.bin 1996/9/19 12:00 BIN 文件 112 KB
dul4sunos.tar.bin 1996/9/19 12:00 BIN 文件 72 KB
dul4sunsol2.tar.bin 1996/9/19 12:00 BIN 文件 64 KB
dul4vaxvms55.exe.bin 1996/9/19 12:00 BIN 文件 122 KB
dul4vaxvms61.exe.bin 1996/9/19 12:00 BIN 文件 49 KB
dul4win95.exe.bin 1996/9/19 12:00 BIN 文件 78 KB
dul4winnt.exe.bin 1996/9/19 12:00 BIN 文件 78 KB
README 1996/9/19 12:00 文件 1 KB

dul初试

Init.dul文件配置

osd_big_endian_flag=false

osd_dba_file_bits=10

osd_c_struct_alignment=32

osd_file_leader_size=1

osd_word_size = 32

dc_columns=2000000

dc_tables=10000

dc_objects=1000000

dc_users=400

dc_segments=100000

Buffer=10485760

control_file = control.txt

db_block_size=8192

export_mode=true

compatible=10

Control.txt文件配置

通过启动数据库到mount执行select ts#,rfile#,name from v$datafile获得

```
[oracle@lunar dul]$ more control.txt
0       1 /u01/oracle/oradata/lunar/system01.dbf
1       2 /u01/oracle/oradata/lunar/undotbs01.dbf
2       3 /u01/oracle/oradata/lunar/sysaux01.dbf
4       4 /u01/oracle/oradata/lunar/users01.dbf
6       5 /u01/oracle/oradata/lunar/datfttuser.dbf
```

Dul开始干活

```
[oracle@luanr dul]$ ./dul

Data UnLoader: 10.2.0.5.13 - Internal Only - on Sun Jun 10 06:39:47 2012
with 64-bit io functions

Copyright (c) 1994 2012 Bernard van Duijnen All rights reserved.

 Strictly Oracle Internal Use Only


Found db_id = 3426707456
Found db_name = lunar

加载数据字典
DUL> BOOTSTRAP;

Unload table
DUL> UNLOAD TABLE hr.test;
```

Q & A