

数据库非常规恢复

Lunar

www.lunar2013.com

<http://blog.csdn.net/lunar2000>

**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，
不得使用，否则后果自负！！**

常见错误

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 [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

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 4"
```

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

```
event = "10015 trace name context forever:  
        10046 trace name context forever, level 4"
```

Event 10046

oradebug setmypid

oradebug EVENT 10046 TRACE NAME CONTEXT FOREVER, LEVEL 12

alter session set db_file_multiblock_read_count=1;

oradebug TRACEFILE_NAME

alter database Open;

oradebug close_trace;

基本概念和名词解释

UBA: Undo block address

RBA: Redo block address

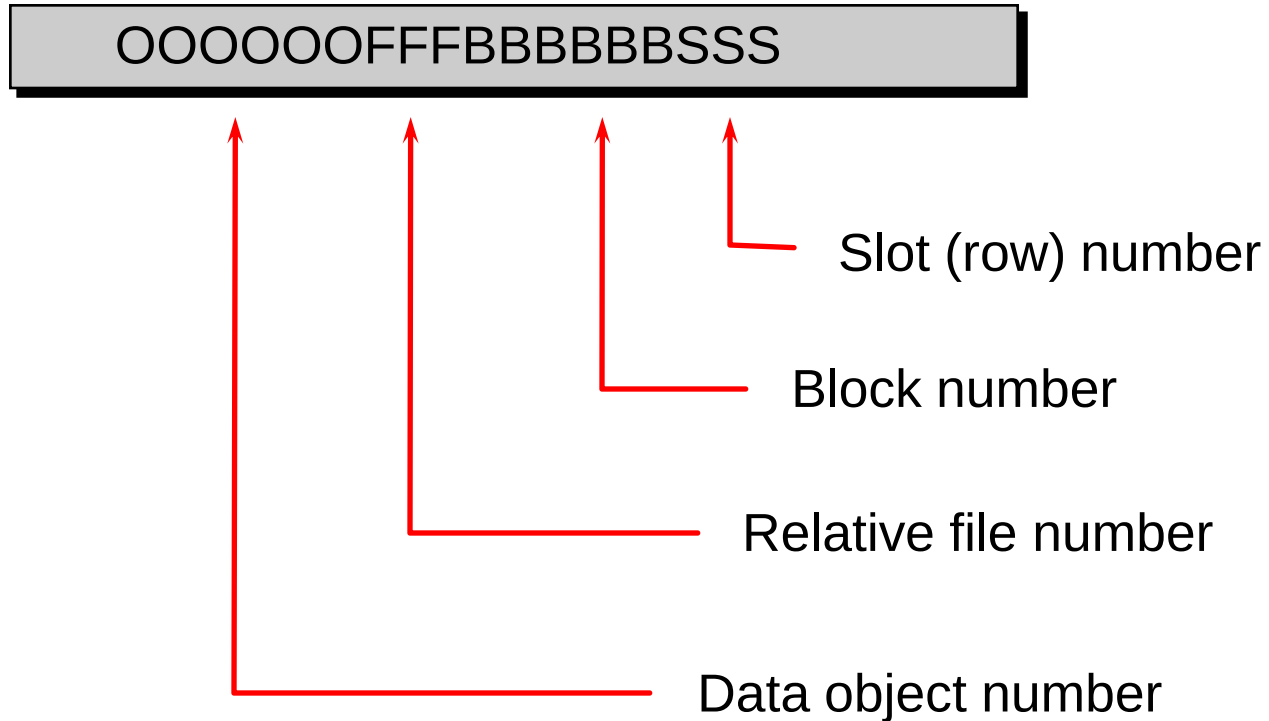
DbA: Data block address

Rdba: Root dba

Xid: Transaction ID

ITL: Interested Transaction List

Oracle ROWID Format



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

Converting DBA's (Database Addresses) to File # and Block # (Doc ID 113005.1)--- External

```
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.trc
SYS@orcl>
```

```
SYS@orcl>oradebug setmypid
Statement processed.
```

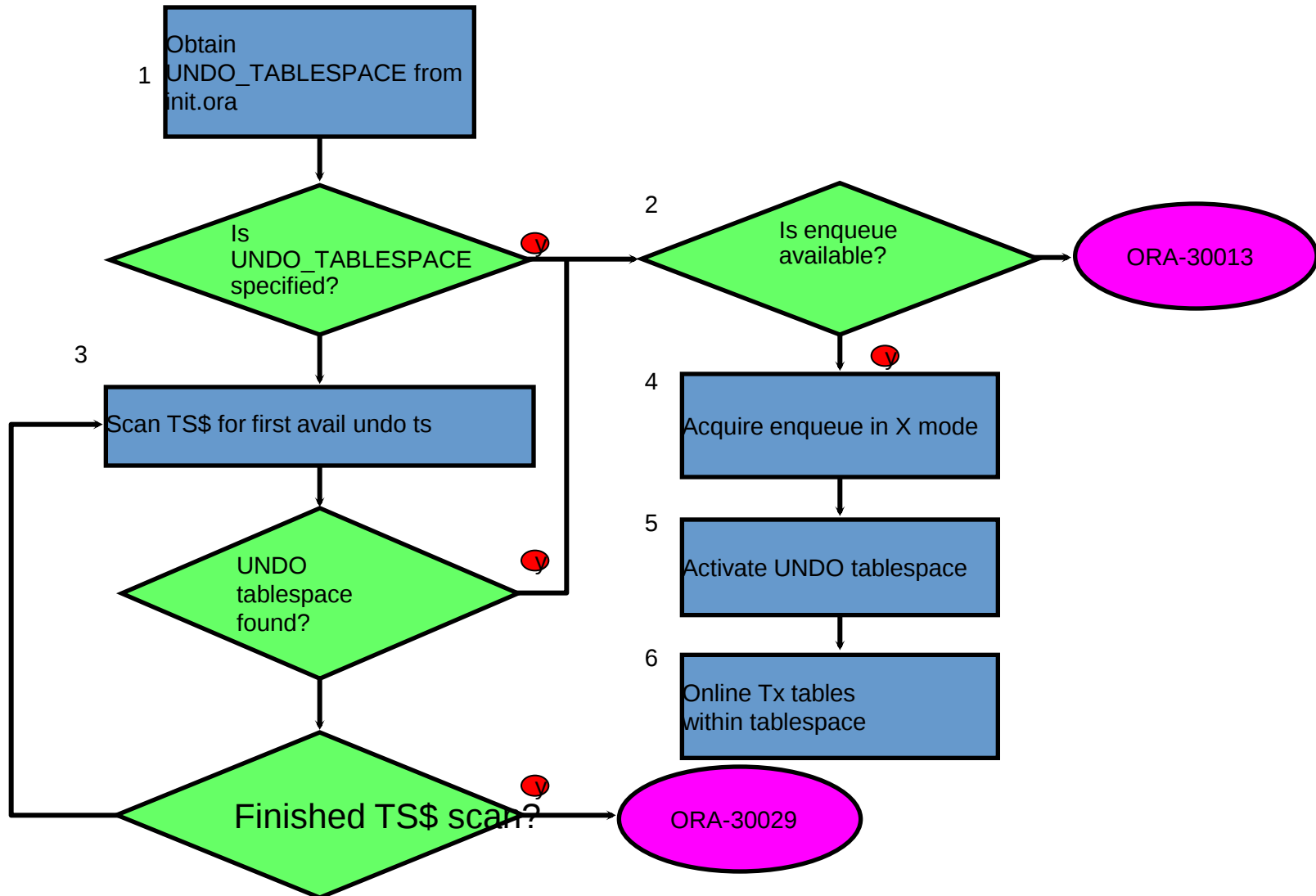
```
SYS@orcl>alter system dump undo block '_SYSSMU5$' xid 5 26 888;
```

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>
```

Instance Startup Using AUM



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#

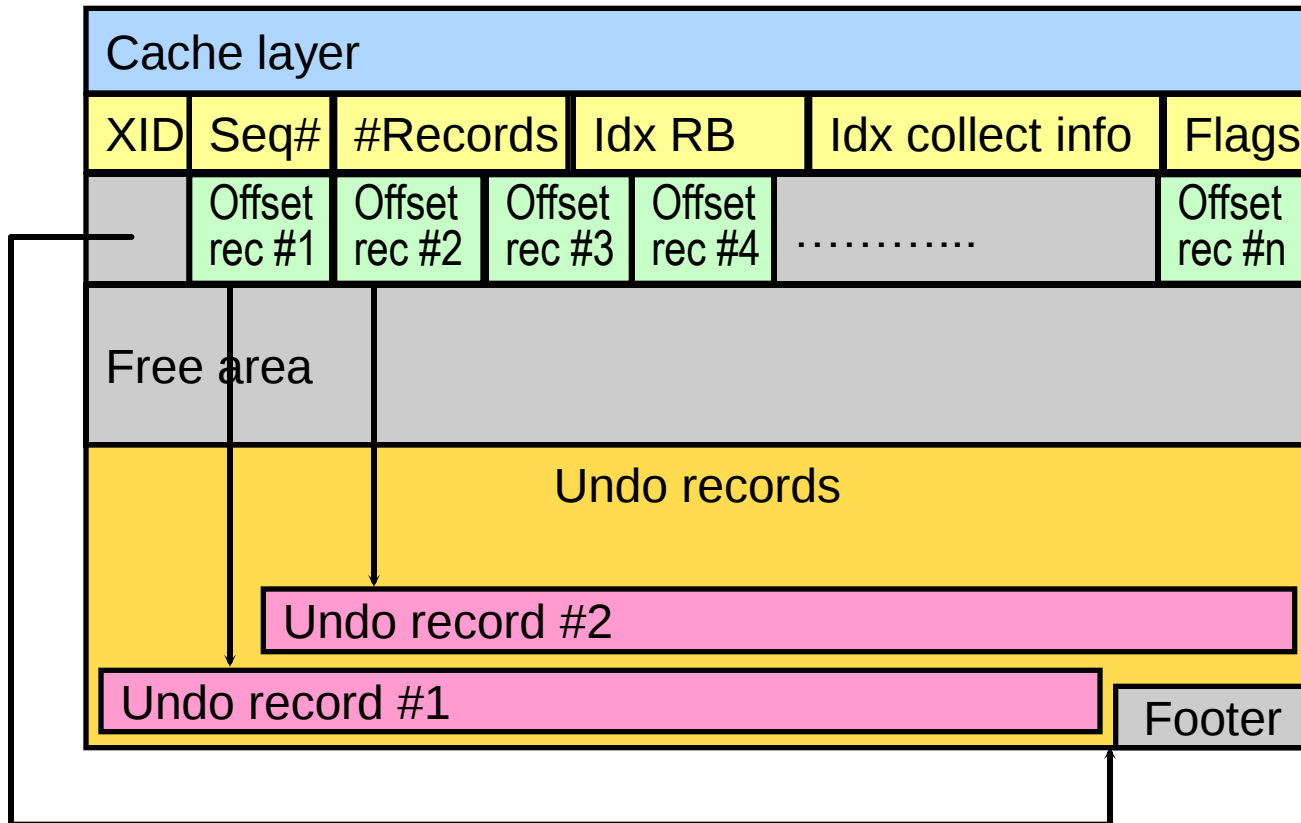
XID = usn# . slot# . wrap#

Undo Block Address

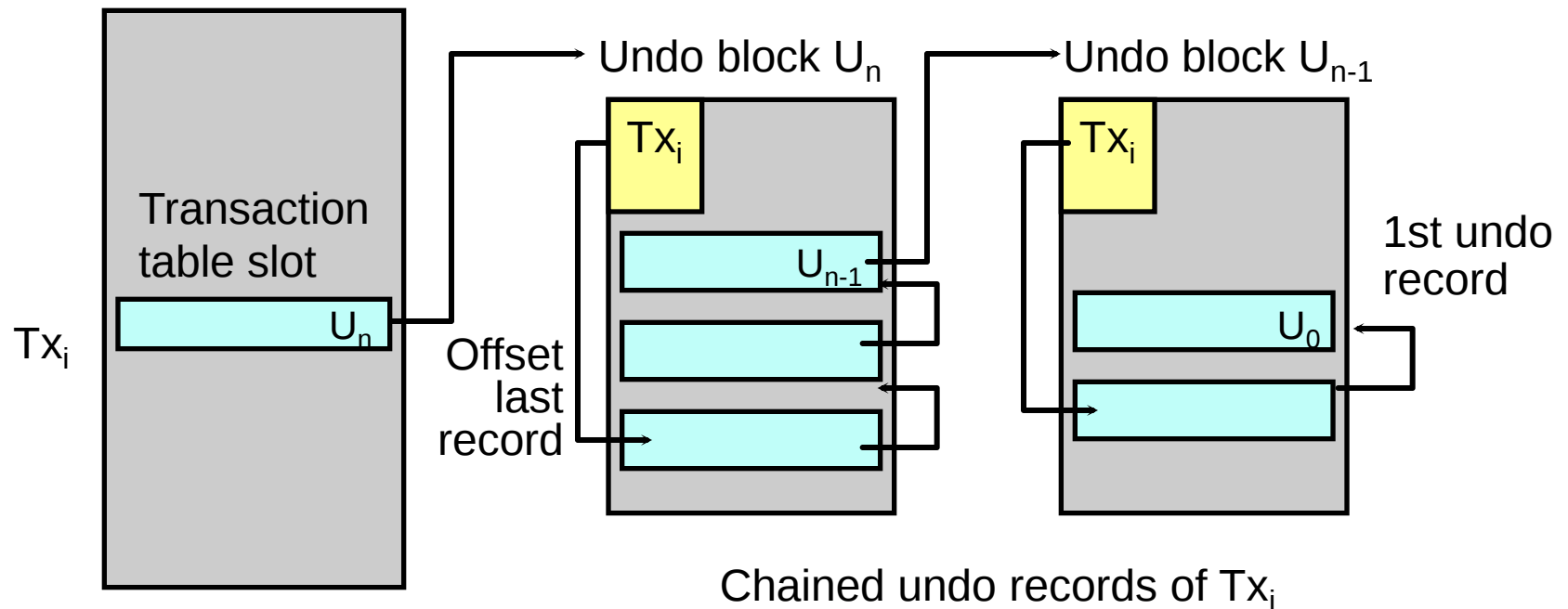
- The undo block address (UBA) uniquely identifies the undo block for a given transaction; it is found within the ITL of the data block.
- A UBA consists of:
 - Data block address (DBA) of the block
 - The sequence number of the block
 - The record number within the block

UBA = DBA. seq#. rec#

Undo Block



Transaction Undo Chain

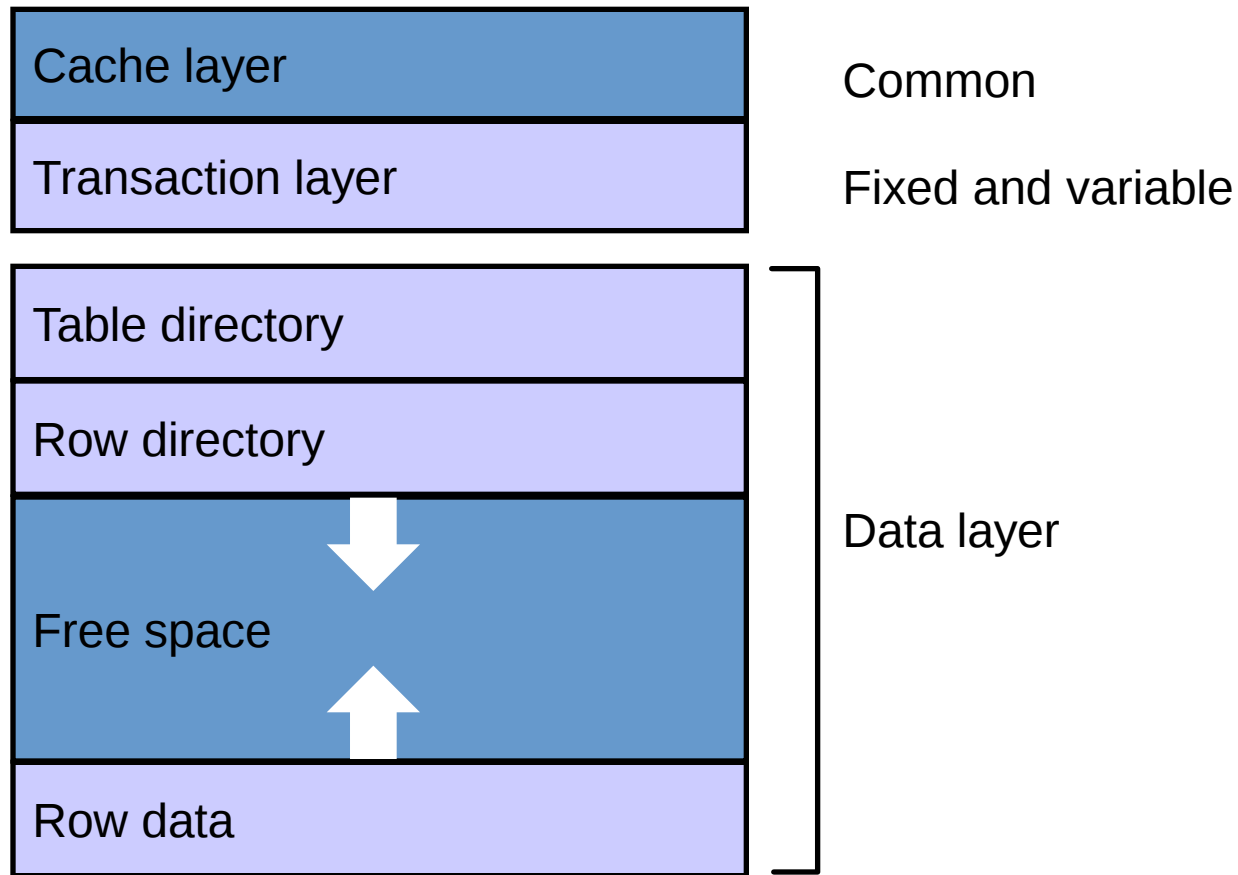


Example: Undo Record Chain Dump

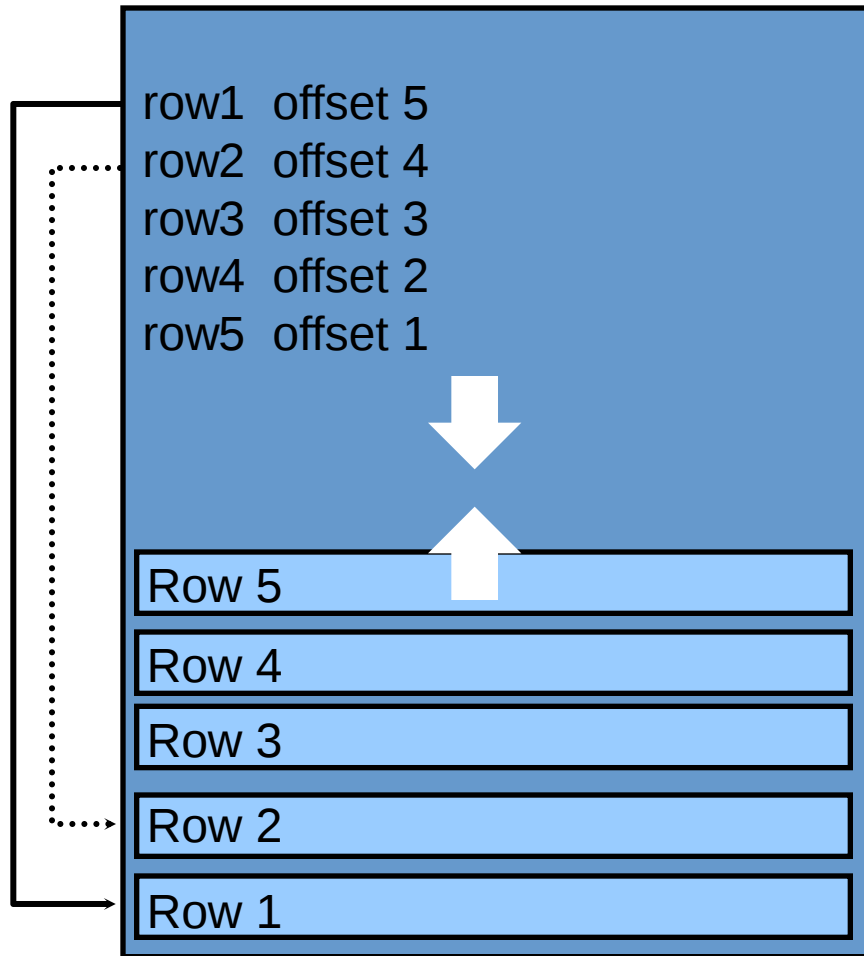
Undo
chain
TX table

```
UNDO BLK:
xid: 0x0002.00a.00000ced  seq: 0x266 cnt: 0x3  irb: 0x3  icl: 0x0  flg:0x000
...
*-----
*Rec #0x1 slt: 0x0a  objn: 11987(0x00002ed3) objd: 11987 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: 0x00c00510.0266.06 ctl max scn:0x0000.0010a135 prv tx scn:0x0000.0010a136
...
*Rec #0x2 slt: 0x0a  objn: 11987(0x00002ed3) objd: 11987 tblspc:4(0x00000004)
*      Layer: 11 (Row)  opc: 1  rci 0x01
Undo type: Regular undo  Last buffer split: No
Temp Object: No
Tablespace Undo: No
rdba: 0x00000000
...
*-----
*Rec #0x3 slt: 0x0a  objn: 11987(0x00002ed3) objd: 11987 tblspc:4(0x00000004)
*      Layer: 11 (Row)  opc: 1  rci 0x02
Undo type: Regular undo  Last buffer split: No
...
```

Data Block Header: Cache Layer



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.

Row Format: Column Data

- Column data is stored as a series of column length values and column data pairs.
- Extract from a logical block dump:

```
tab 0, row 0, @0x6f
tl: 6 fb: --H-FL-- lb: 0x0 cc: 1
col 0: [ 2] c1 02
tab 0, row 1, @0x6d6
tl: 226 fb: --H-FL-- lb: 0x0 cc: 2
col 0: [ 2] c1 03
col 1: [219] 6e 6f 4b 4c 4d ...
```

Flag byte(*fb*): H=header,
F=first, L=last (entire row)

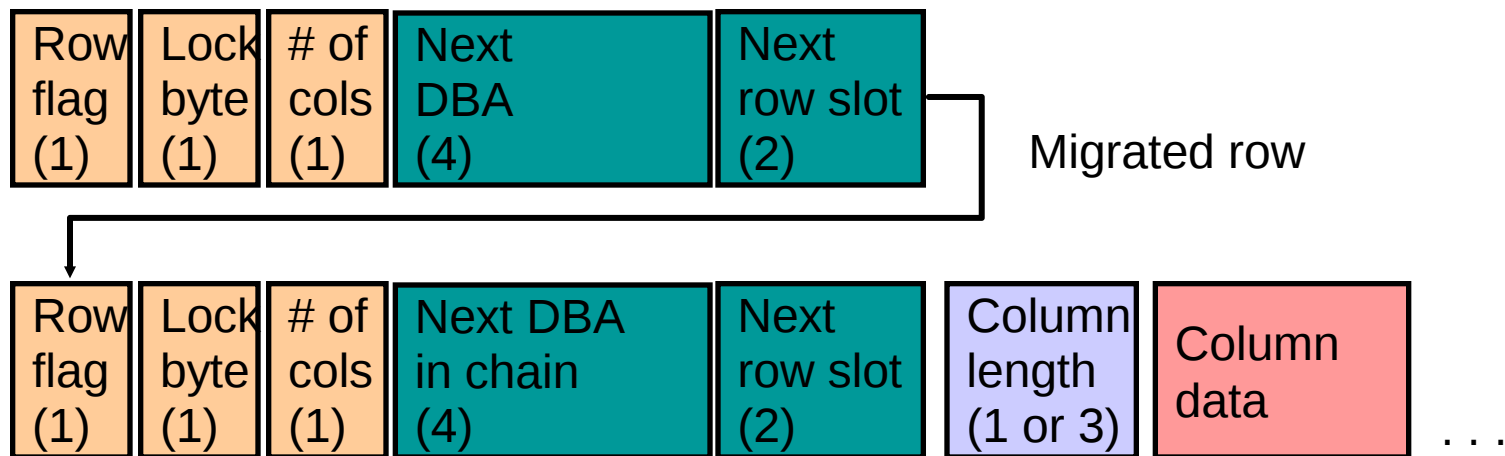
Column count (*cc*) = 1,
a NULL column in position 2

Column length is 2 bytes,
the data bytes are *c1 03*

Row Migration and Chaining

Row *migration*:

A row no longer fits in its original block.



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 (0x7fffffff)
          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	0x0080004de	0x0000.000.00000000	0x00000001	0x00000000	1384046414
0x08	9	0x00	0x0378	0x0018	0x0000.001ba19b	0x0080004e0	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	0x0080004df	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	0x0080004de	0x0000.000.00000000	0x00000003	0x00000000	1384045222
0x0d	9	0x00	0x0378	0xffff	0x0000.001ba1a6	0x0080004e0	0x0000.000.00000000	0x00000001	0x00000000	1384046414
0x0e	9	0x00	0x0377	0x002b	0x0000.001ba185	0x0080004df	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

Db-----》 uba---- 0x008004e0

```

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

UNDO_FILE#
-----
2

SYS@orcl>select dbms_utility.data_block_address_block(to_number('008004e0','xxxxxxxxxxxx')) as undo_block# from dual;

UNDO_BLOCK#
-----
1248

SYS@orcl>

```

oradebug setmypid

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

oradebug close_trace

oradebug tracefile_name

Undo segment 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,ubablkc,ubafilc,ubarec, start_scn from v\$transaction;

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

Undo block xid:

```
*****
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(0x000d72b) objid: 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

ORACLE

保留所有权利。

```

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-----
col 0: [ 5] 4c 55 4e 41 52
col 1: [ 2] c1 4e
col 2: [16] 38 35 39 31 38 37 32 35 37 34 38 38 30 44 37 41
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 6d 08 12 11 22 03
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

```

```

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>

```

```

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*

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 bin 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 site.
A diag patch needs to be requested to have 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/ssbbdpt.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/syslibl
ist` `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/ldf
lags` -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>
```

ORACLE

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 !!! *****

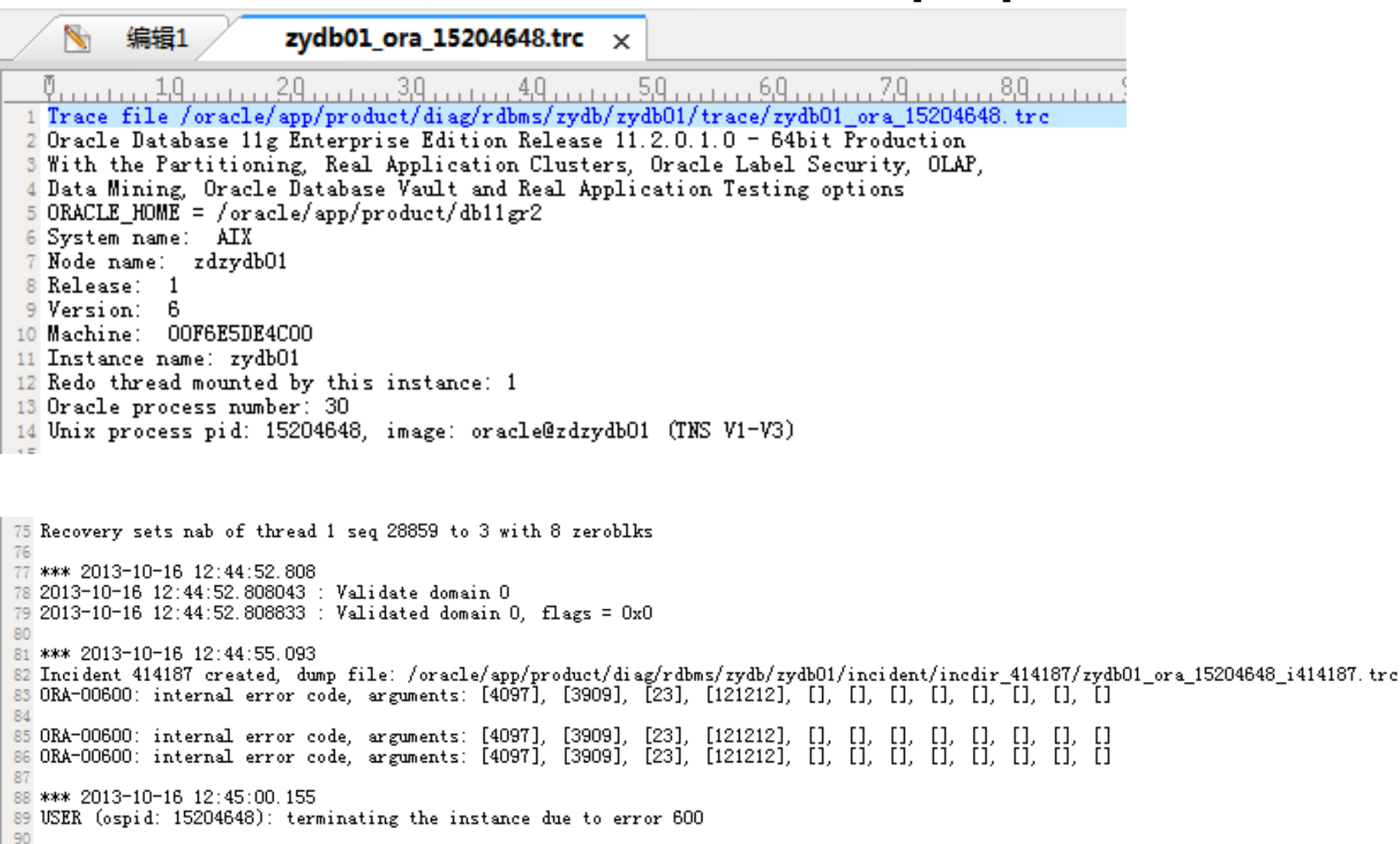
BBED> help
HELP [ <bbed command> | ALL ]

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: ksfedpn: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



```
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: OOF6E5DE4C00
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 rba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 hwmk_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**

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 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 2、I_DEPENDENCY1
3 connect / as sysdba
4 startup upgrade
5 select * from bootstrap$ where sql_text like '%I_DEPENDENCY1%';
6 select obj# from obj$ where name='I_DEPENDENCY1';
7 update ind$ set flags=1024 where obj#=@obj_of_index;
8 REM ==>> ensure only 1 and 1 row is updated
9 commit
10 shutdown abort
11 startup restrict
12 alter index i_dependency1 rebuild;
13 analyze table dependency$ validate structure cascade;
14 alter system disable restricted session;
15
16 3、I_DEPENDENCY2
17 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：
18 (1) 在11.2中操作按如下步骤操作：
19     startup mount
20     create pfile='/tmp/init$ORACLE_SID.ora' from spfile;
21     startup upgrade pfile=/tmp/init$ORACLE_SID.ora
22     select obj# from obj$ where name='I_DEPENDENCY2';
23     update ind$ set flags=1024 where obj#=@obj_of_index;
24     REM ==>> ensure only 1 and 1 row is updated
25     commit
26     shutdown immediate
27 (2) 在原始数据库软件中操作如下步骤：
28     startup upgrade
29     alter index i_dependency2 rebuild;
30     analyze table dependency$ validate structure cascade;
31 (3) 建议重建数据库（expdp/impdp或者TTS等等）
32
```

ORA-600 [4194]

ORA-600 [4194] [a] [b]

Versions: 6.0 - 9.2

Source: ktuc.c

=====

Meaning:

Undo record number mismatch while adding an undo record to an undo block. This is done by the application of redo.

Argument Description:

- a. (ktubhcnt): undo record count - This is the maximum number of undo records that have ever existed within this Undo Block. In other words, it is the High Water Mark for undo records in that undo block. This is from the Undo Block.
- b. (ktudbrec): redo record number - This is the record number for the new undo record that is to be added to the undo block. It should be one greater than the maximum in the undo block currently. This is from the Redo Record.

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`

Active/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)**

Diagnostic Events

- Event **10231**:
“Skip corrupted blocks during a full table scan”
- Event **10232**:
“Dump corrupted blocks in a trace file”
- Event **10233**:
“Skip corrupted data or index blocks on index range scans”

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**

帮助(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;
```

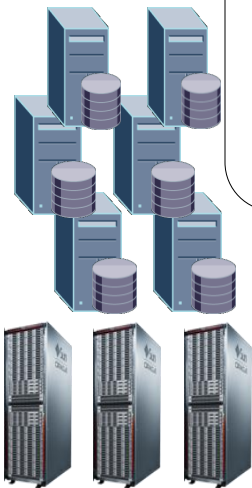
Architected for Protection of Critical Data

Oracle Database Backup Logging Recovery Appliance

Client Databases

Database Delta Push

- DBs access and send only changed data
- Minimal impact on production servers
- Real-time redo ship for near-zero data loss



Cloud Scale

- Scales to 1000s of Clients
- Petabytes of Data
- No expensive backup agents

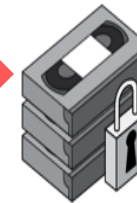
Delta Store

- Validated, compressed database change data
- Fast restores to any point-in-time using deltas
- Built on Exadata scaling & resilience



Autonomous Archive

- Copy to tape: **no** production server load
- Tapes utilized all day
- Restore directly from tape



Tape Library

Complete Data Protection Solution

- Replication for DR
- Unified management: database to tape
- Per-database provisioning

Q & A