**[Current online Redo](http://blog.csdn.net/tianlesoftware/article/details/6261475)****[和](http://blog.csdn.net/tianlesoftware/article/details/6261475)****[Undo](http://blog.csdn.net/tianlesoftware/article/details/6261475)****[损坏的处理方法](http://blog.csdn.net/tianlesoftware/article/details/6261475)**

标签： [system](http://www.csdn.net/tag/system /t _blank)[database](http://www.csdn.net/tag/database /t _blank)[oracle](http://www.csdn.net/tag/oracle /t _blank)[sql](http://www.csdn.net/tag/sql /t _blank)[数据库](http://www.csdn.net/tag/%e6%95%b0%e6%8d%ae%e5%ba%93 /t _blank)[blog](http://www.csdn.net/tag/blog /t _blank)

2011-03-19 17:47 10006人阅读 [评论](http://blog.csdn.net/tianlesoftware/article/details/6261475 /l comments)(4) [收藏](http://) [举报](http://blog.csdn.net/tianlesoftware/article/details/6261475 /l report /o %BE%D9%B1%A8)

本文章已收录于：

IMG_256

IMG_257

分类：

Oracle Troubleshooting*（193）*

IMG_258

IMG_259

[作者同类文章](http://blog.csdn.net/tianlesoftware/article/category/679087 /t _blank)X

IMG_260

版权声明：本文为博主原创文章，未经博主允许不得转载。

**[Oracle](http://lib.csdn.net/base/oracle /o Oracle%D6%AA%CA%B6%BF%E2/t_blank)** 不同故障的恢复方案

Oracle undo 表空间管理

<http://blog.csdn.net/tianlesoftware/archive/2010/07/11/5689558.aspx>

    上面的2篇Blog 讲到了Oracle 一些故障下的恢复方法，但是还有2点，之前没有做详述。即current online redo 损坏的情况和undo的损坏情况。 之所以把他们放一起讲，是因为他们之间有联系。

       在做操作之前，最好把控制文件，redo log 和数据文件做一个冷备份，以防万一.

一.  UNDO 的补充信息

所以这里补充说明一下。 在补充之前先讲一下undo segment的概念。

    在blog：Oracle undo 管理

<http://blog.csdn.net/tianlesoftware/archive/2009/11/30/4901666.aspx>

中提到了Oracle 的undo 有两种管理方式，通过参数undo\_management来设置auto 和 manual。

1.1 当undo\_management被设置成MENUAL时使用系统回滚段, 即将undo records 记录到SYSTEM 表空间下的SYSTEM段。

SQL> col segment\_name format a10

SQL> select segment\_name,tablespace\_name,bytes,next\_extent  from dba\_segments where segment\_type='ROLLBACK';

SEGMENT\_NA TABLESPACE\_NAME       BYTES NEXT\_EXTENT

---------- ------------------------------ ---------- -----------

SYSTEM     SYSTEM                 393216     1048576

    通过上面的这条语句，我们查到了这个用于rollback 的system segment 存在与system 表空间。 默认情况下，只有一个segment，并且它还比较小，所以，如果使用system 段来存储undo records。肯定会影响**[数据库](http://lib.csdn.net/base/mysql /o MySQL%D6%AA%CA%B6%BF%E2/t_blank)的性能。 所以Oracle 是建议使用Undo tablespace 来管理undo records。**

1.2  当undo\_management设置成AUTO时使用UNDO tablespace来管理回滚段。 这个时候，我们将有多个undo segment，并且这些segment 是存放在UNDO 表空间里的。 这样对DB的性能就会提高。

SQL> select segment\_name,tablespace\_name,bytes,next\_extent  from dba\_segments where segment\_type='TYPE2 UNDO';

SEGMENT\_NAME    TABLESPACE\_NAME    BYTES NEXT\_EXTENT

-------------------- -------------------- ---------- -----------

\_SYSSMU1$            UNDOTBS1                1179648       65536

\_SYSSMU2$            UNDOTBS1                1179648       65536

\_SYSSMU3$            UNDOTBS1                2228224       65536

\_SYSSMU4$            UNDOTBS1                1179648       65536

\_SYSSMU5$            UNDOTBS1                 262144       65536

\_SYSSMU6$            UNDOTBS1                1179648       65536

\_SYSSMU7$            UNDOTBS1                1179648       65536

\_SYSSMU8$            UNDOTBS1                1179648       65536

\_SYSSMU9$            UNDOTBS1                1179648       65536

\_SYSSMU10$           UNDOTBS1                1179648       65536

通过以上SQL的查询结果，我们可以看出，有10个undo segment来存放undo records。

    以上我们是通过dba\_segment 表查看的结果。 也可以通过v$rollstat和v$rollname 两个视图来查看信息。 这2个视图会显示所有rollback 段的信息。 包括system段和undo段。

SQL> col name format a15

SQL> select s.usn,n.name,s.extents,s.hwmsize,s.status from v$rollstat s, v$rollname n where s.usn=n.usn;

       USN NAME               EXTENTS    HWMSIZE STATUS

---------- --------------- ---------- ---------- ---------------

         0 SYSTEM                   6     385024 ONLINE

         1 \_SYSSMU1$                3    7659520 ONLINE

         2 \_SYSSMU2$                3    9691136 ONLINE

         3 \_SYSSMU3$                4    7462912 ONLINE

         4 \_SYSSMU4$                3   76668928 ONLINE

         5 \_SYSSMU5$                4    8511488 ONLINE

         6 \_SYSSMU6$                3    7462912 ONLINE

         7 \_SYSSMU7$                3   33480704 ONLINE

         8 \_SYSSMU8$                3    8577024 ONLINE

         9 \_SYSSMU9$                3    7462912 ONLINE

        10 \_SYSSMU10$               3   13754368 ONLINE

11 rows selected.

二.  UNDO 损坏的情况

    了解了第一部分的补充知识后，我们在来看一下undo 损坏的情况。 出现这种情况，大多数是因为异常宕机，在启动的时候报的错误。DB 不能启动。

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

    对于Undo 损坏的情况，能用备份恢复最好，如果不能，就只能通过一些特殊的方法来恢复。

2.1 方法一,使用system segment。

在Blog：

    Oracle undo 表空间管理

<http://blog.csdn.net/tianlesoftware/archive/2010/07/11/5689558.aspx>

提到了一种方法，就是使用SYSTEM 的回滚段, 步骤如下：

（1）用spfile 创建pfile，然后修改参数：

#\*.undo\_tablespace='UNDOTBS1'

#\*.undo\_management='AUTO'

#\*.undo\_tablespace

#\*.undo\_retention

undo\_management='MANUAL'

rollback\_segments='SYSTEM'

（2）用修改之后的pfile，重启DB

SQL> STARTUP MOUNT pfile='F:/initorcl.ora' ;

（3）删除原来的表空间，创建新的UNDO 表空间

SQL> drop tablespace undotbs;

SQL> create undo tablespace undotbs1 datafile '/u01/oradata/undotbs1.dbf' size 10M;

（4）关闭数据库，修改pfile参数，然后用新的pfile创建spfile，在正常启动数据库。

\*.undo\_tablespace='UNDOTBS1'

\*.undo\_management='AUTO'

#undo\_management='MANUAL'

#rollback\_segments='SYSTEM'

2.2. 方法二：跳过损坏的segment

    在方法一里面，我们使用了system segment。 通过第一部分我们了解到，undo segment 有多个，我们可以通过alert log 来查看正在使用的是哪些segment，这些段有可能损坏了。 我们只需要把这些损坏的segment 跳过，先正常启动DB，在创建新的UNDO 表空间，在切换一下。

（1）修改pfile，添加参数：

\*.\_corrupted\_rollback\_segments='\_SYSSMU11$','\_SYSSMU12$','\_SYSSMU13$'

这些字段的值，我们通过alert log 查看。 也可以通过如下命令查看：

#strings system01.dbf | grep \_SYSSMU | cut -d $ -f 1 | sort -u

（2）用修改之后的pfile启动DB

    因为跳过了哪些损坏的segment，所以DB 可以正常启动。

（3）创建新的UNDO 表空间，并切换过来

SQL> create undo tablespace undotbs1 datafile '/u01/oradata/undotbs1.dbf' size 10M;

SQL> alter system set undo\_tablespace=undotbs1;

SQL> drop tablespace undotbs;

（4）修改pfile，创建spfile，并正常启动

删除：

\*.\_corrupted\_rollback\_segments='\_SYSSMU11$','\_SYSSMU12$','\_SYSSMU13$'

以上就是UNDO 出现故障的2种处理方法。

三.  Current online Redo 损毁的处理

    其实在之前的不同故障处理的那篇blog里已经提到了这点。 但这种情况是一种特殊的情况。 所以还是单独拿出来说明一下。

current online log 损坏有两种恢复方法：

（1）如果有归档和备份，可以用不完全恢复。

SQL>startup mount;

SQL>recover database until cancel; 先选择auto，尽量恢复可以利用的归档日志，然后重新执行：

SQL>recover database until cancel; 这次输入cancel，完成不完全恢复,

用resetlogs打开数据：

SQL>alter database open resetlogs； 打开数据库

（2）强制恢复， 这种方法可能会导致数据不一致

sql>startup mount;

sql>alter system set "\_allow\_resetlogs\_corruption"=true scope=spfile;

sql>recover database until cancel;

sql>alter database open resetlogs;

这里主要看2点：

（1）使用了\_allow\_resetlogs\_corruption 参数

（2）这种情况下，可能会报ORA-600[2662]（SCN有关）和 ORA-600[4000]（回滚段有关）的错误。

    使用\_allow\_resetlogs\_corruption参数，强制的打开数据库，可能会导致逻辑的坏块，从而影响数据字典。 所以，即使使用该参数正常打开后，也需要做的一个操作：逻辑导出数据。 重建实例，导入实例。 消除逻辑坏块的可能性。

    如果使用\_allow\_resetlogs\_corruption参数启动报了undo segment的错误而无法启动，处理方法参考第二节中undo 的处理情况。 只要DB 能正常open，就导出数据，重建实例，在导入。