**How to Change the Existing Undo Tablespace to a New Undo Tablespace (Doc ID 431652.1)**

1) Create Undo tablespace UNDOTBS01

SQL> create undo tablespace UNDOTBS01 DATAFILE '+DATAGROUP' SIZE 4g;

2) Create Undo tablespace UNDOTBS02

SQL> create undo tablespace UNDOTBS02 DATAFILE '+DATAGROUP' SIZE 2G;

3) Assign the Newly created UNDOTBS01 tablespace to node1

SQL> ALTER SYSTEM SET UNDO\_TABLESPACE=UNDOTBS01 SCOPE=BOTH SID='test1';

3) Assign the Newly created UNDOTBS02 tablespace to node2

SQL> ALTER SYSTEM SET UNDO\_TABLESPACE=UNDOTBS02 SCOPE=BOTH SID='test2';

SQL>

SQL> show parameter undo

NAME TYPE VALUE  
 ———————————— –  
 undo\_management string AUTO  
 undo\_retention integer 1800  
 undo\_tablespace string UNDOTBS02

SQL>

or

select name,type,value from gv$parameters where name like 'undo%';

To drop tablespace UNDOTBS1(old Undo) see its segment status

等待原UNDO表空间  
段的状态是offline

区的状态是expired

If the old segments are online, then they must be taken offline:

SQL>alter rollback segment "\_SYSSMU3$" offline;  
 SQL>alter rollback segment "\_SYSSMU2$" offline;

SQL>  select owner, segment\_name, tablespace\_name, status from dba\_rollback\_segs where tablespace\_name='UNDOTBS1' 

Run this query to find the actual status of undotbs2. u cant drop undo before creating one and pointing the database to the new one. to drop status should not return as ACTIVE.

 select owner,segment\_name,tablespace\_name,status from dba\_undo\_extents where  tablespace\_name='UNDOTBS1'

Your mentioned query result is.

|  |  |
| --- | --- |
|  | GB STATUS |

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

.135681152 UNEXPIRED

.030944824 EXPIRED

 alter tablespace undotbs1 offline  
  drop tablespace undotbs1 including contents and datafiles;

====================================================

## Undo Lifetime

Undo information has different states during it’s lifecycle, depending on running transactions and retention settings.

There are three states or types of extents in the Undo tablespace: ACTIVE, EXPIRED and UNEXPIRED. Oracle is still using Rollback segments, but with Automatic Undo Management these are completely controlled by Oracle.

### ACTIVE

Active undo extents are used by transactions and will always be active, because they are needed for Rollback. The UNDO\_RETENTION setting is not used here, because one can not say something like: ‘after 900 seconds you are not allowed to rollback anymore…’

You will get ‘**ORA-30036 unable to extend segment in Undo tablespace**‘ errors when no more space is left to store ACTIVE Undo. This will automatically rollback the transaction causing it. The**NOSPACEERRCNT** column in V$UNDOSTAT is a good indication how many times this has occurred.

### EXPIRED

Expired extents are not used by transactions, the data in these extends is committed and the UNDO\_RETENTION time has passed, so it is not needed for Read Consistency.

### UNEXPIRED

Unexpired extents are non-active extents that still honour UNDO\_RETENTION. The transactions belonging to these undo extents are committed, but the retention time has not passed: You still want/need these for Read Consistency!

When the Undo mechanism requires more extents for ACTIVE extents, it is allowed to steal UNEXPIRED extents when there are no EXPIRED extents left for reuse and it can not allocate more free extents (autoextend maxsize reached or fixed tablespace size). One can check the steal-count in**UNXPSTEALCNT** in V$UNDOSTAT.

You will get ‘**ORA-01555 snapshot too old**‘ errors if no Read Consistency information for a query is available. The **SSOLDERRCNT** in V$UNDOSTAT will show a count of these errors.