
Managing Rollback Segments

Elizabeth Boss

Boss Consulting Services Inc.

Abstract

This paper will discuss the management of rollback segments in an Oracle database. Rollback segment creation, modification, and deletion will be presented. Effectively setting the parameters for rollback segments will also be discussed.

Creating Rollback Segments

Rollback segments are used to hold the old images of rows that are modified or deleted; thus allowing an undo of a transaction. Rollback segments are the only type of segment that are created and managed by the database administrator. If a database has tablespaces other than the system tablespace, a minimum of two rollback segments is required. The basic syntax for the CREATE ROLLBACK segment command is:

```
CREATE [PUBLIC] ROLLBACK SEGMENT segment_name
STORAGE (INITIAL n[k|m]
         NEXT    n[k|m]
         MINEXTENTS n
         MAXEXTENTS n
         OPTIMAL n[k|m])
```

A rollback segment is private by default, which means that only the current instance has access to the rollback segment. A private rollback segment is available for a single instance. Private rollback segments are specified in the initialization parameter ROLLBACK_SEGMENTS and are available to all users. Oracle determines the number of public rollback segments to access based upon the initialization parameters TRANSACTIONS and TRANSACTIONS_PER_ROLLBACK_SEGMENT. All users have access to public rollback segments.

To create a rollback segment issue the CREATE ROLLBACK segment command.

```
SQL> CREATE ROLLBACK SEGMENT rbs1
        STORAGE (INITIAL 20K
                NEXT 20k
                MINEXTENTS 20)
        TABLESPACE rollback_data;

Rollback segment created.
```

Optimal Parameter

The Optimal parameter is used to specify the size to which a rollback segment will shrink after automatically extending itself. If the optimal parameter is not specified when a rollback segment is created, any additional extents dynamically allocated for the rollback segment will remain allocated. By setting the optimal parameter the database administrator can control the size of rollback segments. The optimal parameter is set in a number of bytes, kilobytes, or megabytes and is typically set based on the extent size and minimum number of extents. The minimum size for the optimal parameter is $\text{minextents} * \text{extent_size}$ (plus some room for overhead information).

Altering a Rollback Segment

The ALTER ROLLBACK SEGMENT command is used to modify parameters for existing rollback segments. The full syntax for the ALTER ROLLBACK SEGMENT command includes:

```
ALTER ROLLBACK SEGMENT segment_name
[STORAGE(NEXT n
          MAXEXTENTS n)]
[ONLINE|OFFLINE]
[SHRINK TO n]
```

The *segment_name* is the name of the rollback segment to be modified. The *storage* clause allows modification of the next extent size or the maximum number of extents (*maxextents*). The rollback segment may be brought *online* or taken *offline* dynamically. The shrink option allows the rollback segment to be reduced to the size specified or the value set in the OPTIMAL parameter. If no size is specified the value specified by the OPTIMAL parameter is used.

When a rollback segment is created it is initially offline, and not available for use. To make a rollback segment available it must be brought online by issuing an ALTER ROLLBACK SEGMENT command.

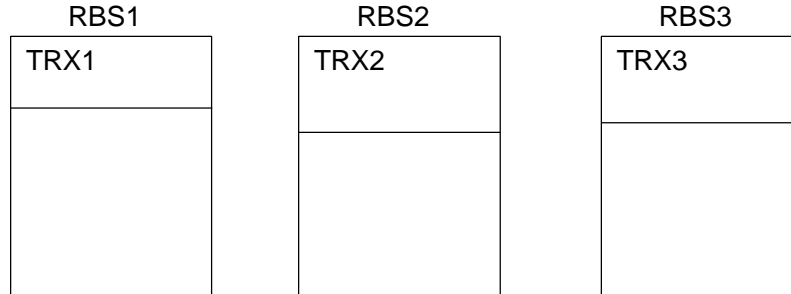
```
SQL> ALTER ROLLBACK SEGMENT
      rbs1 ONLINE;

Rollback segment altered.
```

The rollback segment will remain online until the next time the instance is shut down. To make a rollback segment continuously available it must be specified in the init.ora file in the ROLLBACK_SEGMENTS parameter.

Rollback Segment Assignment

Transactions are assigned to rollback segments in a round-robin manner attempting to keep all rollback segments equally busy. In this example all three rollback segments are equally “busy,” they each have one active transaction.



When transaction 4 requires rollback segment space, it will be assigned to RBS1. If a transaction exceeds the number of blocks in its extent, a second extent is allocated. Rollback segments are filled in a circular fashion so that old data is available for the longest time possible. If a rollback segment extends as far as possible, a wrap will occur, causing old transactions to be overwritten. If the first transaction in the rollback segment is still active (i.e., a it has not been committed), the wrap will fail and the user will receive an error message. A single transaction must always begin and end in the same rollback segment – transactions cannot span rollback segments.

Dropping Rollback Segments

If a rollback segment is no longer needed, or the initial extent size needs to be changed it can be removed by issuing the DROP ROLLBACK SEGMENT command. If the rollback segment is online, it must be taken offline prior to issuing the drop command.

```
SQL> ALTER ROLLBACK SEGMENT rbs1 OFFLINE;
```

```
Rollback segment altered.
```

```
SQL> DROP ROLLBACK SEGMENT rbs1;
```

```
Rollback segment dropped.
```

Monitoring Rollback Segments

Two data dictionary tables contain information regarding rollback segments: DBA_ROLLBACK_SEGS and V\$ROLLSTAT. To obtain a list of rollback segments and their status query the DBA_ROLLBACK_SEGS table.

```
SQL> SELECT segment_name,status
        FROM dba_rollback_segs;
```

SEGMENT_NAME	STATUS
SYSTEM	ONLINE
RB_TEMP	OFFLINE
RB1	ONLINE
B2	ONLINE
RB3	ONLINE
RB4	ONLINE
RB5	ONLINE
RB6	ONLINE
RB7	ONLINE
RB8	OFFLINE
RB9	OFFLINE

To determine the value for the optimal parameter, the dynamic performance table V\$ROLLSTAT must be queried.

```
SQL> SELECT segment_name,
           optsize "Optimal"
        FROM v$rollstat,
           dba_rollback_segs
        WHERE usn = segment_id;
```

SEGMENT_NAME	Optimal
SYSTEM	
RB1	307200
RB2	307200
RB3	307200
RB4	307200
RB5	307200
RB6	307200
RB7	307200

8 rows selected.

The V\$ROLLSTAT dynamic performance table also contains important information regarding shrinks, wraps, and automatic extends. All of these actions are performed dynamically by the database at the expense of processing time. To monitor these activities, query the V\$ROLLSTAT dictionary table.

```
SQL> SELECT name,
           extends,
           wraps,
           shrinks
        FROM v$rollstat r,
           v$rollname n
        WHERE r.usn = n.usn;
```

A large number of extends would indicate the need to increase the extent size for the rollback segment. High numbers of wraps would cause a decreased amount of data available for “old images” required by long running queries and indicate a need for increasing the maximum size of the rollback segment. High numbers of shrinks could be reduced by increasing the size of the optimal parameter. To force a rollback segment to shrink back to the OPTIMAL size issue the following command:

```
SQL> ALTER ROLLBACK SEGMENT rbs_name SHRINK;
```

The SHRINK option also allows the specification of a size (in bytes). The default shrink will be back to the value specified by the OPTIMAL parameter.

Determining Transaction Size

The transaction size in bytes can be determined using the V\$ROLLSTAT dynamic performance table. The WRITES column in the V\$ROLLSTAT dictionary table contains the number of bytes written to the rollback segment. A temporary table may be created to store the beginning and ending bytes in a rollback segment based on a specific transaction.

```
SQL> CREATE TABLE tran_size
      (begin_bytes number,
       end_bytes number);
```

An insert statement can be used to retrieve the current number of bytes in a specific rollback segment and write that number to the column begin_bytes.

```
SQL> INSERT INTO tran_size
      (begin_bytes)
      (SELECT writes
       FROM v$rollstat r,
          v$rollname n
       WHERE r.usn = n.usn
            AND name = 'RB1');
```

Next, use a SET TRANSACTION command to specify the rollback segment that will be used to hold the undo data for the transaction. The SET TRANSACTION command must be the first command issued following a transaction control statement. This means that you must issue a COMMIT or ROLLBACK prior to issuing the SET TRANSACTION command.

```
SQL> COMMIT;

SQL> SET TRANSACTION USE ROLLBACK SEGMENT rbl;
```

The transaction can then be issued.

```
SQL> DELETE FROM invoice
      WHERE invoice_date < '01-JAN-99';
```

Insert the new value from the V\$ROLLSTAT writes column into the tran_size table end_bytes column.

```
SQL> UPDATE tran_size
      SET end_bytes = (SELECT writes
                      FROM v$rollstat r,
                      v$rollname n
                      WHERE r.usn = n.usn
                      AND name = 'RB1');
```

The ending bytes minus the beginning bytes will provide the transaction size.

```
SQL> SELECT end_bytes - begin_bytes "Required Bytes"
      FROM tran_size;
```

Transaction sizes can be useful in determining the extent size for rollback segments. If it were possible to determine the “average” transaction that would occur in the database, along with the largest transaction size those two numbers can be used to set the size of the rollback segment extents.

The number of extents required for each rollback segment would be based on the number of concurrent users accessing the database, and the desired numbers of users that should be assigned to each rollback segment. A typical number of transactions per rollback segment are four. If the number of concurrent users were 40, then the typical number of rollback segments required would be 10.

Conclusion

Rollback Segments are created and managed by the database administrator and contain the old data that allows a user to undo a transaction. Monitor rollback segments through the data dictionary tables V\$ROLLSTAT and DBA_ROLLBACK_SEGS.

About the Author

Elizabeth Boss is president of Boss Consulting Services, Inc. She has more than 15 years of experience in application development and database administration as a database administrator, consultant, instructor, and curriculum developer. As a senior consultant/instructor, Elizabeth has worked directly with customers in all phases and aspects of the design, development, and administration of Oracle systems. She is a frequent presenter at international and local user groups, and, in 1997, was listed among the top 25 speakers at the IOUG-A Conference.

Elizabeth can be reached at:

Boss Consulting Services Inc.
710 Kings Deer Point
Monument, CO 80132
Phone: (877) 489-7745 Fax: (719) 481-5820
Internet: eboss@boss-consulting-inc.com