

Managing Undo

For a default installation, Oracle Database automatically manages undo. There is typically no need for DBA intervention. However, if your installation uses Oracle Flashback operations, you may need to perform some undo management tasks to ensure the success of these operations.

- [What Is Undo?](#)
Oracle Database creates and manages information that is used to roll back, or undo, changes to the database. Such information consists of records of the actions of transactions, primarily before they are committed. These records are collectively referred to as **undo**.
- [Introduction to Automatic Undo Management](#)
Oracle Database can manage undo information and space automatically.
- [Setting the Minimum Undo Retention Period](#)
You specify the minimum undo retention period (in seconds) by setting the `UNDO_RETENTION` initialization parameter.
- [Sizing a Fixed-Size Undo Tablespace](#)
If you decide to use a fixed-size undo tablespace, then the Undo Advisor can help you estimate needed capacity.
- [Managing Undo Tablespaces](#)
You manage undo tablespaces by completing tasks such as creating, altering, and dropping them. You can also switch undo tablespaces and establish user quotas for undo space.
- [Migrating to Automatic Undo Management](#)
If you are currently using rollback segments to manage undo space, Oracle strongly recommends that you migrate your database to automatic undo management.
- [Managing Temporary Undo](#)
By default, undo records for temporary tables are stored in the undo tablespace and are logged in the redo, which is the same way undo is managed for persistent tables. However, you can use the `TEMP_UNDO_ENABLED` initialization parameter to separate undo for temporary tables from undo for persistent tables. When this parameter is set to `TRUE`, the undo for temporary tables is called **temporary undo**.
- [Undo Space Data Dictionary Views](#)
You can query a set of views for information about undo space in the automatic undo management mode.



See Also:

[Using Oracle Managed Files](#) for information about creating an undo tablespace whose data files are both created and managed by Oracle Database.

14.1 What Is Undo?

Oracle Database creates and manages information that is used to roll back, or undo, changes to the database. Such information consists of records of the actions of transactions, primarily before they are committed. These records are collectively referred to as **undo**.

Undo records are used to:

- Roll back transactions when a `ROLLBACK` statement is issued
- Recover the database
- Provide read consistency
- Analyze data as of an earlier point in time by using Oracle Flashback Query
- Recover from logical corruptions using Oracle Flashback features

When a `ROLLBACK` statement is issued, undo records are used to undo changes that were made to the database by the uncommitted transaction. During database recovery, undo records are used to undo any uncommitted changes applied from the redo log to the data files. Undo records provide read consistency by maintaining the before image of the data for users who are accessing the data at the same time that another user is changing it.



See Also:

Oracle Database Concepts

14.2 Introduction to Automatic Undo Management

Oracle Database can manage undo information and space automatically.

- [Overview of Automatic Undo Management](#)
Oracle provides a fully automated mechanism, referred to as automatic undo management, for managing undo information and space. With automatic undo management, the database manages undo segments in an undo tablespace.
- [The Undo Retention Period](#)
The **undo retention period** is the minimum amount of time that Oracle Database attempts to retain old undo information before overwriting it.

14.2.1 Overview of Automatic Undo Management

Oracle provides a fully automated mechanism, referred to as automatic undo management, for managing undo information and space. With automatic undo management, the database manages undo segments in an undo tablespace.

Automatic undo management is the default mode for a newly installed database. An auto-extending undo tablespace named `UNDOTBS1` is automatically created when you create the database with Database Configuration Assistant (DBCA).

You can also create an undo tablespace explicitly. The methods of creating an undo tablespace are explained in "[Creating an Undo Tablespace](#)".

When the database instance starts, the database automatically selects the first available undo tablespace. If no undo tablespace is available, then the instance starts without an undo tablespace and stores undo records in the `SYSTEM` tablespace. This is not recommended, and an alert message is written to the alert log file to warn that the system is running without an undo tablespace.

If the database contains multiple undo tablespaces, then you can optionally specify at startup that you want to use a specific undo tablespace. This is done by setting the `UNDO_TABLESPACE` initialization parameter, as shown in this example:

```
UNDO_TABLESPACE = undotbs_01
```

If the tablespace specified in the initialization parameter does not exist, the `STARTUP` command fails. The `UNDO_TABLESPACE` parameter can be used to assign a specific undo tablespace to an instance in an Oracle Real Application Clusters environment.

The database can also run in *manual undo management mode*. In this mode, undo space is managed through rollback segments, and no undo tablespace is used.

**Note:**

Space management for rollback segments is complex. Oracle strongly recommends leaving the database in automatic undo management mode.

The following is a summary of the initialization parameters for undo management:

Initialization Parameter	Description
<code>UNDO_MANAGEMENT</code>	If <code>AUTO</code> or null, enables automatic undo management. If <code>MANUAL</code> , sets manual undo management mode. The default is <code>AUTO</code> .
<code>UNDO_TABLESPACE</code>	Optional, and valid only in automatic undo management mode. Specifies the name of an undo tablespace. Use only when the database has multiple undo tablespaces and you want to direct the database instance to use a particular undo tablespace.

When automatic undo management is enabled, if the initialization parameter file contains parameters relating to manual undo management, they are ignored.

**Note:**

Earlier releases of Oracle Database default to manual undo management mode. To change to automatic undo management, you must first create an undo tablespace and then change the `UNDO_MANAGEMENT` initialization parameter to `AUTO`. If your Oracle Database is Oracle9i or later and you want to change to automatic undo management, see *Oracle Database Upgrade Guide* for instructions.

A null `UNDO_MANAGEMENT` initialization parameter defaults to automatic undo management mode in Oracle Database 11g and later, but defaults to manual undo management mode in earlier releases. You must therefore use caution when upgrading a previous release to the current release. *Oracle Database Upgrade Guide* describes the correct method of migrating to automatic undo management mode, including information on how to size the undo tablespace.

14.2.2 The Undo Retention Period

The **undo retention period** is the minimum amount of time that Oracle Database attempts to retain old undo information before overwriting it.

- [About the Undo Retention Period](#)
When automatic undo management is enabled, there is always a current **undo retention period**, which is the minimum amount of time that Oracle Database attempts to retain old undo information before overwriting it.
- [Automatic Tuning of Undo Retention](#)
Oracle Database automatically tunes the undo retention period based on how the undo tablespace is configured.
- [Retention Guarantee](#)
To guarantee the success of long-running queries or Oracle Flashback operations, you can enable retention guarantee.
- [Undo Retention Tuning and Alert Thresholds](#)
For a fixed-size undo tablespace, the database calculates the best possible retention based on database statistics and on the size of the undo tablespace.
- [Tracking the Tuned Undo Retention Period](#)
You can determine the current retention period by querying the `TUNED_UNDORETENTION` column of the `V$UNDOSTAT` view.

14.2.2.1 About the Undo Retention Period

When automatic undo management is enabled, there is always a current **undo retention period**, which is the minimum amount of time that Oracle Database attempts to retain old undo information before overwriting it.

After a transaction is committed, undo data is no longer needed for rollback or transaction recovery purposes. However, for consistent read purposes, long-running queries may require this old undo information for producing older images of data blocks. Furthermore, the success of several Oracle Flashback features can also depend upon the availability of older undo information. For these reasons, it is desirable to retain the old undo information for as long as possible.

Old (committed) undo information that is older than the current undo retention period is said to be *expired* and its space is available to be overwritten by new transactions. Old undo information with an age that is less than the current undo retention period is said to be *unexpired* and is retained for consistent read and Oracle Flashback operations.

Oracle Database automatically tunes the undo retention period based on undo tablespace size and system activity. You can optionally specify a minimum undo retention period (in seconds) by setting the `UNDO_RETENTION` initialization parameter. The exact impact this parameter on undo retention is as follows:

- The `UNDO_RETENTION` parameter is ignored for a fixed size undo tablespace. The database always tunes the undo retention period for the best possible retention, based on system activity and undo tablespace size. See "[Automatic Tuning of Undo Retention](#)" for more information.
- For an undo tablespace with the `AUTOEXTEND` option enabled, the database attempts to honor the minimum retention period specified by `UNDO_RETENTION`. When space is low, instead of overwriting unexpired undo information, the tablespace auto-extends. If the `MAXSIZE` clause is specified for an auto-extending undo tablespace, when the maximum

size is reached, the database may begin to overwrite unexpired undo information. The `UNDOTBS1` tablespace that is automatically created by DBCA is auto-extending.

The `UNDO_RETENTION` parameter is not inheritable in a CDB database.

If you want to set the parameter `UNDO_RETENTION` in all PDBs with the same value, use the `CONTAINER=ALL` clause from the `CDB$ROOT`. For example,

```
alter session set container = cdb$root;  
alter system set undo_retention = 2000 container=all scope=both;
```

This modified the parameter value only in memory for all PDBs, and in both the SPFILE and memory for the `CDB$ROOT`. The parameter is not persisted across PDB restarts.

You can modify the `UNDO_RETENTION` parameter so that it is persistent across restarts in each PDB. For example,

```
alter session set container = pdb1;  
alter system set undo_retention = 2000 scope=both;
```

**Note:**

Modifying `UNDO_RETENTION` is only useful if you are using the Oracle Flashback feature, like Flashback Query or Active Dataguard, that require undo to be retained for longer than the longest running query in the system. The purpose of `UNDO_RETENTION` is to enforce, on a best-effort basis, a minimum value of undo retention by the autotuning algorithm, but it is not guaranteed.

14.2.2.2 Automatic Tuning of Undo Retention

Oracle Database automatically tunes the undo retention period based on how the undo tablespace is configured.

Oracle Database automatically tunes the undo retention period based on various parameters such as undo tablespace size, undo generation rate, maximum query length, and the `RETENTION GUARANTEE` setting. A bigger undo tablespace size provides the ability to retain undo for a longer duration. Therefore, it's important to choose the size of a fixed-size undo tablespace or `MAXSIZE` of an `AUTOEXTEND` undo tablespace that is sufficiently large to accommodate your workload. If you choose an undo tablespace size that is too small, the following two errors could occur:

- DML could fail because there is not enough space to accommodate undo for new transactions.
- Long-running queries could fail with a snapshot too old error, which means that there was insufficient undo data for read consistency.

See [Sizing a Fixed-Size Undo Tablespace](#) for more information.

If the rate of undo consumption is high, undo cannot be retained for long by the database. Therefore, the retention will be automatically tuned to a lower value. Alternately, a low undo consumption rate will provide a higher undo retention.

See `V$UNDOSTAT` `UNDOBLKS` column to get the undo blocks consumed over 10-minute intervals.

Whether the undo tablespace is fixed size or configured with the `AUTOEXTEND` option, the database dynamically tunes the undo retention period to be somewhat longer than the longest-running active query on the system. If queries run for a very long duration and the database cannot provide a high retention, it dynamically tunes the undo retention period to a lower best possible retention for that tablespace size and the current system load.

However, the retention period based on the longest-running active query may be insufficient to accommodate Oracle Flashback operations. Oracle Flashback operations resulting in snapshot too old errors are the indicator for which you must intervene to ensure that sufficient undo data is retained to support these operations. To better accommodate Oracle Flashback features, you should set the `UNDO_RETENTION` parameter to a value equal to the longest expected Oracle Flashback operation.

See `V$UNDOSTAT` `MAXQUERYLEN` column to get the longest-running active query duration.

Setting the `UNDO_RETENTION` parameter for Active Data Guard does not have any effect since the physical standby server does not run any DMLs. To accommodate the queries on Active Data Guard, specify the minimum retention on the primary database instance using the `UNDO_RETENTION` parameter and monitor `V$UNDOSTAT`'s `TUNED_UNDORETENTION` on the primary server.

**Note:**

Automatic tuning of undo retention is not supported for LOBs. This is because undo information for LOBs is stored in the segment itself and not in the undo tablespace. For LOBs, the database attempts to honor the minimum undo retention period specified by `UNDO_RETENTION`. However, if space becomes low, unexpired LOB undo information may be overwritten.

**See Also:**

[Setting the Minimum Undo Retention Period](#)
Retention Guarantee

14.2.2.3 Retention Guarantee

To guarantee the success of long-running queries or Oracle Flashback operations, you can enable retention guarantee.

If retention guarantee is enabled, then the specified minimum undo retention is guaranteed; the database never overwrites unexpired undo data even if it means that transactions fail due to lack of space in the undo tablespace. If retention guarantee is not enabled, then the database can overwrite unexpired undo when space is low, thus lowering the undo retention for the system. This option is disabled by default.

WARNING:

Enabling retention guarantee can cause multiple DML operations to fail. Use with caution.

You enable retention guarantee by specifying the `RETENTION GUARANTEE` clause for the undo tablespace when you create it with either the `CREATE DATABASE` or `CREATE UNDO TABLESPACE` statement. Or, you can later specify this clause in an `ALTER TABLESPACE` statement. You disable retention guarantee with the `RETENTION NOGUARANTEE` clause.

You can use the `DBA_TABLESPACES` view to determine the retention guarantee setting for the undo tablespace. A column named `RETENTION` contains a value of `GUARANTEE`, `NOGUARANTEE`, or `NOT APPLY`, where `NOT APPLY` is used for tablespaces other than the undo tablespace.

14.2.2.4 Undo Retention Tuning and Alert Thresholds

For a fixed-size undo tablespace, the database calculates the best possible retention based on database statistics and on the size of the undo tablespace.

For optimal undo management, rather than tuning based on 100% of the tablespace size, the database tunes the undo retention period based on 70% of the tablespace size, or on the warning alert threshold percentage for space used, whichever is lower. (The warning alert threshold defaults to 70%, but can be changed.) Therefore, if you set the warning alert threshold of the undo tablespace below 70%, this may reduce the tuned size of the undo retention period. For more information on tablespace alert thresholds, see ["Managing Tablespace Alerts"](#).

14.2.2.5 Tracking the Tuned Undo Retention Period

You can determine the current retention period by querying the `TUNED_UNDORETENTION` column of the `V$UNDOSTAT` view.

This view contains one row for each 10-minute statistics collection interval over the last 4 days. (Beyond 4 days, the data is available in the `DBA_HIST_UNDOSTAT` view.) `TUNED_UNDORETENTION` is given in seconds.

```
select to_char(begin_time, 'DD-MON-RR HH24:MI') begin_time,
       to_char(end_time, 'DD-MON-RR HH24:MI') end_time, tuned_undoretention
from v$undostat order by end_time;
```

BEGIN_TIME	END_TIME	TUNED_UNDORETENTION
04-FEB-05 00:01	04-FEB-05 00:11	12100
...		
07-FEB-05 23:21	07-FEB-05 23:31	86700
07-FEB-05 23:31	07-FEB-05 23:41	86700
07-FEB-05 23:41	07-FEB-05 23:51	86700
07-FEB-05 23:51	07-FEB-05 23:52	86700

576 rows selected.

See *Oracle Database Reference* for more information about `V$UNDOSTAT`.

14.3 Setting the Minimum Undo Retention Period

You specify the minimum undo retention period (in seconds) by setting the `UNDO_RETENTION` initialization parameter.

As described in "About the Undo Retention Period", the current undo retention period may be automatically tuned to be greater than `UNDO_RETENTION`, or, unless retention guarantee is enabled, less than `UNDO_RETENTION` if space in the undo tablespace is low.

To set the minimum undo retention period:

- Do one of the following:
 - Set `UNDO_RETENTION` in the initialization parameter file.

```
UNDO_RETENTION = 1800
```
 - Change `UNDO_RETENTION` at any time using the `ALTER SYSTEM` statement:

```
ALTER SYSTEM SET UNDO_RETENTION = 2400;
```

The effect of an `UNDO_RETENTION` parameter change is immediate, but it can only be honored if the current undo tablespace has enough space. The `UNDO_RETENTION` parameter is not inheritable in a CDB configuration and should only be changed from its default value to accommodate Oracle Flashback operations or Active Data Guard.

14.4 Sizing a Fixed-Size Undo Tablespace

If you decide to use a fixed-size undo tablespace, then the Undo Advisor can help you estimate needed capacity.

You can access the Undo Advisor through Oracle Enterprise Manager Database Express (EM Express) or through the `DBMS_ADVISOR` PL/SQL package. EM Express is the preferred method of accessing the advisor.

The Undo Advisor relies for its analysis on data collected in the Automatic Workload Repository (AWR). It is therefore important that the AWR have adequate workload statistics available so that the Undo Advisor can make accurate recommendations. For newly created databases, adequate statistics may not be available immediately. In such cases, continue to use the default auto-extending undo tablespace until at least one workload cycle completes.

An adjustment to the collection interval and retention period for AWR statistics can affect the precision and the type of recommendations that the advisor produces. See *Oracle Database Performance Tuning Guide* for more information.

To use the Undo Advisor, you first estimate these two values:

- The length of your expected longest running query
After the database has completed a workload cycle, you can view the Longest Running Query field on the System Activity subpage of the Automatic Undo Management page.
- The longest interval that you will require for Oracle Flashback operations
For example, if you expect to run Oracle Flashback queries for up to 48 hours in the past, your Oracle Flashback requirement is 48 hours.

You then take the maximum of these two values and use that value as input to the Undo Advisor.

Running the Undo Advisor does not alter the size of the undo tablespace. The advisor just returns a recommendation. You must use `ALTER DATABASE` statements to change the tablespace data files to fixed sizes.

The following example assumes that the undo tablespace has one auto-extending data file named `undotbs.dbf`. The example changes the tablespace to a fixed size of 300MB.

```
ALTER DATABASE DATAFILE '/oracle/dbs/undotbs.dbf' RESIZE 300M;
ALTER DATABASE DATAFILE '/oracle/dbs/undotbs.dbf' AUTOEXTEND OFF;
```

Note:

To make the undo tablespace fixed-size, Oracle suggests that you first allow enough time after database creation to run a full workload, thus allowing the undo tablespace to grow to its minimum required size to handle the workload. Then, you can use the Undo Advisor to determine, if desired, how much larger to set the size of the undo tablespace to allow for long-running queries and Oracle Flashback operations.

Note:

Oracle Enterprise Manager Database Express (EM Express) is deprecated, and will be removed in a future Oracle Database release.

- [Activating the Undo Advisor PL/SQL Interface](#)
You can activate the Undo Advisor by creating an undo advisor task through the advisor framework.

14.4.1 Activating the Undo Advisor PL/SQL Interface

You can activate the Undo Advisor by creating an undo advisor task through the advisor framework.

The following example creates an undo advisor task to evaluate the undo tablespace. The name of the advisor is 'Undo Advisor'. The analysis is based on Automatic Workload Repository snapshots, which you must specify by setting parameters `START_SNAPSHOT` and `END_SNAPSHOT`. In the following example, the `START_SNAPSHOT` is "1" and `END_SNAPSHOT` is "2".

```
DECLARE
    tid    NUMBER;
    tname  VARCHAR2(30);
    oid    NUMBER;
BEGIN
    DBMS_ADVISOR.CREATE_TASK('Undo Advisor', tid, tname, 'Undo Advisor Task');
    DBMS_ADVISOR.CREATE_OBJECT(tname, 'UNDO_TBS', null, null, null, 'null', oid);
    DBMS_ADVISOR.SET_TASK_PARAMETER(tname, 'TARGET_OBJECTS', oid);
    DBMS_ADVISOR.SET_TASK_PARAMETER(tname, 'START_SNAPSHOT', 1);
    DBMS_ADVISOR.SET_TASK_PARAMETER(tname, 'END_SNAPSHOT', 2);
    DBMS_ADVISOR.SET_TASK_PARAMETER(tname, 'INSTANCE', 1);
    DBMS_ADVISOR.execute_task(tname);
END;
/
```

After you have created the advisor task, you can view the output and recommendations in the Automatic Database Diagnostic Monitor in EM Express. This information is also available in the

DBA_ADVISOR_* data dictionary views (DBA_ADVISOR_TASKS, DBA_ADVISOR_OBJECTS, DBA_ADVISOR_FINDINGS, DBA_ADVISOR_RECOMMENDATIONS, and so on).

**Note:**

Oracle Enterprise Manager Database Express (EM Express) is deprecated, and will be removed in a future Oracle Database release.

**See Also:**

- ["Using the Segment Advisor"](#) for an example of creating an advisor task for a different advisor
- *Oracle Database Reference* for information about the DBA_ADVISOR_* data dictionary views

14.5 Managing Undo Tablespaces

You manage undo tablespaces by completing tasks such as creating, altering, and dropping them. You can also switch undo tablespaces and establish user quotas for undo space.

- [Creating an Undo Tablespace](#)
Although Database Configuration Assistant (DBCA) automatically creates an undo tablespace for new installations, there may be occasions when you want to manually create an undo tablespace.
- [Altering an Undo Tablespace](#)
You can alter an undo tablespaces using the `ALTER TABLESPACE` statement.
- [Dropping an Undo Tablespace](#)
Use the `DROP TABLESPACE` statement to drop an undo tablespace.
- [Switching Undo Tablespaces](#)
You can switch from using one undo tablespace to another. Because the `UNDO_TABLESPACE` initialization parameter is a dynamic parameter, the `ALTER SYSTEM SET` statement can be used to assign a new undo tablespace.
- [Establishing User Quotas for Undo Space](#)
You can use the Oracle Database Resource Manager to establish user quotas for undo space. The Database Resource Manager directive `UNDO_POOL` allows DBAs to limit the amount of undo space consumed by a group of users (resource consumer group).
- [Managing Space Threshold Alerts for the Undo Tablespace](#)
Oracle Database provides proactive help in managing tablespace disk space use by alerting you when tablespaces run low on available space.

14.5.1 Creating an Undo Tablespace

Although Database Configuration Assistant (DBCA) automatically creates an undo tablespace for new installations, there may be occasions when you want to manually create an undo tablespace.

- [About Creating an Undo Tablespace](#)
When you are creating a database, you can create an undo tablespace with the `CREATE DATABASE` statement. In an existing database, you can create an undo tablespace with the `CREATE UNDO TABLESPACE` statement.
- [Using CREATE DATABASE to Create an Undo Tablespace](#)
You can create a specific undo tablespace using the `UNDO TABLESPACE` clause of the `CREATE DATABASE` statement.
- [Using the CREATE UNDO TABLESPACE Statement](#)
The `CREATE UNDO TABLESPACE` statement is the same as the `CREATE TABLESPACE` statement, but the `UNDO` keyword is specified. The database determines most of the attributes of the undo tablespace, but you can specify the `DATAFILE` clause.

14.5.1.1 About Creating an Undo Tablespace

When you are creating a database, you can create an undo tablespace with the `CREATE DATABASE` statement. In an existing database, you can create an undo tablespace with the `CREATE UNDO TABLESPACE` statement.

There are two methods of creating an undo tablespace. The first method creates the undo tablespace when the `CREATE DATABASE` statement is issued. This occurs when you are creating a new database, and the instance is started in automatic undo management mode (`UNDO_MANAGEMENT = AUTO`). The second method is used with an existing database. It uses the `CREATE UNDO TABLESPACE` statement.

You cannot create database objects in an undo tablespace. It is reserved for system-managed undo data.

Oracle Database enables you to create a single-file undo tablespace. Single-file, or bigfile, tablespaces are discussed in "[Bigfile Tablespaces](#)".

14.5.1.2 Using CREATE DATABASE to Create an Undo Tablespace

You can create a specific undo tablespace using the `UNDO TABLESPACE` clause of the `CREATE DATABASE` statement.

The following statement illustrates using the `UNDO TABLESPACE` clause in a `CREATE DATABASE` statement. The undo tablespace is named `undotbs_01` and one data file, `/u01/oracle/rbdb1/undo0101.dbf`, is allocated for it.

```
CREATE DATABASE rbdb1
  CONTROLFILE REUSE
  .
  .
  .
  UNDO TABLESPACE undotbs_01 DATAFILE '/u01/oracle/rbdb1/undo0101.dbf';
```

If the undo tablespace cannot be created successfully during `CREATE DATABASE`, the entire `CREATE DATABASE` operation fails. You must clean up the database files, correct the error and retry the `CREATE DATABASE` operation.

The `CREATE DATABASE` statement also lets you create a single-file undo tablespace at database creation.

Related Topics

- *Oracle Multitenant Administrator's Guide*

- *Oracle Database SQL Language Reference*

14.5.1.3 Using the CREATE UNDO TABLESPACE Statement

The `CREATE UNDO TABLESPACE` statement is the same as the `CREATE TABLESPACE` statement, but the `UNDO` keyword is specified. The database determines most of the attributes of the undo tablespace, but you can specify the `DATAFILE` clause.

This example creates the `undotbs_02` undo tablespace with the `AUTOEXTEND` option:

```
CREATE UNDO TABLESPACE undotbs_02
  DATAFILE '/u01/oracle/rbdb1/undo0201.dbf' SIZE 2M REUSE AUTOEXTEND ON;
```

You can create multiple undo tablespaces, but only one of them can be active at any one time.



See Also:

Oracle Database SQL Language Reference for the syntax for using the `CREATE UNDO TABLESPACE` statement to create an undo tablespace

14.5.2 Altering an Undo Tablespace

You can alter an undo tablespaces using the `ALTER TABLESPACE` statement.

However, since most aspects of undo tablespaces are system managed, you need only be concerned with the following actions:

- Adding a data file
- Renaming a data file
- Bringing a data file online or taking it offline
- Beginning or ending an open backup on a data file
- Enabling and disabling undo retention guarantee

These are also the only attributes you are permitted to alter.

If an undo tablespace runs out of space, or you want to prevent it from doing so, you can add more files to it or resize existing data files.

The following example adds another data file to undo tablespace `undotbs_01`:

```
ALTER TABLESPACE undotbs_01
  ADD DATAFILE '/u01/oracle/rbdb1/undo0102.dbf' AUTOEXTEND ON NEXT 1M
  MAXSIZE UNLIMITED;
```

You can use the `ALTER DATABASE...DATAFILE` statement to resize or extend a data file.

**See Also:**

- ["Changing Data File Size"](#)
- *Oracle Database SQL Language Reference* for ALTER TABLESPACE syntax

14.5.3 Dropping an Undo Tablespace

Use the DROP TABLESPACE statement to drop an undo tablespace.

The following example drops the undo tablespace undotbs_01:

```
DROP TABLESPACE undotbs_01;
```

An undo tablespace can only be dropped if it is not currently used by any instance. If the undo tablespace contains any outstanding transactions (for example, a transaction died but has not yet been recovered), the DROP TABLESPACE statement fails. However, since DROP TABLESPACE drops an undo tablespace even if it contains unexpired undo information (within retention period), you must be careful not to drop an undo tablespace if undo information is needed by some existing queries.

DROP TABLESPACE for undo tablespaces behaves like DROP TABLESPACE...INCLUDING CONTENTS. All contents of the undo tablespace are removed.

**See Also:**

Oracle Database SQL Language Reference for DROP TABLESPACE syntax

14.5.4 Switching Undo Tablespaces

You can switch from using one undo tablespace to another. Because the UNDO_TABLESPACE initialization parameter is a dynamic parameter, the ALTER SYSTEM SET statement can be used to assign a new undo tablespace.

The following statement switches to a new undo tablespace:

```
ALTER SYSTEM SET UNDO_TABLESPACE = undotbs_02;
```

Assuming undotbs_01 is the current undo tablespace, after this command successfully executes, the instance uses undotbs_02 in place of undotbs_01 as its undo tablespace.

If any of the following conditions exist for the tablespace being switched to, an error is reported and no switching occurs:

- The tablespace does not exist
- The tablespace is not an undo tablespace
- The tablespace is already being used by another instance (in an Oracle RAC environment only)

The database is online while the switch operation is performed, and user transactions can be executed while this command is being executed. When the switch operation completes

successfully, all transactions started after the switch operation began are assigned to transaction tables in the new undo tablespace.

The switch operation does not wait for transactions in the old undo tablespace to commit. If there are any pending transactions in the old undo tablespace, the old undo tablespace enters into a `PENDING OFFLINE` mode (status). In this mode, existing transactions can continue to execute, but undo records for new user transactions cannot be stored in this undo tablespace.

An undo tablespace can exist in this `PENDING OFFLINE` mode, even after the switch operation completes successfully. A `PENDING OFFLINE` undo tablespace cannot be used by another instance, nor can it be dropped. Eventually, after all active transactions have committed, the undo tablespace automatically goes from the `PENDING OFFLINE` mode to the `OFFLINE` mode. From then on, the undo tablespace is available for other instances (in an Oracle Real Application Cluster environment).

If the parameter value for `UNDO TABLESPACE` is set to " (two single quotes), then the current undo tablespace is switched out and the next available undo tablespace is switched in. Use this statement with care because there may be no undo tablespace available.

The following example unassigns the current undo tablespace:

```
ALTER SYSTEM SET UNDO_TABLESPACE = '';
```

14.5.5 Establishing User Quotas for Undo Space

You can use the Oracle Database Resource Manager to establish user quotas for undo space. The Database Resource Manager directive `UNDO_POOL` allows DBAs to limit the amount of undo space consumed by a group of users (resource consumer group).

You can specify an undo pool for each consumer group. An undo pool controls the amount of total undo that can be generated by a consumer group. When the total undo generated by a consumer group exceeds its undo limit, the current `UPDATE` transaction generating the undo is terminated. No other members of the consumer group can perform further updates until undo space is freed from the pool.

When no `UNDO_POOL` directive is explicitly defined, users are allowed unlimited undo space.



See Also:

[Managing Resources with Oracle Database Resource Manager](#)

14.5.6 Managing Space Threshold Alerts for the Undo Tablespace

Oracle Database provides proactive help in managing tablespace disk space use by alerting you when tablespaces run low on available space.

See "[Managing Tablespace Alerts](#)" for information on how to set alert thresholds for the undo tablespace.

In addition to the proactive undo space alerts, Oracle Database also provides alerts if your system has long-running queries that cause `SNAPSHOT TOO OLD` errors. To prevent excessive alerts, the long query alert is issued at most once every 24 hours. When the alert is generated, you can check the Undo Advisor Page of EM Express to get more information about the undo tablespace.

**Note:**

Oracle Enterprise Manager Database Express (EM Express) is deprecated, and will be removed in a future Oracle Database release.

14.6 Migrating to Automatic Undo Management

If you are currently using rollback segments to manage undo space, Oracle strongly recommends that you migrate your database to automatic undo management.

For instructions, see *Oracle Database Upgrade Guide*.

14.7 Managing Temporary Undo

By default, undo records for temporary tables are stored in the undo tablespace and are logged in the redo, which is the same way undo is managed for persistent tables. However, you can use the `TEMP_UNDO_ENABLED` initialization parameter to separate undo for temporary tables from undo for persistent tables. When this parameter is set to `TRUE`, the undo for temporary tables is called **temporary undo**.

- [About Managing Temporary Undo](#)
Temporary undo records are stored in the database's temporary tablespaces and thus are not logged in the redo log. When temporary undo is enabled, some of the segments used by the temporary tablespaces store the temporary undo, and these segments are called **temporary undo segments**.
- [Enabling and Disabling Temporary Undo](#)
You can enable or disable temporary undo for a session or for the system. To do so, set the `TEMP_UNDO_ENABLED` initialization parameter.

14.7.1 About Managing Temporary Undo

Temporary undo records are stored in the database's temporary tablespaces and thus are not logged in the redo log. When temporary undo is enabled, some of the segments used by the temporary tablespaces store the temporary undo, and these segments are called **temporary undo segments**.

When temporary undo is enabled, it might be necessary to increase the size of the temporary tablespaces to account for the undo records.

Enabling temporary undo provides the following benefits:

- Temporary undo reduces the amount of undo stored in the undo tablespaces.
Less undo in the undo tablespaces can result in more realistic undo retention period requirements for undo records.
- Temporary undo reduces the size of the redo log.
Performance is improved because less data is written to the redo log, and components that parse redo log records, such as LogMiner, perform better because there is less redo data to parse.
- Temporary undo enables data manipulation language (DML) operations on temporary tables in a physical standby database with the Oracle Active Data Guard option. However,

data definition language (DDL) operations that create temporary tables must be issued on the primary database.

You can enable temporary undo for a specific session or for the whole system. When you enable temporary undo for a session using an `ALTER SESSION` statement, the session creates temporary undo without affecting other sessions. When you enable temporary undo for the system using an `ALTER SYSTEM` statement, all existing sessions and new sessions create temporary undo.

When a session uses temporary objects for the first time, the current value of the `TEMP_UNDO_ENABLED` initialization parameter is set for the rest of the session. Therefore, if temporary undo is enabled for a session and the session uses temporary objects, then temporary undo cannot be disabled for the session. Similarly, if temporary undo is disabled for a session and the session uses temporary objects, then temporary undo cannot be enabled for the session.

Temporary undo is enabled by default for a physical standby database with the Oracle Active Data Guard option. The `TEMP_UNDO_ENABLED` initialization parameter has no effect on a physical standby database with Active Data Guard option because of the default setting.

**Note:**

Temporary undo can be enabled only if the compatibility level of the database is 12.0.0 or higher.

**See Also:**

- ["Creating a Temporary Table"](#)
- ["About the Undo Retention Period"](#)
- *Oracle Database Reference* for more information about the `TEMP_UNDO_ENABLED` initialization parameter
- *Oracle Data Guard Concepts and Administration*
- *Oracle Database Concepts* for more information about temporary undo segments

14.7.2 Enabling and Disabling Temporary Undo

You can enable or disable temporary undo for a session or for the system. To do so, set the `TEMP_UNDO_ENABLED` initialization parameter.

To enable or disable temporary undo:

1. In SQL*Plus, connect to the database.

If you are enabling or disabling temporary undo for a session, then start the session in SQL*Plus.

If you are enabling or disabling temporary undo for the system, then connect as an administrative user with the `ALTER SYSTEM` system privilege in SQL*Plus.

See ["Connecting to the Database with SQL*Plus"](#).

2. Set the `TEMP_UNDO_ENABLED` initialization parameter:

- To enable temporary undo for a session, run the following SQL statement:

```
ALTER SESSION SET TEMP_UNDO_ENABLED = TRUE;
```

- To disable temporary undo for a session, run the following SQL statement:

```
ALTER SESSION SET TEMP_UNDO_ENABLED = FALSE;
```

- To enable temporary undo for the system, run the following SQL statement:

```
ALTER SYSTEM SET TEMP_UNDO_ENABLED = TRUE;
```

After temporary undo is enabled for the system, a session can disable temporary undo using the `ALTER SESSION` statement.

- To disable temporary undo for the system, run the following SQL statement:

```
ALTER SYSTEM SET TEMP_UNDO_ENABLED = FALSE;
```

After temporary undo is disabled for the system, a session can enable temporary undo using the `ALTER SESSION` statement.

You can also enable temporary undo for the system by setting `TEMP_UNDO_ENABLED` to `TRUE` in a server parameter file or a text initialization parameter file. In this case, all new sessions create temporary undo unless temporary undo is disabled for the system by an `ALTER SYSTEM` statement or for a session by an `ALTER SESSION` statement.



See Also:

- *Oracle Database Reference* for more information about the `TEMP_UNDO_ENABLED` initialization parameter
- *Oracle Data Guard Concepts and Administration* for information about enabling and disabling temporary undo in an Oracle Data Guard environment

14.8 Undo Space Data Dictionary Views

You can query a set of views for information about undo space in the automatic undo management mode.

In addition to views listed here, you can obtain information from the views available for viewing tablespace and data file information. See "[Data Files Data Dictionary Views](#)" for information on getting information about those views.

The following dynamic performance views are useful for obtaining space information about the undo tablespace:

View	Description
V\$UNDOSTAT	Contains statistics for monitoring and tuning undo space. Use this view to help estimate the amount of undo space required for the current workload. The database also uses this information to help tune undo usage in the system. This view is meaningful only in automatic undo management mode.

View	Description
V\$TEMPUNDOSTAT	Contains statistics for monitoring and tuning temporary undo space. Use this view to help estimate the amount of temporary undo space required in the temporary tablespaces for the current workload. The database also uses this information to help tune temporary undo usage in the system. This view is meaningful only when temporary undo is enabled.
V\$ROLLSTAT	For automatic undo management mode, information reflects behavior of the undo segments in the undo tablespace
V\$TRANSACTION	Contains undo segment information
DBA_UNDO_EXTENTS	Shows the status and size of each extent in the undo tablespace.
DBA_HIST_UNDOSTAT	Contains statistical snapshots of V\$UNDOSTAT information.

The V\$UNDOSTAT view is useful for monitoring the effects of transaction execution on undo space in the current instance. Statistics are available for undo space consumption, transaction concurrency, the tuning of undo retention, and the length and SQL ID of long-running queries in the instance.

Each row in the view contains statistics collected in the instance for a ten-minute interval. The rows are in descending order by the BEGIN_TIME column value. Each row belongs to the time interval marked by (BEGIN_TIME, END_TIME). Each column represents the data collected for the particular statistic in that time interval. The first row of the view contains statistics for the (partial) current time period. The view contains a total of 576 rows, spanning a 4 day cycle.

The following example shows the results of a query on the V\$UNDOSTAT view.

```
SELECT TO_CHAR(BEGIN_TIME, 'MM/DD/YYYY HH24:MI:SS') BEGIN_TIME,
       TO_CHAR(END_TIME, 'MM/DD/YYYY HH24:MI:SS') END_TIME,
       UNDOTSN, UNDOBLKS, TXNCOUNT, MAXCONCURRENCY AS "MAXCON"
FROM v$UNDOSTAT WHERE rownum <= 144;
```

BEGIN_TIME	END_TIME	UNDOTSN	UNDOBLKS	TXNCOUNT	MAXCON
10/28/2004 14:25:12	10/28/2004 14:32:17	8	74	12071108	3
10/28/2004 14:15:12	10/28/2004 14:25:12	8	49	12070698	2
10/28/2004 14:05:12	10/28/2004 14:15:12	8	125	12070220	1
10/28/2004 13:55:12	10/28/2004 14:05:12	8	99	12066511	3
...					
10/27/2004 14:45:12	10/27/2004 14:55:12	8	15	11831676	1
10/27/2004 14:35:12	10/27/2004 14:45:12	8	154	11831165	2

144 rows selected.

The preceding example shows how undo space is consumed in the system for the previous 24 hours from the time 14:35:12 on 10/27/2004.