

Automatic SecureFiles Shrink

The Oracle Database SecureFiles Shrink feature provides manual and automatic methods to free the unused space in SecureFiles LOB segments and release the space back to the containing tablespace. This chapter explains how to use the automatic method called Automatic SecureFiles Shrink with Oracle Database.

- [About Manual SecureFiles Shrink](#)
- [About Automatic SecureFiles Shrink](#)
- [Automatic SecureFiles Shrink Features](#)
Automatic SecureFiles Shrink has the following features:
- [SecureFiles Shrink and Undo Retention](#)
This section discusses how the Auto Shrink feature treats different flavors of undo retention.
- [Enable Automatic SecureFiles Shrink](#)
Automatic SecureFiles Shrink affects out-of-line SecureFile LOBs. Automatic SecureFiles Shrink does not have any effect on the BasicFiles LOBs and inline SecureFiles LOBs.
- [Disable Automatic SecureFiles Shrink](#)
By default, the Automatic SecureFiles Shrink feature is disabled.
- [Targets and Limits](#)
- [Selection Criteria for SecureFiles LOB Segments to Shrink](#)
Here are the criteria that automatic SecureFiles shrink uses for selecting SecureFiles LOB segments to shrink.
- [Automatic SecureFiles Shrink Task](#)
Automatic SecureFiles Shrink performs a series of steps to complete the shrink of SecureFiles LOB segments.
- [Checking Progress](#)

16.1 About Manual SecureFiles Shrink

Use the `ALTER TABLE ... SHRINK SPACE` statement to manually shrink a SecureFiles LOB segment. You can also use tools, such as Segment Advisor or a PL/SQL procedure, such as `DBMS_SPACE.SPACE_USAGE` to return information about SecureFiles space usage before deciding on the SecureFiles LOB segments to shrink.

The following points are important when opting for the manual shrink method:

- The manual SecureFiles shrink operation is an online DDL with part of the operations being offline, where offline means concurrent DML are blocked until the shrink activity on the critical section ends. The concurrent DML statements do not fail with ORA-54, but are blocked.
- The manual SecureFiles shrink operation disregards any flavor of undo retention and treats it as if the retention is equal to none. The user cannot expect the LOB retention feature to provide the usual guarantees after invoking the shrink operation. The user may see the ORA-1555 `snapshot too old` error message in queries. Run the shrink operation with caution if this is a concern.

Use *shrink_clause* on SecureFiles LOB segments from release 21c and onward. There are two ways to invoke *shrink_clause*:

1. The following command targets the *specified* LOB column and all its partitions.

```
ALTER TABLE <table_name> MODIFY LOB <lob_column> SHRINK SPACE
```

2. The following command cascades the shrink operation for *all* LOB columns and its partitions in the specified table.

```
ALTER TABLE <table_name> SHRINK SPACE CASCADE
```



See Also:

- SPACE_USAGE Procedures
- ALTER TABLE

16.2 About Automatic SecureFiles Shrink

SecureFiles LOB segments can potentially become the largest consumer of space in a database. It may not be feasible for administrators to spend their time checking each SecureFiles LOB segment to shrink. Automatic SecureFiles Shrink uses a framework that enables automatic selection of SecureFiles LOB segments to shrink based on a set criteria and it runs Automatic SecureFiles Shrink in the background.

Automatic SecureFiles Shrink is designed to minimize the functional and performance impact on concurrent workloads. While shrink runs automatically on a SecureFiles LOB segment, all DML and DDL statements that involve the segment will succeed. Space is gradually freed in the SecureFiles LOB segment and the performance impact is minimal.

Automatic SecureFiles Shrink does not have any effect on the BasicFiles LOBs and in-lined SecureFiles LOBs.



See Also:

Reclaiming Unused Space

16.3 Automatic SecureFiles Shrink Features

Automatic SecureFiles Shrink has the following features:

Managed by Database

The database provides background execution and resource management infrastructure for Automatic SecureFiles Shrink. The database also executes the shrink task automatically, at a system-determined interval, within a bounded runtime.

**See Also:**

Managing Automated Database Maintenance Tasks

Integrates with Pre-Allocation

Automatic SecureFiles Shrink integrates with pre-allocation seamlessly without affecting performance. Automatic SecureFiles Shrink avoids the SecureFiles LOB segments that are recently pre-allocated. Segment pre-allocation is performed in the background for segments that have high demand for free space.

Works with DDL and DML

Automatic SecureFiles Shrink targets only idle segments and skips active SecureFiles LOB segments. User driven DDL and DML statements do not fail and face minimal performance impact when Automatic SecureFiles Shrink works in the background. If Automatic SecureFiles Shrink for a SecureFiles LOB segment comes across locked rows, it skips the locked rows because locked rows are indicative of DML activity or waiting on locked rows may cause deadlocks with user transactions. Automatic SecureFiles Shrink always acquires row locks in the `NOWAIT` mode to avoid deadlock with user transactions.

Targets Idle LOB Segments

To avoid unnecessary block accesses, Automatic SecureFiles Shrink filters SecureFiles LOB segments based on information available in System Global Area (SGA). Automatic SecureFiles Shrink selects only idle SecureFiles LOB segments and skips active LOB segments to minimize performance impact on active SecureFiles LOB segments.

Covers All SecureFiles LOB Segments

The Automatic SecureFiles Shrink task covers all SecureFiles LOB segments in a PDB over several intervals and this includes user-created LOB segments and the SecureFiles LOB segments that are created using features, such as JSON and DBFS.

Performs Shrinks in Iterations

Automatic SecureFiles Shrink does not free all the free space in the selected SecureFiles LOB segments at once. Instead, the Automatic SecureFiles shrink task frees a modest amount of space at every shrink call (iteration). The trickle threshold limit defines the amount of space to shrink in every iteration. Over time, the amount of free space in idle SecureFiles LOB segments approaches the minimum that is specified for pre-allocation.

Executes in the Background

All steps involved in Automatic SecureFiles Shrink, including the selection of SecureFiles LOB segments to shrink, run in the background. After Automatic SecureFiles Shrink is enabled, it comes into effect with the start of a database instance. No directive regarding how Automatic SecureFiles Shrink should operate is required.

Honors Undo Retention

Automatic SecureFiles Shrink respects the undo retention period. It does not allow a query to fail within the undo retention period because an affected SecureFiles LOB segment has been freed, relocated, or reused as a part of an Automatic SecureFiles Shrink task. Unexpired blocks are freed only after the undo retention time.

16.4 SecureFiles Shrink and Undo Retention

This section discusses how the Auto Shrink feature treats different flavors of undo retention.

As part of the Automatic SecureFiles Shrink feature, used blocks in an extent are relocated, and the extents are freed to the tablespace. After an extent is freed to the tablespace, it may be reused and overwritten. Freeing and reusing an extent during shrink can change the expected behavior of undo retention.

Automatic SecureFiles Shrink moves data in a SecureFiles LOB segment to create free space. The feature honors the LOB retention setting by not reusing space until the minimum retention requirement has been met. In practice, concurrent queries may run into `ORA-1555 Snapshots too old` more often compared to the same queries on an idle segment. Automatic SecureFiles Shrink for Autonomous Database skips SecureFiles LOB segments configured with the `RETENTION` parameter set to `MAX`.

16.5 Enable Automatic SecureFiles Shrink

Automatic SecureFiles Shrink affects out-of-line SecureFile LOBs. Automatic SecureFiles Shrink does not have any effect on the BasicFiles LOBs and inline SecureFiles LOBs.

Automatic SecureFiles Shrink is not enabled by default.

- In on-premises environments, run the following command to enable the Automatic SecureFiles Shrink feature.

```
exec DBMS_SPACE.SECUREFILE_SHRINK_ENABLE();
```

In Autonomous Cloud environments, contact your system administrator to enable Automatic SecureFiles Shrink.

16.6 Disable Automatic SecureFiles Shrink

By default, the Automatic SecureFiles Shrink feature is disabled.

- In on-premises environments, run the following command to disable the Automatic SecureFiles Shrink feature.

```
exec DBMS_SPACE.SECUREFILE_SHRINK_DISABLE();
```

In Autonomous Cloud environments, contact your system administrator to disable Automatic SecureFiles Shrink.

16.7 Targets and Limits

Automatic SecureFiles Shrink follows a set of targets and limits to operate optimally.

- [Automatic SecureFiles Shrink Targets](#)
An Automatic SecureFiles Shrink target determines the amount of space that needs to be freed in one automatic shrink run.

- [LOB Segment Idle Time Limit](#)
The LOB segment idle time limit is the minimum time limit a LOB segment can be idle before a shrink can start.

16.7.1 Automatic SecureFiles Shrink Targets

An Automatic SecureFiles Shrink target determines the amount of space that needs to be freed in one automatic shrink run.

In manual shrink, the shrink command continues to free space until there is no free space left. Automatic SecureFiles Shrink sets a shrink target for every shrink run. The target is based on:

- Pre-allocation threshold that determines the minimum limit below which an Automatic SecureFiles Shrink task cannot free space for a segment.
- Trickle threshold that specifies the maximum limit that Automatic SecureFiles Shrink can free space for a LOB segment in one iteration.
- [Pre-Allocation Threshold](#)
A pre-allocation threshold value is used to compute the targeted minimum free space for a LOB segment.
- [Automatic SecureFiles Shrink Trickle Threshold](#)
The trickle threshold controls the maximum amount of space per segment to be freed by one automatic shrink iteration.

16.7.1.1 Pre-Allocation Threshold

A pre-allocation threshold value is used to compute the targeted minimum free space for a LOB segment.

Pre-allocation threshold values are represented in percentages of free space out of the LOB segment size. For example, for a 2 TB LOB segment, the minimum amount of free space is 20 GB (2 TB * 1%). If the actual free space is less than the pre-allocation threshold, pre-allocation triggers in and adds extents to the segment. By default, unexpired space is not considered as free space.



Note:

Automatic SecureFiles Shrink does not free space that is less than the pre-allocation minimum threshold for a LOB segment.

[Pre-Allocation Threshold](#) shows the default pre-allocation threshold values for different segment sizes.



Note:

You can adjust the thresholds using optimizations in the future Oracle Database releases.

Table 16-1 Pre-Allocation Threshold

Segment Size	Actual Threshold	Minimum Free Space
<= 1 GB	10%	100 MB
<= 10 GB	5%	500 MB
<= 100 GB	2%	2 GB
> 100 GB	1%	1 GB

16.7.1.2 Automatic SecureFiles Shrink Trickle Threshold

The trickle threshold controls the maximum amount of space per segment to be freed by one automatic shrink iteration.

The shrink run on a LOB segment frees up space incrementally. Therefore, the trickle threshold is an incremental shrink amount, with the shrink task freeing limited space in iterations without concurrent DML and DDL statements facing any noticeable performance impact due to higher latency.

**Note:**

Automatic SecureFiles Shrink does not free space that is less than the pre-allocation minimum threshold for a LOB segment.

16.7.2 LOB Segment Idle Time Limit

The LOB segment idle time limit is the minimum time limit a LOB segment can be idle before a shrink can start.

The LOB segment idle time limit specifies the minimum amount of time that a LOB segment must be idle to qualify for Automatic SecureFiles Shrink. An idle LOB segment is defined as one that has neither user-driven DML statements nor pre-allocation activity in the past N hours. LOB Segment Idle Time Limit defines the N hours. The default value for LOB segment idle time limit is 1440 minutes, which is 24 hours.

16.8 Selection Criteria for SecureFiles LOB Segments to Shrink

Here are the criteria that automatic SecureFiles shrink uses for selecting SecureFiles LOB segments to shrink.

The Automatic SecureFiles Shrink task excludes the following SecureFiles LOB segments when choosing the SecureFiles LOB segments to shrink:

- The SecureFiles LOB segment is not an idle segment as per LOB Segment Idle Time Limit.
- The SecureFiles LOB segment does not contain extra free space greater than the pre-allocation threshold.
- The SecureFiles LOB segment has `RETENTION MAX`, which means the segment keeps as many unexpired blocks as possible.

- The SecureFiles LOB segment is currently being shrunk.
- The SecureFiles LOB segment does not have enough expired free space that is no longer needed for lob retention requirement. Space that is still needed for lob retention is treated as used space.
- The SecureFiles LOB segment has failed a previous shrink task. Previous shrink calls have failed to free space from the SecureFiles LOB segment. Automatic SecureFiles Shrink identifies the LOB segments that it failed to shrink previously and avoids such segments.

16.9 Automatic SecureFiles Shrink Task

Automatic SecureFiles Shrink performs a series of steps to complete the shrink of SecureFiles LOB segments.

When enabled, a shrink task is performed as one instance of the background action performed on AutoTask. The task runs every 30 minutes and performs the following steps:

1. A shrink task has 60 minutes at the start of the task. As the task progresses, it tracks both the time spent so far and the average duration of a shrink call. The latter is used to predict how long the next shrink call would take. If the time left is not enough for another call, the shrink task exits. If a shrink call goes over the 60-minute mark, it is terminated.
2. Automatic SecureFiles Shrink fetches the next batch of SecureFiles LOB segments from internal catalog tables (which is ordered by the object identifier). The last object identifier in the previous shrink task is used as the starting point for the next shrink task.
3. Automatic SecureFiles Shrink applies the criteria filters from the Selection Criteria for SecureFiles LOB segment to remove the segments that do not qualify for the shrink task.



See Also:

[Selection Criteria for SecureFiles LOB Segments to Shrink](#)

4. Once the qualified segment is found, the shrink task can start work on the segment.
5. Before starting the shrink, the shrink target is computed. The shrink target is based on the Pre-Allocation Threshold and the Automatic SecureFiles Shrink Trickle Threshold.
6. Automatic SecureFiles Shrink runs the shrink command. The `ALTER TABLE ... SHRINK SPACE` command is executed using the OCI interface.
7. Automatic SecureFiles Shrink updates the timestamp for the next shrink. This timestamp indicates the earliest time when Automatic SecureFiles Shrink can select this SecureFiles LOB segment again. If space was freed successfully, the timestamp uses the current time. Otherwise, the shrink is assigned a time in the future. If shrink is not successful, a penalty time is assessed to avoid Automatic SecureFiles Shrink from selecting the same segment in future shrink tasks.

**See Also:**

- [LOB Segment Idle Time Limit](#)
- [Pre-Allocation Threshold](#)
- [Automatic SecureFiles Shrink Trickle Threshold](#)

16.10 Checking Progress

The most straightforward way of checking the result of Automatic SecureFiles Shrink is to compute the combined size of the SecureFiles LOB segments in the database. Alternatively, you can observe space saving at a micro level using the `V$SECUREFILE_SHRINK` table, which reports one row for each SecureFiles LOB segment that is shrunk.

**See Also:**

`V$SECUREFILE_SHRINK` to see the results of the previous shrink calls.