

# Shrink Database and shrink file with Low Priority

Last updated by | Ricardo Marques | Feb 15, 2023 at 9:17 AM PST

---

## Contents

- [Shrink Database with Low Priority](#)
  - [What's new?](#)
  - [Permissions](#)
- [Using WAITATLOW\\_PRIORITY option with Shrink](#)
  - [T-SQL syntax for Shrink Database](#)
  - [T-SQL syntax for Shrink File](#)
  - [Examples](#)
- [Error Handling](#)
  - [1. ABORTAFTERWAIT = SELF on MAX\\_DURATION timeout.](#)
  - [2. ABORTAFTERWAIT = BLOCKERS on MAX\\_DURATION tim...](#)
- [Telemetry](#)
  - [1. Shrink WLP timeout failure](#)
  - [2. Cumulative Shrink WLP wait time for the shrink command](#)
- [References](#)

## Shrink Database with Low Priority

Shrink database and shrink file acquire a schema modify lock (Sch-M) when moving or deleting an Index Allocation Map (IAM) page. Queries acquire schema stability locks (Sch-S) which protect against schema modifications while they execute and keep this lock until they complete. These two are conflicting locks.

Queries holding Sch-S locks do not conflict with each other. However, a long running query holding a Sch-S lock blocks a subsequent Shrink operation. A query requiring a Sch-S lock that comes after the Shrink operation is queued up behind Shrink and is processed in order of first come first serve (FCFS) by the SQL lock manager. This results in the query indirectly being blocked by the long running query and significantly reduces the query performance of the database engine which is not a desirable scenario.

This is the current behavior on SQL Server 2019 and previous versions of SQL Server, where the shrink database command can lead to blocking and lock waits especially with active maintenance such as managing indexes and on any busy active SQL environment.

As a result, customers have to wait and schedule their shrink database and data file maintenance operations to after-hours as any long running reader operations essentially made it impossible to shrink data files during regular business hours.

## What's new?

Shrink using "Wait at Low Priority" (WLP) mode helps improve latency and performance of the execution of concurrent queries. When shrink is executed in WLP mode, new queries requiring Sch-S locks are not blocked by the waiting shrink operation. They jump the queue and continue execution as Sch-S locks are non-conflicting with other Sch-S locks. Shrink will eventually timeout and abort if it does not get the Sch-M lock due to the large number of queries or long running queries holding Sch-S lock.

Using shrink database wait at low priority does not affect customer workloads.


The only expense is delaying the database shrink operation(s).

## Permissions

Requires membership in the sysadmin fixed server role or the db\_owner fixed database role.

The BLOCKERS option requires the login to have ALTER ANY CONNECTION permission.

## Using WAIT\_AT\_LOW\_PRIORITY option with Shrink

WAIT\_AT\_LOW\_PRIORITY indicates that the shrink operation will wait for low priority locks, allowing other operations to proceed while the shrink operation is waiting. This feature is similar to the [WAIT\\_AT\\_LOW\\_PRIORITY with online index operations](#) , with some differences.

- You cannot modify MAX\_DURATION. The wait duration is 60000 milliseconds (1 minute).
- You cannot specify ABORT\_AFTER\_WAIT option NONE.

### MAX\_DURATION = 1 [MINUTES]

This is parameter cannot be changed. The default default is 1 minute and cannot be increased or decreased.


It is the wait time (an integer value specified in minutes) that the shrink operation Sch-M lock request will wait with low priority when executing the command. If the operation is blocked for the duration, the specified ABORT\_AFTER\_WAIT action will be executed. MAX\_DURATION time is always in minutes, and the word MINUTES can be omitted.

### ABORT\_AFTER\_WAIT = [SELF | BLOCKERS } ]

This is an optional parameter. The default value when not specified = SELF

- SELF - Exit the shrink operation currently being executed without rolling back completed changes. It throws an error message.
- BLOCKERS - Kill all user transactions that block the shrink operation so that the operation can continue. The BLOCKERS option requires the login to have ALTER ANY CONNECTION permission.

## T-SQL syntax for Shrink Database

See the public documentation on [DBCC SHRINKDATABASE](#) .

```

DBCC SHRINKDATABASE
( database_name | database_id | 0
  [ , target_percent ]
  [ , { NOTRUNCATE | TRUNCATEONLY } ]
)
[ WITH
  NO_INFOMSGS ,
  {
    [ WAIT_AT_LOW_PRIORITY
      [ (
        <wait_at_low_priority_option_list>
      )]
    ]
  }
]

< wait_at_low_priority_option > ::=
MAX_DURATION = { 'timeout' } [ MINUTES ]
| , ABORT_AFTER_WAIT = { SELF | BLOCKERS }

```

## T-SQL syntax for Shrink File

```

DBCC SHRINKFILE
(
  { file_name | file_id }
  { [ , EMPTYFILE ]
  | [ [ , target_size ] [ , { NOTRUNCATE | TRUNCATEONLY } ] ]
}
)
[ WITH
  NO_INFOMSGS ,
  {
    [ WAIT_AT_LOW_PRIORITY
      [ (
        <wait_at_low_priority_option_list>
      )]
    ]
  }
]

< wait_at_low_priority_option > ::=
MAX_DURATION = { 'timeout' } [ MINUTES ]
| , ABORT_AFTER_WAIT = { SELF | BLOCKERS }

```

## Examples

- Shrink a database with WAIT\_AT\_LOW\_PRIORITY

The following example attempts to reduce the size of the data and log files in the AdventureWorks2022 database to allow for 20% free space in the database. If a lock cannot be obtained within one minute, the shrink operation will abort.

```
DBCC SHRINKDATABASE ([AdventureWorks2022], 20) WITH WAIT_AT_LOW_PRIORITY (ABORT_AFTER_WAIT = SELF);
```

- Shrink a database file with WAIT\_AT\_LOW\_PRIORITY

The following example attempts to shrink the size of a data file in the current user database to 1 MB. The sys.database\_files catalog view is queried to obtain the file\_id of the data file, in this example, file\_id 5. If a lock

cannot be obtained within one minute, the shrink operation will abort.

```
USE AdventureWorks2022;
GO

SELECT file_id, name
FROM sys.database_files;
GO

DBCC SHRINKFILE (5, 1) WITH WAIT_AT_LOW_PRIORITY (ABORT_AFTER_WAIT = SELF)
```

## Error Handling

### 1. ABORT\_AFTER\_WAIT = SELF on MAX\_DURATION timeout.

#### Example:

```
DBCC SHRINKDATABASE(0) WITH WAIT_AT_LOW_PRIORITY (MAX_DURATION = 1, ABORT_AFTER_WAIT = SELF)
```

Msg 49516, Level 16, State 1, Line 134

Shrink timeout waiting to acquire schema modify lock in WLP mode to process IAM pageID 1:2865 on database ID 5



In this example, the error is thrown and the Shrink operation is aborted.

Error: 49516

Cause: User tried to shrink when there is excessive read, data manipulation or other activities which require a schema shared lock.

Action: Reduce read activity and retry shrink.

### 2. ABORT\_AFTER\_WAIT = BLOCKERS on MAX\_DURATION timeout.

#### Example:

```
DBCC SHRINKDATABASE(0) WITH WAIT_AT_LOW_PRIORITY (MAX_DURATION = 1, ABORT_AFTER_WAIT = BLOCKERS)
```

In this example, the error is thrown and the blocking operation is aborted.

A connection that blocked above command because it was holding a conflicting schema lock will be terminated with the following error:

Msg 1219, Level 16, State 1, Line 8

Your session has been disconnected because of a high priority DDL operation.

There is also the following message in the errorlog:

```
spid67      Process ID 66 was killed by an ABORT_AFTER_WAIT = BLOCKERS DDL statement on database_id = 5, objec
```



Error: 1219

Cause: The session was killed because it was blocking an ABORT = BLOCKERS lock request.

Action: Reconnect.

## Telemetry

### 1. Shrink WLP timeout failure

This data is captured in MonSqlShrinkInfo kusto view. For example:

```
MonSqlShrinkInfo
| where AppName contains {AppName}
| where TIMESTAMP >= startTime and TIMESTAMP <= endTime
| where event contains "shrink_file_telemetry"
| where tag contains "shrink_failure_reasons"
| where failure_reason contains "shrink timeout waiting to acquire schema modify lock in WLP mode"
```

### 2. Cumulative Shrink WLP wait time for the shrink command

```
MonSqlShrinkInfo
| where AppName contains {AppName}
| where TIMESTAMP >= startTime and TIMESTAMP <= endTime
| where event contains "shrink_file_telemetry"
| where status contains "shrink_wlp_wait_time"
```

## References

[DBCC SHRINKDATABASE](#) 

[DBCC SHRINKFILE](#) 

**How good have you found this content?**

