

15.6.3.7 Disabling Tablespace Path Validation

At startup, [InnoDB](#) scans directories defined by the [innodb_directories](#) variable for tablespace files. The paths of discovered tablespace files are validated against the paths recorded in the data dictionary. If the paths do not match, the paths in the data dictionary are updated.

The [innodb_validate_tablespace_paths](#) variable, introduced in MySQL 8.0.21, permits disabling tablespace path validation. This feature is intended for environments where tablespaces files are not moved. Disabling path validation improves startup time on systems with a large number of tablespace files. If [log_error_verbosity](#) is set to 3, the following message is printed at startup when tablespace path validation is disabled:

```
[InnoDB] Skipping InnoDB tablespace path validation.
Manually moved tablespace files will not be detected!
```



Warning

Starting the server with tablespace path validation disabled after moving tablespace files can lead to undefined behavior.

15.6.3.8 Optimizing Tablespace Space Allocation on Linux

As of MySQL 8.0.22, you can optimize how [InnoDB](#) allocates space to file-per-table and general tablespaces on Linux. By default, when additional space is required, [InnoDB](#) allocates pages to the tablespace and physically writes NULLs to those pages. This behavior can affect performance if new pages are allocated frequently. As of MySQL 8.0.22, you can disable [innodb_extend_and_initialize](#) on Linux systems to avoid physically writing NULLs to newly allocated tablespace pages. When [innodb_extend_and_initialize](#) is disabled, space is allocated to tablespace files using [posix_fallocate\(\)](#) calls, which reserve space without physically writing NULLs.

When pages are allocated using [posix_fallocate\(\)](#) calls, the extension size is small by default and pages are often allocated only a few at a time, which can cause fragmentation and increase random I/O. To avoid this issue, increase the tablespace extension size when enabling [posix_fallocate\(\)](#) calls. Tablespace extension size can be increased up to 4GB using the [AUTOEXTEND_SIZE](#) option. For more information, see [Section 15.6.3.9, “Tablespace AUTOEXTEND_SIZE Configuration”](#).

[InnoDB](#) writes a redo log record before allocating a new tablespace page. If a page allocation operation is interrupted, the operation is replayed from the redo log record during recovery. (A page allocation operation replayed from a redo log record physically writes NULLs to the newly allocated page.) A redo log record is written before allocating a page regardless of the [innodb_extend_and_initialize](#) setting.

On non-Linux systems and Windows, [InnoDB](#) allocates new pages to the tablespace and physically writes NULLs to those pages, which is the default behavior. Attempting to disable [innodb_extend_and_initialize](#) on those systems returns the following error:

```
Changing innodb_extend_and_initialize not supported on this platform. Falling
back to the default.
```

15.6.3.9 Tablespace AUTOEXTEND_SIZE Configuration

By default, when a file-per-table or general tablespace requires additional space, the tablespace is extended incrementally according to the following rules:

- If the tablespace is less than an extent in size, it is extended one page at a time.
- If the tablespace is greater than 1 extent but smaller than 32 extents in size, it is extended one extent at a time.