

SET_VAR Hint Applies	No
Type	Enumeration
Default Value	crc32
Valid Values	innodb crc32 none strict_innodb strict_crc32 strict_none

Specifies how to generate and verify the [checksum](#) stored in the disk blocks of [InnoDB tablespaces](#). The default value for [innodb_checksum_algorithm](#) is [crc32](#).

Versions of [MySQL Enterprise Backup](#) up to 3.8.0 do not support backing up tablespaces that use CRC32 checksums. [MySQL Enterprise Backup](#) adds CRC32 checksum support in 3.8.1, with some limitations. Refer to the [MySQL Enterprise Backup 3.8.1 Change History](#) for more information.

The value [innodb](#) is backward-compatible with earlier versions of MySQL. The value [crc32](#) uses an algorithm that is faster to compute the checksum for every modified block, and to check the checksums for each disk read. It scans blocks 32 bits at a time, which is faster than the [innodb](#) checksum algorithm, which scans blocks 8 bits at a time. The value [none](#) writes a constant value in the checksum field rather than computing a value based on the block data. The blocks in a tablespace can use a mix of old, new, and no checksum values, being updated gradually as the data is modified; once blocks in a tablespace are modified to use the [crc32](#) algorithm, the associated tables cannot be read by earlier versions of MySQL.

The strict form of a checksum algorithm reports an error if it encounters a valid but non-matching checksum value in a tablespace. It is recommended that you only use strict settings in a new instance, to set up tablespaces for the first time. Strict settings are somewhat faster, because they do not need to compute all checksum values during disk reads.

The following table shows the difference between the [none](#), [innodb](#), and [crc32](#) option values, and their strict counterparts. [none](#), [innodb](#), and [crc32](#) write the specified type of checksum value into each data block, but for compatibility accept other checksum values when verifying a block during a read operation. Strict settings also accept valid checksum values but print an error message when a valid non-matching checksum value is encountered. Using the strict form can make verification faster if all [InnoDB](#) data files in an instance are created under an identical [innodb_checksum_algorithm](#) value.

Table 15.26 Permitted [innodb_checksum_algorithm](#) Values

Value	Generated checksum (when writing)	Permitted checksums (when reading)
none	A constant number.	Any of the checksums generated by none , innodb , or crc32 .
innodb	A checksum calculated in software, using the original algorithm from InnoDB .	Any of the checksums generated by none , innodb , or crc32 .