

**NAME**

innochecksum – offline InnoDB file checksum utility

**SYNOPSIS**

**innochecksum** [*options*] *file\_name*

**DESCRIPTION**

**innochecksum** prints checksums for InnoDB files. This tool reads an InnoDB tablespace file, calculates the checksum for each page, compares the calculated checksum to the stored checksum, and reports mismatches, which indicate damaged pages. It was originally developed to speed up verifying the integrity of tablespace files after power outages but can also be used after file copies. Because checksum mismatches cause InnoDB to deliberately shut down a running server, it may be preferable to use this tool rather than waiting for an in-production server to encounter the damaged pages.

**innochecksum** cannot be used on tablespace files that the server already has open. For such files, you should use CHECK TABLE to check tables within the tablespace. Attempting to run **innochecksum** on a tablespace that the server already has open will result in an “Unable to lock file” error.

If checksum mismatches are found, you would normally restore the tablespace from backup or start the server and attempt to use **mysqldump** to make a backup of the tables within the tablespace.

Invoke **innochecksum** like this:

```
shell> innochecksum [options] file_name
```

**innochecksum** Options. PP **innochecksum** supports the following options. For options that refer to page numbers, the numbers are zero-based.

- **--help, -?**

Displays command line help. Example usage:

```
shell> innochecksum --help
```

- **--info, -I**

Synonym for **--help**. Displays command line help. Example usage:

```
shell> innochecksum --info
```

- **--version, -V**

Displays version information. Example usage:

```
shell> innochecksum --version
```

- **--verbose, -v**

Verbose mode; prints a progress indicator to the log file every five seconds. In order for the progress indicator to be printed, the log file must be specified using the **--log** option. To turn on verbose mode, run:

```
shell> innochecksum --verbose
```

To turn off verbose mode, run:

```
shell> innochecksum --verbose=FALSE
```

The **--verbose** option and **--log** option can be specified at the same time. For example:

```
shell> innochecksum --verbose --log=/var/lib/mysql/test/logtest.txt
```

To locate the progress indicator information in the log file, you can perform the following search:

```
shell> cat ./logtest.txt | grep -i "okay"
```

The progress indicator information in the log file appears similar to the following:

```
page 1663 okay: 2.863% done
page 8447 okay: 14.537% done
page 13695 okay: 23.568% done
page 18815 okay: 32.379% done
page 23039 okay: 39.648% done
page 28351 okay: 48.789% done
page 33023 okay: 56.828% done
page 37951 okay: 65.308% done
page 44095 okay: 75.881% done
page 49407 okay: 85.022% done
page 54463 okay: 93.722% done
...
```

- **--count, -c**

Print a count of the number of pages in the file and exit. Example usage:

```
shell> innochecksum --count ../data/test/tab1.ibd
```

- **--start-page=num, -s num**

Start at this page number. Example usage:

```
shell> innochecksum --start-page=600 ../data/test/tab1.ibd
```

or:

```
shell> innochecksum -s 600 ../data/test/tab1.ibd
```

- **--end-page=num, -e num**

End at this page number. Example usage:

```
shell> innochecksum --end-page=700 ../data/test/tab1.ibd
```

or:

```
shell> innochecksum --p 700 ../data/test/tab1.ibd
```

- **--page=num, -p num**

Check only this page number. Example usage:

```
shell> innochecksum --page=701 ../data/test/tab1.ibd
```

- **--strict-check, -C**

Specify a strict checksum algorithm. Options include innodb, crc32, and none.

In this example, the innodb checksum algorithm is specified:

```
shell> innochecksum --strict-check=innodb ../data/test/tab1.ibd
```

In this example, the crc32 checksum algorithm is specified:

```
shell> innochecksum -C crc32 ../data/test/tab1.ibd
```

The following conditions apply:

- If you do not specify the **--strict-check** option, **innochecksum** validates against innodb, crc32 and none.
- If you specify the none option, only checksums generated by none are allowed.
- If you specify the innodb option, only checksums generated by innodb are allowed.
- If you specify the crc32 option, only checksums generated by crc32 are allowed.
- **--no-check, -n**

Ignore the checksum verification when rewriting a checksum. This option may only be used with the **innochecksum --write** option. If the **--write** option is not specified, **innochecksum** will terminate.

In this example, an innodb checksum is rewritten to replace an invalid checksum:

```
shell> innochecksum --no-check --write innodb ../data/test/tab1.ibd
```

- **--allow-mismatches, -a**

The maximum number of checksum mismatches allowed before **innochecksum** terminates. The default setting is 0. If **--allow-mismatches=N**, where  $N \geq 0$ ,  $N$  mismatches are permitted and **innochecksum** terminates at  $N+1$ . When **--allow-mismatches** is set to 0, **innochecksum** terminates on the first checksum mismatch.

In this example, an existing innodb checksum is rewritten to set **--allow-mismatches** to 1.

```
shell> innochecksum --allow-mismatches=1 --write innodb ../data/test/tab1.ibd
```

With **--allow-mismatches** set to 1, if there is a mismatch at page 600 and another at page 700 on a file with 1000 pages, the checksum is updated for pages 0–599 and 601–699. Because **--allow-mismatches** is set to 1, the checksum tolerates the first mismatch and terminates on the second mismatch, leaving page 600 and pages 700–999 unchanged.

- **--write=name, -w num**

Rewrite a checksum. When rewriting an invalid checksum, the **--no-check** option must be used together with the **--write** option. The **--no-check** option tells **innochecksum** to ignore verification of the invalid checksum. You do not have to specify the **--no-check** option if the current checksum is valid.

An algorithm must be specified when using the **--write** option. Possible values for the **--write** option are:

- innodb: A checksum calculated in software, using the original algorithm from InnoDB.
- crc32: A checksum calculated using the crc32 algorithm, possibly done with a hardware assist.

- **none**: A constant number.

The **--write** option rewrites entire pages to disk. If the new checksum is identical to the existing checksum, the new checksum is not written to disk in order to minimize I/O.

**innochecksum** obtains an exclusive lock when the **--write** option is used.

In this example, a crc32 checksum is written for tab1.ibd:

```
shell> innochecksum -w crc32 ../data/test/tab1.ibd
```

In this example, a crc32 checksum is rewritten to replace an invalid crc32 checksum:

```
shell> innochecksum --no-check --write crc32 ../data/test/tab1.ibd
```

- **--page-type-summary, -S**

Display a count of each page type in a tablespace. Example usage:

```
shell> innochecksum --page-type-summary ../data/test/tab1.ibd
```

Sample output for **--page-type-summary**:

```
File:../data/test/tab1.ibd
=====PAGE TYPE SUMMARY=====
#PAGE_COUNT PAGE_TYPE
=====
    2    Index page
    0    Undo log page
    1    Inode page
    0    Insert buffer free list page
    2    Freshly allocated page
    1    Insert buffer bitmap
    0    System page
    0    Transaction system page
    1    File Space Header
    0    Extent descriptor page
    0    BLOB page
    0    Compressed BLOB page
    0    Other type of page
=====
Additional information:
Undo page type: 0 insert, 0 update, 0 other
Undo page state: 0 active, 0 cached, 0 to_free, 0 to_purge, 0 prepared, 0 other
```

- **--page-type-dump, -D**

Dump the page type information for each page in a tablespace to stderr or stdout. Example usage:

```
shell> innochecksum --page-type-dump=/tmp/a.txt ../data/test/tab1.ibd
```

- **--log, -l**

Log output for the **innochecksum** tool. A log file name must be provided. Log output contains checksum values for each tablespace page. For uncompressed tables, LSN values are also provided. The **--log** replaces the **--debug** option, which was available in earlier releases. Example usage:

```
shell> innochecksum --log=/tmp/log.txt ../data/test/tab1.ibd
```

or:

```
shell> innochecksum -l /tmp/log.txt ../data/test/tab1.ibd
```

- – option.

Specify the – option to read from standard input. If the – option is missing when “read from standard in” is expected, **innochecksum** will output **innochecksum** usage information indicating that the “–” option was omitted. Example usages:

```
shell> cat t1.ibd | innochecksum -
```

In this example, **innochecksum** writes the crc32 checksum algorithm to a.ibd without changing the original t1.ibd file.

```
shell> cat t1.ibd | innochecksum --write=crc32 - > a.ibd
```

Running **innochecksum** on Multiple User-defined Tablespace Files.PP The following examples demonstrate how to run **innochecksum** on multiple user-defined tablespace files (.ibd files).

Run **innochecksum** for all tablespace (.ibd) files in the “test” database:

```
shell> innochecksum ../data/test/*.ibd
```

Run **innochecksum** for all tablespace files (.ibd files) that have a file name starting with “t”:

```
shell> innochecksum ../data/test/t*.ibd
```

Run **innochecksum** for all tablespace files (.ibd files) in the data directory:

```
shell> innochecksum ../data/*/*.ibd
```

### Note

Running **innochecksum** on multiple user-defined tablespace files is not supported on Windows operating systems, as Windows shells such as **cmd.exe** do not support glob pattern expansion. On Windows systems, **innochecksum** must be run separately for each user-defined tablespace file. For example:

```
cmd> innochecksum.exe t1.ibd
cmd> innochecksum.exe t2.ibd
cmd> innochecksum.exe t3.ibd
```

Running **innochecksum** on Multiple System Tablespace Files.PP By default, there is only one InnoDB system tablespace file (ibdata1) but multiple files for the system tablespace can be defined using the `innodb_data_file_path` option. In the following example, three files for the system tablespace are defined using the `innodb_data_file_path` option: ibdata1, ibdata2, and ibdata3.

```
shell> ./bin/mysqld --no-defaults --innodb-data-file-path="ibdata1:10M;ibdata2:10M;ibdata3:10M:autoextend"
```

The three files (ibdata1, ibdata2, and ibdata3) form one logical system tablespace. To run **innochecksum** on multiple files that form one logical system tablespace, **innochecksum** requires the – option to read tablespace files from standard input, which is equivalent to concatenating multiple files to create one single file. For the example provided above, the following **innochecksum** command would be used:

```
shell> cat ibdata* | innochecksum -
```

Refer to the **innochecksum** options information for more information about the “–” option.

### Note

Running **innochecksum** on multiple files in the same tablespace is not supported on Windows operating systems, as Windows shells such as **cmd.exe** do not support glob pattern expansion. On Windows systems, **innochecksum** must be run separately for each system tablespace file. For example:

```
cmd> innochecksum.exe ibdata1
cmd> innochecksum.exe ibdata2
cmd> innochecksum.exe ibdata3
```

## COPYRIGHT

Copyright © 1997, 2019, Oracle and/or its affiliates. All rights reserved.

This documentation is free software; you can redistribute it and/or modify it only under the terms of the GNU General Public License as published by the Free Software Foundation; version 2 of the License.

This documentation is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with the program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA or see <http://www.gnu.org/licenses/>.

## SEE ALSO

For more information, please refer to the MySQL Reference Manual, which may already be installed locally and which is also available online at <http://dev.mysql.com/doc/>.

## AUTHOR

Oracle Corporation (<http://dev.mysql.com/>).