



Linux Ubuntu 26.04 Manual Pages on command 'mysqlimport.1'

\$ man mysqlimport.1

MYSQLIMPORT(1) MySQL Database System MYSQLIMPORT(1)

NAME

mysqlimport - a data import program

SYNOPSIS

mysqlimport [options] db_name textfile1 ...

DESCRIPTION

The mysqlimport client provides a command-line interface to the LOAD DATA SQL statement.

Most options to mysqlimport correspond directly to clauses of LOAD DATA syntax. See

Section 15.2.9, ?LOAD DATA Statement?.

Invoke mysqlimport like this:

```
mysqlimport [options] db_name textfile1 [textfile2 ...]
```

For each text file named on the command line, mysqlimport strips any extension from the file name and uses the result to determine the name of the table into which to import the file's contents. For example, files named patient.txt, patient.text, and patient all would be imported into a table named patient.

mysqlimport supports the following options, which can be specified on the command line or in the [mysqlimport] and [client] groups of an option file. For information about option files used by MySQL programs, see Section 6.2.2.2, ?Using Option Files?.

? --help, -?

```
????????????????????????????????????
```

? Command-Line Format ? --help ?

```
????????????????????????????????????
```

Display a help message and exit.

? --bind-address=ip_address

??

? Command-Line Format ? --bind-address=ip_address ?

??

On a computer having multiple network interfaces, use this option to select which interface to use for connecting to the MySQL server.

? --character-sets-dir=dir_name

??

? Command-Line Format ? --character-sets-dir=path ?

??

? Type ? String ?

??

? Default Value ? [none] ?

??

The directory where character sets are installed. See Section 12.15, "Character Set Configuration".

? --columns=column_list, -c column_list

??

? Command-Line Format ? --columns=column_list ?

??

This option takes a list of comma-separated column names as its value. The order of the column names indicates how to match data file columns with table columns.

? --compress, -C

??

? Command-Line Format ? --compress[={OFF|ON}] ?

??

? Deprecated ? Yes ?

??

? Type ? Boolean ?

??

? Default Value ? OFF ?

??

Compress all information sent between the client and the server if possible. See

Section 6.2.8, 'Connection Compression Control'.

This option is deprecated. Expect it to be removed in a future version of MySQL. See the section called 'Configuring Legacy Connection Compression'.

? --compression-algorithms=value

??

? Command-Line Format ? --compression-algorithms=value ?

??

? Type ? Set ?

??

? Default Value ? uncompressed ?

??

? Valid Values ? ?

? ? zlib ?

? ? ?

? ? zstd ?

? ? ?

? ? uncompressed ?

??

The permitted compression algorithms for connections to the server. The available algorithms are the same as for the `protocol_compression_algorithms` system variable. The default value is `uncompressed`.

For more information, see Section 6.2.8, 'Connection Compression Control'.

? --debug[=debug_options], -# [debug_options]

??

? Command-Line Format ? --debug[=debug_options] ?

??

? Type ? String ?

??

? Default Value ? d:t:o ?

??

Write a debugging log. A typical `debug_options` string is `d:t:o,file_name`. The default is `d:t:o`.

This option is available only if MySQL was built using `WITH_DEBUG`. MySQL release

binaries provided by Oracle are not built using this option.

? --debug-check

??

? Command-Line Format ? --debug-check ?

??

? Type ? Boolean ?

??

? Default Value ? FALSE ?

??

Print some debugging information when the program exits.

This option is available only if MySQL was built using WITH_DEBUG. MySQL release

binaries provided by Oracle are not built using this option.

? --debug-info

??

? Command-Line Format ? --debug-info ?

??

? Type ? Boolean ?

??

? Default Value ? FALSE ?

??

Print debugging information and memory and CPU usage statistics when the program exits.

This option is available only if MySQL was built using WITH_DEBUG. MySQL release

binaries provided by Oracle are not built using this option.

? --default-character-set=charset_name

??

? Command-Line Format ? --default-character- ?

? ? set=charset_name ?

??

? Type ? String ?

??

Use charset_name as the default character set. See Section 12.15, "Character Set Configuration".

? --default-auth=plugin

??

? Command-Line Format ? --default-auth=plugin ?

??

? Type ? String ?

??

A hint about which client-side authentication plugin to use. See Section 8.2.17,

?Pluggable Authentication?.

? --defaults-extra-file=file_name

??

? Command-Line Format ? --defaults-extra-file=file_name ?

??

? Type ? File name ?

??

Read this option file after the global option file but (on Unix) before the user option

file. If the file does not exist or is otherwise inaccessible, an error occurs. If

file_name is not an absolute path name, it is interpreted relative to the current

directory.

For additional information about this and other option-file options, see

Section 6.2.2.3, ?Command-Line Options that Affect Option-File Handling?.

? --defaults-file=file_name

??

? Command-Line Format ? --defaults-file=file_name ?

??

? Type ? File name ?

??

Use only the given option file. If the file does not exist or is otherwise inaccessible,

an error occurs. If file_name is not an absolute path name, it is interpreted relative

to the current directory.

Exception: Even with --defaults-file, client programs read .mylogin.cnf.

For additional information about this and other option-file options, see

Section 6.2.2.3, ?Command-Line Options that Affect Option-File Handling?.

? --defaults-group-suffix=str

??

? Command-Line Format ? --defaults-group-suffix=str ?

??

? Type ? String ?

??

Read not only the usual option groups, but also groups with the usual names and a suffix of str. For example, mysqlimport normally reads the [client] and [mysqlimport] groups. If this option is given as --defaults-group-suffix=_other, mysqlimport also reads the [client_other] and [mysqlimport_other] groups.

For additional information about this and other option-file options, see Section 6.2.2.3, ?Command-Line Options that Affect Option-File Handling?.

? --delete, -D

??

? Command-Line Format ? --delete ?

??

Empty the table before importing the text file.

? --enable-cleartext-plugin

??

? Command-Line Format ? --enable-cleartext-plugin ?

??

? Type ? Boolean ?

??

? Default Value ? FALSE ?

??

Enable the mysql_clear_password cleartext authentication plugin. (See Section 8.4.1.4, ?Client-Side Cleartext Pluggable Authentication?.)

? --fields-terminated-by=..., --fields-enclosed-by=...,

--fields-optionally-enclosed-by=..., --fields-escaped-by=...

??

? Command-Line Format ? --fields-terminated-by=string ?

??

? Type ? String ?

??

??

? Command-Line Format ? --fields-enclosed-by=string ?

??

? Type ? String ?

??

??

? Command-Line Format ? --fields-optionally-enclosed- ?

? ? by=string ?

??

? Type ? String ?

??

??

? Command-Line Format ? --fields-escaped-by ?

??

? Type ? String ?

??

These options have the same meaning as the corresponding clauses for LOAD DATA. See
Section 15.2.9, ?LOAD DATA Statement?.

? --force, -f

??

? Command-Line Format ? --force ?

??

Ignore errors. For example, if a table for a text file does not exist, continue
processing any remaining files. Without --force, mysqlimport exits if a table does not
exist.

? --get-server-public-key

??

? Command-Line Format ? --get-server-public-key ?

??

? Type ? Boolean ?

??

Request from the server the public key required for RSA key pair-based password
exchange. This option applies to clients that authenticate with the
caching_sha2_password authentication plugin. For that plugin, the server does not send

the public key unless requested. This option is ignored for accounts that do not authenticate with that plugin. It is also ignored if RSA-based password exchange is not used, as is the case when the client connects to the server using a secure connection.

If `--server-public-key-path=file_name` is given and specifies a valid public key file, it takes precedence over `--get-server-public-key`.

For information about the `caching_sha2_password` plugin, see Section 8.4.1.2, “Caching SHA-2 Pluggable Authentication”.

? `--host=host_name, -h host_name`

??

? Command-Line Format ? `--host=host_name` ?

??

? Type ? String ?

??

? Default Value ? localhost ?

??

Import data to the MySQL server on the given host. The default host is localhost.

? `--ignore, -i`

??

? Command-Line Format ? `--ignore` ?

??

See the description for the `--replace` option.

? `--ignore-lines=N`

??

? Command-Line Format ? `--ignore-lines=#` ?

??

? Type ? Numeric ?

??

Ignore the first N lines of the data file.

? `--lines-terminated-by=...`

??

? Command-Line Format ? `--lines-terminated-by=string` ?

??

? Type ? String ?

??

This option has the same meaning as the corresponding clause for LOAD DATA. For example, to import Windows files that have lines terminated with carriage return/linefeed pairs, use --lines-terminated-by="\r\n". (You might have to double the backslashes, depending on the escaping conventions of your command interpreter.) See Section 15.2.9, "LOAD DATA Statement".

? --local, -L

??

? Command-Line Format ? --local ?

????????????????????????????????????

? Type ? Boolean ?

????????????????????????????????????

? Default Value ? FALSE ?

????????????????????????????????????

By default, files are read by the server on the server host. With this option, mysqlimport reads input files locally on the client host.

Successful use of LOCAL load operations within mysqlimport also requires that the server permits local loading; see Section 8.1.6, "Security Considerations for LOAD DATA LOCAL".

? --lock-tables, -l

????????????????????????????????????

? Command-Line Format ? --lock-tables ?

????????????????????????????????????

Lock all tables for writing before processing any text files. This ensures that all tables are synchronized on the server.

? --login-path=name

????????????????????????????????????

? Command-Line Format ? --login-path=name ?

????????????????????????????????????

? Type ? String ?

????????????????????????????????????

Read options from the named login path in the .mylogin.cnf login path file. A "login path" is an option group containing options that specify which MySQL server to connect to and which account to authenticate as. To create or modify a login path file, use the

mysql_config_editor utility. See mysql_config_editor(1).

For additional information about this and other option-file options, see
Section 6.2.2.3, "Command-Line Options that Affect Option-File Handling".

? --no-login-paths

??

? Command-Line Format ? --no-login-paths ?

??

Skips reading options from the login path file.

See --login-path for related information.

For additional information about this and other option-file options, see
Section 6.2.2.3, "Command-Line Options that Affect Option-File Handling".

? --low-priority

??

? Command-Line Format ? --low-priority ?

??

Use LOW_PRIORITY when loading the table. This affects only storage engines that use only
table-level locking (such as MyISAM, MEMORY, and MERGE).

? --no-defaults

??

? Command-Line Format ? --no-defaults ?

??

Do not read any option files. If program startup fails due to reading unknown options
from an option file, --no-defaults can be used to prevent them from being read.

The exception is that the .mylogin.cnf file is read in all cases, if it exists. This
permits passwords to be specified in a safer way than on the command line even when
--no-defaults is used. To create .mylogin.cnf, use the mysql_config_editor utility. See
mysql_config_editor(1).

For additional information about this and other option-file options, see
Section 6.2.2.3, "Command-Line Options that Affect Option-File Handling".

? --password[=password], -p[password]

??

? Command-Line Format ? --password[=password] ?

??

? Type ? String ?

??

The password of the MySQL account used for connecting to the server. The password value is optional. If not given, mysqlimport prompts for one. If given, there must be no space between --password= or -p and the password following it. If no password option is specified, the default is to send no password.

Specifying a password on the command line should be considered insecure. To avoid giving the password on the command line, use an option file. See Section 8.1.2.1, ?End-User Guidelines for Password Security?.

To explicitly specify that there is no password and that mysqlimport should not prompt for one, use the --skip-password option.

? --password1[=pass_val] The password for multifactor authentication factor 1 of the MySQL account used for connecting to the server. The password value is optional. If not given, mysqlimport prompts for one. If given, there must be no space between --password1= and the password following it. If no password option is specified, the default is to send no password.

Specifying a password on the command line should be considered insecure. To avoid giving the password on the command line, use an option file. See Section 8.1.2.1, ?End-User Guidelines for Password Security?.

To explicitly specify that there is no password and that mysqlimport should not prompt for one, use the --skip-password1 option.

--password1 and --password are synonymous, as are --skip-password1 and --skip-password.

? --password2[=pass_val] The password for multifactor authentication factor 2 of the MySQL account used for connecting to the server. The semantics of this option are similar to the semantics for --password1; see the description of that option for details.

? --password3[=pass_val] The password for multifactor authentication factor 3 of the MySQL account used for connecting to the server. The semantics of this option are similar to the semantics for --password1; see the description of that option for details.

? --pipe, -W

??

? Command-Line Format ? --pipe ?

????????????????????????????????????

? Type ? String ?

????????????????????????????????

On Windows, connect to the server using a named pipe. This option applies only if the server was started with the `named_pipe` system variable enabled to support named-pipe connections. In addition, the user making the connection must be a member of the Windows group specified by the `named_pipe_full_access_group` system variable.

? --plugin-dir=dir_name

??

? Command-Line Format ? --plugin-dir=dir_name ?

??

? Type ? Directory name ?

??

The directory in which to look for plugins. Specify this option if the `--default-auth` option is used to specify an authentication plugin but `mysqlimport` does not find it. See Section 8.2.17, `?Pluggable Authentication?`.

? --port=port_num, -P port_num

??

? Command-Line Format ? --port=port_num ?

??

? Type ? Numeric ?

??

? Default Value ? 3306 ?

??

For TCP/IP connections, the port number to use.

? --print-defaults

??

? Command-Line Format ? --print-defaults ?

??

Print the program name and all options that it gets from option files.
For additional information about this and other option-file options, see Section 6.2.2.3, `?Command-Line Options that Affect Option-File Handling?`.

? --protocol={TCP|SOCKET|PIPE|MEMORY}

??

? Command-Line Format ? --protocol=type ?

??

? Type ? String ?

??

? Default Value ? [see text] ?

??

? Valid Values ? ?

? ? TCP ?

? ? ?

? ? SOCKET ?

? ? ?

? ? PIPE ?

? ? ?

? ? MEMORY ?

??

The transport protocol to use for connecting to the server. It is useful when the other connection parameters normally result in use of a protocol other than the one you want.

For details on the permissible values, see Section 6.2.7, ?Connection Transport Protocols?.

? --replace, -r

??

? Command-Line Format ? --replace ?

??

The --replace and --ignore options control handling of input rows that duplicate existing rows on unique key values. If you specify --replace, new rows replace existing rows that have the same unique key value. If you specify --ignore, input rows that duplicate an existing row on a unique key value are skipped. If you do not specify either option, an error occurs when a duplicate key value is found, and the rest of the text file is ignored.

? --server-public-key-path=file_name

??

? Command-Line Format ? --server-public-key- ?

? ? path=file_name ?

??

? Type ? File name ?

??

The path name to a file in PEM format containing a client-side copy of the public key required by the server for RSA key pair-based password exchange. This option applies to clients that authenticate with the sha256_password (deprecated) or caching_sha2_password authentication plugin. This option is ignored for accounts that do not authenticate with one of those plugins. It is also ignored if RSA-based password exchange is not used, as is the case when the client connects to the server using a secure connection.

If --server-public-key-path=file_name is given and specifies a valid public key file, it takes precedence over --get-server-public-key.

For sha256_password (deprecated), this option applies only if MySQL was built using OpenSSL.

For information about the sha256_password and caching_sha2_password plugins, see Section 8.4.1.3, "SHA-256 Pluggable Authentication", and Section 8.4.1.2, "Caching SHA-2 Pluggable Authentication".

? --shared-memory-base-name=name

??

? Command-Line Format ? --shared-memory-base-name=name ?

??

? Platform Specific ? Windows ?

??

On Windows, the shared-memory name to use for connections made using shared memory to a local server. The default value is MYSQL. The shared-memory name is case-sensitive.

This option applies only if the server was started with the shared_memory system variable enabled to support shared-memory connections.

? --silent, -s

??

? Command-Line Format ? --silent ?

??

Silent mode. Produce output only when errors occur.

? --socket=path, -S path

??

? Command-Line Format ? --socket={file_name|pipe_name} ?

??

? Type ? String ?

??

For connections to localhost, the Unix socket file to use, or, on Windows, the name of the named pipe to use.

On Windows, this option applies only if the server was started with the `named_pipe` system variable enabled to support named-pipe connections. In addition, the user making the connection must be a member of the Windows group specified by the `named_pipe_full_access_group` system variable.

? `--ssl*` Options that begin with `--ssl` specify whether to connect to the server using encryption and indicate where to find SSL keys and certificates. See the section called `?Command Options for Encrypted Connections?`.

? `--ssl-fips-mode={OFF|ON|STRICT}`

??

? Command-Line Format ? `--ssl-fips-mode={OFF|ON|STRICT}` ?

??

? Deprecated ? Yes ?

??

? Type ? Enumeration ?

??

? Default Value ? OFF ?

??

? Valid Values ? ?

? ? OFF ?

? ? ?

? ? ON ?

? ? ?

? ? STRICT ?

??

Controls whether to enable FIPS mode on the client side. The `--ssl-fips-mode` option differs from other `--ssl-xxx` options in that it is not used to establish encrypted connections, but rather to affect which cryptographic operations to permit. See Section 8.8, `?FIPS Support?`.

These --ssl-fips-mode values are permitted:

- ? OFF: Disable FIPS mode.
- ? ON: Enable FIPS mode.
- ? STRICT: Enable ?strict? FIPS mode.

Note

If the OpenSSL FIPS Object Module is not available, the only permitted value for --ssl-fips-mode is OFF. In this case, setting --ssl-fips-mode to ON or STRICT causes the client to produce a warning at startup and to operate in non-FIPS mode.

This option is deprecated. Expect it to be removed in a future version of MySQL.

? --tls-ciphersuites=ciphersuite_list

??

? Command-Line Format ? --tls- ?

? ? ciphersuites=ciphersuite_list ?

??

? Type ? String ?

??

The permissible ciphersuites for encrypted connections that use TLSv1.3. The value is a list of one or more colon-separated ciphersuite names. The ciphersuites that can be named for this option depend on the SSL library used to compile MySQL. For details, see Section 8.3.2, ?Encrypted Connection TLS Protocols and Ciphers?.

? --tls-sni-servername=server_name

??

? Command-Line Format ? --tls-sni-servername=server_name ?

??

? Type ? String ?

??

When specified, the name is passed to the libmysqlclient C API library using the MYSQL_OPT_TLS_SNI_SERVERNAME option of mysql_options(). The server name is not case-sensitive. To show which server name the client specified for the current session, if any, check the Tls_sni_server_name status variable.

Server Name Indication (SNI) is an extension to the TLS protocol (OpenSSL must be compiled using TLS extensions for this option to function). The MySQL implementation of SNI represents the client-side only.

? --tls-version=protocol_list

??

? Command-Line Format ? --tls-version=protocol_list ?

??

? Type ? String ?

??

? Default Value ? ?

? ? TLSv1,TLSv1.1,TLSv1.2,TLSv1.3 ?

? ? (OpenSSL 1.1.1 or ?

? ? higher) ?

? ? ?

? ? TLSv1,TLSv1.1,TLSv1.2 ?

? ? (otherwise) ?

??

The permissible TLS protocols for encrypted connections. The value is a list of one or more comma-separated protocol names. The protocols that can be named for this option depend on the SSL library used to compile MySQL. For details, see Section 8.3.2,

?Encrypted Connection TLS Protocols and Ciphers?.

? --user=user_name, -u user_name

??

? Command-Line Format ? --user=user_name, ?

??

? Type ? String ?

??

The user name of the MySQL account to use for connecting to the server.

? --use-threads=N

??

? Command-Line Format ? --use-threads=# ?

??

? Type ? Numeric ?

??

Load files in parallel using N threads.

? --verbose, -v

????????????????????????????????

? Command-Line Format ? --verbose ?

????????????????????????????????

Verbose mode. Print more information about what the program does.

? --version, -V

????????????????????????????????

? Command-Line Format ? --version ?

????????????????????????????????

Display version information and exit.

? --zstd-compression-level=level

????????????????????????????????

? Command-Line Format ? --zstd-compression-level=# ?

????????????????????????????????

? Type ? Integer ?

????????????????????????????????

The compression level to use for connections to the server that use the zstd compression algorithm. The permitted levels are from 1 to 22, with larger values indicating increasing levels of compression. The default zstd compression level is 3. The compression level setting has no effect on connections that do not use zstd compression.

For more information, see Section 6.2.8, ?Connection Compression Control?.

Here is a sample session that demonstrates use of mysqlimport:

```
$> mysql -e 'CREATE TABLE impptest(id INT, n VARCHAR(30))' test
```

```
$> ed
```

```
a
```

```
100   Max Sydow
```

```
101   Count Dracula
```

```
.
```

```
w impptest.txt
```

```
32
```

```
q
```

```
$> od -c impptest.txt
```

```
0000000  1 0 0 \t M a x   S y d o w \n 1 0
```

```
0000020  1 \t C o u n t   D r a c u l a \n
```

0000040

```
$> mysqlimport --local test imptest.txt
```

test.imptest: Records: 2 Deleted: 0 Skipped: 0 Warnings: 0

```
$> mysql -e 'SELECT * FROM imptest' test
```

```
+-----+-----+
| id  | n      |
+-----+-----+
| 100 | Max Sydow  |
| 101 | Count Dracula |
+-----+-----+
```

COPYRIGHT

Copyright ? 1997, 2026, Oracle and/or its affiliates.

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/>).

MySQL 8.4

03/23/2026

MYSQLIMPORT(1)