`--xml, -X`
This option exports results in an XML format.

mysqld Server

mysqld

`mysqld [options]`

When *mysqld* starts, various options can be used to alter the server's behavior. Although you don't need to know all of the server options available or use them—quite often the default settings are fine—as a database administrator, it's useful to know what options exist for various categories that may be related to your needs.

Options may be given at the command line when starting or restarting the server. However, it's common practice to enter them into a configuration file. On Unix-based systems, the main configuration file typically is */etc/my.cnf*. For Windows systems, the main file is usually either *c:\systems\my.ini* or *c:\my.conf*. Options are entered on separate lines and follow a *variable=value* format. Some options are binary and can be enabled by just including the option at the command line when starting the server or in the options file with no value (or an equals sign followed by no value).

Within the options file, options are grouped under headings contained within square brackets. The *mysqld* daemon reads options from the configuration file under the headings of `[mysqld]` and `[server]` as it's started. For more recent versions of the MySQL server, the group `[mysqld-5.0]` is also read. Groups are read in the order mentioned here, and the last setting for an option read is the one used. To get a list of options that *mysqld* is using on a particular server, enter the following line from the command line (results follow):

```
$ mysqld --print-defaults
/usr/libexec/mysqld would have been started with the following arguments:
--datadir=/data/mysql --socket=/var/lib/mysql/mysql.sock
--old-passwords=1
```

As the resulting message indicates, the `--print-defaults` options draws information from the options files and indicates the options and what their values would be if the MySQL server were restarted. However, if the options files were changed since MySQL was started, or if MySQL was started from the command line or with command-line options from a script on the server, this output will not reflect those options. Basically, the results of `--print-defaults` do not reflect the current settings, just the options it finds in the options files for the relevant server groups. To determine the current server options that have been used—other than the default options—while a server is running, you can enter the following command from a Unix system (sample results follow):

```
$ ps aux | grep mysql

mysql 27670 0.2 3.2 124252 17296 ? Sl Aug21 25:06
/usr/libexec/mysqld --defaults-file=/etc/my.cnf --basedir=/usr
--datadir=/data/mysql --user=mysql --pid-file=/var/run/mysqld/mysqld.pid
--skip-locking --socket=/var/lib/mysql/mysql.sock
```

If you see an option that you don't see in your default options file, it may be coming from a different options file. You may even be running a different installation of `mysqld` than you think. In such a situation, you would have to specify the path to the `mysqld` you want to use when starting the server.

In the following sections of this chapter, options are grouped by their use:

Location
: These options specify where the server can find files and directories it needs.

Security and connection
: These options are related to user and database security, limits on connections, and how clients connect to the server.

Global
: These options affect server behavior, and are stored in global variables.

Logs
: These options relate to server logs.

Performance optimization
: This section contains several options that could be included in other categories, but they are worth considering together because they can affect the speed of the database.

Replication
: These options are strictly related to replication.

Storage engine specific options
: These options concerning storage engines (formerly known as table types) are grouped into subsections based on the specific storage engines to which they relate.

Some options are listed in more than one section because they have more than one use relative to the sections listed.

The options are shown as they would be entered from the command line. If an option is used in a configuration file, the long form should be used and the double-dash prefix should be omitted. For example, `--basedir=/data/mysql` would be entered from the command line. However, in a configuration file the same option would read as `basedir=/data/mysql` on its own separate line.

The syntax for listing options is as follows:

`--option=value`
: An option that requires a value

`--option[=value]`
: An option that can take a value, but does not require one

`--option[=value]`
: A binary option that is to be given without a value

A few options have single-letter abbreviations, also called short forms. The short form is shown in parentheses after the long form.

As new versions of MySQL are released, more options may be added. To get a list for your version, type `mysqld --verbose --help` from the command line on the server host.

For many of the options, there is a system variable with the same name as the option, but without the leading double-dashes. For some options, the dashes within the name

will need to be changed to underscores (e.g., the variable associated with `--setting-example` would be `setting_example`). Before changing the value or the setting of a variable, it's often a good idea to see what the variable is set to. You can do this by entering a statement like this:

```
SHOW VARIABLES LIKE 'setting_example';
```

Location

Some `mysqld` options allow you to instruct MySQL where files are located and what network settings should be used when clients connect to it remotely. An alphabetical list of these options follows, along with the syntax and an explanation of each. This list does not include storage system specific options related to file paths. See the section for the particular storage engine's options later in this chapter:

`--basedir=`*path*, `-b` *path*
: If you've installed more than one version of MySQL on your server or if you have moved the binary files for MySQL, you will need to specify the base directory for the MySQL installation. This option is particularly necessary if you're using `mysqld_safe` to keep the *mysqld* daemon running; list this option under the `[mysqld_safe]` group heading.

`--character-sets-dir=`*path*
: This option specifies the absolute path to the directory containing character sets. By default, this directory is in the subdirectory *charsets* in the directory where MySQL is installed (e.g., */usr/share/mysql/charsets/*).

`--datadir=`*path*, `-h` *path*
: If you want to put your datafiles for MySQL (i.e., database directories and table files) in a different directory from the default, you need to use this option. This is useful especially if you want the data on a different hard drive. Within the directory that you name, MySQL will create subdirectories for each database. If you use this option, be sure that the *mysql* user on the filesystem has permissions to read and write to the directory. Generally, you would make it the owner of the directory.

`--init-file=`*filename*
: If you have a set of SQL commands that you must execute every time you restart the server, rather than enter them manually you could put them in a file and use this option to tell MySQL to execute them for you at startup. Each SQL statement in the file must be on a separate line. Unfortunately, you cannot include comments in the file. You could put them in a separate text file in the same directory, perhaps with a similar same filename (e.g., *init.sql* and *init.txt*).

`--secure-file-priv=`*path*
: Use this option to restrict the importing of files to the given path. This is related to the `SELECT...INTO OUTFILE` and `LOAD DATA` statements, as well as the `LOAD_FILE( )` function. This option is available as of version 5.1.17 of MySQL.

`--pid-file=`*filename*
: Instead of starting `mysqld` directly, the common method used lately is to start the script `mysqld_safe`. It will in turn start `mysqld` and make sure it keeps running. Thus, if `mysqld` crashes, `mysqld_safe` will automatically restart it. To keep track of the system process for `mysqld`, the `mysqld_safe` program will record the process identification number in a file called *mysqld.pid*. With this option, you can tell MySQL where to put that file.

`--plugin-dir=path`
: This option sets the directory where plugins on the server are placed. It's available as of version 5.1.2 of MySQL.

`--skip-symbolic-links`
: This option is used to disable symbolic links. The reverse is to enable them through `--symbolic-links`. Prior to version 4.0.3 of MySQL, this option was `--skip-symlink`.

`--slave-load-tmpdir=value`
: This option specifies the directory where a slave server stores temporary files when the `LOAD DATA INFILE` statement is executed.

`--slow-query-log-file=filename`
: See the "Performance optimization" section later in this chapter.

`--socket=filename`
: Socket files are used on Unix systems. With this option, you may specify the path and filename of the socket file. If you don't use this option, recent versions of MySQL place the socket file in the data directory of MySQL. On Windows systems, this option may be used to provide the pipe name (*MySQL* by default) for local connections. Just as with the `--port` option, the `--socket` option may be used for multiple instances of MySQL. You could issue one `mysqld_safe` command with the default socket file and another with an option such as `--socket=mysqld_test.sock` to indicate a test server. A second server that you assign to the same socket file will refuse to start because otherwise the daemons would conflict with each other. Incidentally, it's not necessary to specify a separate port and socket file, but most administrators do it all the same.

`--symbolic-links, -s`
: This option enables symbolic links at the filesystem level for database directories and table files. MySQL expects to find the files in its data directory, but if you want to store the data in other directories in order to find more space or spread reads and writes around, this option allows you to create links in the data directory that point to where the data actually is stored. On Windows systems, this allows you to create shortcuts to databases (e.g., `database.sym`). On Unix systems with MyISAM tables, this option allows you to specify a different directory for a table's location with the `DATA DIRECTORY` or `INDEX DIRECTORY` options of both the `ALTER TABLE` and `CREATE TABLE` SQL statements. When the table is renamed or deleted, the related files that are symbolically linked will be renamed or deleted, respectively.

`--sync-frm`
: This option instructs the server to synchronize the *.frm* files with the filesystem when a table is created. This slows down table creation slightly, but is more stable than leaving it in memory only.

`--temp-pool`
: This option instructs the server to utilize a small set of names for temporary file-naming rather than unique names for each file.

`--tmpdir=path, -t path`
: If you want to control where MySQL places its temporary files, specify this option. You can give multiple file paths in a colon-separated list. When you're using a storage engine such as InnoDB to create tablespaces over multiple files and you're working with huge tables of data that would exceed the filesystem limits, this option is useful for working around those limits. For instance, if you have a system with a

file or directory size limit of 4 MB, you can provide two directories with the `--tmpdir` option and thereby double your physical table limitations to 8 MB. The directories could even be on separate filesystems that your operating system mounts.

Security and connections

These `mysqld` server options relate to security, user-related settings, and the network connections clients make to the server:

`--allow-suspicious-udfs[={0|1}]`
: As of version 5.0.3 of MySQL, the server requires user-defined functions to be named with an acceptable suffix—*function_name*`_add( )`, *function_name*`_clear( )`, *function_name*`_deinit( )`, *function_name*`_init( )`, *function_name*`_reset( )`, etc.—and won't load functions that fail to adhere to that standard. However, you can disable that security protection by giving this option a value of 0. A value of 1 enables it and is the default.

`--automatic-sp-privileges[={0|1}]`
: By default, this option is set to 1 and therefore gives users the `ALTER ROUTINE` and the `EXECUTE` privileges for any stored routine that the user has created, as long as the user and those routines exist. If you set this option to 0, the user does not get those privileges and therefore cannot alter or execute routines. However, you can explicitly grant users those privileges, as with other MySQL privileges.

`--back-log=`*value*
: When the primary thread of the MySQL server gets many connection requests simultaneously, they are backlogged while the server begins new threads. Use this option to set the number of connections that may be backed up. The number cannot exceed the system value for TCP/IP connections related to the `listen( )` system function.

`--bind-address=`*address*
: This option specifies the IP address the server binds to. It's used to restrict network access on a host with multiple IP addresses.

`--bootstrap`
: This option isn't normally used by administrators. It's used by the `mysql_install_db` script to create the necessary privileges tables without the *mysqld* daemon running.

`--character-set-client-handshake`
: Use this option at the command line only (not available in the options file) to instruct the server not to ignore strange characters that it receives (perhaps due to a character set mismatch) from the client. Use `--skip-character-set-client-handshake` to disable this option because it's set by default.

`--chroot=`*path*
: This option runs the daemon with `chroot( )` from the filesystem so as to start it in a closed environment for additional security. This is a recommended security measure.

`--connect-timeout=`*value*
: This option may be used to change the number of seconds that the server should wait for a connection packet before terminating the connection and returning *Bad Handshake*. As of version 5.1.23, the related variable is set to five seconds by default.

If clients display messages saying that they lost the connection to the server, you might try increasing this value.

`--des-key-file=`*`filename`*

This option instructs the server to obtain the default keys from the given file when the MySQL functions `DES_ENCRYPT( )` or `DES_DECRYPT( )` are used.

`--enable-named-pipe`

This option enables support for named pipe connections with the `mysqld-nt` and `mysqld-max-nt` servers, which support them. It's used only with Windows NT, 2000, XP, and 2003 systems; do not use it on non-Windows systems (e.g., Linux or Mac OS X). Use the `--socket` option with this one to specify the path and name of the pipe.

`--init-connect='`*`string`*`'`

This option specifies one or more SQL statements, all combined in a single *`string`*, that are to be executed each time a client connects to the server. It will not allow SQL statements to be executed for users with the `SUPER` privilege.

`--init-file=`*`filename`*

This option indicates a file containing SQL statements that are to be executed when the server is started. This option will not work if the `--disable-grant-options` option is enabled. SQL statements need to be on separate lines, and comments are not permitted in the file.

`--interactive-timeout=`*`value`*

For interactive clients (clients using `mysql_real_connect( )` with the `CLIENT_INTERACTIVE` flag), this option sets the number of seconds of inactivity allowed before closing the connection.

`--local-infile[={0|1}]`

The SQL statement `LOAD DATA INFILE` can import data from a file on either the server's host or the client's host. By adding the `LOCAL` option, the client instructs the server to import locally from the client machine. This has the potential to be a security problem, though, because the file being loaded could have malicious code. Therefore, some administrators for public servers want to prevent clients from being able to import files local to the client, while still allowing them to import files located on the server. Use this option and set it to 0 to disable importing files local to the client. By default this is set to 1.

`--max-allowed-packet=`*`value`*

See the "Performance optimization" section later in this chapter.

`--max-connect-errors=`*`value`*

If the client has problems connecting and the number of attempts exceeds the value of the MySQL variable `max_connect_errors` (10 by default), the host address for the client will be blocked from further attempts. Use this option to change that value of that variable. To reset blocked hosts, run the `FLUSH HOSTS` statement on the server.

`--max-connections=`*`value`*

Clients are not permitted to have more connections than the number specified by the variable `max_connections`. By default it's either 100 or 150, depending on your version. Use this option to change that value.

`--max-user-connections=value`
: This option limits the number of connections per user account. Set the value to 0 to disable the limit and thereby allow a single user to create as many connections as MySQL and the operating system allow.

`--net-buffer-length=value`
: Memory is allocated by MySQL for each thread's connection and results. The amount initially allocated for each of these buffers is controlled by the variable `net_buffer_length`. You can use this option to change the value, but you normally shouldn't. Each buffer can expand as needed until it reaches the limit specified in `max_allowed_packet`, but when each thread finishes its work, the buffers contract again to their initial sizes.

`--net-read-timeout=value`
: This option sets the number of seconds the server will wait for a response from the client while reading from it before terminating the connection. Use `--net-write-timeout` to set the amount of time the server should wait when writing to a client before terminating. The timeouts apply only to TCP/IP connections and not to connections made through a socket file, a named pipe, or shared memory.

`--net-retry-count=value`
: If the connection to the client is interrupted while the server is reading, the server will try to reestablish the connection a number of times. That number can be set with this option.

`--net-write-timeout=value`
: This option sets the number of seconds the server will wait for a response from the client while writing to it before terminating the connection. Use `--net-read-timeout` to set the amount of time the server should wait when reading from a client before terminating. The timeouts apply only to TCP/IP connections and not to connections made through a socket file, a named pipe, or shared memory.

`--old-passwords`
: This option permits clients to continue to use passwords that were created before version 4.1 of MySQL, along with the old, less secure encryption method in use in earlier versions.

`--old-protocol, -o`
: This option has the server use version 3.20 protocol of MySQL for compatibility with older clients.

`--old-style-user-limits`
: Prior to version 5.0.3 of MySQL, user resource limits were based on each combination of user and host. Since then, user resources are counted based on the user regardless of the host. To continue to count resources based on the old method, use this option.

`--one-thread`
: This option instructs the server to run only one thread, which is needed when debugging a Linux system using older versions of the *gdb* debugger.

`--port=port, -P port`
: This option specifies the port on which the server will listen for client connections. By default, MySQL uses port 3306. However, if you want to use a separate port, you may specify one with this option. This feature can be useful if you are running more than one instance of MySQL on your server. For example, you might use port 3306

for your regular MySQL server and port 3307 for a particular department's databases, as well as 3308 for testing a new version of MySQL.

`--port-open-timeout=value`
: As of version 5.1.5 of MySQL, this option may be used to set the number of seconds the server should wait for a TCP/IP port to become available. This usually comes into play when the server has been restarted.

`--safe-show-database`
: This option hides database names that a user does not have permission to access.

`--safe-user-create`
: This option prevents a user from creating new users without the `INSERT` privilege for the user table in the `mysql` database.

`--secure`
: This option enables reverse host lookup of IP addresses, which provides some defense against spoofing domain names but adds overhead to each remote connection.

`--secure-auth`
: This option prevents authentication of users with passwords created prior to version 4.1 of MySQL.

`--secure-file-priv=path`
: See the "Location" section earlier in this chapter.

`--skip-automatic-sp-privileges`
: This option disables the `--automatic-sp-privileges` option, which is related to users automatically being granted `ALTER ROUTINE` and `EXECUTE` privileges on stored procedures that they create.

`--skip-character-set-client-handshake`
: This option disables the `--character-set-client-handshake` option.

`--skip-grant-tables`
: This option instructs the server not to use the grants table and thus give all users full access. This option presents a security risk. It may be used if the *root* password is lost so that you may log in without it and then reset the password. Restart the server without this option or run the `FLUSH PRIVILEGES` statement from the monitor to reenable privileges.

`--skip-host-cache`
: This option disables the use of the internal host cache, which requires a DNS lookup for each new connection.

`--skip-name-resolve`
: This option requires a client's IP address to be named in the privileges tables for tighter security and faster connections.

`--skip-networking`
: This option prevents network connections of clients and allows only local connections.

`--skip-show-database`
: This option prevents the `SHOW DATABASES` statement from being executed by users without the specific privilege.

`--skip-ssl`
: This option specifies that an SSL connection should not be used, if SSL is enabled by default.

`--ssl`
: This option specifies the use of SSL-protected connections. It requires the server to be SSL-enabled. If this option is enabled on the utility by default, use `--skip-ssl` to disable it.

`--ssl-ca=`*`pem_file`*
: This option specifies the file (i.e., the *pem* file) that provides a list of trusted SSL CAs.

`--ssl-capath=`*`path`*
: This option specifies a directory of files that provide trusted SSL certificates (i.e., *pem* files).

`--ssl-cert=`*`filename`*
: This option specifies the SSL certificate file for SSL connections.

`--ssl-cipher=`*`ciphers`*
: This option gives a list of ciphers that may be used for SSL encryption.

`--ssl-key=`*`filename`*
: This option specifies the SSL key file for secure connections.

`--ssl-verify-server-cert`
: This option has the client verify its certificate with the server during an SSL connection. It is available as of version 5.1.11 of MySQL.

`--standalone`
: If MySQL is running Windows NT, this option instructs the server not to run as a service.

`--thread-handling={one-thread|one-thread-per-connection}`
: This option specifies the thread handling model that the server is to use. The `one-thread` option is basically used for debugging; `one-thread-per-connection` is the default. This option is available as of version 5.1.17 of MySQL.

`--user=`*`user`*`, -u `*`user`*
: This option instructs the client to access MySQL under a username different from the current system user.

Global

Following is a list of global server options related to the server's behavior:

`--ansi, -a`
: This option instructs the server to use standard American National Standards Institute (ANSI) SQL syntax instead of MySQL syntax.

`--auto-increment-increment[=`*`value`*`]`
: This option and the `--auto-increment-offset` option are used when replicating a master to a master server. They determine the amount by which an `AUTO_INCREMENT` column is increased with each new row inserted into any table in the system. By default, the variable associated with this option is set to 1. Each can be set to a value from 1 to 65535. If either option is set to 0, they both will be set back to 1. If either is set to a non-integer value, it will remain unchanged. If either is set to a negative value or a value in excess of 65535, they both will then be set to 65535. Don't use these options with MySQL Cluster, as they cause problems.

`--auto-increment-offset[=value]`
: This option sets the starting number for `AUTO_INCREMENT` columns on all tables on the server. Each successive row inserted into tables will be incremented by the value of the `auto-increment-increment` system variable. If that variable is set to a number lower than the value set by this option, the value of the `auto-increment-offset` system variable (set by this option) will be ignored. See the description of the `--auto-increment-increment` option previously for more restrictions on this option.

`--character-set-server=set, -C`
: This option makes the server use a particular character set by default for its calculations. It's available as of version 4.1.3 of MySQL.

`--character-set-filesystem=value`
: This option specifies the character set that the filesystem uses. It was added in version 5.1.6 of MySQL.

`--completion-type=[=0|1|2]`
: The SQL statements `COMMIT` and `ROLLBACK` support an optional `AND CHAIN` parameter that automatically begins a new transaction at the same isolation level after the end of the transaction completed by these statements. If this option is set to 1, this chaining effect will be the default setting for those SQL statements. Similarly, if this option is set to 2, the default setting for the statements will be `RELEASE`, which causes the server to disconnect after each transaction is terminated. A value of 0, which is the default, does nothing.

`--console`
: On Windows systems, this option has the server display error messages to `stdout` and `std.err` even if `--log-error` is enabled.

`--core-file`
: This option instructs the server to create a core file if the daemon dies. Some systems require the `--skip-stack-trace` option to be set as well. Some systems also require the `--core-file-size` option when using `mysqld_safe`. On Solaris systems, if the `--user` option is used also, the server will not create the core file.

`--date-format=value`
: The variable associated with this option is not yet implemented. It's expected to be used to set the default date format for the MySQL server.

`--datetime-format=value`
: The variable associated with this option is not yet implemented. It's expected to be used to set the default datetime format for the MySQL server.

`--default-week-format=value`
: The variable associated with this option is not yet implemented. It's expected to be used to set the default format for the days of the week on the MySQL server.

`--debug[=options], -# options]`
: This option is used to get a trace file of the daemon's activities. The debug options are typically `d:t:o,filename`. See Table 16-1 at the end of the list of options for the `mysqldump` utility later in this chapter for an explanation of these flags and others that may be used. MySQL has to be compiled for debugging using the `--with-debug` option when configuring.

`--default-character-set=character_set`
: This option is used to specify the default character set. This option is deprecated as of version 4.1.3 of MySQL. Use the `--character-set-server` option instead.

`--default-collation=collation`
This option specifies the collation to use as the default. This option is deprecated as of version 4.1.3 of MySQL. Use the `--collation-server` option instead.

`--default-time-zone=zone`
This option specifies the default time zone for the server. The filesystem time zone is used by default.

`--div-precision-increment=value`
This option sets the number of decimal places to show in the results of dividing numbers. The variable associated with this option (`div_precision_increment`) has a default value of 4. You can set it from 0 to 30.

`--enable-pstack`
This option instructs the server to print a symbolic stack trace if the server fails and exits.

`--exit-info[=flags], -T [flags]`
This option displays debugging information when the server exits.

`--external-locking`
This option allows system locking. Be careful when using it on a platform with problems with `lockd`, such as Linux, because the *mysqld* daemon may deadlock and require rebooting the server to unlock it. This option was previously called `--enable-locking`.

`--flush`
This option flushes all changes to disk after each SQL statement instead of waiting for the filesystem to do the writes at regular intervals.

`--flush-time=seconds`
This option sets the `flush_time` variable, which specifies the number of seconds a table can remain open before it's closed and flushed to free resources and to synchronize data. For current operating systems, this option shouldn't be used because it will slow the server. A value of 0 disables it and is the default.

`--gdb`
This option is recommended when debugging the MySQL daemon. It enables a handler for `SIGINT`, which is necessary for the server daemon to be stopped with Ctrl-C at debugging breakpoints. It also disables core file handling as well as stack tracing.

`--group-concat-max-len=value`
This option sets the maximum length of a value created by the `GROUP_CONCAT( )` function.

`--language=[language|pathname]`
This option specifies the language the daemon should use to display messages. It can be the name of a language or a pathname to the language file.

`--lower-case-table-names[=0|1|2]`
If this option is set to 1, database and table names will be saved in lowercase letters on the server, and MySQL will not consider case when given database and table names. A value of 2 causes databases and tables to be stored on the filesystem in filenames with uppercase and lowercase based on what it is given when they are created. However, they will be treated as lowercase. A value of 0 disables these

features, but you shouldn't set it to 0 if using a case-insensitive filesystem, such as Windows.

`--max-error-count=value`
: When errors, warnings, and notes are generated, they are stored by the server to be displayed when the `SHOW ERRORS` or `SHOW WARNINGS` statements are executed. This option limits the number of messages that will be stored. The default value is 64.

`--max-join-size=value`
: This option sets the maximum number of rows in a join. By default, this option is set very high. You may want to lower it if you suspect abuse from users. To reset it to the default value, enter a value of `DEFAULT`. If you set this option to any other value, it causes the system variable `SQL_BIG_SELECTS` to be set to 0. If the `SQL_BIG_SELECTS` variable is then set to another value, this option's setting is ignored.

`--max_length_for_sort_data=value`
: This option sets the maximum size of data that can be sorted with the `ORDER BY` clause.

`--max_prepared_stmt_count=value`
: This option sets the maximum number of prepared statements allowed on the server. Values from 0 to 1000000 (one million) are accepted; the default is 16382. If you set the value lower than the current number of prepared statements, existing ones will be unaffected. But when they are removed, new ones cannot be added until the total count falls below the value given with this option. This option is available as of version 5.1.10 of MySQL.

`--new, -n`
: At the time of this writing, this option is used to test queries before upgrading from version 4.0 to 4.1.

`--open_files_limit=value`
: This option specifies the maximum number of files the daemon can keep open, which may require it to close tables more often than is optimal.

`--help, -?`
: This option displays basic help information. It displays more information when combined with the `--verbose` option.

`--read_only`
: If this option is used, users cannot add, change, or delete data on the server, unless they have `SUPER` privileges. The other exception is that updates from slave threads are allowed. This option does not carry to the slaves. It can be set on slaves independently from the master and may be useful to keep slaves synchronized properly.

`--safe-mode=value`
: This option disables some optimizations at startup.

`--set-variable variable = value, -O variable = value`
: This option sets a server variable. Enter `mysqld --verbose --help` to see the current values for particular server variables.

`--skip-external-locking`
: Previously called `--skip-locking`, this option prevents system locking.

`--skip-locking`
: This option disables system locking of the server.

`--skip-new`
: This option instructs the server not to use new options—i.e., options that are enabled by default but are still in beta testing mode.

`--sql-mode=`*value*
: This option covers a number of possible ways of interpreting SQL statements, mostly for compatibility with other database engines. Multiple values may be given in a comma-separated list.

`--sql_auto_is_null={0|1}`
: If you enable this option by setting it to 1, you can give the name of a column that uses `AUTO_INCREMENT` in `WHERE` clauses with a condition of NULL to find the last inserted row. For example, `SELECT...WHERE client_id IS NULL;` will return the row that was last inserted into a table where `client_id` is the primary key. A value of 0 for this option will disable it. The option is useful when interfacing with ODBC applications (e.g., MS Access).

`--sql_big_selects={0|1}`
: Disable (set to 0) this option to prevent large `SELECT` statements from being executed. Large statements are defined as joins whose results would exceed the maximum number of rows set by the `--max_join_size` option. The default value of 1 enables large SQL statements. Setting the `--max_join_size` option to something other than `DEFAULT` will reset this option back to 0.

`--sql_buffer_result={0|1}`
: If this option is set to 1, the results of `SELECT` statements will be sent to a buffer before being returned to the client. This slows the results, but unlocks the associated tables faster for the use of other clients. The default setting of 0 disables this option.

`--sql-safe-updates={0|1}`
: This option, when set to 1, is useful in helping to prevent inadvertent deletion of multiple and possibly all rows in a table. It requires that `DELETE` and `UPDATE` statements contain a `WHERE` clause with a key column and value. The default value of 0 disables the option.

`--sql_select_limit={`*value*`|DEFAULT}`
: This option limits the number of rows returned from a `SELECT` statement when the `LIMIT` clause hasn't been given. The value of `DEFAULT` means that there is no limit.

`--sysdate-is-now`
: The `SYSDATE( )` function returns the date and time in which the function was executed by MySQL within an SQL statement. It doesn't return the time that the SQL statement started, as the `NOW( )` function does. If you want `SYSDATE( )` to return the same time as `NOW( )`, use this option. See the description of `SYSDATE( )` in Chapter 12 for an example.

`--tc-heuristic-recover={COMMIT|ROLLBACK}`
: This option is not yet implemented by MySQL. It will relate to the heuristic recovery process when it is implemented.

`--time_format=`*value*
: The variable associated with this option is not yet implemented. It's expected to be used to set the default time format for the MySQL server.

`--transaction-isolation=`*option*
: This option sets the default transaction isolation level. The available levels are `READ-UNCOMMITTED`, `READ-COMMITTED`, `REPEATABLE-READ`, or `SERIALIZABLE`.

`--updatable_views_with_limit={0|1}`
: Set this option to 1 to prevent updates to views that do not contain all of the columns of the primary key of the underlying table; the option applies only when the SQL statement contains a `LIMIT` clause. If set to the default value of 1, only a warning is returned and the update is not prevented.

`--version`
: This option displays the version of MySQL that is running on the server.

`--version_compile_machine`
: This option displays the type of machine on which MySQL was compiled.

`--version_compile_os`
: This option displays the type of operating system on which MySQL was compiled.

Logs

These `mysqld` server options relate to general logs created by MySQL. For storage engine specific logs, see the "Storage engine specific options" section later in this chapter.

`--binlog-do-db=`*`value`*
: This option limits the binary log to entries created by SQL statements executed against the database given, and only when it is the default database. If the user sets the default database to another database, but executes SQL statements affecting the database given with this option, those statements will not be written to the binary log. Additional databases may be specified with multiple instances of this option. Despite this option, though, `ALTER DATABASE`, `CREATE DATABASE`, and `DROP DATABASE` statements for the given database will be logged regardless of the default database setting.

`--binlog-ignore-db=`*`value`*
: This option omits entries from the binary log for SQL statements executed against the database given, but only when it is the default database. So when the user sets the default database to another database, but executes SQL statements affecting the database given with this option, those statements will be written to the binary log. Additional databases may be specified with multiple instances of this option. Despite this option, though, `ALTER DATABASE`, `CREATE DATABASE`, and `DROP DATABASE` statements for the given database will be logged regardless of the default database setting.

`--log[=`*`filename`*`]`, `-l [`*`filename`*`]`
: This option instructs the server to log connection information and queries to the given file, or to the default (*host.log*) if none is given.

`--log-bin[=`*`filename`*`]`
: This option records database changes to a binary log to the filename given. If a filename isn't provided, the default name of *host-bin.index* will be used, where *host* is the hostname of the server and *index* is a numeric count.

`--log-bin-trust-function-creators[={0|1}]`
: By default, if binary logging is enabled, when creating a stored procedure you have to state whether the function is deterministic and whether it will modify data. If this option is specified without a value or with a value of 1, this requirement is disabled. If set to 0, which is the default setting, the requirement is enabled.

`--log-error[=filename]`
: This option activates logging of error messages and server startup messages to the filename given. The default name for the log if none is specified is *host.err*, where *host* is the server's hostname.

`--log-long-format, -O`
: This option instructs the server to be more verbose in logs. This is the default setting as of version 4.1 of MySQL. Use the `--log-short-format` option to disable this option.

`--log-short-format`
: This option instructs the server to be less verbose in logs. It is available as of version 4.1 of MySQL.

`--log-queries-not-using-indexes`
: See "Performance optimization" later in this chapter.

`--log-slave-updates`
: This option is used on a slave server to instruct it to write to its own binary log any updates to data made from SQL threads. The option requires that the `--log-bin` option be used on the slave. With this method, it's possible to have a slave act as master to a slave under it.

`--log-slow-admin-statements`
: See "Performance optimization."

`--log-slow-queries[=filename]`
: See "Performance optimization."

`--log-tc=filename`
: This option specifies the filename of the memory-mapped transaction coordinator log. The default filename is *tc.log*, located in the data directory for MySQL.

`--log-tc-size=size`
: This option specifies the size of the memory-mapped transaction coordinator log. The default is 24 KB.

`--log-update[=filename]`
: Activates logging of updates to the filename given. This feature is deprecated in favor of binary logging.

`--log-warnings, -W`
: This option activates logging of warning messages. Prior to version 4.0 of MySQL, this option was invoked with the `--warnings` option. After version 4.1.2, this option is enabled by default and can be disabled with the `--skip-log-warnings` option.

`--long_query_time=value`
: See "Performance optimization."

`--max-binlog-dump-events`
: This option is used by the MySQL test suite for testing and debugging replication.

`--relay-log=filename`
: See "Replication" later in this chapter.

`--relay-log-index=filename`
: See "Replication" later in this chapter.

`--relay-log-info-file=filename`
: See "Replication" later in this chapter.

`--relay-log-purge[={0|1}]`
See "Replication" later in this chapter.

`--relay-log-space-limit=`*value*
See "Replication" later in this chapter.

`--skip-log-warnings`
This option disables the `--log-warnings` feature so that warning messages are not logged.

`--skip-stack-trace`
This option prevents the writing of stack traces.

`--slow-query-log[={0|1}]`
See "Performance optimization."

`--slow-query-log-file=`*filename*
See "Performance optimization."

`--sporadic-binlog-dump-fail`
This option is used by the MySQL test suite for testing and debugging replication.

`--sql_log_bin={0|1}`
The default value of 1 for this option has clients log to the binary log. A value of 0 disables it.

`--sql_log_off={0|1}`
The default value of 0 for this option has clients log to the general query log. A value of 1 disables it and general logging is not done for the client.

`--sql_notes={0|1}`
If this option is set to the default of 1, note-level warning messages are logged. A value of 0 disables it.

`--sql_warnings={0|1}`
If this option is set to 1, warning messages for single row `INSERT` statements generate an information string. The default value of 0 disables it.

`--sql_quote_show_create={0|1}`
If this option is set to the default of 1, identifiers in statements will be quoted in the logs. This can be necessary for certain slave servers that may require identifiers to be contained within quotes. A value of 0 disables it.

`--sync_binlog={0|1}`
If this option is set to a value of 1, the server will synchronize every write to the binary log to the disk. The default value of 0 disables this feature.

Performance optimization

These `mysqld` server options relate to improving server performance. Before changing a server's setting, you should make note of its current setting, and then use the `BENCHMARK( )` function to determine performance before changes are made. After implementing the new server setting, run the `BENCHMARK( )` function again to compare the results. This is just one of many ways in which you might test a server's performance before and after making changes to its settings. The important thing is not to assume that a particular setting will improve performance and to be aware that a change could cause other problems. Test and monitor changes to be sure. For performance options that are specific to InnoDB, see the "InnoDB" subsection of the "Storage engine specific options" section.

`--big-tables`
: This option instructs the server to save temporary results sets to a file to solve problems where results are large and error messages indicate that tables are full.

`--bulk_insert_buffer_size=`*value*
: When bulk inserting data into an existing table that already contains data, the MyISAM storage engine uses a special buffer to make the process faster. You can use this option to set the size of that buffer to improve performance. The default value is 8 MB. A value of 0 disables the buffer.

`--concurrent-insert[={0|1|2}]`
: If this option is set to its default of 1, the MyISAM storage engine will allow simultaneous inserting and selecting of data, but only if there are no free spaces on the filesystem within the datafile. A setting of 2 for this option allows concurrent reading and writing despite spaces in the datafile. It just writes the new rows to the end of the datafile if reads are occurring while the server is trying to write. If no concurrent reads are taking place, the server will get a write lock on the table and make use of the blank space. A value of 0 for this option disables concurrent inserting and reading.

`--delayed_insert_limit=`*value*
: If an `INSERT` statement is entered with the `DELAYED` parameter, the server delays entering rows if there are `SELECT` statements already running against the table. When the table is free, the server will then insert the delayed rows. This option causes the server to enter a fixed number of rows before rechecking to see whether new `SELECT` statements are queued. If there are, it will delay the inserts again.

`--delayed_insert_timeout=`*value*
: When an `INSERT` statement has been issued with the `DELAYED` parameter, the server will wait for the outstanding `SELECT` statements against the table to finish running before executing it. Use this option to set the number of seconds that the server should wait before terminating the `INSERT` statement.

`--delay-key-write[=`*option*`]`
: This option instructs the server how to handle key buffers between writes for MyISAM tables. The choices are `OFF`, `ON`, and `ALL`. The `ON` choice delays writes for tables created with `DELAYED KEYS`. The `ALL` choice delays writes for all MyISAM tables. MyISAM tables should not be accessed by another server or clients such as `myisamcheck` when the `ALL` choice is used; it may cause corruption of indexes.

`--delay-key-write-for-all-tables`
: This option instructs the server not to flush key buffers between writes for MyISAM tables. As of version 4.0.3 of MySQL, use `--delay-key-write=ALL` instead.

`--delayed_queue_size=`*value*
: When an `INSERT` statement has been entered with the `DELAYED` parameter, the server will wait for the outstanding `SELECT` statements against the table to finish running before executing it. Use this option to set the maximum number of rows that the server should queue from inserts. Any additional rows will not be queued, and the `INSERT` statements will have to wait until the queue is reduced.

`--join_buffer_size=`*value*
: This option sets the size of the buffer file to use for joins in which an index is not used. The maximum value for this option is 4 GB, but on 64-bit operating systems, as of version 5.1.23, a larger buffer size may be possible.

`--key_buffer_size=value`
: This option sets the key cache size. This is a buffer used by MyISAM tables for index blocks. The maximum value for this option is 4 GB, but on 64-bit operating systems, as of version 5.1.23, a larger buffer size may be possible. Execute the `SHOW STATUS` statement on the server to see the settings for the key cache.

`--key_cache_age_threshold=value`
: This option sets the point at which a buffer will be switched from what is known as a hot subchain in the key cache to a warm one. Lower values cause the switching to occur faster. The default value is 300. The lowest value allowed is 100.

`--key_cache_block_size=value`
: This option sets the size of blocks in the key cache. The values are in bytes. The default is 1024.

`--key_cache_division_limit=value`
: This option sets the division point between hot and warm subchains in the key cache. The value given represents a percentage of the whole buffer. The default value is 100. A value of 1 to 100 is allowed.

`--large-pages`
: This option enables large pages in memory.

`--log-slow-admin-statements`
: If this option is enabled, administrative SQL statements that take too long to execute will be logged. These include statements such as `ALTER TABLE`, `CHECK TABLE`, and `OPTIMIZE TABLE`.

`--log-slow-queries[=filename]`
: This option instructs the server to log queries that take longer than the number of seconds specified in the value of the `long_query_time` variable. If *`filename`* is specified, entries are recorded in the log file named.

`--log-queries-not-using-indexes`
: When used with the `--log-slow-queries` option, this option causes all queries that do not use indexes to be logged to the slow query log. It is available as of version 4.1 of MySQL.

`--long_query_time=value`
: This option sets the number of seconds that a query can take to execute before it's considered a slow query. If the `--log-slow-queries` option is in use, queries that exceed the number of seconds set by this option will be logged.

`--low-priority-updates`
: This option sets all SQL statements that modify data to a lower priority than `SELECT` statements, by default.

`--max_allowed_packet=value`
: This option sets the maximum size of a packet or a generated string. If using `BLOB` or `TEXT` columns, the variable associated with this option should be at least as large as the largest entry for the column. To determine this, you can execute `SHOW TABLE STATUS LIKE 'table';` on the server and look for the `Max_data_length` field. The maximum size allowed for this option is 1 GB. The `--net_buffer_length` option sets the initial size of buffer packets.

`--max_delayed_threads=value`
: This option sets the maximum number of threads the server can use to handle delayed inserts. See the `--delayed_insert_limit` and `--delayed_insert_timeout` options earlier in this chapter for more information.

`--max_seeks_for_key=value`
: When MySQL searches a table for data based on a `WHERE` clause using an index, it expects to have to search a certain number of rows in the index. You can adjust this expectation with this option. A lower value causes the MySQL optimizer to give preference to indexes over table scans.

`--max_sort_length=value`
: This option sets the maximum number of bytes the server can examine in each field when sorting `BLOB` or `TEXT` columns. Any bytes of data beyond the value set for this option are ignored in sorting. The default is 1024.

`--max_sp_recursion_depth[=value]`
: This option sets the maximum depth to which a stored procedure can invoke itself. The default is 0, which disables all recursion, and the maximum depth allowed is 255.

`--max_tmp_tables=value`
: This is a new option that has not yet been implemented. When it is, you will be able to use it to limit the number of temporary tables that a client can have open at one time.

`--max_write_lock_count=value`
: This option limits the number of write locks that may be made without allowing reads to be performed.

`--multi_range_count=value`
: This option sets the maximum number of ranges that may be sent to a table handler at one time for a range select. The default is 256.

`--memlock`
: This option is used on filesystems that support `mlockall( )` system calls (e.g., Solaris) to lock the daemon in memory and thereby avoid the use of disk swapping in an attempt to improve performance. Requires the daemon to be started by *root*, which may be a security problem.

`--optimizer_prune_level[={0|1}]`
: This option sets the behavior of the optimizer when it tries to reduce or remove plans that don't seem to be useful. A value of 0 disables heuristics and instructs the optimizer to search as much as possible. The default value of 1 enables heuristics and thereby instructs the optimizer to prune plans.

`--optimizer_search_depth[=value]`
: This option sets the maximum depth of searches performed by the query optimizer. A lower number will make for better queries, but it will take longer to perform. A higher number should make queries faster. If the value is set to 0, the server will attempt to decide on the best setting.

`--preload_buffer_size=value`
: This option sets the size of the buffer used to hold preloaded indexes. The default is 32768 (32 KB).

`--query_alloc_block_size=value`
: This option sets the size of memory blocks that are allocated for use in parsing and executing a statement.

`--query_cache_limit=value`
: This option sets the maximum size of the query cache in bytes. The default is 1 MB.

`--query_cache_min_res_unit=value`
: This option sets the minimum size in bytes of blocks used for the query cache. The default is 4096 (4 KB).

`--query_cache_size=value`
: This option sets the maximum size in bytes of the cache used for query results. The default is 0. Values should be given in multiples of 1024 (1 KB).

`--query_cache_type={0|1|2}`
: This option sets the type of query cache to use on the server. A value of 0 causes the query cache not to be used. The default value of 1 causes all queries to be cached except `SELECT` statements that include the `SQL_NO_CACHE` parameter. A value of 2 means that no queries will be cached except `SELECT` statements that include the `SQL_CACHE` parameter.

`--query_cache_wlock_invalidate[={0|1}]`
: If a table is locked, but the results of querying the same table are already contained in the query cache, the results of a query will be returned if this option is set to 0, the default. Setting it to 1 will disable this feature and users will have to wait for the write lock to be released before reading the table and the related query cache data.

`--query_prealloc_size=value`
: This option sets the size of the persistent buffer used for parsing and executing statements.

`--range_alloc_block_size=value`
: This option sets the size of blocks of memory allocated for range queries.

`--read_buffer_size=value`
: This option sets the size in bytes of the buffer to use for each thread when doing sequential scans. The default value is 131072; the maximum is 2 GB.

`--read_rnd_buffer_size=value`
: Rows that are sorted by an index are read into a buffer to minimize disk activity. You can set the size of this buffer with this option to a maximum of 2 GB.

`--safemalloc-mem-limit=value`
: This option is used to simulate a memory shortage when the server has been compiled with the `--with-debug=full` option.

`--shared-memory`
: This option allows shared memory connections by Windows clients locally. It is available as of version 4.1 of MySQL.

`--shared-memory-base-name=name`
: This option sets the name to use for shared memory connections in Windows. It is available as of version 4.1 of MySQL.

`--skip-concurrent-insert`
: This option prevents simultaneous `SELECT` and `INSERT` statements for MyISAM tables.

`--skip-delay-key-write`
: This option disregards tables marked as `DELAY_KEY_WRITE`. As of version 4.0.3 of MySQL, use `--delay-key-write=OFF` instead.

`--skip-safemalloc`
: This option prevents the server from checking for memory overruns when performing memory allocation and memory freeing activities.

`--skip-thread-priority`
: This option prevents prioritizing of threads.

`--slow-query-log[={0|1}]`
: Slow queries are ones that take more than the number of seconds set by the `--long_query_time` option. A value of 1 for this option enables the logging of slow queries; the default value of 0 disables it. This option is available as of version 5.1.12 of MySQL.

`--slow-query-log-file=`*`filename`*
: This option sets the name of the slow query log file. By default it's *`host_name-slow.log`*. This option is available as of version 5.1.12 of MySQL.

`--slow_launch_time`
: This option causes a thread's `Slow_launch_threads` status to be updated to reflect whether a thread takes too long to launch.

`--sort_buffer_size=`*`value`*
: This option sets the size of the buffer each thread should use when sorting data for a query. The maximum value for this option is 4 GB, but on 64-bit operating systems, as of version 5.1.23, a larger buffer size may be possible.

`--table_lock_wait_timeout=`*`value`*
: This option sets the number of seconds that the server should wait to get a table lock before it terminates and returns an error. The timeout is related only to connections with active cursors. The default value is 50.

`--table_open_cache=`*`value`*
: This option sets the maximum number of open tables allowed for all threads. Prior to version 5.1.3, this option was called `--table_cache`. Executing the `FLUSH TABLES` statement will close any open tables and reopen any in use.

`--thread_cache_size=`*`value`*
: With this option, you can set the number of threads that the server should cache for reuse. This may lead to quicker connection times for new connections that are made by clients.

`--thread_concurrency=`*`value`*
: The value of the variable associated with this option is used by applications to provide a hint regarding the number of threads that the server should run concurrently. It's used on Solaris systems in conjunction with the `thr_setconcurrency( )` system function.

`--thread_stack=`*`value`*
: This option sets the size of the stack for each thread. The default value is 192 KB.

`--tmp_table_size=`*`value`*
: This option sets the maximum size of internal, in-memory temporary tables. This option is not related to MEMORY tables, though.

`--transaction_alloc_block_size=`*value*
: The memory pool described under the `--transaction_prealloc_size` option is increased as needed in increments. The amount of increments is drawn from the value of the `transaction_alloc_block_size` server variable. This option can be used to change that variable.

`--transaction_prealloc_size=`*value*
: A memory pool is used to temporarily store activities related to transactions. The size of that pool expands as needed. Initially, it is set to the size of the value of the server variable `transaction_prealloc_size`. This option can be used to set that variable higher to improve performance.

`--wait_timeout=`*value*
: This option sets the number of seconds that the server will wait before terminating a nonresponsive connection based on TCP/IP or a socket file. This option is not associated with connections through named pipes or shared memory.

Replication

An alphabetical list follows of `mysqld` server options related to replication. Many also appear earlier in Chapter 8. Although these options can be set at the command line when starting the server, and some can also be set with SQL statements while the server is running, as a general policy the options should be given in the server's options file (e.g., *my.cnf* or *my.ini*, depending on your system). Otherwise, there's a chance that the options may be lost when the server is restarted, in which case replication may fail or at least not function as you want:

`--abort-slave-event-count=`*value*
: This option is used by the MySQL test suite for testing and debugging replication.

`--disconnect-slave-event-count=`*value*
: This option is used by the MySQL test suite for testing and debugging replication.

`--init_slave='`*string*`'`
: Use this option on the server to specify one or more SQL statements, all combined in a single *string*, that are to be executed by the slave each time its SQL thread starts.

`--log-slave-updates`
: This option is used on a slave server to instruct it to write to its own binary log any updates to data made from SQL threads. It requires that the `--log-bin` option be used on the slave. With this method it's possible to have a slave act as master to a slave under it.

`--master-connect-retry=`*seconds*
: This option sets the number of seconds that a slave thread may sleep before trying to reconnect to the master. The default is 60 seconds. This value is also included in the *master.info* file. If that file exists and is accessible, the value contained in it will override this option.

`--master-host=`*host*
: This option is superseded by the same information in the *master.info* file and is necessary for replication. It that file doesn't exist or is inaccessible, this option may be used to set the hostname or IP address of the master server.

`--master-info-file=`*filename*
: This option sets the name of the master information file. This file is described in detail in Chapter 8 in the section "Replication Process." By default this file is named *master.info* and is located in the data directory of MySQL.

`--master-password=`*password*
: If the *master.info* file doesn't exist or is inaccessible, this option may be used to set the password used by the slave thread for accessing the master server.

`--master-port=`*port*
: This option sets the port number on which the master will listen for replication. By default it's 3306. The value for this variable in the *master.info* file, if available, will override this option.

`--master-retry-count=`*value*
: This option specifies the number of times the slave should try to connect to the master if attempts fail. The default value is 86400. The interval between retries is set by the option `--master-connect-retry`. Retries are initiated when the slave connection times out for the amount of time set with the `--slave-net-timeout` option.

`--master-ssl`
: This option is similar to `--ssl` in the "Security and connections" section earlier in this chapter, but it applies to a slave's SSL connection with the master server.

`--master-ssl-ca[=`*value*`]`
: This option is similar to `--ssl-ca` in the "Security and connections" section earlier in this chapter, but it applies to a slave's SSL connection with the master server.

`--master-ssl-capath[=`*value*`]`
: This option is similar to `--ssl-capath` in the "Security and connections" section earlier in this chapter, but it applies to a slave's SSL connection with the master server.

`--master-ssl-cert[=`*value*`]`
: This option is similar to `--ssl-cert` in the "Security and connections" section earlier in this chapter, but it applies to a slave's SSL connection with the master server.

`--master-ssl-cipher[=`*value*`]`
: This option is similar to `--ssl-cipher` in the "Security and connections" section earlier in this chapter, but it applies to a slave's SSL connection with the master server.

`--master-ssl-key[=`*value*`]`
: This option is similar to `--ssl-key` in the "Security and connections" section earlier in this chapter, but it applies to a slave's SSL connection with the master server.

`--master-user=`*value*
: This option sets the name of the user account that the slave thread uses to connect to the master server for replication. The user given must have the `REPLICATION SLAVE` privilege on the master. This option is overridden by the *master.info* file.

`--max-binlog-dump-events=`*value*
: This option is used by the MySQL test suite for testing and debugging replication.

`--read_only`
: This option prevents users from adding, changing, or deleting data on the server, except for users with `SUPER` privileges. The other exception is that updates from slave threads are allowed. This option does not carry over from a master to its slaves.