**4.2.2.2 Using Option Files**

Most MySQL programs can read startup options from option files (sometimes called configuration files). Option files provide a convenient way to specify commonly used options so that they need not be entered on the command line each time you run a program.

To determine whether a program reads option files, invoke it with the --help option. (For [**mysqld**](https://dev.mysql.com/doc/refman/8.0/en/mysqld.html), use [--verbose](https://dev.mysql.com/doc/refman/8.0/en/server-options.html#option_mysqld_verbose) and [--help](https://dev.mysql.com/doc/refman/8.0/en/server-options.html#option_mysqld_help).) If the program reads option files, the help message indicates which files it looks for and which option groups it recognizes.

Note

A MySQL program started with the --no-defaults option reads no option files other than .mylogin.cnf.

A server started with the [persisted\_globals\_load](https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_persisted_globals_load) system variable disabled does not read mysqld-auto.cnf.

Many option files are plain text files, created using any text editor. The exceptions are:

* The .mylogin.cnf file that contains login path options. This is an encrypted file created by the [**mysql\_config\_editor**](https://dev.mysql.com/doc/refman/8.0/en/mysql-config-editor.html) utility. See [Section 4.6.7, “**mysql\_config\_editor** — MySQL Configuration Utility”](https://dev.mysql.com/doc/refman/8.0/en/mysql-config-editor.html). A “login path” is an option group that permits only certain options: host, user, password, port and socket. Client programs specify which login path to read from .mylogin.cnf using the [--login-path](https://dev.mysql.com/doc/refman/8.0/en/option-file-options.html#option_general_login-path) option.

To specify an alternative login path file name, set the MYSQL\_TEST\_LOGIN\_FILE environment variable. This variable is used by the **mysql-test-run.pl** testing utility, but also is recognized by [**mysql\_config\_editor**](https://dev.mysql.com/doc/refman/8.0/en/mysql-config-editor.html) and by MySQL clients such as [**mysql**](https://dev.mysql.com/doc/refman/8.0/en/mysql.html), [**mysqladmin**](https://dev.mysql.com/doc/refman/8.0/en/mysqladmin.html), and so forth.

* The mysqld-auto.cnf file in the data directory. This JSON-format file contains persisted system variable settings. It is created by the server upon execution of [SET PERSIST](https://dev.mysql.com/doc/refman/8.0/en/set-variable.html) or [SET PERSIST\_ONLY](https://dev.mysql.com/doc/refman/8.0/en/set-variable.html) statements. See [Section 5.1.9.3, “Persisted System Variables”](https://dev.mysql.com/doc/refman/8.0/en/persisted-system-variables.html). Management of mysqld-auto.cnf should be left to the server and not performed manually.
* [Option File Processing Order](https://dev.mysql.com/doc/refman/8.0/en/option-files.html#option-file-order)
* [Option File Syntax](https://dev.mysql.com/doc/refman/8.0/en/option-files.html#option-file-syntax)
* [Option File Inclusions](https://dev.mysql.com/doc/refman/8.0/en/option-files.html#option-file-inclusions)

**Option File Processing Order**

MySQL looks for option files in the order described in the following discussion and reads any that exist. If an option file you want to use does not exist, create it using the appropriate method, as just discussed.

Note

For information about option files used with NDB Cluster programs, see [Section 22.3, “Configuration of NDB Cluster”](https://dev.mysql.com/doc/refman/8.0/en/mysql-cluster-configuration.html).

On Windows, MySQL programs read startup options from the files shown in the following table, in the specified order (files listed first are read first, files read later take precedence).

**Table 4.1 Option Files Read on Windows Systems**

| **File Name** | **Purpose** |
| --- | --- |
| %WINDIR%\my.ini, %WINDIR%\my.cnf | Global options |
| C:\my.ini, C:\my.cnf | Global options |
| *BASEDIR*\my.ini, *BASEDIR*\my.cnf | Global options |
| defaults-extra-file | The file specified with [--defaults-extra-file](https://dev.mysql.com/doc/refman/8.0/en/option-file-options.html#option_general_defaults-extra-file), if any |
| %APPDATA%\MySQL\.mylogin.cnf | Login path options (clients only) |
| *DATADIR*\mysqld-auto.cnf | System variables persisted with [SET PERSIST](https://dev.mysql.com/doc/refman/8.0/en/set-variable.html) or [SET PERSIST\_ONLY](https://dev.mysql.com/doc/refman/8.0/en/set-variable.html) (server only) |

In the preceding table, %WINDIR% represents the location of your Windows directory. This is commonly C:\WINDOWS. Use the following command to determine its exact location from the value of the WINDIR environment variable:

C:\> echo %WINDIR%

%APPDATA% represents the value of the Windows application data directory. Use the following command to determine its exact location from the value of the APPDATA environment variable:

C:\> echo %APPDATA%

*BASEDIR* represents the MySQL base installation directory. When MySQL 8.0 has been installed using MySQL Installer, this is typically C:\*PROGRAMDIR*\MySQL\MySQL 8.0 Server where *PROGRAMDIR* represents the programs directory (usually Program Files on English-language versions of Windows), See [Section 2.3.3, “MySQL Installer for Windows”](https://dev.mysql.com/doc/refman/8.0/en/mysql-installer.html).

*DATADIR* represents the MySQL data directory. As used to find mysqld-auto.cnf, its default value is the data directory location built in when MySQL was compiled, but can be changed by [--datadir](https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_datadir) specified as an option-file or command-line option processed before mysqld-auto.cnf is processed.

On Unix and Unix-like systems, MySQL programs read startup options from the files shown in the following table, in the specified order (files listed first are read first, files read later take precedence).

Note

On Unix platforms, MySQL ignores configuration files that are world-writable. This is intentional as a security measure.

**Table 4.2 Option Files Read on Unix and Unix-Like Systems**

| **File Name** | **Purpose** |
| --- | --- |
| /etc/my.cnf | Global options |
| /etc/mysql/my.cnf | Global options |
| *SYSCONFDIR*/my.cnf | Global options |
| $MYSQL\_HOME/my.cnf | Server-specific options (server only) |
| defaults-extra-file | The file specified with [--defaults-extra-file](https://dev.mysql.com/doc/refman/8.0/en/option-file-options.html#option_general_defaults-extra-file), if any |
| ~/.my.cnf | User-specific options |
| ~/.mylogin.cnf | User-specific login path options (clients only) |
| *DATADIR*/mysqld-auto.cnf | System variables persisted with [SET PERSIST](https://dev.mysql.com/doc/refman/8.0/en/set-variable.html) or [SE PERSIST\_ONLY](https://dev.mysql.com/doc/refman/8.0/en/set-variable.html) (server only) |

In the preceding table, ~ represents the current user's home directory (the value of $HOME).

*SYSCONFDIR* represents the directory specified with the [SYSCONFDIR](https://dev.mysql.com/doc/refman/8.0/en/source-configuration-options.html#option_cmake_sysconfdir) option to **CMake** when MySQL was built. By default, this is the etc directory located under the compiled-in installation directory.

MYSQL\_HOME is an environment variable containing the path to the directory in which the server-specific my.cnf file resides. If MYSQL\_HOME is not set and you start the server using the [**mysqld\_safe**](https://dev.mysql.com/doc/refman/8.0/en/mysqld-safe.html) program, [**mysqld\_safe**](https://dev.mysql.com/doc/refman/8.0/en/mysqld-safe.html) sets it to *BASEDIR*, the MySQL base installation directory.

*DATADIR* represents the MySQL data directory. As used to find mysqld-auto.cnf, its default value is the data directory location built in when MySQL was compiled, but can be changed by [--datadir](https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_datadir) specified as an option-file or command-line option processed before mysqld-auto.cnf is processed.

If multiple instances of a given option are found, the last instance takes precedence, with one exception: For [**mysqld**](https://dev.mysql.com/doc/refman/8.0/en/mysqld.html), the *first* instance of the [--user](https://dev.mysql.com/doc/refman/8.0/en/server-options.html#option_mysqld_user) option is used as a security precaution, to prevent a user specified in an option file from being overridden on the command line.

**Option File Syntax**

The following description of option file syntax applies to files that you edit manually. This excludes .mylogin.cnf, which is created using [**mysql\_config\_editor**](https://dev.mysql.com/doc/refman/8.0/en/mysql-config-editor.html) and is encrypted, and mysqld-auto.cnf, which the server creates in JSON format.

Any long option that may be given on the command line when running a MySQL program can be given in an option file as well. To get the list of available options for a program, run it with the --help option. (For [**mysqld**](https://dev.mysql.com/doc/refman/8.0/en/mysqld.html), use [--verbose](https://dev.mysql.com/doc/refman/8.0/en/server-options.html#option_mysqld_verbose) and [--help](https://dev.mysql.com/doc/refman/8.0/en/server-options.html#option_mysqld_help).)

The syntax for specifying options in an option file is similar to command-line syntax (see [Section 4.2.2.1, “Using Options on the Command Line”](https://dev.mysql.com/doc/refman/8.0/en/command-line-options.html)). However, in an option file, you omit the leading two dashes from the option name and you specify only one option per line. For example, --quick and --host=localhost on the command line should be specified as quick and host=localhost on separate lines in an option file. To specify an option of the form --loose-*opt\_name* in an option file, write it as loose-*opt\_name*.

Empty lines in option files are ignored. Nonempty lines can take any of the following forms:

* #*comment*, ;*comment*

Comment lines start with # or ;. A # comment can start in the middle of a line as well.

* [*group*]

*group* is the name of the program or group for which you want to set options. After a group line, any option-setting lines apply to the named group until the end of the option file or another group line is given. Option group names are not case-sensitive.

* *opt\_name*

This is equivalent to --*opt\_name* on the command line.

* *opt\_name*=*value*

This is equivalent to --*opt\_name*=*value* on the command line. In an option file, you can have spaces around the = character, something that is not true on the command line. The value optionally can be enclosed within single quotation marks or double quotation marks, which is useful if the value contains a # comment character.

Leading and trailing spaces are automatically deleted from option names and values.

You can use the escape sequences \b, \t, \n, \r, \\, and \s in option values to represent the backspace, tab, newline, carriage return, backslash, and space characters. In option files, these escaping rules apply:

* A backslash followed by a valid escape sequence character is converted to the character represented by the sequence. For example, \s is converted to a space.
* A backslash not followed by a valid escape sequence character remains unchanged. For example, \S is retained as is.

The preceding rules mean that a literal backslash can be given as \\, or as \ if it is not followed by a valid escape sequence character.

The rules for escape sequences in option files differ slightly from the rules for escape sequences in string literals in SQL statements. In the latter context, if “*x*” is not a valid escape sequence character, \*x* becomes “*x*” rather than \*x*. See [Section 9.1.1, “String Literals”](https://dev.mysql.com/doc/refman/8.0/en/string-literals.html).

The escaping rules for option file values are especially pertinent for Windows path names, which use \ as a path name separator. A separator in a Windows path name must be written as \\ if it is followed by an escape sequence character. It can be written as \\ or \ if it is not. Alternatively, / may be used in Windows path names and will be treated as \. Suppose that you want to specify a base directory of C:\Program Files\MySQL\MySQL Server 8.0 in an option file. This can be done several ways. Some examples:

basedir="C:\Program Files\MySQL\MySQL Server 8.0"

basedir="C:\\Program Files\\MySQL\\MySQL Server 8.0"

basedir="C:/Program Files/MySQL/MySQL Server 8.0"

basedir=C:\\Program\sFiles\\MySQL\\MySQL\sServer\s8.0

If an option group name is the same as a program name, options in the group apply specifically to that program. For example, the [mysqld] and [mysql] groups apply to the [**mysqld**](https://dev.mysql.com/doc/refman/8.0/en/mysqld.html) server and the [**mysql**](https://dev.mysql.com/doc/refman/8.0/en/mysql.html) client program, respectively.

The [client] option group is read by all client programs provided in MySQL distributions (but *not* by [**mysqld**](https://dev.mysql.com/doc/refman/8.0/en/mysqld.html)). To understand how third-party client programs that use the C API can use option files, see the C API documentation at [Section 28.7.6.50, “mysql\_options()”](https://dev.mysql.com/doc/refman/8.0/en/mysql-options.html).

The [client] group enables you to specify options that apply to all clients. For example, [client] is the appropriate group to use to specify the password for connecting to the server. (But make sure that the option file is accessible only by yourself, so that other people cannot discover your password.) Be sure not to put an option in the [client] group unless it is recognized by *all* client programs that you use. Programs that do not understand the option quit after displaying an error message if you try to run them.

List more general option groups first and more specific groups later. For example, a [client] group is more general because it is read by all client programs, whereas a [mysqldump] group is read only by [**mysqldump**](https://dev.mysql.com/doc/refman/8.0/en/mysqldump.html). Options specified later override options specified earlier, so putting the option groups in the order [client], [mysqldump] enables [**mysqldump**](https://dev.mysql.com/doc/refman/8.0/en/mysqldump.html)-specific options to override [client] options.

Here is a typical global option file:

[client]

port=3306

socket=/tmp/mysql.sock

[mysqld]

port=3306

socket=/tmp/mysql.sock

key\_buffer\_size=16M

max\_allowed\_packet=128M

[mysqldump]

quick

Here is a typical user option file:

[client]

# The following password will be sent to all standard MySQL clients

password="my password"

[mysql]

no-auto-rehash

connect\_timeout=2

To create option groups to be read only by [**mysqld**](https://dev.mysql.com/doc/refman/8.0/en/mysqld.html) servers from specific MySQL release series, use groups with names of [mysqld-5.7], [mysqld-8.0], and so forth. The following group indicates that the [sql\_mode](https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_sql_mode) setting should be used only by MySQL servers with 8.0.x version numbers:

[mysqld-8.0]

sql\_mode=TRADITIONAL

**Option File Inclusions**

It is possible to use !include directives in option files to include other option files and !includedir to search specific directories for option files. For example, to include the /home/mydir/myopt.cnf file, use the following directive:

!include /home/mydir/myopt.cnf

To search the /home/mydir directory and read option files found there, use this directive:

!includedir /home/mydir

MySQL makes no guarantee about the order in which option files in the directory are read.

Note

Any files to be found and included using the !includedir directive on Unix operating systems *must* have file names ending in .cnf. On Windows, this directive checks for files with the .ini or .cnf extension.

Write the contents of an included option file like any other option file. That is, it should contain groups of options, each preceded by a [*group*] line that indicates the program to which the options apply.

While an included file is being processed, only those options in groups that the current program is looking for are used. Other groups are ignored. Suppose that a my.cnf file contains this line:

!include /home/mydir/myopt.cnf

And suppose that /home/mydir/myopt.cnf looks like this:

[mysqladmin]

force

[mysqld]

key\_buffer\_size=16M

If my.cnf is processed by [**mysqld**](https://dev.mysql.com/doc/refman/8.0/en/mysqld.html), only the [mysqld] group in /home/mydir/myopt.cnf is used. If the file is processed by [**mysqladmin**](https://dev.mysql.com/doc/refman/8.0/en/mysqladmin.html), only the [mysqladmin] group is used. If the file is processed by any other program, no options in /home/mydir/myopt.cnf are used.

The !includedir directive is processed similarly except that all option files in the named directory are read.

If an option file contains !include or !includedir directives, files named by those directives are processed whenever the option file is processed, no matter where they appear in the file.

For inclusion directives to work, the file path should not be specified within quotes and should have no escape sequences. For example, the following statements provided in my.ini will read the option file myopts.ini:

!include C:/ProgramData/MySQL/MySQL Server/myopts.ini

!include C:\ProgramData\MySQL\MySQL Server\myopts.ini

!include C:\\ProgramData\\MySQL\\MySQL Server\\myopts.ini

On Windows, if !include */path/to/extra.ini* is the last line in the file, make sure that a newline is appended at the end or the line will be ignored.