

3 GCC Command Options

When you invoke GCC, it normally does preprocessing, compilation, assembly and linking. The “overall options” allow you to stop this process at an intermediate stage. For example, the `-c` option says not to run the linker. Then the output consists of object files output by the assembler. See [Options Controlling the Kind of Output](#).

Other options are passed on to one or more stages of processing. Some options control the preprocessor and others the compiler itself. Yet other options control the assembler and linker; most of these are not documented here, since you rarely need to use any of them.

Most of the command-line options that you can use with GCC are useful for C programs; when an option is only useful with another language (usually C++), the explanation says so explicitly. If the description for a particular option does not mention a source language, you can use that option with all supported languages.

The usual way to run GCC is to run the executable called `gcc`, or *machine-gcc* when cross-compiling, or *machine-gcc-version* to run a specific version of GCC. When you compile C++ programs, you should invoke GCC as `g++` instead. See [Compiling C++ Programs](#), for information about the differences in behavior between `gcc` and `g++` when compiling C++ programs.

The `gcc` program accepts options and file names as operands. Many options have multi-letter names; therefore multiple single-letter options may *not* be grouped: `-dv` is very different from `‘-d -v’`.

You can mix options and other arguments. For the most part, the order you use doesn’t matter. Order does matter when you use several options of the same kind; for example, if you specify `-l` more than once, the directories are searched in the order specified. Also, the placement of the `-l` option is significant.

Many options have long names starting with `‘-f’` or with `‘-w’`—for example, `-fmove-loop-invariants`, `-wformat` and so on. Most of these have both positive and negative forms; the negative form of `-ffoo` is `-fno-foo`. This manual documents only one of these two forms, whichever one is not the default.

Some options take one or more arguments typically separated either by a space or by the equals sign (`‘=’`) from the option name. Unless documented otherwise, an argument can be either numeric or a string. Numeric arguments must typically be small unsigned decimal or hexadecimal integers. Hexadecimal arguments must begin with the `‘0x’` prefix. Arguments to options that specify a size threshold of some sort may be arbitrarily large decimal or hexadecimal integers followed by a byte size suffix designating a multiple of bytes such as `kB` and `KiB` for kilobyte and kibibyte, respectively, `MB` and `MiB` for megabyte and mebibyte, `GB` and `GiB` for gigabyte and gibibyte, and so on. Such arguments are designated by *byte-size* in the following text. Refer to the NIST, IEC, and other relevant national and international standards for the full listing and explanation of the binary and decimal byte size prefixes.

See [Option Index](#), for an index to GCC’s options.

- [Option Summary](#)
- [Options Controlling the Kind of Output](#)
- [Compiling C++ Programs](#)
- [Options Controlling C Dialect](#)
- [Options Controlling C++ Dialect](#)
- [Options Controlling Objective-C and Objective-C++ Dialects](#)
- [Options to Control Diagnostic Messages Formatting](#)
- [Options to Request or Suppress Warnings](#)
- [Options That Control Static Analysis](#)

- [Options for Debugging Your Program](#)
- [Options That Control Optimization](#)
- [Program Instrumentation Options](#)
- [Options Controlling the Preprocessor](#)
- [Passing Options to the Assembler](#)
- [Options for Linking](#)
- [Options for Directory Search](#)
- [Options for Code Generation Conventions](#)
- [GCC Developer Options](#)
- [Machine-Dependent Options](#)
- [Specifying Subprocesses and the Switches to Pass to Them](#)
- [Environment Variables Affecting GCC](#)
- [Using Precompiled Headers](#)
- [C++ Modules](#)

Next: [C Implementation-Defined Behavior](#), Previous: [Language Standards Supported by GCC](#), Up: [Introduction](#)
[\[Contents\]](#)[\[Index\]](#)