

On Windows 98 and Windows ME this generally requires editing the system `autoexec.bat` file, and a reboot is needed for the changes to take effect.

Windows XP Users Note: MinGW currently allows a mere 8 megabytes for the heap. If your application requires access to more memory, try compiling with: `-Wl,--heap=0x01000000`. Use larger hexadecimal values for `--heap` until your program runs.

Running G95

G95 determines how an input file should be compiled based on its extension. Allowable file name extensions for Fortran source files are limited to `.f`, `.F`, `.for`, `.FOR`, `.f90`, `.F90`, `.f95`, `.F95`, `.f03` and `.F03`. The filename extension determines whether Fortran sources are to be treated as fixed form, or free format. Files ending in `.f`, `.F`, `.for`, and `.FOR` are assumed to be fixed form source compatible with old f77 files. Files ending in `.f90`, `.F90`, `.f95`, `.F95`, `.f03` and `.F03` are assumed to be free source form. Files ending in uppercase letters are pre-processed with the C preprocessor by default, files ending in lowercase letters are not pre-processed by default.

The basic options for compiling Fortran sources with g95 are:

- c Compile only, do not run the linker.
- v Show the actual programs invoked by g95 and their arguments. Particularly useful for tracking path problems.
- o Specify the name of the output file, either an object file or the executable. An `.exe` extension is automatically added on Windows systems. If no output file is specified, the default output file is named `a.out` on unix, or `a.exe` on Windows systems.

Simple examples:

```
g95 -c hello.f90
```

Compiles `hello.f90` to an object file named `hello.o`.

```
g95 hello.f90
```

Compiles `hello.f90` and links it to produce an executable `a.out` (on unix), or `a.exe` (on MS Windows systems).

```
g95 -c h1.f90 h2.f90 h3.f90
```

Compiles multiple source files. If all goes well, object files `h1.o`, `h2.o` and `h3.o` are created.

```
g95 -o hello h1.f90 h2.f90 h3.f90
```

Compiles multiple source files and links them together to an executable file named `hello` on unix, or `hello.exe` on MS Windows systems.

Option Synopsis

<code>g95</code>	<code>[-c -S -E]</code>	Compile & assemble Produce assembly code List source
	<code>[-g] [-pg]</code>	Debug options
	<code>[-O[n]]</code>	Optimization level, $n = 0, 1, 2, 3$
	<code>[-s]</code>	Strip debug info
	<code>[-Wwarn] [-pedantic]</code>	Warning switches
	<code>[-Idir]</code>	Include directory to search
	<code>[-Ldir]</code>	Library directory to search
	<code>[-D macro[=value]...]</code>	Define macro
	<code>[-U macro]</code>	Undefine macro
	<code>[-f option ...]</code>	General compile options
	<code>[-m machine-option ...]</code>	Machine specific options. See GCC manual
	<code>[-o outfile]</code>	Name of outfile
	<code>infile</code>	