

**NAME**

winegcc – Wine C and C++ MinGW Compatible Compiler

**SYNOPSIS**

**winegcc** [*options*] *infile*...

**DESCRIPTION**

**winegcc** is a gcc wrapper which tries to provide a MinGW compatible compiler under Linux. This is most useful to Win32 developers who can simply take their MinGW code from Windows, and recompile it without modifications under Winelib on Linux. **winegcc** accepts mostly the same options as **winegcc**.

The goal of **winegcc** is to be able to simply replace **gcc/g++/windres** with **winegcc/wineg++/wrc** in a MinGW Makefile, and just recompile the application using Winelib under Wine. While typically there are small adjustments that must be made to the application source code and/or Makefile, it is quite easy to do them in a fashion that is compatible between the MinGW and Wine environments.

This manual will document only the differences from **gcc**; please consult the **gcc** manual for more information on those options.

**OPTIONS**

**gcc options:** All gcc options are supported, and are passed along to the backend compiler.

**-b,--target** *target*

Specify the target architecture triplet for cross-compiling. **winegcc** will then invoke *target-gcc* instead of **gcc**.

**--wine-objdir** *dir*

Specify the Wine object directory. This is used when building Wine itself, to use the includes and libraries from inside the build tree.

**--winebuild** *name*

Specifies the path and name of the **winebuild** binary that will be launched automatically by **winegcc**. If not set, **winegcc** will look for a file named *winebuild* in the path. This takes precedence over the **WINEBUILD** environment variable.

**-fno-short-wchar**

Override the underlying type for `wchar_t` to be the default for the target, instead of using short unsigned int, which is the default for Win32.

**-mconsole**

This option passes '`--subsystem console`' to **winebuild**, to build console applications. It is the default.

**-mno-cygwin**

Use Wine implementation of MSVCRT, instead of linking against the host system `libc`. This is necessary for the vast majority of Win32 applications, as they typically depend on various features of MSVCRT. This switch is also used by the MinGW compiler to link against MSVCRT on Windows, instead of linking against Cygwin `libc`. Sharing the syntax with MinGW makes it very easy to write Makefiles that work under Wine, MinGW+MSYS, or MinGW+Cygwin.

**-municode**

Set the default entry point of the application to be the Unicode **wmain()** instead of the standard **main()**.

**-mwindows**

This option adds `-lgdi32`, `-lcomdlg32`, and `-lshell32` to the list of default libraries, and passes '`--subsystem windows`' to **winebuild** to build graphical applications.

**-nodefaultlibs**

Do not use the standard system libraries when linking. These include at a minimum `-lkernel32`, `-luser32`, `-ladvapi32`, and any default libraries used by the backend compiler. The **-mwindows** option augments the list of default libraries as described above.

**-nostartfiles**

Do not add the winecrt0 library when linking.

**-Wb,*option***

Pass an option to winebuild. If *option* contains commas, it is split into multiple options at the commas.

**ENVIRONMENT****WINEBUILD**

Specifies the path and name of the **winebuild** binary that will be launched automatically by **winegcc**. If not set, **winegcc** will look for a file named *winebuild* in the path.

**DEFINES**

winegcc defines `__WINE__`, for code that needs to know when it is being compiled under Wine. It also defines `WIN32`, `_WIN32`, `__WIN32`, `__WIN32__`, `__WINNT`, and `__WINNT__` for compatibility with MinGW.

**BUGS**

The `dllimport/dllexport` attributes are not supported at the moment, due to lack of support for these features in the ELF version of gcc.

Static linking is not currently supported against Wine DLLs. As a result, the `-static`, `--static`, and `-Wl,-static` options will generate an error.

Bugs can be reported on the **Wine bug tracker** <<https://bugs.winehq.org>>.

**AUTHORS**

**winegcc** was written by Dimitrie O. Paun.

**AVAILABILITY**

**winegcc** is part of the Wine distribution, which is available through WineHQ, the **Wine development headquarters** <<https://www.winehq.org/>>.

**SEE ALSO**

**gcc(1)**, **winebuild(1)**, **wrc(1)**, **wine(1)**,  
**Wine documentation and support** <<https://www.winehq.org/help>>.