

C Compilers

Most of these [compilers](#) do C++ and C. Just rename the files to have .C extensions.

- [Microsoft Visual C++ 2008 Express](#). Not everyone likes Microsoft but there's no denying that they do provide very good code with an excellent [IDE](#). It needs [.NET](#) though you can compile for win 32 (but no MFC). We've already produced Instructions on [downloading and installing](#) it. it requires free registration.
- [Linkfrm](#). This is more of a computer science project developed at the IPD at the Universität Karlsruhe to improve the quality of software with visualization of of program structures. It includes c parser, a C compiler, which can parse C89 and C99 as well as many GCC and some MSVC extensions. The handled [GCC](#) extensions include `__attribute__`, inline assembler, computed goto and statement expressions. See [this post](#) for more about the project.
- [Open Watcom](#). Getting a bit long in the tooth and the IDE isn't great but runs on Windows 2000 (probably 98) as well as newer Windows.
- [GCC](#). The classic [open source](#) C compiler for Linux and many other operating systems (and Windows under Cygwin or Ming). A project that has been around forever. Excellent open source quality software. It doesn't come with an IDE (which are generally platform dependent) but there are loads out there eg [MonoDevelop](#) on Linux.
- [Digital Mars C/C++ Compiler](#). Their IDE costs (\$42.55) but the Basic C/C++ Win 32 compiler is free. You should also download the free STLPort as well as it contains the standard library including `<iostream>`. You'd probably also find the free [STLSoft](#) and Garbage Collection downloads useful.
- [Xcode](#). This is for Apple Macs and is their version of GCC but purely for Apple's own Mac OS Operating System. It has excellent documentation and [SDKs](#) for Mac and iPhone. If you have a Mac this is what you use.
- [Tiny C - Compiler](#). TinyCC (aka TCC) is a small fast C compiler that is meant to be self-relying: you do not need an external assembler or linker because TCC does that for you. With the aid of another library it can be used as a backend code generator. TCC compiles so fast that even for big projects Make files may not be necessary.
- [Portable C Compiler](#). This was developed from one of the earliest C Compilers and at the start of the 80s most c compilers were based on it. Portability was designed into it from the start in contrast to Dennis Ritchie's C compiler which was very hardware dependent. It's now being developed to be C99 compatible.
- [Failsafe C](#). A Japanese project from the Research Team for Software Security at the Research Center for Information Security (RCIS), National Institute of Advanced Industrial Science and Technology (AIST), JAPAN, this version of C for Linux supports over 500 functions (not C99 or Widechar). It provides complete protection against memory block over-boundary accesses making it as safe as Java and C#.

- [Pelles C](#) is a free development kit for Windows and Windows Mobile containing an optimizing C compiler, a macro assembler, a linker, a resource compiler, a message compiler, a make utility and install builders for both Windows and Windows Mobile. It also has an IDE with project management, debugger, source code editor and resource editors for dialogs, menus, string tables, accelerator tables, bitmaps, icons, cursors, animated cursors, animation videos (AVI's without sound), versions and XP manifests.
- [CC65](#) is an open source cross development package for 65(C)02 systems, including a powerful macro assembler, a C compiler, linker, librarian and several other tools. It includes support for Commodore C64, the GEOS operating system for the Commodore C64, the Commodore C128, the Commodore C16, C116 and Plus/4, the Commodore P500, the Commodore 600/700 family of computers, the Apple][, the Atari 8bit machines, the Oric Atmos, the Nintendo Entertainment System (NES), the Supervision Game Console and the Atari Lynx Console.
- [lcc](#) is a retargetable compiler for Standard C. It generates code for the ALPHA, SPARC, MIPS R3000, and Intel x86 and its successors. It's been compiling production programs since 1988 and used by hundreds of C programmers. Addison-Wesley published a book about it, documenting how it works back in 1995 that you can still buy.
- [SDCC](#) is also retargettable, and optimizing ANSI - C compiler targeting the Intel 8051, Maxim 80DS390, Zilog Z80 and the Motorola 68HC08 based MCUs. It has the ability to add inline assembler code anywhere in a function, as well reporting on the complexity of a function to help decide what should be re-written in assembler and comes with the source level debugger SDCDB.
- [Borland C++ 5.5](#)(Turbo C) is only a 8.5 MB download. It includes the compiler bcc32, 32 bit linker (tlink32), Borland Resource Compiler / Binder (brc32, brcc32), C++ Win32 Preprocessor (cpp32) and a few other utilities for importing definitions from libraries, and about dlls, exes plus a .hlp file.
- [nesC](#) is an extension to the C programming language designed to embody the structuring concepts and execution model of TinyOS. [TinyOS](#) is an event-driven operating system designed for sensor network nodes that have very limited resources (e.g., 8K bytes of program memory, 512 bytes of RAM).
- [Cc386](#) is a free Win 32 C compiler, that supports C99 (or will do soon). This has been put together by David Lindauer over 8 years and includes the source code for the compiler and tools. It also includes an IDE which provides compilation, editing and debugging. A very impress achievement for one individual.