

Q: What is NeXT's position on C++?

A: In Release 2 and later, NeXT provides a C++ compiler bundled with the extended release software. This includes a port of the GNU G++ compiler which has been extended to recognize Objective-C constructs. The compiler is based on G++ version 1.36.4, which implements version 2.0 of the C++ language, as specified by AT&T. It is distinguished from AT&T's cfront compiler in that it produces "native code" (or machine code) rather than C code. Thus it is faster than AT&T's.

The Objective-C++ compiler consists of the compiler driver (/bin/cc++), the compiler proper (/lib/cc1++), a post-linker (/lib/collect), a symbol <sup>a</sup>demangler<sup>o</sup> (/bin/g++-filt), an enhanced symbol table tool (/bin/nm++), and an enhanced profiler (/usr/ucb/gprof++). In addition, the debugger provides support for debugging C++ (see the Debugger release notes).

InterfaceBuilder does not support C++. It generates template code in Objective-C, and its palette objects are Objective-C objects. The AppKit also continues to be Objective-C based. However, you can integrate InterfaceBuilder nib files and C++ and Objective-C language code together into one program in the NeXT environment. See `../Objective_C/integrating_C++_code.rtf` for details on how to do this.

Support for C++ is intended to give developers with a significant investment in C++ object classes a migration path to the NeXT development environment. We recommend that you use Objective-C for any application you are developing from scratch.

There is an example located in `/NextDeveloper/Examples/AppKit/CalculatorLab++` which illustrates the integration of InterfaceBuilder nib files, Objective-C source code, and C++ source code into one program.

See also:

`../Objective_C/integrating_C++_code.rtf` for a technical description of how to integrate Objective C with C++.

The `/NextDeveloper/Examples/Appkit/CalculatorLab++` example.

QA591

Valid for 2.0, 3.0