



## Overview

---

C++ is the latest object-oriented programming language from Apple Computer for use with the Macintosh Programmer's Workshop (MPW<sup>®</sup>) development environment.

Apple's implementation of C++ fully supports the industry standard for object-oriented C++ as defined by AT&T's C++ Release 2.0. Apple has enhanced the language to support the Macintosh Toolbox and Operating System, Object Pascal-based

functions and procedures (such as those found in MacApp<sup>®</sup>), and the Standard Apple Numerics Environment (SANE<sup>®</sup>). MPW C++ can be debugged at the C++ source level using Apple's Symbolic Application Debugging Environment (SADE<sup>®</sup>). MPW C++ outperforms typical CFront implementations because it integrates portions of Apple's own MPW C compiler into CFront and adds precompiled headers.

Applications built using MPW C++ can run on the complete line of Apple<sup>®</sup> Macintosh<sup>®</sup> personal computers, and can take advantage of the powerful hardware found in the high-end Macintosh models. MPW C++ provides full support for object-oriented programming for C-based applications. The use of object-oriented programming techniques help to reduce development time while it increases the reliability of the resulting applications.

# Features

# Benefits

- Support for object-oriented programming

- Reduces development time
- Makes it easier to maintain applications
- Increases the reliability of applications
- Facilitates the creation of reusable code
- Offers a better model for building applications than procedural programming can provide

- Based on AT&T's C++ Release 2.0

- Provides data abstraction, multiple inheritance, and message-passing capabilities
- Offers operator overloading and protected variables within classes
- Provides strong type-checking for C-based applications

- Extensions for the Macintosh environment

- Supplies full access to the Macintosh Toolbox and operating system
- Supports Object Pascal functions and procedures in order to be compatible with MacApp
- Provides access to SANE for numerical accuracy
- Supports SADE for C++ source-level debugging
- Includes Apple's Command interface for ease of use

- Integration of CFront tool with MPW and MPWC

- Includes the MPWC scanner and preprocessor
- Allows MPWC++ to produce tokenized C, resulting in reduced build times
- Allows the use of MultiFinder® memory for building large applications
- Can automatically mark all functions and procedures in source files

- Faster than original CFront

- Supports precompiled headers, resulting in up to 50 percent reductions in build times when using MacApp
- Allows MPWC++ to produce tokenized C, resulting in reduced build times

- Support for multilingual applications

- Lets you call Object Pascal functions and procedures from MPWC++
- Allows C++ to be used with MacApp, further enhancing the programmer's productivity

- Sample programs

- Provides examples of two stand-alone, MultiFinder® "system-compatible" applications
- Provides an example of an MPW tool that is written in C++
- Can be used as learning aids or as the foundation for actual applications and tools

# Product Details

---

## Object-Oriented Language Extensions

The MPWC++ system offers object-oriented programming to programmers using C. Multiple inheritance, operator overloading, protected variables and members within classes are but a few of the object-oriented facilities of MPWC++.

## C++ Translator

C++ source code is translated into C source code by the CFront tool. The resulting C source code is then compiled by MPWC. All of this is "automated" by CPlus, an MPW script provided with MPWC++. CPlus calls both CFront and MPWC, passing appropriate parameters. Doing this results in a complete compilation of C++ source code.

MPWC++ uses the same preprocessor and scanner as MPWC. This allows MPWC++ to output tokenized C source code (as well as "standard" C source code), a capability that reduces the build times typically associated with C++.

A larger reduction in build times is achieved through the use of precompiled headers. Standard header files for C must be recompiled each time they are needed. MPWC++ provides an option for "dumping" compiled headers to a disk file the first time they are compiled. Each time these headers are needed, MPWC++ can "load" them in their precompiled form. This results in savings of up to 50 percent on build times when used with MacApp.

MPWC is available from the Apple Programmers and Developers Association (APDA).

## Source-Level Debugging

MPWC++ works with Apple's Symbolic Application Debugging Environment (SADE). SADE can be used at either the source or the assembly level to debug applications and MPW tools. During compilation, MPWC++ can create the symbol files that are needed by SADE to debug C++ applications at the C++ source-code level. This allows C++ programmers to harness the powerful scripting language of SADE during the development cycle in order to further increase application reliability and decrease development time.

SADE is available from APDA.

## Libraries

MPWC++ includes libraries for complex math and I/O stream processing. Apple has completely redone the Complex library. It retains the functionality of AT&T's Complex library and expands on it, using SANE as the basis for superior numerical accuracy.

## Unmangler

Error messages produced while linking C++ based files can be very cryptic. MPWC++ comes with a tool for converting these "mangled" error messages into messages that are much easier to read. Also included is a resource that allows the MacsBug debugger to "unmangle" C++ function names.

## Sample Programs

Three sample programs are included with MPWC++. Two of them are complete Macintosh applications and the third is a counting tool for MPW. These samples make excellent starting points for the development of other applications and tools.

## MacApp Class Library

Apple's third-generation class library, MacApp, provides an object-oriented framework that implements the standard Macintosh user interface, including scrollable, resizable windows and multipage printing. MacApp fosters development of robust, professional-quality applications by providing you with extensive memory management support, exception-handling mechanisms, support for Undo commands, and a large body of ready-to-use, high-quality code your application can inherit.

MacApp 2.0 allows programmer to use C++ in place of Object Pascal for their own applications. This is accomplished through the use of special C++ interface files, because MPWC++ can call Object Pascal-based procedures and functions.

For more information on MacApp, refer to the MacApp data sheet (order number M0243LL/A) or contact APDA.

## Training and Support

Apple has been offering courses in C++ programming since late 1989. For details, please contact

Registrar

Apple Developer University

20525 Mariani Avenue

MS75-2B

Cupertino, CA 95014

(408) 974-6215

AppleLink®: DEVUNIV



# Macintosh Programmer's Workshop C++

---

## System Requirements

To use MPW C++ you need the following:

- An Apple Macintosh Plus, Macintosh SE, or Macintosh II personal computer with at least 2MB RAM (4MB or more highly recommended). A 68020- or 68030-based Macintosh is recommended.
- A hard disk
- Macintosh system software version 6.0.2 or later
- MPW version 3.1 or later
- MPW C++ version 3.1 or later

---

## Ordering Information

MPW C++ version 3.1  
APDA<sup>®</sup> Order No. M0346LL/B

With your order, you'll receive the following:

- Two disks containing the MPW C++ translator, C++ interfaces and libraries, an unmangler for C Front error messages, and sample programs
- Macintosh Programmer's Workshop C++ Reference
- AT&T C++ Release 2.0 Product Reference
- AT&T C++ Release 2.0 Library Manual
- AT&T C++ Release 2.0 Selected Readings

---

## Apple Programmers and Developers Association

Apple Computer, Inc.  
20525 Mariani Avenue MS33-G  
Cupertino, CA 95014  
US: 1-800-282-2732  
Canada: 800-637-0029  
Int'l: 408-562-3910  
TLX: 171-576

AppleLink<sup>®</sup> address: APDA  
CompuServe: 766,2045  
MCI: POstrom  
Fax: (408) 562-3971  
GEnie: ADEVELOPER3