



Xcode

The fastest way to create Mac OS X applications.

Features

High-performance development

- You can work in your choice of development languages—Xcode supports C, C++, Objective-C, Java, and AppleScript.
- Rapid builds: Compiler optimizations and a new precompiled header mechanism deliver compile performance improvements, while a new “native” build system intelligently manages project dependencies, minimizing preprocessing.
- Fix and Continue lets you make changes to applications while they run in the debugger, reducing debugging turnaround time.
- Predictive compile begins compiling source files while you edit, reducing perceived compile times.
- Distributed build enables you to harness the power of other computers on your network to cut build times.

Streamlined development workflow

- Smart Groups automatically organize key project elements, making navigation of complex development projects easy.
- Integrated documentation window provides a single point of access for all developer documentation on Mac OS X.
- Fast search throughout the interface makes it easy to find exactly what you’re looking for.
- Code completion puts function and method calls, variables, and class names at your fingertips to reduce errors and let you focus on writing your code.
- New import tools and source tree support make it easy to migrate complex projects from CodeWarrior.

New Apple technologies

- GCC 3.3 and advanced performance tools help you optimize applications for the PowerPC G5.

Technology Brief
Mac OS X: Xcode

Developers are constantly challenged to do more in their applications. Responding to this challenge demands substantial improvements in developer productivity. At the heart of Xcode, Apple’s new integrated development environment, is a completely new approach to making developers productive. Apple has streamlined the most common and time-consuming developer tasks through the fusion of familiar user interface concepts and a unique mix of performance technologies. Xcode provides the easiest way to take advantage of new technologies from Apple, while providing migration tools that speed the transition from legacy development tools.

Xcode brings together three essential elements to make developers more productive:

- The power of proven UNIX-based development tools like GCC and GDB
- A groundbreaking user interface that delivers a new level of developer productivity and ease of use
- Unique high-performance development and debugging technologies

Xcode at Work

The Project window is a central dashboard for your development projects.

- 1 Smart Groups organize every aspect of your project and provide quick access.
- 2 The status bar tells you what’s going on in the background.
- 3 The detail view presents data for each Smart Group in tabular form.
- 4 Fast searching refines results as you type.

A solid foundation

Thousands of developers on dozens of platforms use GCC to create world-class applications every day. Xcode includes an enhanced version of GCC 3.3 that provides significant performance gains as well as features that assist in debugging. Thanks in part to a new compiler machine model developed in partnership with IBM, Xcode enables you to optimize your applications for the massively parallel execution core of the PowerPC G5 processor.

In addition to the optimizations for G5 code generation, GCC 3.3 in Xcode includes a new precompiled header mechanism that reduces build time, a new C++-like “throw/catch” exception model for Objective-C and Objective-C++, and support for Predictive Compile.

A New Developer Experience

Xcode is the first development tool built specifically for Mac OS X. By taking advantage of the Aqua user experience and drawing on the best ideas from Apple’s consumer-oriented digital media applications, Xcode delivers a new standard in developer productivity.



The Xcode user interface provides powerful organization and navigation features to get you where you want to go in your project, while providing unobtrusive status information to stay out of your way and let you focus on the task at hand.

Smart Groups

Smart Groups are a powerful new way of organizing the elements of a development project, providing easy navigation of even the most complex projects. Similar to Smart Playlists in iTunes, Smart Groups allow you to view files in the project filtered by rules.

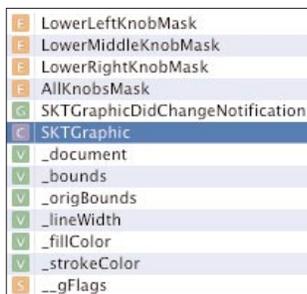
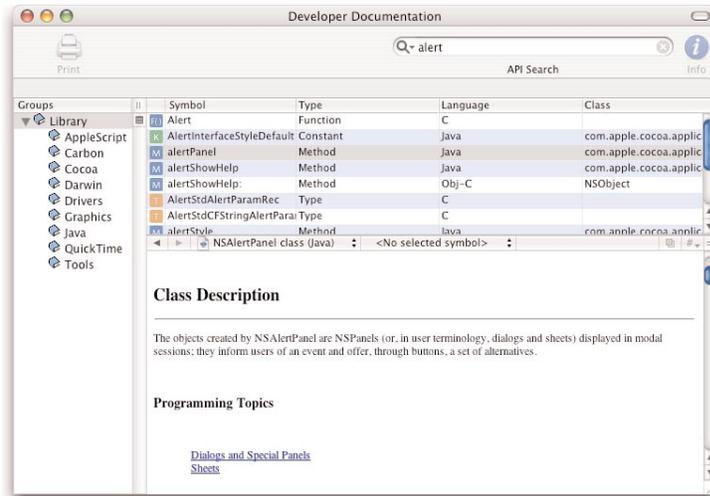
Xcode has a number of built-in Smart Groups representing common project management and organization tasks.

- **File Types.** A conventional hierarchical view of the files in your development project.
- **Targets.** Settings related to your build target, including different styles such as “development” and “deployment.”
- **Errors and Warnings.** A single place where you can list, search, and navigate the errors and warnings from a build.
- **Executables.** Executable build products, organized in one location.
- **Project Symbols.** All of the symbols (classes, function and method calls, variables) in your project in one searchable and navigable list. The detail view provides information such as symbol type (with color-coded icons for easy recognition).
- **Project Bookmarks.** Just like bookmarks for the web, project bookmarks let you save a location so that you can reach it later by simply clicking the bookmark.
- **Find Results.** Xcode saves the find results during each session in this Smart Group, where you can easily reach them again and even search within the find results to locate exactly what you’re looking for.

Help when you need it

Mac OS X provides a wide range of powerful APIs for developers, and in most cases there's an API that's perfect for your application. The challenge: How do you search the more than 500 books and 75,000 API calls in Carbon, Cocoa, Java, and UNIX to find just what you're looking for?

Xcode integrates all of the Mac OS X developer documentation in one location, with rapid searching as well as browsing, so that you can find what you're looking for quickly and easily.



Status symbols

Throughout the Xcode user interface there are lists of symbols—functions, methods, variables, classes, and other elements of your project's code. Sometimes navigating them by name doesn't give you an obvious result, or there is more than one type of symbol with the same name. That's why every symbol list in Xcode is keyed with color-coded icons identifying the symbol type. It's a small detail, but it can be a big help.

In addition, the Xcode editor window provides shortcuts to search headers and documentation while you're coding. The result is that you spend less time searching and more time writing.

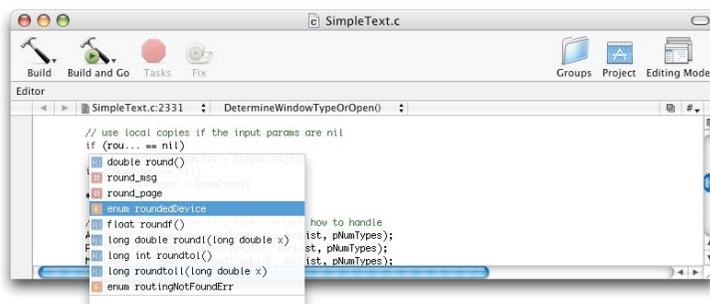
Fast search

In large, complex projects, scrolling around to find the files you're looking for can be frustrating. Find features in Xcode use a live project index, so you get instant results that are always up to date.

Powerful editing

In designing Xcode, Apple's engineers worked very hard to make sure that you can stay focused on the task at hand, yet have access the full power of the development environment. Nothing exemplifies that better than the Xcode editor window.

- You can set and edit breakpoints, as well as view warnings and errors right in the gutter of the editor window.
- The status bar provides feedback on what's currently happening in your project.
- Navigation pop-ups take you to the part of the code you're looking for.



The Xcode editor is clearly designed to help developers to stay focused and be more productive. In addition to convenient project navigation features, the editor provides syntax coloring and auto-indenting for a number of development languages, including C, C++, Objective-C, Java, AppleScript, Perl, Python, and several other scripting languages.

To maximize the benefits of developer “motor memory,” Xcode lets you configure key bindings in the editor and the rest of the IDE to help you work more efficiently. Full sets of predefined key bindings are provided for easy transition from Apple’s own Project Builder IDE and MPW as well as other popular tools such as CodeWarrior and BBEdit.

Think big

When Apple’s engineers began designing Xcode, they wanted to make sure that it could tackle the largest applications—some of which contain millions of lines of code. How can you build an application of that size without taking days? The answer is in the Distributed Build feature. You can call on other computers on your network—idle workstations or dedicated compile servers—to each build a piece of your application. Thanks to Rendezvous, it’s easy to configure and use.

The Distributed Build feature is based in part on the open source distcc project. Xcode enhances distcc with extensions to support easy configuration and service discovery using Rendezvous, as well as support for Objective-C.

Distributed builds can be configured with a few mouse clicks in the Xcode interface, and can also be used from the command-line `xcodebuild` utility or automated via AppleScript.

Built for Speed

Good performance is a requirement for any modern development tool. Compile performance is most important during the edit-compile-debug cycle, where developers often sit and wait for a build to finish in order to run the debugger and see whether their code changes have had the desired effect. This process of debugging turnaround has a number of critical phases:

- **Find the source of the problem.** Working inside the debugger, you watch variables and step through your code to find the source of an anomaly.
- **Stop the debugger.** After identifying the problem, you drop out of the debugger to begin working on the fix.
- **Edit the code.** You open the source file where you suspect there’s a problem and make the changes necessary to resolve it.
- **Compile the changes.** Just editing the code doesn’t fix the problem; you need to recompile the changes to the source code into executable form.
- **Relink the application.** Once the compiler has finished doing its work, you’re left with a number of binary object files that need to be linked into an executable. Even for a small change, this can be one of the most time-consuming phases of debugging turnaround.
- **Restart the debugger.** It’s finally time to relaunch the application in the debugger so that you can verify whether the changes you have made address the problem.

One of the most obvious problems with this cycle is that a large amount of time is spent waiting for various processes to complete. This isn’t productive time, and unfortunately it’s often not a good background task, because you need to find out quickly whether your changes work so that you can either move on to the next issue or go back and reexamine the problem at hand.

Teamwork counts

While making individual developers productive is a very important part of the Xcode story, Apple didn't lose sight of the needs of organizations. Xcode provides built-in support for source code management using either CVS or Perforce so that you get all of the benefits of team development.

Source code management integration is provided within the Xcode user interface for all common operations, including adding and removing files, check-in/check-out, and version comparison.

CodeWarrior migration

Bringing your existing Mac OS X application into Xcode is easy with the new CodeWarrior project importer. Xcode imports your project in place so it still works with your old tools. The CodeWarrior Project Import Wizard does more than read your project files—it integrates with CodeWarrior itself via AppleScript to migrate advanced project model information and settings. Support for features like source trees and cross-project references mean that you can focus on your code rather than on how to migrate project structure.

The designers of Xcode looked at the steps in debugging turnaround and systematically applied optimizations to each of them. The result is a dramatic improvement in debugging turnaround time, with a corresponding improvement in developer productivity.

| Turnaround phase | Xcode optimizations |
|--------------------------|--|
| Find the problem. | Debugging data formatters make reading data easy. The Errors and Warnings Smart Group organizes build problems for quick access and resolution. |
| Stop/Start the debugger. | Fix and Continue lets you change your application while it's still running in the debugger, eliminating the need to stop and restart. |
| Edit the code. | Context-sensitive code completion helps to prevent typos and puts important API reference information at your fingertips. |
| Compile. | The unique Predictive Compile feature begins compiling your source code while you type, dramatically reducing perceived compile times. |
| Link. | Zero Link links only the objects needed to launch the application. For the most common types of development builds, Zero Link can eliminate the linking phase of a build altogether. |

Scripting and Automation

As great as interactive development tools are, there are things like overnight builds that you'd rather not have to be around for. Xcode has a number of automation methods to suit the needs and working styles of different development organizations.

- **AppleScript automation.** The Xcode AppleScript dictionary provides access to all of the elements of project structure, meaning that scripts can control almost any aspect of the environment. You can use AppleScript to do more than just automate common build actions—you can actually extend the user interface and behavior of Xcode through scripting.
- **Command-line tools.** Should you prefer traditional UNIX scripting, Xcode provides a powerful set of command-line tools. The central tool for build automation, `xcodebuild`, gives you control over numerous build options and can automate the process from compile to install. Unlike some IDEs that tightly bind the graphical environment to the build tools to achieve best performance, Xcode provides the same infrastructure beneath both the IDE and command-line tools, ensuring great performance regardless of how you prefer to work.

In addition to automation of common tasks through Apple events and command-line tools, Xcode provides a user-configurable script menu that can execute scripts in any language and even return the results of a script operation to the text buffer. You can bind your own scripts to shortcut keys in the Xcode user interface for instant keyboard access to custom functions.

AppleScript Studio

AppleScript Studio is a part of Xcode that enables developers to create native Mac OS X applications in AppleScript. This allows for rapid development of applications that take full advantage of the Cocoa environment, building on the foundation of Apple events. Apple events enable communication between applications or even over the network, giving developers an easy way to leverage the power of hundreds of AppleScript-enabled applications on Mac OS X.

Be More Productive

Xcode represents a big step forward for Mac OS X developers. A groundbreaking new user interface, combined with the latest technologies for accelerating development, gives you the productivity gains you need to work smarter, spend less time waiting, and tackle new challenges to stay ahead of the competition.

System Requirements

To use Xcode, you need the following:

- Macintosh computer with Mac OS X v10.3 or later
- 384MB of RAM (512MB recommended)
- CD or DVD drive for installation
- 1GB of available disk space for installing applications, documentation, and examples

Mac OS X Version 10.3 “Panther”: Power of UNIX, Simplicity of Macintosh

With Xcode, Apple’s integrated development environment, Mac OS X Panther delivers a new standard in developer productivity. In addition, Panther offers more than 150 new features and innovations, including iChat AV for personal video conferencing, Exposé for instantly finding any window, and a new Finder for easy access to everything you need. It’s like having an all-new Mac.

For More Information

For more information about Xcode, visit www.apple.com/xcode. For more information about Mac OS X Panther, visit www.apple.com/macosx.

© 2003 Apple Computer, Inc. All rights reserved. Apple, the Apple logo, AppleScript, AppleScript Studio, Aqua, Carbon, Cocoa, iTunes, Mac, Macintosh, and Mac OS are trademarks of Apple Computer, Inc., registered in the U.S. and other countries. Finder, iChat, Panther, Rendezvous, and Xcode are trademarks of Apple Computer, Inc. Java is a trademark or registered trademark of Sun Microsystems, Inc., in the U.S. and other countries. Objective-C is a trademark of NeXT Software, Inc., registered in the U.S. and other countries. PowerPC is a trademark of International Business Machines Corporation, used under license therefrom. Other product and company names mentioned herein may be trademarks of their respective companies. Product specifications are subject to change without notice. October 2003 L31954A