



About Xcode Tools Compatibility and Installation

Xcode Tools Release for Mac OS X v10.4 (Tiger)

Contents

- Introduction
- Compatibility with Mac OS X Versions
- What's New
- Deprecated Features
- Installation
- Troubleshooting

Introduction

Xcode Tools includes everything you need to develop software for Mac OS X, including the programming tools, system API documentation, and Mac OS X interfaces, libraries, and example source code.

For the latest information and updates to Xcode Tools, documentation, code samples, and technical notes, please visit the Apple Developer Connection's Mac OS X pages at <http://developer.apple.com/macosx>.

We encourage developers to join the Apple Developer Connection, which provides the most convenient access to Mac OS X development resources, including technical support and pre-release software. For information visit <http://www.apple.com/developer>.

Compatibility with Mac OS X Versions

Xcode Tools 2.1 will run on Mac OS X v10.4 (Tiger) and later. It will not install or run on earlier versions of Mac OS X, and is not supported on development or preview versions of Mac OS X v10.4 (Tiger). Xcode Tools supports development for Mac OS X v10.2 (Jaguar), Mac OS X v10.3 (Panther), Mac OS X v10.4 (Tiger), or Universal binaries for PowerPC and Intel architectures on Mac OS X v10.4.1 using Mac OS X SDK support.

What's New

Xcode 2.1 has substantial changes, enhancements, and bug fixes from Xcode 2.0.

- The project file format is now much more readable and less prone to SCM conflicts. It also supports the new Build Configurations feature. Because of the magnitude of the changes, the project file extension is changed in Xcode 2.1 to ".xcodeproj". Older versions of Xcode will not read .xcodeproj files, but Xcode 2.1 can convert older project files to .xcodeproj format.
- Build Styles are replaced in Xcode 2.1 by **Build Configurations**, a more "what-you-see-is-what-you-get" approach to target settings. You can set per-configuration settings directly in the target's inspector, and subprojects are built with the same build configuration as the master project.
- Built products from different configurations are now built in **per-configuration build directories**, and can even have different product names per-configuration. That means that building your Release build doesn't overwrite your Debug build any more.
- A build configuration can be based on a **build configuration file**, a text file that provides base settings for one or more configurations. This means that your settings can be viewed, edited, searched, committed to SCM, and even compared as text files.
- Dependency analysis** is now much more reliable. You don't need to clean before building as often, and files won't be recompiled unnecessarily.
- You can configure targets and projects to use **Shared Precompiled Headers** to minimize the building of precompiled headers.
- Xcode now supports **Preprocessing Info.plist files** to perform macro expansion and substitution using common header files.
- You can now create targets in your projects that perform **Unit Testing** of other built products at build time. Using test frameworks for C, C++, and Objective-C, you can report test failures and regressions in newly-built code at the time you build it.
- The ability to **drag and drop** items in Xcode is significantly enhanced. You can now drag any file or folder into any build setting that expects a file path; drag a target into the Target Dependencies list of any aggregate target; and drag groups or file references into build phases.
- Distributed builds** have been updated to no longer allow remote execution of arbitrary programs. The list of programs that a compile server will allow is contained in /private/etc/compilers.
- There are many refinements and additions in the Xcode user interface that will be familiar to users of other IDEs. There's now a **Targets tab** in each file inspector to show and set what targets that file is included in. The "Built" column in the Groups and Files and Details views can now be clicked to **Touch** a file to cause it to be rebuilt. In the File Editor you can now **Unlock** a file that is locked in the file system. The **Build**, **Preprocess**, and **Show Assembly Code** commands now work on multiple

- selections.
- The Xcode debugger now supports **conditional breakpoints**, **breakpoint actions**, and **watchpoints** in the debugger interface. The breakpoint actions can log a message, execute a script, speak a phrase, or **visualize** your program flow in a Class Model diagram.
- Viewing variables in the debugger now has extended support for **Booleans**, **Pascal strings**, and **OSType** data types.
- The debugger console now performs **tab-completion** of symbol names in the current context.
- Source Code Management now uses an **Online/Offline** master switch (as is used in Mail) to control connecting and disconnecting from an SCM system. All SCM systems now have basic support for **wraper** (project files, nibs, model files, RTFD documents, etc.).
- The **AppleScript** dictionary in Xcode continues to grow, in this version adding commands to **make new target**, **upgrade project file**, and perform various SCM operations, .

WebObjects 5.3 The WebObjects developer tools are now included with the Xcode Tools. For more information see: <http://developer.apple.com/releasesnotes/WebObjects/index.html>

- EOModels can now be edited within Xcode with a new EOModeler plugin that integrates CoreData modeling tools.
- WebObjects Builder has UI enhancements and generates HTML 4.0.1 code.
- WebObjects Runtime now supports HTML 4.0.1.
- NSArray, NSDictionary and NSSet now implement the java.util.Collection interfaces.
- Axis 1.1 has been integrated with the Direct To Webservice feature.
- WebObjects is qualified against Oracle 10g using the 10.1.0.2 jdbc drivers.

Automator

- AppleScript action projects can now be debugged using the AppleScript debugger.
- A new Automator Shell Script Action project template has been added, allowing the creation of Automator actions using your favorite scripting language.

gcc 4.0

- Bug fixes to improve stability and correctness.
- Synced with the released FSF version.

Deprecated Features

- Build Styles have been removed from Xcode and replaced with Build Configurations.
- The AppleScript **build** and **clean** commands no longer support a **with settings** optional parameter.
- gcc no longer supports the **-fnew-ra** flag.

Installation

Installing Xcode Tools The XcodeTools.mpkg package contains several separate sub-packages in the Packages folder that allow flexibility in installation and updating:

- Developer Tools (DeveloperTools.pkg) - This package contains Xcode, gdb, and the other tools necessary for Macintosh development.
- GCC 4.0 (gcc4.0.pkg) - This is the latest version of gcc. This compiler is necessary for 64-bit development.
- GCC 3.3 (gcc3.3.pkg) - This package contains the gcc 3.3 compiler.
- Software Development Kits - These packages provide the header files and link libraries necessary for Macintosh development. All Macintosh development require the Mac OS X SDK, which provides the API for basic Macintosh technology such as Carbon, Cocoa, and command-line tools. The other SDK packages provide API for extended Macintosh technologies, such as QuickTime or OpenGL, etc.
- Java 1.4 Developer Tools - These packages are used when developing Java 1.4 applications for Macintosh.
- Developer Documentation (DevDocumentation.pkg) - This package contains the core Macintosh developer documentation for technologies such as Carbon, Cocoa, QuickTime, WebKit, etc.
- Developer Examples (DevExamples.pkg) - The package contains example code for a wide variety of Macintosh technologies. Note that these examples may require any of the above Software Development Kits to be installed before they will compile.
- Packages for Mac OS X v10.2, Mac OS X v10.3, and Mac OS X v10.4 (Tiger) to allow cross development. (MacOSX10.2.8.pkg, MacOSX10.3.9.pkg, and MacOSX10.4.pkg). These can be installed together or independently, and can be installed separately from the installation of the rest of Xcode. (CrossDevelopment.mpkg)
- CHUD (CHUD.pkg) - The Computer Hardware Understanding Developer Tools.
- WebObjects (WebObjects.mpkg) - Xcode 2.1 contains the packages necessary for WebObjects 5.3 development.

For more information see <http://developer.apple.com/tools/performance/>

Installation instructions for Xcode 2.1

- 1) Boot into Mac OS X.
- 2) Insert the Xcode Tools DVD or the Xcode Tools CD 1.
- 3) Double-click on the XcodeTools.mpkg file located at the top-level.
- 4) Follow the instructions in the installer. If you need to install the Cross-Development packages, CHUD, WebObjects, or the Java 1.4 Reference Documentation, you must select "Custom Install", and check those packages.
- 5) Authenticate as the administrative user. The first user you create when setting up Mac OS X has administrator privileges by

default.

NOTE: Xcode Tools MUST be installed on the same hard drive partition from which you booted Mac OS X. The Installer prevents installation on any other partition.

If you encounter any problems in installation, including not having enough disk space to do the installation, please refer to the "Troubleshooting" section below.

Once you have installed the Xcode Tools package, you can access the documentation by launching Xcode and choosing any of the items in the Help menu, or by launching Help Viewer and clicking on the Developer Center link at the bottom of the Help Center welcome page. Developer applications, such as Xcode and Interface Builder, are installed in /Developer/Applications.

It is highly recommended that you copy any of the examples in /Developer/Examples to another directory before opening them and modifying them. If a particular example is modified such that it no longer works, you must remove the example, and reinstall the entire examples package by double-clicking on Packages/DevExamples.pkg on the Xcode Tools installation disk.

Installation instructions for WebObjects 5.3

- 1) Double click on the Xcode installer
- 2) Select 'Continue' at the Welcome panel
- 3) Select 'Continue' after reading the Software Licensing Agreement
- 4) Select your destination drive
- 5) Select 'Custom Install' and select the 'WebObjects' packages.
- 6) Select the 'Upgrade' button

NOTE: You do not need a previous version of WebObjects on your system before installing WebObjects 5.3

Troubleshooting

- If you have a small amount of disk space on your boot volume, you can set up a symbolic link for any part of the /Developer hierarchy. The installer will respect the symbolic link and install the files correctly. For more information on symbolic links, see "man ln".
- In some configurations upgraded from Jaguar to Tiger, the Project Builder IDE and Xcode IDE will both be available in the /Developer/Applications folder. The Project Builder IDE is not supported for use on Panther, but it will work for some uses. Some of the issues in its operation are subtle and difficult to detect. It is recommended that you discontinue use of Project Builder and use Xcode instead.
- Interface Builder can be used with Project Builder, or Xcode, but not both at the same time. If you use Interface Builder and need to switch between Project Builder and Xcode, you must quit Interface Builder and relaunch it to switch to the other IDE.
- Uninstalling Xcode Tools

To uninstall Xcode Tools on your system, type in a Terminal window:

```
$ sudo perl /Developer/Tools/uninstall-devtools.pl
```

To uninstall from the CD or disk image that contains Xcode Tools, type in a Terminal window:

```
$ sudo perl /Volumes/Xcode\ Tools/Utilities/uninstall-devtools.pl
```

NOTE: Running the uninstall script from Xcode Tools will uninstall any previous installations of Mac OS X Developer Tools or Xcode Tools releases.

NOTE: After running the uninstall script, you should restart your system before installing the Xcode Tools again. The uninstaller that ships with previous versions of Xcode Tools or the Developer Tools will not clean everything off of your system properly. You should either use the one on the Xcode Tools 2.1 DVD, or you should use the one installed when you installed Xcode Tools 2.1.