



About Xcode Developer Tools

Xcode 3.0 Developer Tools for Mac OS X v10.5 (Leopard)

Contents

- Introduction
- Compatibility with Mac OS X Versions
- What's New
- Installation
- Troubleshooting
- Deprecation Notice

Introduction

Xcode 3.0 developer tools for Mac OS X v10.5 (Leopard) gives developers a head start in creating applications for the next major release of Mac OS X. This Xcode distribution includes everything required to develop software for Mac OS X v10.5, including the Xcode IDE, programming tools, system API documentation, Mac OS X interfaces, libraries, and example source code. For the latest information and updates to the Xcode developer tools, documentation, code samples, and technical notes, please visit the Apple Developer Connection (ADC) tools pages at <http://developer.apple.com/tools/>.

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://developer.apple.com/>.

Compatibility with Mac OS X Versions

The Xcode 3.0 developer tools for Mac OS X v10.5 (Leopard) will run exclusively on the version of Mac OS X v10.5 with which it is distributed. The Xcode 3.0 developer tools supports development targeting Mac OS X v10.3 (Panther) and Universal applications for Mac OS X v10.4 (Tiger) and Mac OS X v10.5 (Leopard) using the Mac OS X SDK support.

What's New

Xcode 3.0 developer tools for Mac OS X v10.5 (Leopard) introduces many new features to improve developer productivity, provides greater debugging control, and leverages Leopard's new system-wide 64-bit capabilities.

Xcode Installing on Mac OS X 10.5

The Xcode 3.0 developer tools release now supports installing multiple versions of the Xcode developer tools on Mac OS X 10.5. When installing, the default location for the Xcode 3.0 developer tools continues to be `/Developer`; however, you may install Xcode 3.0 developer tools to any other directory or volume, including external drives. The Xcode directory can also be named something other than "Developer". As with previous releases, the folder hierarchy inside the installed Xcode directory should not be altered or rearranged.

Throughout this document, `<Xcode directory>` will refer to the path in which the developer tools are installed.

NOTE: Now that developer frameworks and libraries are laid down inside the movable `<Xcode directory>` it is not possible to relocate or copy applications in the `<Xcode directory>/Applications` hierarchy since the developer frameworks they depend on will not be found. If you need to refer to a developer application from a different directory, it is best to use an alias or symlink.

You may also have multiple versions of the Xcode developer tools (2.5 or later) installed on a single system running Mac OS X v10.5 (Leopard). The Xcode IDE will use the SDKs, build tools, and auxiliary applications from the Xcode directory it is launched from whenever possible, rather than those from the

root system.

The XcodeTools 3.0 installer also optionally installs the standard system development tools and interfaces into `/usr`, so conventional makefile- and config-based builds will operate correctly.

NOTE: There is no support for more than one version of these system (command line) tools per system; the last installed set of system tools replaces any previously installed set.

To support these changes there has been a re-structuring of the Xcode directory to move developer tools content out of the underlying system and into a single top-level folder. Some of the more noticeable changes are:

- The Xcode directory now has a `usr` subdirectory that includes all command line developer content. For example, now `xcodebuild` is found at `<Xcode directory>/usr/bin/xcodebuild`, where previously it was located at `/usr/bin/xcodebuild`. Similarly, developer man pages, libraries, and other files can be found in the appropriate locations under `<Xcode directory>/usr`.
- In order to continue to support the UNIX model of developer tools being found in the boot volume and provide backward compatibility with external build systems, Xcode includes an optional install of this content into the boot volume. System headers and libraries are also provided for those projects that have not migrated to using SDKs.
- In an effort to better support multiple installations of the Xcode developer tools, the versions of `xcodebuild`, `xcodeindex`, `instruments`, `ibtool`, `opendiff`, and `agvtool` placed in `/usr/bin` are now shim scripts that work with the `xcode-select` tool (also located in `/usr/bin/`) and reference the "default" version of the Xcode developer tools. Consult the `xcode-select` man page for more information.
- Previously, supporting files for Xcode and other developer applications were located in `/Library/Application Support/Apple/Developer Tools`. Now that content is located inside the Xcode directory in `<Xcode directory>/Library/`, with each application having its own folder (so Xcode support files are located at `<Xcode directory>/Library/Xcode`). In addition, we also support finding support files outside the Xcode directory at `/Library/Application Support/Developer/<tools version>` for support files needed for a specific version of the tools (`/Library/Application Support/Developer/3.0` for the Xcode 3.0 developer tools) and `/Library/Application Support/Developer/Shared` for support files that are not specific to a given tools version. We encourage users to place additional support files (like custom file or project templates) inside `<Xcode directory>/Library/` to allow the support files to move with the folder instead of being tied to the boot volume.
- Since we have added a `usr` folder hierarchy to the Xcode directory, `<Xcode directory>/Tools` is being deprecated in Leopard Xcode developer tools, and will be removed in a future release of the tools. References to tools that used to exist in `<Xcode directory>/Tools` should be updated to find their content in `<Xcode directory>/usr/bin`.
- Deprecated in a previous release, `ocvs` is no longer shipped as part of Xcode. Users of wrapped CVS repositories should transition to Subversion or CVS. In a related note, Subversion now ships as a part of Mac OS X v10.5 proper.
- The Java reference documentation for Java 1.4.2 and J2SE 5 is no longer part of the Xcode developer tools install and will be made available as a separate download on the ADC website. The Core Java Reference Library is still available via RSS subscription in Xcode's documentation window.
- The `ant`, `junit`, and `maven` command line tools are now part of Mac OS X v10.5 proper, so they are no longer shipped as part of Xcode developer tools.
- Jar Bundler and Applet Launcher have moved to `/usr/share/java` and are part of Mac OS X v10.5 proper, but are still accessible via symlinks in `<Xcode directory>/Applications/Utilities`.

Developer Productivity

- Objective-C 2.0: The Xcode IDE and toolset fully support the new Objective-C 2.0 features, including garbage collection, properties, and for...in iteration syntax. Note that Objective-C 2.0 is

only compatible with the Mac OS X v10.5 (Leopard). The Xcode IDE is compiled and runs as a fully garbage collected application. For additional information, see the "The Objective-C 2.0 Programming Language" guide, available via Documentation in the Help menu of Xcode.

- **New Xcode Editor:** New features include improved performance when loading large files and a streamlined edit/build/debug process. For instance, errors and warnings can now be viewed inline within your code in "message bubbles", allowing you to easily correct mistakes without switching windows. Additionally, a new feature called Code Focus combines new code folding features along with visual cues to highlight the block structure of your code.
- **Refactoring for Objective-C:** Refactoring enables wide-scale restructuring of your project and class design in a single, safe operation.
- **Project Snapshots:** Enables you to save your complete project state, or restore your project to a previous state. Together with refactoring, these provide a safe mechanism to make dramatic changes to your entire project.

Debugging Control

- **In-Place Debugging:** Now any editor window in Xcode can be used as a debugging interface. When you click "Go" to launch your application, debugging controls such as stepping and a thread selector will appear at the top of the editor window. Hovering over a variable in your code will use the new debugger tooltip feature to display its current value.
- **DWARF:** DWARF is now the default debugging format for both new and existing projects. If you do not want to use DWARF, you can manually change the debugging format in the project by setting the "Debug Information Format" build setting to Stabs.

64-bit Leopard Support

- Mac OS X v10.5 (Leopard) significantly enhances support for 64-bit applications, including UI frameworks. As part of this, the Xcode 3.0 developer tools enables creation of 32 and 64-bit Universal applications for Intel and PowerPC architectures.

Interface Builder 3.0

- Interface Builder has undergone its first major update in over a decade and includes numerous improvements to the application appearance and workflow. The new version features major improvements in areas like Object palettes, Inspectors, and creating connections between objects. It also has support for adding Core Animation–based behaviors to elements. Finally, Interface Builder 3 makes it easy to integrate custom objects and refactor your nib files.

Performance Analysis

- **Instruments:** New in Leopard, Instruments provides a wide variety of capabilities, conveniently packaged as separate instruments, to monitor, visualize, and correlate data collected from your application and the OS over time. ObjectAlloc.app and Sampler.app have been removed from developer tools and replaced by Instrument Templates with similar functionality. Instruments also leverages DTrace and allows you to create your own custom DTrace instruments. This enables you to quickly spot problems and relationships within your application.

Dashcode

- Dashcode is a new application that includes everything you need to create great Dashboard widgets. It provides a number of templates, starting you off with a useful, fully-functioning widget. From there you can change the look and controls within your widget using the canvas. More advanced developers can write new code with the source code editor with Code Sense and debug it with the JavaScript debugger. Dashcode also takes care of the packaging and deployment of your widget to Dashboard once you've perfected it.
- The Dashcode installer package is included at the top level of the Xcode Tools folder in the

Leopard installation DVD and is installed by default. Dashcode installs into `<Xcode directory>/Applications`.

Please see the complete Xcode 3.0 release notes for more detailed information on the above features, and the many other new features in this release.

Installing Xcode Developer Tools

The XcodeTools installer has been changed for Leopard to allow for simple installation of the Xcode developer tools. This reorganized structure gives you all the options you need when installing, and obviates the need to install content from individual packages.

The installation choices in the "Customize..." panel have been reorganized into six items:

- Developer Tools Essentials - Xcode, Interface Builder, Instruments, Dashcode, Quartz Composer, GCC 4.0.1, and other essential developer tools. Also installs the Mac OS X v10.4 (Universal) and Mac OS X v10.5 SDKs, as well as sample source code.
- Core Reference Library - An Xcode documentation set consisting of Apple's Mac OS X and Developer Tools technical resources, including Guides, Reference, Release Notes, Sample Code, Technical Notes, and Technical Q&As.
- Developer Tools System Components - CHUD Tools (including Shark) for measuring and optimizing software performance on Mac OS X, hardware bringup, and system benchmarking. Also includes support for enabling distributed builds and Instruments/DTrace integration. Note: the CHUD Tools are installed into `/Developer` on the boot volume.
- UNIX Development Support - Optional content to allow command-line development from the boot volume. Installs a duplicate of the GCC compiler and command line tools included with the Developer Tools Essentials package into the boot volume. It also installs header files, libraries, and other resources for developing software using Mac OS X into the boot volume. This package is provided for compatibility with shell scripts and makefiles that require access to the developer tools in specific system locations. This content is not relocatable and will only be installed onto the boot volume.
- Mac OS X v10.3.9 Support - Adds support for developing applications that target Mac OS X v10.3.9 APIs. Includes the Apple version of the GCC 3.3 compiler and the Mac OS X v10.3.9 SDK. Note: GCC 3.3 is not relocatable and is installed on the boot volume. This package is not selected by default.
- WebObjects - Installs WebObjects development tools, examples, and documentation. Note: WebObjects is not relocatable and is installed into `/Developer` on the boot volume. This package is not selected by default.

Step-by-Step Instructions

1. Boot into a partition with the Mac OS X v10.5 (Leopard) installed.
2. Insert the Mac OS X v10.5 (Leopard) DVD.
3. Double-click on `XcodeTools.mpkg`, located inside `Optional Installs/Xcode Tools`.
4. Follow the instructions in the Installer. If you need to install Mac OS X 10.3.9 Support or WebObjects, 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.

Once you have installed the Xcode developer tools, you can access the documentation by launching Xcode and choosing any of the items in the Help menu. Developer applications such as Xcode, Instruments, and Interface Builder are installed in `<Xcode directory>/Applications`.

Uninstalling Xcode 3.0 Developer Tools

To uninstall Xcode developer tools on the boot volume along with the Xcode directory, from a Terminal window type:

```
$ sudo <Xcode directory>/Library/uninstall-devtools --mode=all
```

If you just want to remove the underlying developer content on the boot volume, but leave the Xcode directory untouched, from a Terminal window type:

```
$ sudo <Xcode directory>/Library/uninstall-devtools --mode=systemsupport
```

If you just want to just remove the UNIX development support on the boot volume, but leave the Xcode directory and it's supporting files untouched, from a Terminal window type:

```
$ sudo <Xcode directory>/Library/uninstall-devtools --mode=unixdev
```

Finally, to just uninstall the Xcode directory you can simply drag it to the trash, or from a Terminal window type:

```
$ sudo <Xcode directory>/Library/uninstall-devtools --mode=xcodedir
```

NOTE: The uninstaller that ships with previous versions of the Xcode developer tools will not clean everything off of your system properly. You should either use the one provided on the Mac OS X v10.5 (Leopard) DVD, or you should use the one installed when you installed the Xcode 3.0 developer tools.

Deprecation Notice

NOTE: As of the release of Xcode 2.4, the Cocoa Java bridge has been deprecated. This means that, while still supported with Xcode 2.4, future releases of Xcode may not support the bridge or other dependent features. Due to this, the following additional WebObjects developer applications are also being deprecated:

- EOModeler
- EOModeler Plugin
- WebObjects Builder
- WebServices Assistant
- RuleEditor
- WOALauncher

This also means that the Xcode Java bridge templates have been deprecated, and should not be used for new Java bridge-based development.

The following directories inside the Xcode directory are deprecated and will be removed in a future version of the Xcode developer tools:

- <Xcode directory>/Tools (content will move to <Xcode directory>/usr/bin)
- <Xcode directory>/Makefiles (content will move to <Xcode directory>/usr/share/makefiles in a future release)
- <Xcode directory>/Private (content will be removed in a future release)

© 2007 Apple Inc. All rights reserved. Xcode, Apple, the Apple logo, Mac, Mac OS, and Macintosh are trademarks of Apple Inc., registered in the U.S. and other countries.