



About Xcode Developer Tools

iPhone SDK and Xcode 3.1.2 Developer Tools for iPhone OS 2.2.1

Contents

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

Introduction

iPhone SDK for iPhone OS 2.2.1 is an update and bug fix release to the iPhone OS developer tools. This release includes new, optional GCC and LLVM compilers, updated project assistant, a simplified toolbar, and additional security enhancements.

The iPhone SDK includes the complete Xcode 3.1.2 developer tools for Mac OS X development as well as the SDK components for iPhone OS development.

We encourage developers to apply to the iPhone Developer Program for access to additional support resources, as well as provisioning resources to enable development directly on an iPhone or iPod touch. For more information visit:

<http://developer.apple.com/iphone/program/>

Compatibility with Mac OS X Versions

iPhone SDK requires an Intel-based Mac running Mac OS X Leopard version 10.5.4 or later. Xcode 3.1.2 is also available separately for Mac-only development and is compatible with Intel and PowerPC Macs running Mac OS X Leopard 10.5 and later.

What's New

- SDK support for targeting non-Mac OS X platforms, including iPhone OS SDK.
- GCC 4.2 & LLVM GCC 4.2 optional compilers for use with Mac OS X 10.5 SDK
- Updated assistants to create new projects, targets, and source files
- Toolbar uses a single popup to choose platform, target, and debug/release
- Integrated SCM support now works with Subversion 1.5

About SDKs

Xcode 3.1 extended the concept of an SDK (software development kit) to enable Xcode to build, debug, and deploy applications for platforms completely different from that of the host computer. This is possible by bundling a complete set of low-level compilers and tools within the SDK, in addition to platform frameworks.

Within Xcode 3.1 you can choose specific Mac OS X SDKs to create applications that run on particular versions of Mac OS X, or use the iPhone SDK to build applications for the iPhone Simulator, the iPhone, or iPod touch. The tools included with the SDK will upload your application to a device, and debug directly from the IDE.

Xcode for Mac development is included as part of the iPhone SDK installer, and is also available separately with only the Mac OS X SDKs for Mac-only developers.

New in Xcode IDE

- The assistant interface has been completely revamped, making it easier to create "New Project", "New Target", and "New File" resources. This release of Xcode also adds new assistant templates for both Mac OS X and iPhone OS applications.
- A new Overview toolbar item lets you see and set the project's active Target, Configuration, Executable, Architecture, and SDK with a single control.
- It is now possible to set a global "override" SDK as the Active SDK, making it easy to test the build of your entire project against a different platform.
- New interface shortcuts have been added to the Build Setting inspectors making it easy to set the desired compiler, architectures, and SDKs
- You can now add Libraries and Frameworks directly into the Target window and relate them to the project's SDK.
- The Xcode UI now allows you to designate a library or framework as weak-linked, to support building with newer SDKs while targeting older OS versions.
- New "Edit all in scope" option within the Xcode source editor makes it easy to update all instances of the selected symbol within the current scope of code.
- You can now have conditional build settings assigned for any combination of architecture and SDK.
- LLVM GCC 4.2 is a new optional compiler based on the LLVM.org open source project. LLVM GCC 4.2 provides an LLVM-based back-end optimizer using the GCC 4.2 front-end parser. This compiler is both source and binary compatible with GCC 4.2 and requires the Mac OS X 10.5 SDK or "Current OS" SDK.
- GCC 4.2 is a new optional compiler with stricter C and C++ language enforcement, security enhancements, performance optimizations, and bug fixes over GCC 4.0. Requires the Mac OS X 10.5 SDK or "Current OS" SDK

New in Interface Builder 3.1

- New drag & drop functionality allows easily re-parenting objects with the normal drag gesture
- The document outline view has been enhanced to support full drag re-ordering.
- A new string table interface allows localizers to make quick text changes across documents.
- Generates XIB files that are more diff friendly while retaining full backwards compatibility with Interface Builder 3.0.
- Now supports previewing image resource content in Carbon controls.

New in Dashcode

- You can now use Dashcode to create web applications optimized for the iPhone, as well as Dashboard widgets.

For the latest security information visit: <http://support.apple.com/kb/HT1222>

For more detailed information please see the complete Xcode 3.1.2 release notes.

Installation

The Xcode installer provides three main options for configuring the installation:

- **Developer Tools Essentials.** Contains the essential components of Xcode. It's installed in `/Developer` by default or another directory of your choice, including external volumes.
- **System Tools.** Optional tools, includes Shark and other CHUD performance tools. Its components are always placed in `/Developer`.
- **UNIX Development Support.** Optional command-line tools used for UNIX-based development. Its components are always placed in `/usr`.

Throughout this document `<xcode>` refers to the path in which the Developer Tools Essentials components are installed.

You can now have both Xcode 3.1 and Xcode 3.0 installed. The Xcode installer supports installing multiple versions of the Developer Tools Essentials components on Mac OS X v10.5. You can move or rename the `<xcode>` directory, but must not alter its internal structure. If you need to refer to a developer application from a different directory, it is best to use an alias or symbolic link for the full path.

NOTE: Only one version of the System Tools and the UNIX Development Support components can be installed on a computer at a time. The last installed set of these components replaces any previously installed set.

Installing iPhone SDK Developer Tools

The Xcode installer has been changed for Leopard to allow for simple installation of the 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 five items:

- Developer Tools Essentials - Xcode, Interface Builder, Instruments, Dashcode, Quartz Composer, GCC 4.0.1, GCC 4.2, LLVM GCC 4.2, and other essential developer tools. Also installs the Mac OS X 10.4 (Universal) and Mac OS X 10.5 SDKs, as well as sample source code.
- System Tools - 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 onto the boot volume at `/Developer`.
- 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 onto the boot volume at `/Developer`. This package is not selected by default.

Step-by-Step Instructions

Note: By default, all versions of Xcode (and iPhone SDK) install in `/Developer` with previous versions automatically upgraded. If you want to keep a previously installed version of Xcode and still install Xcode 3.1.2, you must either do a "Custom Install" and specify a new `<Xcode>` directory name (e.g. `/Xcode3.1.2`) or you must move the pre-existing `/Developer` directory to a new name (e.g. `/Xcode3.0`) before you install Xcode 3.1.2. See step 5 below.

1. Boot into a partition with the Mac OS X v10.5.4 or later installed.
2. Download the iPhone SDK DVD software.

3. Double-click on iPhone SDK.
4. Follow the instructions in the Installer.
5. To install Xcode 3.1.2 in a directory other than the default /Developer, you must specify a new <Xcode> directory name on the "Custom Install" pane; e.g. Xcode3.1.2.
 - Select the folder icon under "Location" next to the "Developer Tools Essentials" package name.
 - Select "Other..." at the top of the pop-up.
 - Navigate to the location for the new <Xcode> folder. Select "New Folder".
 - Enter a folder name in the "New Folder" window; e.g. Xcode3.1.2. Select "Create".
 - Finally, select "Choose" in the "Install Xcode Tools" pane.
6. If you need to install Mac OS X 10.3.9 Support or WebObjects, you must select "Customize", and check those packages on the "Custom Install" panel.
7. 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>/Applications.

Uninstalling Xcode Developer Tools

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

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

To remove the underlying developer content on the boot volume, but leave the <Xcode> directory and supporting files untouched, from a Terminal window type:

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

To just remove the UNIX development support on the boot volume, but leave the <Xcode> directory and supporting files untouched, from a Terminal window type:

```
$ sudo <Xcode>/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>/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 installed within the Xcode 3.1.2 developer tools.

Deprecation Notice

NOTE: As of the release of Xcode 2.4, the Cocoa Java bridge has been deprecated and is no longer supported. Due to this, the following additional WebObjects developer applications have been removed:

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

Xcode Java bridge templates have been deprecated, and should not be used for new Java bridge-based development.

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

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

© 2008 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.