About Xcode Developer Tools

Xcode 3.2.1 for Mac OS X 10.6 Snow Leopard

Contents

Introduction

Xcode 3.2 developer tools for Mac OS X 10.6 Snow Leopard require Snow Leopard as the host platform, but can target past OS releases of Mac OS X including 10.4 Tiger and 10.5 Leopard.

Xcode 3.2 includes a new Build and Analyze feature that adds Message Bubbles that call out code problems found with the new static code analyzer. Additionally, the analysis is reported within the new Build Results window.

We encourage developers to join the Apple Developer Connection. It provides the most convenient access to Apple Inc. development resources, technical support and pre-release software. For information visit `http://developer.apple.com/`.

What's New

Xcode 3.2 developer tools for Mac OS X 10.6 has the following improvements:

- Static code analysis is fully integrated within the Xcode IDE via the Build and Analyze option under the Build menu or via custom build settings
- New Build Results window collects warnings and errors in a persistent log
- GCC 4.2 is the primary system compiler for the 10.6 SDK
- The next-generation LLVM compiler is included using two different front ends - the Clang compiler is a new, leading-edge front end that also provides the basis for the new static analyzer. For maximum compatibility, the GCC LLVM compiler utilizes the LLVM back-end with the GCC 4.2 front-end parser.
- New template chooser and project assistants
- New opening window for quick access to recent projects

Xcode 3.2 Installer

The Xcode 3.2 installer provides five main options for configuring the installation:

- **Essentials.** Contains the essential components of Xcode, installed by default in `/Developer` or another directory of your choice, including external volumes.
- **System Tools.** Includes CHUD performance tools. Its components are always placed in `/Developer`.
- **UNIX Dev Support.** Command-line tools used for UNIX-based development. Its components are always placed in `/usr` - only one version installed at a time.
- **Documentation.** Configures Xcode to automatically download updates to developer documentation. If left unchecked the documentation will be read from the web instead of the local disk. Can be changed via Preferences.
- **Mac OS X 10.4 Support.** This option, off by default, adds support for developing applications that target Mac OS X 10.4 APIs.

Developer Productivity

- A new Overview toolbar item lets you see and set the project's active Target, Configuration, Executable, Architecture, and SDK with a single control.
- Setting Architectures and SDKs are now done with provided pop-up lists rather than manually entered strings or long paths.
- It is now easier to add known frameworks to your project
- The source code editor now allows you to "Edit All In Scope", to edit all instances of a symbol in a particular scope.
- Build settings can be set for any combination of architecture and SDK.
- Using the Clang compiler requires the 10.6 SDK
- Using GCC LLVM or GCC 4.2 requires the 10.5 SDK or 10.6 SDK

Dashcode

- Dashcode now supports creating of web applications optimized for Safari and mobile Safari, in addition to Dashboard widgets. Choose the Safari templates to create these optimized interfaces for your site content.

Please see the complete Xcode 3.2 release notes for more detailed information.

About Software Developer Kits and Platform Targets

A Software Developer Kit (SDK), an extension to Xcode, lets you develop software for a particular operating system platform. In addition to Mac OS X software products, you can now develop applications that run on other operating system platforms. Just as you use the default Mac OS X SDKs to create applications that run on Mac OS X, other operating system SDKs will allow you to build applications for those operating systems, upload them to a device and debug them directly from the IDE.

To develop software for a specific version of a platform, you use specialized SDKs. For example to target Mac OS X 10.4 or later you use the Mac OS X 10.4 SDK.

Installation

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

The Xcode installer supports installing multiple versions of the Essentials components on Mac OS X 10.6. 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 or the full path.

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

Installing Xcode Developer Tools

The Xcode installer for Mac OS X 10.6 allows for simple installation of the Xcode developer tools. This 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 are organized into five items:

- 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 bring-up, and system benchmarking. This choice also includes Instruments and DTrace integration. CHUD Tools are always installed into `/Developer` on the boot volume.
- UNIX Dev 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 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.
- Documentation **-** This option, selected by default, configures Xcode to automatically check for and download updates to developer documentation. If unchecked the documentation will not be updated. Instead it will be accessed directly from the web and is unavailable when disconnected from the Internet. This option can be changed at any time within Xcode Preferences by checking the box labeled "Check & Install Updates Automatically" in the Documentation Sets portion of the Documentation pane.

- Mac OS X 10.4 Support **-** This option, off by default, adds support for developing applications that target Mac OS X 10.4 APIs.

Step-by-Step Instructions

Note: The default install location for Xcode 3.2 is still `/Developer`. If you want to keep a set of developer tools previously installed in `/Developer` and do a new install of Xcode 3.2, you must either do a "Custom Install" and specify a new <Xcode> directory name (e.g. Xcode3.2) or you must move the pre-existing `/Developer` directory to a new name before you install Xcode 3.2. See step 5 below.

1. Boot into a partition with Mac OS X 10.6 installed.
2. If necessary, download the Xcode 3.2 Developer DVD software.
3. Double-click on `Xcode.mpkg`.
4. Follow the instructions in the Installer.
5. To install Xcode 3.2 in a directory other than the default `/Developer`, you must specify a new <Xcode> directory name on the "Custom Install" pane; e.g. Xcode3.2.
   - Select the folder icon under "Location" next to the "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. Xcode3.2. Select "Create".
   - Finally, select "Choose" in the "Install Xcode" pane.
6. 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 Dev Support components 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 use the one installed when you installed the Xcode 3.2 developer tools.

Deprecation Notice

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`)