
Xcode Installation Guide

[Tools > Xcode](#)



2008-05-27



Apple Inc.
© 2008 Apple Inc.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

.Mac is a registered service mark of Apple Inc.

Apple, the Apple logo, Dashcode, Leopard, Mac, Mac OS, Macintosh, Quartz, WebObjects, and Xcode are trademarks of Apple Inc., registered in the United States and other countries.

Intel and Intel Core are registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Java and all Java-based trademarks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

UNIX is a registered trademark of The Open Group

Simultaneously published in the United States and Canada.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

Introduction **Introduction to Xcode Installation Guide** 7

Organization of This Document 7

Chapter 1 **Xcode Installation Basics** 9

The Xcode Tools Disk 9

Xcode Tools Installer 10

Installation Instructions 12

The Xcode Tools Uninstaller 12

Chapter 2 **Xcode Installation Details** 15

The Install Destinations and Directories 15

Customizing Xcode Installations 16

Xcode 2.5 Support 17

Intel Compiler Support 17

Xcode Directory Structure 17

 Changes in the Xcode Directory 18

Xcode Command-Line Tools Installation 19

UNIX Command-Line Tools Installation 20

Document Revision History 23

Figures and Tables

Chapter 1 **Xcode Installation Basics** 9

- Figure 1-1 Xcode Tools disk 10
- Figure 1-2 Xcode Tools installer 11

Chapter 2 **Xcode Installation Details** 15

- Table 2-1 Xcode Tools install destinations 15
- Table 2-2 Xcode Tools install directories 16

Introduction to Xcode Installation Guide

Important: This is a preliminary document. Although it has been reviewed for technical accuracy, it is not final. Apple is supplying this information to help you adopt the technologies and programming interfaces described herein. This information is subject to change, and software implemented according to this document should be vetted against final documentation. For information about updates to this and other developer documentation, you can check the [ADC Reference Library Revision List](#). To receive notification of documentation updates, you can sign up for a free Apple Developer Connection Online membership and receive the bi-weekly ADC News e-mail newsletter. (See <http://developer.apple.com/products/> for more details about ADC membership.)

Xcode is a collection of applications, command-line tools, frameworks, SDKs, and other resources that facilitate the development of Mac OS X applications. In Mac OS X v10.5 and later, you can install multiple versions of the Xcode developer tools. 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 in any other directory or volume.

Some users may install Xcode to develop UNIX applications. They generally use the command-line to perform most of their development tasks. With more than one release of Xcode installed on a computer, some steps must be taken to ensure that scripts that assume all UNIX tools are in the `/usr` directory invoke the desired UNIX tools.

Xcode Installation Guide has several objectives:

- Identify Xcode installation issues
- Describe the directory structure Xcode uses
- List the file system locations Xcode searches to allow users to customize their development environment
- Discuss ways to select the Xcode release used for command-line-based development

The intended audience for *Xcode Installation Guide* includes all users of Xcode 2.5 or later, as well as developers of UNIX applications. Much of the information in this document is targeted at developers who need to use more than one Xcode release on their computers.

The information in this document applies to systems using Mac OS X v10.5 or later and Xcode 2.5 or later.

Organization of This Document

This document contains the following chapters:

- [“Xcode Installation Basics”](#) (page 9) discusses the basic concepts and procedures for installing and uninstalling Xcode Developer Tools.
- [“Xcode Installation Details”](#) (page 15) discusses the install destinations and directories, customizing an installation, and command-line-based development.

INTRODUCTION

Introduction to Xcode Installation Guide

Xcode Installation Basics

Beginning in Mac OS X v10.5, Xcode supports installing multiple versions of the Xcode developer tools. When installing, the default location for the Xcode developer tools continues to be `/Developer`; however, you can install Xcode 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 subdirectory hierarchy inside the installed Xcode directory should not be altered or rearranged.

Installing Xcode on a Macintosh computer is an easy affair: just run the Xcode Tools installer. You can install more than one release of Xcode, which provides you with one or more Xcode environments. You can take advantage of multiple Xcode environments to work on projects using a release of Xcode other than the release you typically use, or to run a test build of a future release of Xcode.

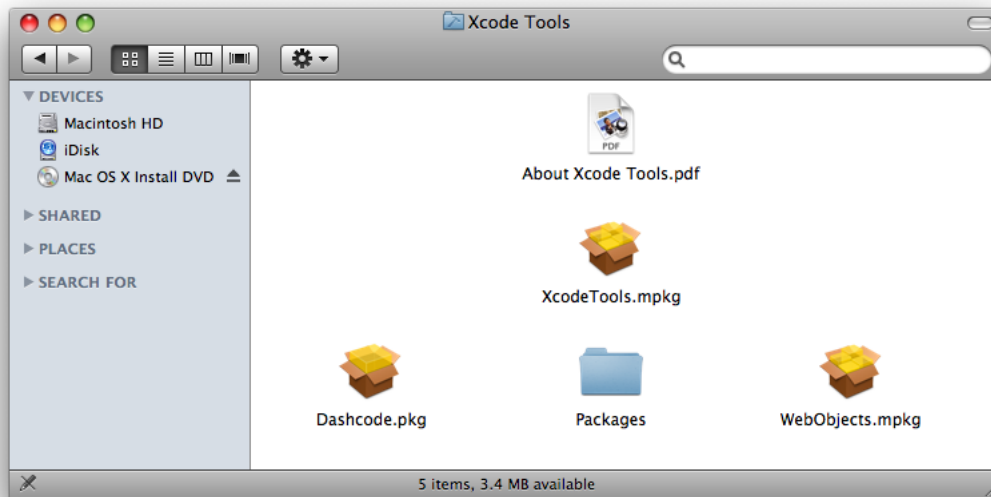
After installing Xcode, the essential Xcode tools are contained in a single directory, known as the `<Xcode>` directory. By default, the Xcode Tools installer sets `/Developer` as the `<Xcode>` directory, but users can choose a different location for this directory.

Note: Throughout this document, `<Xcode>` refers to the path in which the developer tools are installed.

The Xcode Tools Disk

The Xcode Tools disk packages a set of applications, command-line tools, SDKs, and other resources used to develop software that runs on Mac OS X.

Figure 1-1 Xcode Tools disk



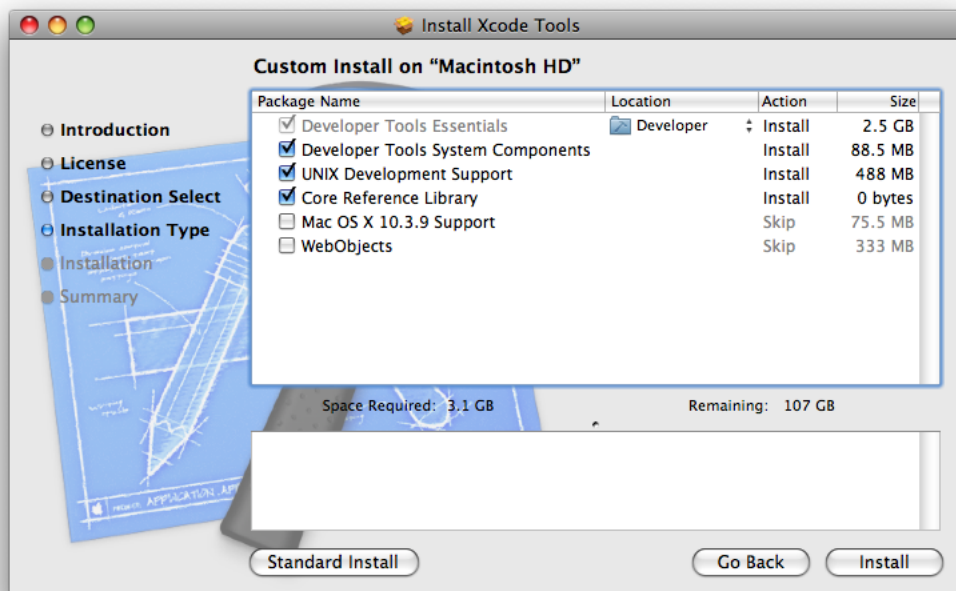
The Xcode Tools disk includes a set of UNIX developer tools, which include GCC, GDB and others. An Xcode installation contains a `<Xcode>/usr` directory with these tools. To support traditional UNIX-based development, you have the option of installing a second set of the UNIX developer tools in `/usr`. By default, this option is selected. You can also modify your shell environment to use the Xcode-provided UNIX tools.

You can customize one of all the Xcode environments on your computer by moving files between the file-system locations Xcode programs look in when searching for development resources. For example, instead of having the ADC Core Reference Library documentation set on all the Xcode releases on a computer, you can move it to your home directory or the local domain of your file system to save space.

Xcode Tools Installer

The Xcode Tools installer gives you all the options you need when installing and obviates the need to install content from individual packages. The installation choices in the "Custom Install..." panel have been organized into six groups, as shown in Figure 1-2.

Figure 1-2 Xcode Tools installer



The first group, Developer Tools Essentials, is always installed; the remaining groups are optional. The list that follows explains what each group contains and where it is stored.

- **Developer Tools Essentials** (always installed). Contains the main applications, command-line tools, and other resources that make up the Xcode development environment. This group includes the Xcode application, Interface Builder, Instruments, Dashcode, Quartz Composer, GCC 4.0.1, and the Mac OS X v10.4 (Universal) and Mac OS X v10.5 SDKs, as well as sample source code.
- **Developer Tools System Components** (selected by default). The CHUD performance 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: CHUD is installed into `/Developer` on the boot volume.
- **UNIX Development Support** (selected by default). 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 group 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 is installed only onto the boot volume.
- **Core Reference Library** (selected by default). 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.
- **Mac OS X 10.3.9 Support** (selectable). 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.
- **WebObjects** (selectable). Installs WebObjects development tools, examples, and documentation. Note: WebObjects is not relocatable and is installed into `/Developer` on the boot volume.

Installation Instructions

To install the Xcode 3.0 Developer Tools using the Mac OS X v10.5 (Leopard) DVD:

1. Boot into a partition with Mac OS X v10.5 (Leopard) installed.
 2. Insert the Mac OS X v10.5 (Leopard) Install DVD.
 3. Double-click the file `XcodeTools.mpkg`, located inside the directory `Optional Installs/Xcode Tools`.
 4. Follow the instructions in the Installer. The Installer provides several options:
 - If you want to install Xcode Tools in a different directory, or if you want to install Xcode Tools in the standard `/Developer` directory on a different volume, you must select **Customize**. Then choose a new location from the **Location** menu item for **Developer Tools Essentials**.
- Note:** You should avoid using the **Change Install Location** button. You need to use a customized install instead.
- If you want to install Mac OS X 10.3.9 Support or WebObjects, you must select **Customize** and check those install groups.
 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>/Applications`.

The Xcode Tools Uninstaller

Because the Xcode Tools installer places files across several file-system domains, when you need to uninstall Xcode you should run its uninstall script. To uninstall an Xcode release, you must run the `uninstall-devtools` script, located in `<Xcode>/Library`. The script accepts one argument, `--mode`. Its value depends on what install groups you want to uninstall.

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

```
sudo <Xcode>/Library/uninstall-devtools --mode=all
```
- To remove the underlying developer content on the boot volume but leave the Xcode directory untouched, from a Terminal window type:

```
sudo <Xcode>/Library/uninstall-devtools --mode=systemsupport
```
- To remove the UNIX development support on the boot volume but leave the Xcode directory and its supporting files untouched, from a Terminal window type:

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

- To uninstall only the Xcode directory, you can simply drag it to the trash or from a Terminal window type:

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

For example, to remove the Developer Tools System Components file as well as the <Xcode> directory, run these commands:

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

Note that the `uninstall-devtools` script removes only the directories listed in [Table 2-2](#) (page 16). You must manually remove other directories Xcode may use, such as `~/Library/Developer/Xcode_3.0`.

Note: You should use either the uninstaller script provided on the Mac OS X v10.5 (Leopard) DVD or the script provided in Xcode 3.0 or later.

Xcode Installation Details

The main application in Xcode is `Xcode.app`, but the term Xcode is also used to refer to all the items installed by the Xcode Tools installer. After installing Xcode, the essential Xcode tools are contained in a single directory, known as the `<Xcode>` directory. By default, the Xcode Tools installer sets `/Developer` as the `<Xcode>` directory, but users can choose a different location for this directory. The user is free to rename the folder either during or after the install.

Even when a user chooses a custom location, unless the Developer Tools System Components choice is disabled, the installer will install the CHUD performance tools in `/Developer`. This is done to prevent multiple versions of CHUD from being installed on one system; only the last version of CHUD installed is functional. CHUD depends on components installed into the system by the Xcode Tools installer. You may move or rename the `/Developer` directory containing the CHUD tools, as long as the directory stays on the same machine and the same system partition is booted.

Important: The Developer Tools System Components, Unix Development Support, and Mac OS X 10.3.9 Support installer choices install content outside of the `<Xcode>` directory. If the `<Xcode>` directory is copied from one computer to another, these components will be missing or nonfunctional until the Xcode Tools installer is run on that computer.

If the user has installed the developer system components, the `<Xcode>` directory may be located with the `xcode-select` tool. For more information, see [“Xcode Command-Line Tools Installation”](#) (page 19).

The Install Destinations and Directories

The Xcode Tools installer adds content to the directories described in this section.

Table 2-1 lists the directories in which the installer places the content of each install group.

Table 2-1 Xcode Tools install destinations

Install group	Destination
Developer Tools Essentials	<code><Xcode></code>
Developer Tools System Components	<code>/Developer</code> , <code>/usr</code> , <code>/Library</code> , <code>/System/Library/LaunchDaemons</code>
UNIX Development Support	<code>/usr</code>
Core Reference Library	<code><Xcode>/Documentation/DocSets/com.apple.ADC_ - Reference_Library.CoreReference.docset</code>

Install group	Destination
Mac OS X 10.3.9 Support	<Xcode>/usr/bin/gcc-3.3, <Xcode>/SDKs/MacOSX10.3.9.sdk

Table 2-2 lists the directories created by the Xcode Tools installer.

Table 2-2 Xcode Tools install directories

Directory	Description
<Xcode>	The default is /Developer.
/Library/Developer/<Xcode_release_number>	Content in this location is specific to one Xcode release.
/Library/Developer/Shared	Content in this location is used by all releases of Xcode.
/Developer	When the Developer Tools System Components group is installed, the CHUD performance tools are installed in this location.

Customizing Xcode Installations

Whether you're a standard user or an administrator, you may want to customize an Xcode installation. For example, on a system with multiple Xcode installations, you may want to consolidate documentation in one place to reduce space usage. You may have developed a custom project template that needs to be accessible to several Xcode installations.

Note that for a particular Xcode release, Xcode applications look first in their <Xcode> directory, then the user's home directory, and then the local domain. They do not look in, or have any knowledge of, other <Xcode> directories on the computer.

User-Managed Directories allow users to customize their Xcode environments:

- ~/Library/Application Support/Developer/<Xcode_release_number>/
- ~/Library/Application Support/Developer/Shared/

For example, a user might want to install an Xcode project template in this directory:

- ~/Library/Application Support/Developer/Shared/Xcode/Project Templates/

The primary type of content users need to install in these locations are templates. In general, templates should go in the Shared location. Content in the Shared location is used by all releases of Xcode, while content in an Xcode_release_number location is only used by that release. Users generally want their custom templates available no matter which version of Xcode is installed.

Administrator-Managed Directories allow users with administrator privileges to customize Xcode environments:

- /Library/Application Support/Developer/<Xcode_release_number>/

- `/Library/Application Support/Developer/Shared/`
- `/Library/Developer/Shared`. The contents of this directory are used by all Xcode releases on the computer.

The Shared Directory is a special case. If a system contains multiple Xcode releases, the `/Library/Developer/Shared` directory contains content for only one Xcode release—the last release installed. For example, if you install Xcode 3.0 and then Xcode 3.1 on the same computer, with the Developer Tools System Components group selected for both installs, `/Library/Developer/Shared` contains only Xcode 3.1 content.

The last release installed is the only release which is able to host distcc-distributed builds (see the "Share my computer for shared workgroup builds" option in Xcode's Distributed Builds preferences).

Xcode 2.5 Support

In Mac OS X v10.5 and later, developers can install and run Xcode 2.5 side-by-side with later versions of Xcode. Xcode 2.5 is the first version of the developer tools to support this feature. Apple recommends installing it alongside a later version if you need to support 2.x users or 2.x-only features (such as targeting Mac OS X 10.2.8).

When Xcode 2.5 is installed in Mac OS X v10.4, it resides in `/Developer` and behaves similarly to Xcode 2.4 and earlier releases. When Xcode 2.5 is installed in Mac OS X v10.5, you can choose the installation location. The default location is `/Xcode2.5`, so it does not remove or interfere with your Xcode 3 installation. In Mac OS X v10.5, Xcode 2.5 does not install the Developer Tools System Components, Unix Development Support, or WebObjects support. The Mac OS X v10.2.8 & v10.3.9 Support choice (off by default) installs some limited content in `/usr`.

Xcode 2.5 does not install performance tools in Mac OS X v10.5. Apple recommends using the Xcode 3 performance tools.

Intel Compiler Support

Xcode tools are not officially extensible. That is, there is no official support in Xcode for third-party tools such as compilers. The only exception is Intel and their C++ Compiler for Mac OS X. The installation of this compiler in Xcode is documented and supported by Intel.

For more information, see [Intel C++ Compiler Installation Guide](#).

Xcode Directory Structure

The `<Xcode>` directory, the main directory of the Xcode environment, contains the essential applications, command-line tools, and resources needed for developing software for Mac OS X. These are some of its subdirectories:

Applications	Contains the applications and utilities used for software development.
Documentation	Contains API reference and high-level documentation.
Examples	Contains some example applications. You may incorporate sample code from these examples into your applications.
Library	Contains frameworks and resources used by Xcode applications.
Platforms	Contains the command-line tools and resources used for developing software for the platforms Xcode supports. Present in Xcode 3.1 and later.
SDKs	Contains the command-line tools and resources used for developing software using a specific SDK.
usr	Contains the Xcode UNIX command-line tools, libraries, man pages, and other resources.

The Xcode Tools 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.

Changes in the Xcode Directory

In Xcode 3.0 and later, the Xcode directory has been re-structured to move developer tools content out of the underlying system and into a single top-level folder. Some of the more noticeable changes are listed here.

- The Xcode directory now has a `<Xcode>/usr` subdirectory that includes all command line developer content. For example, now `xcodebuild` is found at `<Xcode>/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>/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 `agvttool` 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(1)` 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>/Library/`, with each application having its own folder. For example, support files for the Xcode application are located at `<Xcode>/Library/Xcode`. In addition, support files needed for a specific version of the tools can be placed outside the Xcode directory at `/Library/Application Support/Developer/<tools version>` for support files needed for a specific version of the tools and `/Library/Application Support/Developer/Shared` for support files that are not specific to

a given tools version. You are encouraged to place additional support files (like custom file or project templates) inside `<Xcode>/Library/` to allow the support files to move with the folder instead of being tied to the boot volume.

- Because a `<Xcode>/usr` folder hierarchy has been added to the Xcode directory, `<Xcode>/Tools` is being deprecated in Xcode 3.0 developer tools and will be removed in a future release of the tools. References to tools that used to exist in `<Xcode>/Tools` should be updated to find their content in `<Xcode>/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 the Xcode 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>/Applications/Utilities`.

Xcode Command-Line Tools Installation

Shimmed tools are command-line tools that are installed in `<Xcode>/usr/bin` but that have a counterpart in `/usr/bin`. This counterpart is not a complete program, but a shim or launcher that invokes one of its counterparts in an `<Xcode>` directory. These shims invoke the actual tools in the `<Xcode>` directory selected for CLI-based development, known as the Xcode CLI directory.

These are the Xcode *shimmed tools*:

- `agvtool`
- `ibtool`
- `xcodebuild`
- `xcodeindex`
- `instruments`
- `opendiff`

The shims determine which `<Xcode>` directory is the Xcode CLI directory through `xcode-select`. When writing software which needs to invoke Apple developer tools, your software should invoke `xcode-select -print-path` to find the path to your preferred developer tools install.

You specify the Xcode CLI directory on a shell session using the `DEVELOPER_DIR` environment variable or `xcode-select`.

When you install more than one Xcode release on your computer and need to use shimmed tools in scripts that are unaware of the location of the `<Xcode>` directory that contains the real tools, set the `DEVELOPER_DIR` environment variable to point to the desired `<Xcode>` directory or use the `xcode-select -switch` command.

For example, if you have Xcode 2.5 and Xcode 3.1 installed on your computer in `/Xcode_2.5` and `/Xcode_3.1`, respectively, and want to use the Xcode 2.5 shimmed tools in a script or in a shell session, do either of the following:

- Set the `DEVELOPER_DIR` environment variable:

```
export DEVELOPER_DIR="/Xcode_2.5"
```

- Use `xcode-select`:

```
sudo xcode-select -switch /Xcode_2.5
```

Note: If you use both approaches, the environment variable overrides the `xcode-select -switch` command.

The Xcode Tools installer will set `xcode-select` to the `<Xcode>` directory it is installing unless the Developer Tools System Components choice is turned off.

See the man page `xcode-select` for more information.

UNIX Command-Line Tools Installation

When a user installs the UNIX Development Support group, the Xcode UNIX tools, libraries, and man pages are installed in `/usr` (in addition to `<Xcode>/usr`). This content can be used for traditional UNIX software development.

If you install more than one Xcode release and want to use the UNIX tools from one of those releases instead of the ones in `/usr`, change the `PATH` and `MANPATH` environment variables to list `<Xcode>/usr` first.

For example, if you want to use the UNIX tools in `<Xcode>/usr` instead of the ones in `/usr` when invoking those tools using relative paths (`gcc`, not `/usr/bin/gcc`), change the environment variables `PATH` and `MANPATH` (if you want to use the corresponding man pages) using one of two methods:

- You can use `xcode-select` to construct the paths:

```
export PATH=`xcode-select --print-path`/usr/bin:${PATH}
export MANPATH=`xcode-select --print-path`/usr/share/man:${MANPATH}
```

- Rely on the `DEVELOPER_DIR` environment variable. If you are writing a script for a Run Script phase in Xcode or building your UNIX project with an External Target in Xcode (external targets can be used to kick off makefiles or other external build systems), Xcode sets `DEVELOPER_DIR` to the path of the developer directory with which you're building.

You can use `DEVELOPER_DIR` to construct the paths as follows:

```
export PATH=${DEVELOPER_DIR}/usr/bin:${PATH}
export MANPATH=${DEVELOPER_DIR}/usr/share/man:${MANPATH}
```

If you're not going through Xcode at all, you can set `DEVELOPER_DIR` in your own environment and then rely on it when constructing paths.

When writing software that uses the Xcode UNIX tools, Apple recommends using the copy of those tools installed in `<Xcode>/usr` instead of `/usr`, because your users may not have tools in `/usr` if they have disabled the optional UNIX Development Support choice in the Xcode installer. Your software can find the path to the `<Xcode>` directory with the `xcode-select` command (see previous section).

Note: Even if you don't install the UNIX Development Support group, you can use this mechanism to use the UNIX tools in an `<Xcode>` directory for traditional UNIX-based development.

Document Revision History

This table describes the changes to *Xcode Installation Guide*.

Date	Notes
2008-05-27	New document that describes general installation issues for Xcode users.

REVISION HISTORY

Document Revision History