

Xcode Glossary



Developer

Contents

Introduction 3

Organization of This Document 3

Glossary 4

Document Revision History 12

Introduction

The Xcode Glossary defines the most common terms needed to understand Apple's Xcode application. Some of these terms are specific to Xcode while others are commonly used in the developer community.

Organization of This Document

This document contains one chapter, "[Glossary](#)" (page 4), that defines Xcode terms in alphabetical order.

Glossary

anchor For documentation sets, a location within an HTML file. When loading the documentation node's landing page, Xcode scrolls to the location of this anchor. See also [documentation node](#).

action line The code line indicated by the pointer at the time you choose a debugging command from the shortcut menu.

action object The object on which you want to perform an action.

activity viewer A control in the center of the workspace toolbar that shows the progress of the currently executing task, and the outcome of the last task. It also displays workspace issues.

API reference search A search type that looks through the available reference for a symbol name.

apple_ref An informal name for a token identifier that uses the prefix `//apple_ref`. See [token identifier](#).

archives organizer A pane in the Organizer that lets you share product archives with others, and submit them to iTunes Connect. Use this pane to ensure your application archives pass essential validation tests before submitting them to iTunes Connect.

attaching The process of starting a debugging session on a process that's already running and was not launched by Xcode.

Automator action A loadable bundle that performs discrete tasks that users can link together in a workflow using the Automator application.

base URL An alternate location for documentation nodes whose documentation files reside on the web. The default location is in the installed documentation set bundle or its fallback web location.

bold text The font style that Xcode uses in panes of Project and Target Info windows to indicate build settings specified at the current level. Build settings that are not in bold text are specified at lower layers.

breakpoint action An action to perform when a program reaches a certain point in its execution, such as logging output to the console. The default breakpoint action is to pause program execution

browse navigator, documentation organizer A browser of the documentation organizer in which you can traverse a hierarchy of categories in documentation sets until you get to a list of documents. Compare [search navigator](#), [documentation organizer](#).

bookmarks navigator A navigator in the documentation organizer that lets you view your bookmarks.

breakpoint navigator A navigator in the workspace window that lets you add exception and symbolic breakpoints, and manage breakpoints across various domains.

BSD Short for Berkeley Software Distribution. BSD is an operating system that provides low-level features such as networking and thread management. It also includes a command-shell environment for managing system resources. OS X includes a BSD application environment.

build The process Xcode uses to create a product from a target.

build client The computer that performs a build operation. This is the computer that runs the Xcode or the `xcodebuild` instance that carries out the build command.

build configuration A named set of build settings that tells Xcode how to build a product. The typical build configurations are Debug and Release, but you can define additional build configurations. See also [build setting](#).

build configuration file A file that contains build setting definitions. You can use it to share a set of build setting definitions among all the individuals working on a team. You can also use a configuration file to quickly configure targets and projects with common build setting definitions.

build phase A set of operations performed on a group of files as part of building a product.

build server A computer that a build host uses to perform compilation tasks. Build servers do not need to run Xcode or `xcodebuild` to aid a build host, but they must at least be running the same version of the Mac OS as the build server.

build set The host names of build servers to which a build client distributes compilation tasks. To distribute a build, you must define at least one build set on the build client, or use the Bonjour set when available.

build setting A variable that contains the information for building a product. For each operation performed in the build process—such as compiling Objective-C source files—build settings control how that operation is performed.

build setting name A label that identifies a build setting. It is similar to the names of environment variables in a command shell.

build setting specification The information Xcode uses to determine the value of a build setting at build time.

build setting title The label used to display the build setting in the Xcode user interface.

bundle A file system directory that stores executable code and the software resources related to that code.

call stack A collection of stack frames that positions the most recent calls on the top of the stack.

Carbon An OS X application environment that uses procedural programming interfaces derived from earlier versions of the Mac OS.

clean Removes all the product files, as well as any object files (`.o` files) or other intermediate files created during the build process.

check out To obtain a managed copy of a project or workspace from a source control repository.

client In source control management, the program a developer uses to interact with the repository.

Cocoa An OS X development environment that uses Objective-C programming interfaces that are based on the integration of Open Step, Apple technologies, and Java.

code completion Suggests likely completions as you type an identifier or a keyword. The suggestions are based on the text you have already typed and the surrounding context within the file.

code focus A feature of the text editor that highlights the scope of the selected source code.

code folding A feature of the text editor that hides code you don't want to see.

command-line utility A tool without a graphical user interface, typically used in the command-line environment.

commit To publish changes in an Xcode project to a repository.

compatibility version One of two numbers used to track minor version information in a framework. This number marks changes to the public interfaces. The compatibility version typically lags behind the current version. Compare [current version](#).

completion list A list that Xcode builds for a typed token and that you can display when using code completion. See also [code completion](#).

composite SDK A group of additional SDKs used by a target. Xcode creates and caches the composite at build time.

conditional build setting Setting values that apply only when one or more conditions are met (for example, the product is being built using a particular SDK). Xcode uses these definitions when generating executable code for a particular architecture or for a particular variant of the product.

configuration file See [build configuration file](#).

configuration unit A group of configuration files joined together by `#include` directives.

console A pane in the debug area that lets you see program output, and interact with the debugger.

Core Data A technology for managing the relationships and persistence of managed objects, whose backing store is usually a database or a file.

cross-development Creating software that can be deployed on, and take advantage of features from, specified versions of iOS and OS X, including versions different from the one you are developing on.

current version One of two numbers used to track minor version information in a framework. It tracks individual builds of your framework and is mostly for internal use by your team. Compare [compatibility version](#).

data formatter A string that specifies how variables are displayed in debugger datatips and the variable list in the debugger.

debugger A process that lets you pause a program and examine its state.

debug datatip A way of viewing and modifying the contents of variables using a progressive disclosure mechanism.

default build configuration The configuration Xcode uses when a project does not have a definition for the active build configuration.

devices organizer A pane in the Organizer window that lets you manage your iOS development devices.

documentation node A documentation file or a folder of files within a documentation set. Each documentation node is associated with a location that identifies the file to display you select that node in the documentation window. A node may be a single document, a collection of documents, or a single HTML documentation page.

documentation search A search composed of a search term, search type, and search scope.

documentation feed An RSS-style web feed that Xcode can check periodically to determine when a publisher makes available updates to a documentation set or releases a new documentation set.

dSYM file A file that stores an executable's debugging information to minimize the size of the executable file without compromising the program's debugging experience.

documentation set A subset of the reference library packaged as a standard OS X bundle. Each documentation set contains HTML-based content as well as indexes into that content, which Xcode uses to perform quick documentation searches. Also called a doc set.

documentation organizer A pane in the Organizer window designed for browsing and searching developer documentation.

dynamic library A library for which binding of undefined symbols is delayed until execution. Code in dynamic shared libraries can be shared by multiple, concurrently running programs.

Encapsulate transformation A refactoring operation that creates accessors for the transformation item, reduces its visibility, and changes code that directly accesses the item to use the accessors instead.

executable environment Defines how Xcode should execute a product in response to the Go command.

execute condition A state that triggers a breakpoint.

Extract transformation A refactoring operation that creates a function or method with the selected code as its body.

focus box A box that code focus uses to delineate scopes.

focus center Code in the text editor at the lowest scope. Code focus highlights a source file's scope levels using a grayscale.

focus ribbon A vertical strip to the right of the gutter in the text editor. You can use the focus ribbon to change where the focus box is.

framework A bundle that encapsulates shared resources, such as a dynamic shared library, nib files, image files, localized strings, header files, and reference documentation in a single package.

full-text search A search option that looks for the documents whose content matches a search term. The search term can be a word or phrase or may be an elaborate expression using Boolean operators and wildcard characters

gutter A vertical strip on the left side of the content pane in the editor. You can use it to quickly locate items in a file. A gutter can display numbers, errors and warning, and breakpoints.

Info window A window that lets you manipulate information about a project item, such as files, targets, and the project itself.

installation package A file package you use to deliver your product. Your customer opens the installation package with the Installer application, which places the package's payload in the appropriate locations in your customer's file system.

ignore condition A state that triggers a breakpoint only if the condition is true.

kernel extension A piece of code that can be dynamically loaded into the OS X kernel. For example, a driver is a kernel extension that supports one or more devices.

log navigator A navigator in the workspace navigator area that lists the tasks and sessions for which there are logs and transcripts.

major version A framework version specifier that designates a framework incompatible with programs linked with a previous version of the framework's dynamic shared library. Compare [minor version](#).

managed files The files stored in a source control repository.

message bubble A small window that displays a project message in place—that is, in the location the message applies to, such as a codeline that contains an error.

minor version A framework version specifier that designates a framework compatible with programs linked with later builds of the framework within the same major version. Compare [major version](#).

jump bar The bar along the top of an editor that contains a number of controls that you can use to move between open files, jump to symbols, and open related files.

navigator area, workspace window The area on the left side of the workspace window that lets you access several facets of your projects and workspaces.

nib file An Interface Builder document such as `MainMenu.xib`. Nib files typically define and lay out objects for the graphical interface of a product.

nodes file A file that describes the hierarchical structure of the documentation set. It defines the table of contents that users see in the browser view of the Xcode Documentation window and the relationships between entries in the documentation set hierarchy.

ordered tasks Tasks that contain inputs that are the outputs of other tasks or outputs that are the inputs of other tasks. Compare [unordered tasks](#).

pages One of the three views provided by the layout.

path For documentation sets, the page to display when the user selects the documentation node or the folder containing that page. This path is interpreted relative to the base URL of the documentation node.

per-file compiler flags A flag that you can use to customize the build process of source files of a particular type.

prebinding A process that Xcode applies when a program is built, by which the static linker replaces references to external symbols with the addresses of the symbols in the referenced libraries or tells the dynamic linker to resolve the references when a program is loaded or when a symbol is referenced.

product reference A special type of file reference that refers to the build system output for a particular target. A product reference lets you view your target's products in the project navigator.

Project editor The editor for viewing and editing information kept at the project level, such as general information, project build settings, and build phases.

project root The directory at which source control operations are rooted and that serves as the origin of a project hierarchy. By default, a project root is the project directory.

refactoring The process of modifying source code for the purpose of improving its readability and maintainability while retaining the program's functionality and behavior. See also [transformation](#).

repository A directory tree or database that contains the files managed by a source control system.

repository configuration A set of data that tells Xcode how to use a particular source control client tool to access a specific repository.

Quick Help A lightweight window that provides a condensed view of the API reference and links to related documentation for the selected system or build setting.

repositories organizer A pane in the Organizer window that lets you configure and navigate your source control repositories.

revision number A value that indicates a particular version of a project that's been committed to the repository, also known as a version number.

visibility, header file The scope of the header file in a project: public, private, or project.

SCM See [source control](#).

SDK See [Software Development Kit \(SDK\)](#).

search navigator, documentation organizer A browser of the documentation organizer in which you can search documentation sets until you get to a list of documents Compare [browse navigator](#), [documentation organizer](#).

search result The set of documents that meet search criteria.

server In source control, the process that modifies the repository.

setter In refactoring, the method to use to set the value of the transformation item.

snapshot A view of the state of the files in project. Snapshots allow you to experiment freely with refactoring operations.

snapshot store The set of snapshots taken from one or more projects with the same project root.

Software Development Kit (SDK) A complete set of header files and stub libraries as shipped in a particular version of iOS or OS X.

source control A set of tools and procedures developers for managing files and changes made to them over time. Also known as source control management (SCM) or version control.

source file A file used to build a product. Source files include source code files, resource files, image files, and others.

source group A group inside a project group in the source window that contains references to actual files somewhere on the hard disk.

sparse SDK An SDK that is not a system SDK. Sparse SDKs may be provided by third parties, or you can build them yourself.

stack frame Information the debugger stores about each function call that your program makes.

Standard Apple Plug-ins Plug-ins for standard Apple applications, such as Interface Builder, Address Book, and Quartz Composer, and preference panes.

static analyzer A tool that detects potential software bugs in source code without running the corresponding executable.

static library A library for which all referenced symbols are bound at link time.

strip style Specifies the level of stripping performed when dead-code stripping is enabled. There are three levels of stripping available: all symbols, nonglobal symbols, debugging symbols.

stub library A library used for linking purposes. Stub libraries contain exported symbols only; they do not contain executable code.

syntax-aware indenting A feature of the text editor that gives you a number of ways to control code layout. When you use syntax-aware indenting, the editor automatically indents your code as you type.

syntax formatting Formatting that uses different fonts and colors to identify particular elements in a source code file, such as keywords and comments.

target The instructions for building a finished product from a set of files in your project—for example, a framework, library, application, or command-line tool. Each target builds a single product.

target dependency Specifies the order for Xcode to build targets.

target template A template that specifies the a target product type, a list of default build phases, and default definitions for some build settings. A target template typically includes all build settings and build phases required to build an instance of the specified product.

text macro A menu item or keystroke that lets you insert common constructs and blocks of code.

text editor A view that displays a file for editing. The editor can a pane in a window or a standalone window.

title search A search type that finds the documents whose titles start with, contain, or match the search term.

token (1) In code completion, a string that does not contain spaces. (2) For documentation sets, the element used to associate a symbol with its reference documentation. The element includes a unique

identifier representing the symbol, the location of the reference documentation for that symbol, summary information about that symbol, and information about related documentation and symbols.

tokens file A file that associates symbol names with locations in documentation. A tokens file is used to create the symbol index for a documentation set, which support fast API lookup.

token identifier The unique identifier for a symbol described in a tokens file.

transformation A refactoring operation that modifies source code.

universal binary An executable file that can contain code and data for more than one architecture. You can create a single binary file that runs on both PowerPC-based and Intel-based Macintosh computers. The Architectures (ARCHS) build setting lets you specify which architectures Xcode builds for.

umbrella framework A grouping of several related frameworks.

unit test A piece of code that exercises some part of your application. A unit test provides a specific input and expects your code to return a specific output.

unordered tasks Tasks with no inputs and outputs, or tasks whose inputs are not the outputs of other tasks and whose outputs are not the inputs of other tasks. Compare [ordered tasks](#).

version number See [revision number](#).

versioned bundle A type of bundle that allows for multiple versions of framework code and header files to be stored inside the bundle.

views In Cocoa, objects that support drawing.

watchpoint A place in code that pauses execution of the program whenever the value of the watched item changes.

workgroup build A build that use several computers to compile source files.

Xcode project A group that contains the source files, libraries, media, and other resources needed to build your product.

Document Revision History

This table describes the changes to *Xcode Glossary*.

Date	Notes
2010-12-23	Update terminology for Xcode 4.0.
2009-08-28	Added terms. Made minor corrections. Added “ static analyzer ” (page 9).
2009-01-20	New document that defines the most common terms needed to understand the Xcode application.



Apple Inc.
Copyright © 2010 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.

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.

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

Apple, the Apple logo, Bonjour, Carbon, Cocoa, iTunes, Mac, Mac OS, Macintosh, Objective-C, OS X, Quartz, and Xcode are trademarks of Apple Inc., registered in the U.S. and other countries.

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

Java is a registered trademark of Oracle and/or its affiliates.

PowerPC and the PowerPC logo are trademarks of International Business Machines Corporation, used under license therefrom.

iOS is a trademark or registered trademark of Cisco in the U.S. and other countries and is used under license.

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.