

About Debugging with Xcode

Important: This documentation contains preliminary information about an API or technology in development. This information is subject to change, and software implemented according to this documentation should be tested with final operating system software.

Finding and eliminating problems in your code is a critical part of the development process. The Xcode debugger is preset with useful features for general purpose debugging and runs automatically when your app is launched. The debugger helps you:

- Identify and locate the problem
- Examine the control flow and data structures of running code to find the cause
- Devise a solution and edit your code accordingly
- Run the revised app and confirm that the fix works

Prerequisites

You should be familiar with app design and programming concepts. Some familiarity with Xcode is also recommended; see *Xcode Overview*.

See Also

Every year, several sessions at the Apple Worldwide Developer Conference are devoted to debugging that expand upon the material in this guide and add to it with useful techniques. These sessions are available for you in the Apple developer libraries at the Apple Developer website and are easy to find by filtering on “debug.”

The following recent WWDC presentations focus on using the Xcode debugger and related tools:

- WWDC 2013: Debugging with Xcode: Detect and fix performance problems using the Xcode graphical debugger.
- WWDC 2013: Advanced Debugging with LLDB: Debug using Terminal and the Xcode graphical debugger.
- WWDC 2014: Debugging with Xcode 6: Learn how apps enqueue work, explore and fix user interfaces, add custom Quick Look support.
- WWDC 2015: Advanced Debugging and the Address Sanitizer: Learn how to use advanced breakpoint actions to explore and fix your app, and see how the address sanitizer finds memory corruption bugs at run time.
- WWDC 2016: Visual Debugging with Xcode: Discover new enhancements for debugging autolayout issues at run time, how issues inside complex objects can be easily debugged, and see how the enhanced FPS gauge find bottlenecks with SpriteKit and SceneKit apps. Find and fix leaked or abandoned memory from within your debugging workflow with memory graph debugging.
- WWDC 2016: Thread Sanitizer and Static Analysis: Learn how to use the thread sanitizer to find data races and other concurrency bugs. See how the static analyzer can now search for localizability issues, check nullability, and find memory leaks.

A good primer on debugging in general is *The 9 Indispensable Rules for Finding Even the Most Elusive Software and Hardware Problems* by David J. Agans.

Copyright © 2016 Apple Inc. All Rights Reserved. Terms of Use | Privacy Policy | Updated: 2016-09-13