Documentation

Language: Swift

Updates / Xcode updates

Article

Xcode updates

Learn about important changes to Xcode.

Overview

Browse notable changes in Xcode.

June 2025

The latest version of Xcode includes the following new features.

Optimizations

- Download components, such as Metal toolchains and Simulator runtimes, only when Xcode detects that you need them. For more information, see Downloading and installing additional Xcode components.
- Download smaller Simulator runtimes that no longer contain Intel-based Mac support by default.

Workspace and editing

- In the source editor, add as many tabs as you want, and pin files to a tab.
- Find clusters of words across files in your project using the Multiple Words search option in the Find navigator. Enter a set of words and Xcode finds the clusters in proximity to each other across your files, sorting the occurrences by relevance.
- Use Voice Control to instruct Xcode and write Swift code. Voice Control understands Swift syntax, adds spaces where needed, and enters expressions correctly.

- Use Icon Composer to create a single file representation of your app icon for iOS, iPadOS, macOS, and watchOS. Take full advantage of the new material and appearances on these platforms. For more information, see Creating your app icon using Icon Composer.

Intelligence

- Use a coding assistant to explain, write, and fix Swift code for you from prompts and project files in Xcode. You can configure models in the Intelligence settings and switch between them in the coding assistant. For more information, see Writing code with intelligence in Xcode.
- Use the playground macro to quickly iterate on code snippets directly in Xcode and display the live execution results in a canvas tab. Ask the coding assistant to generate playgrounds for you about symbols in your project. For more information, see Running code snippets using the playground macro.
- In the source editor, use the Coding Tools popover for common actions to selected code, such as explain, document, and generate a preview or playground. Enter a prompt about the code in the text field.
- When using code completion in the source editor, click the disclosure triangle to select one of multiple signatures for a base method.
- For issues and warnings that appear in the Issue navigator and source editor, such as deprecation warnings, click the Generate button in the Fix-It popover to let the coding assistant fix it for you.

String catalogs

- Add localized strings to string catalogs and access them in code using type-safe Swift symbols that also appear in code completion.
- Let Xcode use on-device models to generate comments for localized strings in your code that assist translators.

Debugging and performance

- When debugging Swift concurrency code, step into or out of an `await` call to follow a task onto a new thread.
- Add a missing usage description key to your project directly from the debugger when your app stops abruptly because it accesses a private resource without the person’s permission. The debugger takes you to the Signing & Capabilities pane where you can edit all the usage description keys in one place.

- Use the SwiftUI template in Instruments to find long-running view body updates and other performance issues in your SwiftUI views. For more information, see Understanding and improving SwiftUI performance.
- Identify situations where your app causes a device to consume high amounts of energy with the Power Profiler instrument. For more information, see Measuring your app's power use with Power Profiler.
- Discover and analyze situations where your app doesn't use the processor at highest effectiveness with the CPU Counters instrument.

Organizer

- Compare your apps metrics with recommended values in the Metrics Organizer. For more information, see Analyzing the performance of your shipping app.
- Prioritize performance work by identifying trending insights in the Metrics Organizer.

Security

- Adopt Enhanced Security in your apps and extensions to take advantage of compiler and runtime capabilities that can help to address some security issues. For more information, see Enabling enhanced security for your app.

Testing

- Use the improved UI automation recording feature to build UI tests for your app. For more information, see Recording UI automation for testing.
- Use Thread Performance Checker to detect situations where your tests use main-thread-only APIs on background threads or where your tests create background tasks that depend on tasks with lower quality of service.

June 2024

Xcode 16 includes SDKs for iOS 18, iPadOS 18, macOS 15, tvOS 18, and watchOS 11, and the following new features.

Source Editor

- Leverage the new predictive code completion engine, which predicts the code you need. Powered by a unique model specifically trained for Swift and Apple SDKs, code completion now takes context such as function names and comments into account to provide more thorough code suggestions.
- Match user input to the best action more accurately with Improved Quick Actions.
- Leverage the repeat command with Vim key bindings.

Swift

- Enable strict concurrency in Swift 6 to check for the risk of data races at compile time.

Testing

- Write your unit tests using Swift Testing and get richer output and more actionable test results. Take advantage of its new features, such as parameterized testing, to create more tests cases with less code, annotate tests, and modify runtime behaviors.
- Generate test plans in the Test Plan Editor based on the tags you define and attach to the test you write using Swift Testing.
- Gain insights into device and tag specific failure patterns in the tests results when testing across multiple configurations and run destinations. Xcode displays these insights in the Report navigator in Xcode under the Insights section of a Test Report.
- Export video, still frames, or screenshots from the details of a Test Report under the Reports navigator in Xcode.

Previews

- Iterate designs quickly with the improved SwiftUI preview. Leverage new preview APIs for declaring state and reusing data across previews.

Asset management

- Add new Dark and Tinted app icons for iOS.

Devices and simulator

- Test FaceTime and SharePlay experiences for visionOS using the upgraded Xcode simulator.

- Get started with your projects more quickly with a smaller, more efficient Xcode. Simulator updates resume even after your download gets interrupted, and the new Settings -> Components panel helps you manage your downloadable components.

Performance and metrics

- Identify areas of launch time improvement. Discover where your iOS app spends its time at launch with the launch diagnostics that come with Xcode Organizer.
- Use signature trend insights to understand how the impact of disk usage issues changes over time for your iOS app.
- Surface runtime issues for launch time, hang rate, and disk usage with the Thread performance checker.
- Leverage the Flame Graph feature in Instruments to better visualize performance hotspots in your app.

Debugging

- Reduce the size of your `.dSYM` bundles and allow for faster lookups by adapting the DWARF5 default symbol debugging format.
- Visualize a threads backtrace interactively in the source editor using the new Unified Backtrace view in Xcode Debugger. You can view the source code corresponding to each frame from the backtrace in a single editor tab, along with program counter annotations and data tips for local variables that have values available in each frame.
- Interactively debug crash logs in LLDB with or without the existence of a local copy of the corresponding Xcode project file and source code. You can supply Xcode with matching symbol-rich executables or dSYM files to create a readable version of the contents of the crash log. These files can then load into a crash debugging session using the “Load Symbols” contextual menu item in the debug navigator.
- Inspect properties and get a snapshot of your running app’s entity hierarchy with RealityKit debugger.
- Leverage the `@DebugDescription` macro to convert compatible Swift `debugDescription` properties into LLDB type summaries.
- Debug devices remotely from the command line in LLDB with the `device` command.

Projects and workspaces

- Leverage the New Empty File feature from the Project Navigator’s context menu to quickly create a new Swift file without any confirmation dialogs.
- Leverage the Copy, Paste, and Duplicate feature from the Edit menu to create a new file from an existing file.
- Cut text from the Source Editor, and then use the New File from Clipboard command while holding the option key in the Project Navigator’s context menu to quickly extract part of a source file into a new file.
- Minimize project file changes and avoid conflicts with buildable folder references. Convert an existing group to a buildable folder with the Convert to Folder context menu item in the Project Navigator. Buildable folders only record the folder path into the project file without enumerating the contained files, minimizing diffs to the project when your team adds or removes files, and avoiding source control conflicts. To use a folder as an opaque copiable resource, the default behavior before Xcode 16, uncheck the Build Folder Contents option in the File Inspector.
- Create groups with associated folders by default when using the New Group and New Group from Selection commands in the Project Navigator. To create a group without a folder, hold the Option key in the context menu to reveal the New Group without Folder variant of the command.

Xcode Cloud

- Define custom aliases to set up and manage centralized Xcode and macOS configurations, and apply them across multiple workflows.
- View coverage data from test runs in Xcode Cloud by opening the build report under the Reports navigator in Xcode and navigating to Coverage.
- Configure webhooks that connect Xcode Cloud to other services and tools.

StoreKit testing in Xcode

- Configure app policies, such as user license agreements and localized privacy policies, to display in StoreKit views using the `StoreKit.configuration` file.
- Configure win-back offers in the `StoreKit.configuration` file to test win-back offers for autorenewable subscriptions.

Localization

- Use new features in the string catalog, such as inline diagnostics and the ability to mark strings as not to be translated, and jump between source code and translations for improved localization workflows.

Build system

- Discover improved parallelism, more detailed compile time error messages, and improved debugger performance with explicit modules.
- Enable low-overhead security-critical checks at runtime with the hardened C++ standard library.

Documentation

- Create documentation links to on-page headings and topic sections.
- Combine overloaded methods by adding `--enable-experimental-overloaded-symbol-presentation` to “Other DocC Flags” (`OTHER_DOCC_FLAGS`).
- Build documentation for C++ projects using Product > Build Documentation.
- Separate content using a horizontal rule by adding `---` or `***` in a line.

June 2023

Xcode 15 includes SDKs for iOS 17, iPadOS 17, macOS 14, tvOS 17, and watchOS 10, and the following new features.

Xcode IDE

- Install just the platforms you need. Platform runtimes are separated into individual installations. Select the platforms you develop for when you download Xcode from the developer website or when you launch Xcode for the first time. You can add or remove platform runtimes at any time. See Downloading and installing additional Xcode components.
- Access and configure capabilities granted for your team through App Store Connect right in your Xcode project settings. See Capabilities.
- Use the new Integrate menu to stage and commit source code repository changes, and to create and manage your Xcode Cloud workflows.
- Generate privacy reports for app archives based on privacy manifests in your app and third-party SDKs your app links to. See Describing data use in privacy manifests.
- Verify the code signature of XCFrameworks in your project. The Xcode build system fails with an error if the signature changes or is removed. See Verifying the origin of your XCFrameworks.

Related sessions from WWDC23

Session 10165: What’s new in Xcode 15

Code

- Use code completion more effectively. Xcode uses more sources of input for code completion, prioritizes completions based on context, and improves the display of parameter options.
- View the expanded form of your Swift macros in the source editor, and set breakpoints in code that a macro generates. For more information about Swift macros, see Applying Macros.
- Create bookmarks for lines of code and saved queries. Create to-do lists by organizing bookmarks into groups, and mark items complete as you address them.
- View and stage changes in the gallery view in the source editor.
- Validate the ability to link with modules at build time. Module verification is enabled by default, but you can enable and disable verification by setting `Enable Module Verifier` in build settings. See Build settings reference.
- Use mergeable dynamic libraries to get app launch times similar to static linking in release builds, without losing dynamically linked build times in debug builds. See Configuring your project to use mergeable libraries.

Related sessions from WWDC23

Session 10166: Write Swift macros

Session 10268: Meet mergeable libraries

Interface

- Use the `#Preview` Swift macro to generate previews for all UI technologies, including SwiftUI, AppKit, UIKit, and WidgetKit.
- Choose between devices connected to your Mac or from Simulator runtimes when rendering previews. From the Preview canvas, choose the device or Simulator runtime from the popup menu.
- Interact with your macOS app’s interface in the Preview canvas — similar to how you can on other platforms — to test controls, logic, animations, and text input.

- Select entries from a widget's timeline in the WidgetKit preview to view transition animations between those entries.
- Use a string catalog to localize and translate all your app's text in a visual editor right in Xcode. Host translations, configure pluralization messages for different regions and locales, and change how text appears on different devices, all in one place. See Localizing and varying text with a string catalog.
- Automatically generate symbols for assets in your asset catalogs so you don't need to reference those assets by string names.

Related sessions from WWDC23

Session 10252: Build programmatic UI using Xcode Previews

Documentation

- Preview your docs in real time. Access the new Documentation Preview assistant from the assistant jump bar.
- Get better help quickly. Quick Help offers a new visual style, support for images, and dynamic font size adjustments based on your editor font.
- Produce richer documentation. DocC now supports videos, more layout configuration options, and theming.
- Use DocC to document Swift extensions.

Related sessions from WWDC23

Session 10244: Create rich documentation with Swift DocC

Tuning and debugging

- Locate the most relevant `os_log` for debugging with the new structured debug console. The console provides insight into the origin of logs and lets you filter by metadata in addition to log messages. Jump right to your source code from individual log messages. Metadata and source code information for `os_log` messages is available for devices and Simulator runtimes running macOS 13.3 and later, iOS 17 and later, iPadOS 17 and later, watchOS 10 and later, and tvOS 17 and later.
- Diagnose testing issues more quickly and easily. Test reports include new insights that help you correlate failures between tests and configurations, so you can identify common underlying

issues. Reports also include videos that help you inspect and diagnose issues with your view hierarchy.

- Xcode uses a new device connectivity stack to target devices running iOS 17 and later, iPadOS 17 and later, tvOS 17 and later, and Apple Watch devices running watchOS 8.7.1 and later when paired with an iPhone running iOS 17 and later.
- Manage devices that the new connectivity stack supports, including Apple Watch, in Xcode's Devices and Simulators window.
- Manage and interact with devices that the new connectivity stack supports from your shell scripts with `devicectl`. For information on `devicectl`, run `xcrun devicectl help`.
- Connect and communicate wirelessly between a Mac running Xcode or Instruments and Apple Watch Series 6 and later when using the new device connectivity stack. The initial pairing process and some other tools, such as Console and Accessibility Inspector, still require the watch's companion phone with a wired connection to a Mac.
- Xcode 15 no longer supports developing on Apple Watch (1st generation), Apple Watch Series 1, and Apple Watch Series 2 that are paired to iPhones running iOS 17 and later.

Related sessions from WWDC23

Session 10226: Debug with structured logging

Session 10175: Fix failures faster with Xcode test reports

Distribution and continuous integration

- Use streamlined options in the Xcode Organizer window to distribute your app using recommended settings. Choose from several preconfigured methods of distribution or create a custom configuration for your needs. See Distributing your app for beta testing and releases.
- Notarize macOS apps in Xcode Cloud. See Creating a workflow that builds your app for distribution.
- Add text files to your Xcode project to provide notes to beta testers about what to test through Xcode Cloud. See Including notes for testers with a beta release of your app.
- Provide feedback on issues you encounter when building with Xcode Cloud. The system prepopulates the feedback form with build-specific context and attachments Apple can use to triage a bug. See Reporting feedback for Xcode Cloud.
- Xcode Server is no longer available as a part of Xcode. Use Xcode Cloud instead.

Related sessions from WWDC23

Session 10224: Simplify distribution in Xcode and Xcode Cloud

See Also

Technology updates

Accelerate updates

Learn about important changes to Accelerate.

Accessibility updates

Learn about important changes to Accessibility.

ActivityKit updates

Learn about important changes in ActivityKit.

AdAttributionKit Updates

Learn about important changes to AdAttributionKit.

App Clips updates

Learn about important changes in App Clips.

App Intents updates

Learn about important changes in App Intents.

AppKit updates

Learn about important changes to AppKit.

Apple Intelligence updates

Learn about important changes to Apple Intelligence.

AppleMapsServerAPI Updates

Learn about important changes to AppleMapsServerAPI.

Apple Pencil updates

Learn about important changes to Apple Pencil.

ARKit updates

Learn about important changes to ARKit.

Audio Toolbox updates

Learn about important changes to Audio Toolbox.

AuthenticationServices updates

Learn about important changes to AuthenticationServices.

AVFAudio updates

Learn about important changes to AVFAudio.

AVFoundation updates

Learn about important changes to AVFoundation.