

WWDC 2023 Viewing Guide

Where Do I Start?

The Vision Pro got a lot of attention in the keynote but watch the Platforms State of the Union for a summary of what's new in Swift, SwiftUI, Swift Data, WidgetKit and more.

- ☐ [Platforms State of the Union \(SOTU\)](#)

You don't have to learn everything new today!



There are about 170 sessions and a huge amount of new stuff to learn. You may feel like you're getting left behind. Don't let it overwhelm you. You've got time. Many of the new APIs require iOS 17 and Vision Pro doesn't ship until next year. Don't burn yourself out trying to keep up!

Watching the sessions



I watch the videos using the Apple Developer app. It's available on macOS and iOS. The video player supports from 0.5x to 2x playback, many videos have transcripts and support copying of the onscreen [sample code](#).

There are a lot of sessions, but many are short (10-20 minutes). There's no padding and Apple's engineers get to the point quickly.

Swift

What's new in Swift and the Swift Package Manager:

- ☐ [What's new in Swift](#) Recap of changes in Swift 5.9. Generic APIs with parameter packs. Macros to extend language and remove boilerplate. Swift implementation of Foundation. Non-copyable types. C++ interop. CMake support for Swift.
- ☐ [Generalise APIs with parameter packs](#) Parameter packs let you write generic implementations for variable argument functions removing the workaround of using multiple overloads.

WWDC 2023 Viewing Guide

Swift (continued)

- ☐ [Beyond the basics of structured concurrency](#) Prefer structured tasks. Create with `async let` or task groups. Cancelled when they go out of scope. You still need to check for cancellation.
- ☐ [Mix Swift and C++](#) Prefer structured tasks. Create with `async let` or task groups. Cancelled when they go out of scope. You still need to check for cancellation.

Swift Macros

Swift macros extend Swift without needing to extend the compiler.

- ☐ [Write Swift macros](#) Examples of writing macros like `#stringify`. Start with Swift macro package template. Xcode can expand the macro to show the result. Write unit tests for your macros using `assertMacroExpansion`. Print the syntax tree in the debugger.
- ☐ [Expand on Swift macros](#) This session helped me see the possibilities of Swift macros. See the `SwiftSyntax` documentation when working with the Swift syntax tree to implement a macro. Create a diagnostic for errors and warnings. Don't use outside information, only use what the compiler provides.

SwiftUI

Lots of changes to SwiftUI including the new `Observable` macro.

- ☐ [What's new in SwiftUI](#) Spatial computing scenes. Interactive Widgets. MapKit. Pie charts. Subscription store views. `@Observable` macro. `SwiftData`. Inspector view. Hide and reorder table columns and collapse sections. Key frame animator API. Sensory Feedback API. Visual effects. Metal shaders. Animate SF Symbols. Scroll view effects. Container relative frames. HDR images. Accessibility Zoom. Bordered button shapes. On-key press modifier.
- ☐ [Discover Observation in SwiftUI](#) A big change. Migrate from `ObservableObject` to `@Observable` macro. No more `@Published`. View is only invalidated when a property it uses changes. Use `@State`, `@Environment` or `@Bindable`.
- ☐ [Meet MapKit for SwiftUI](#) New SwiftUI API to make it easier to integrate maps.

WWDC 2023 Viewing Guide

SwiftUI (continued)

- ☐ [Explore SwiftUI animation](#) Smooth spring animations are the default in iOS 17. CustomAnimation protocol.
- ☐ [Animate with springs](#) Why springs are a good fit for animations. Presets for smooth, snappy, bouncy. Spring model for more control.
- ☐ [Wind your way through advanced animations in SwiftUI](#) Building complex, multi-step animations.
- ☐ [Beyond scroll views](#) New API to manage margins, content offset and transitions. scrollTargetBehavior to align scroll to page or view. contentRelativeFrame to size relative to container. scrollPosition to programmatically scroll. scrollTransition to add visual effects to views.
- ☐ [Inspectors in SwiftUI: Discover the details](#) Inspector view shows detail for selected content. Grouped style by default. Full height or nested under toolbar.
- ☐ [Demystify SwiftUI performance](#) Extract views to reduce dependencies and switch to new Observable. In iOS 17 SwiftUI has optimized for filtering and scrolling lists. List needs to know all its IDs up front, so avoid conditional views. New streamlined initializer for Table that back deploys.
- ☐ [Explore pie charts and interactivity in Swift Charts](#) Swift Charts get pie charts. Increase inner radius for a donut chart. New selection modifier to handle gesture recognition. Single value or range selection. Use scrollTargetBehavior to snap scroll to chart values.
- ☐ [The SwiftUI cookbook for focus](#) Controlling focus appearance, observing focus movement, responding to keyboard input with custom controls. New modifiers to allow a custom control to be focusable and set default focus (previously on macOS only).

UIKit

UIKit continues to get improvements including new trait handling.

- ☐ [What's new in UIKit](#) Preview macro works for UIKit. viewIsAppearing. Back-deploys to iOS 13. Custom traits. Animated SF Symbols. Empty state view configuration. Per-locale UIImage. UIDocumentViewController. New PencilKit inks. Keyboard control of scroll views. Collection view performance and layout improvements. UIView spring animation API. Status bar adaptive styles. Drop content onto app to launch. Fractional page progress control. Palette menus.

WWDC 2023 Viewing Guide

UIKit (continued)

- ☐ [Unleash the UIKit trait system](#) Define custom traits. New closure based API for working with trait collections. Traits now flow through view hierarchy. Use new `viewIsAppearing` in place of `viewWillAppear` (back to iOS 13). New trait override property. `traitCollectionDidChange` is deprecated. Bridged to SwiftUI environment keys.
- ☐ [What's new with text and text interactions](#) New input language switcher, selection handles and loupe for `UITextView`s and `UITextField`s. `UITextViewDelegate` can customize primary action or provide a menu for text items. TextKit 2 list and bullet support. `UITextField` and `UILabel` better handle languages with variable line height.
- ☐ [Keep up with the keyboard](#) Keyboard now runs outside of app process, can change timing of notifications. Keyboard layout guide is the recommended approach. New properties to track bottom of screen and account for accessory view. English language keyboard predictive text.
- ☐ [Build better document-based apps](#) New `UIDocumentViewController`. Configures toolbar for sharing, dragging, undo and renaming. SwiftUI `DocumentGroup` supports all these features with no additional code.

macOS

- ☐ [What's new in AppKit](#) Customize table columns. New Inspector sidebar, back deploys to Big Sur. Menus re-written to fully use Cocoa. Palette menus. Cooperative App Activation. `CADisplayLink`. Five new system colors. Most `NSViews` no longer clip to bounds by default. `NSImage`, `NSColor` and `NSSound` conform to `Transferable`.
- ☐ [Create seamless experiences with virtualization](#) Creating Mac and Linux virtual machines in macOS Sonoma. Pause, save and resume virtual machine. Attach network storage. Rosetta 2 uses caching daemon to run x86_64 binaries faster in Linux VMs.
- ☐ [Protect your Mac app with environment constraints](#) Environment constraints describe how code is expected to run on the system. Useful when app has multiple processes or loads code signed by different teams.
- ☐ [What's new in ScreenCaptureKit](#) Presenter Overlay shows the presenter over the shared content. Picker to choose windows, apps or displays to share. New Screenshot API.

WWDC 2023 Viewing Guide

watchOS

- ☐ [Design and build for watchOS 10](#) Keep it brief and focused. More emphasis on Digital Crown to navigate. Three foundational layout grids: dial-based, infographic, and lists. Vibrant full-screen background materials and gradients.
- ☐ [Meet watchOS 10](#) New user interface to take advantage of larger, brighter displays. Vertical pagination for apps with multiple tabs. Keep navigation shallow.
- ☐ [Update your app for watchOS 10](#) Code-along updating an app for watchOS 10. See the sample code. NavigationSplitView for source and detail lists. TabView to break content into full-screen views. Container background. Materials.

tvOS

- ☐ [Discover Continuity camera for tvOS](#) Using an iPhone or iPad as a camera and mic for a tvOS app using continuity camera.

Developer Tools

- ☐ [What's new in Xcode 15](#) Smaller download with optional simulators. New code completion. Symbol name generation. String catalogs. Documentation preview. New preview API. Bookmarks navigator. Improved source control navigator. New test reports with insights. OSLog integration. Xcode Cloud test flight notes, notarization, and internal testing. Framework signature verification.
- ☐ [Build programmatic UI with Xcode Previews](#) New preview also works for UIKit and AppKit. Preview Widget timeline. Creating a small preview app with required entitlements.
- ☐ [Fix failures faster with Xcode test reports](#) New test report with insights to show patterns across configurations and destinations.
- ☐ [Verify app dependencies with digital signatures](#) Dependency signature verification to reduce risk of maliciously modified external dependency.

WWDC 2023 Viewing Guide

Developer Tools (continued)

DocC

- ☐ [Create rich documentation with Swift-DocC](#) Documentation Preview editor for real-time view of documentation. Document extensions on external types. DocC extended Markdown. Quick navigation support in Swift-DocC websites.

Build System

- ☐ [Meet mergeable libraries](#) Using dynamic libraries helps build times but can slow down launch times. Mergeable libraries give similar launch times to static libraries. Manual merging allows a fine-grained approach.

Performance and Debugging

- ☐ [Debug with structured logging](#) New debugger console hides metadata by default. The p and po commands now perform a "Do What I Mean Print". Use OSLog instead of print debugging.
- ☐ [Analyze hangs with instruments](#) Analyzing and fixing hangs. Aim for less than 100ms for instant, above 250 ms is a micro hang and above 500ms a proper hang. Use Time Profiler, os_signposts, View Body, and Swift Concurrency Tasks instruments.

Xcode Cloud

- ☐ [Create practical workflows in Xcode Cloud](#) Case studies for using Xcode Cloud from a solo developer up to large teams with legacy code.

Playgrounds

- ☐ [Prototype with Xcode playgrounds](#) This is how I originally tried to use playgrounds to prototype in an Xcode project. Time to try again?

Localisation

- ☐ [Discover String catalogs](#) Replace strings and stringsdicts files with string catalogs. Back-deployable to any OS. Right click to migrate localizable files. Use Swift tools version 5.9 to use with Swift Packages.

WWDC 2023 Viewing Guide

Localisation (continued)

- ☐ [Unlock the power of grammatical agreement](#) Added support for European Portuguese and German. New properties in String Catalog to specify words that agree with concept or argument. `TermOfAdress` property and `referenceConcept` property to reflect personal pronouns.

Accessibility

- ☐ [Perform accessibility audits for your app](#) Automate accessibility audits by running them from UI tests. Call `performAccessibilityAudit`. Decide which categories to audit and which issues to ignore.
- ☐ [Build accessible apps with SwiftUI and UIKit](#) New `isToggle` trait for custom buttons that act like switches. Accessibility Notifications. Zoom action. Direct touch options to silence VoiceOver. Accessibility content shape for custom shapes. Block based setters to keep values updated.
- ☐ [Create accessible spatial experiences](#) VisionOS contains large list of familiar accessibility features reimaged for spatial computing. VoiceOver uses spatial audio to provide cues to an objects location. Use Reduce Motion to avoid rapid, bouncing or zooming motions which can be dizzying. Dwell control to interact without using hands. Pointer control to move focus without using eyes. Support closed captions.
- ☐ [Meet Assistive Access](#) Assistive access is a mode that lightens cognitive load. Home screen shows only enabled apps with larger app icons and text. Large back button at bottom of screen. Apps should adapt to reduced screen size and respect safe area.

App Store and Distribution

- ☐ [What's new in StoreKit2 and StoreKit Testing in Xcode](#) New API for promoted in-app purchases. StoreKit migrating to SHA-256 for signing receipts. New SwiftUI views for in-app purchases.
- ☐ [Meet StoreKit for SwiftUI](#) SwiftUI views to build declarative in-app purchase UI for `StoreView`, `ProductView`, and `SubscriptionStoreView`. Cross platform, see sample code.

WWDC 2023 Viewing Guide

App Store and Distribution (continued)

- ☐ [Explore testing in-app purchases](#) StoreKit Testing in Xcode for offline testing. App Store sandbox for end-to-end testing.
- ☐ [What's new in App Store pre-orders](#) Soft launch pre-orders to specific regions.
- ☐ [What's new in App Store pricing](#) 800 price points with extra 100 on request. Choose price for base region, generate other regions. Existing apps default to US base. Use temporary price changes for promotions.
- ☐ [What's new in App Store server APIs](#) Up to 900 price points. Xcode Cloud tester notes. Use test accounts as family group to test family sharing. New xrOS data privacy collection types. Regional pre-orders.
- ☐ [Meet the App Store Server Library](#) App Store Server Library supports Swift, Java, Node.js and Python. Beta available on GitHub.
- ☐ [Explore App Store Connect for spatial computing](#) Can make existing iPad and iPhone apps available without submitting a new build. Extra data collection privacy settings.
- ☐ [What's new in App Clips](#) App clip size up to 50Mb. The iOS 16 15Mb limit still applies for NFC tags or App Clip codes. 10Mb limit still applies for iOS 15. Apple creates a default link when you publish an app clip.
- ☐ [What's new in Background Assets](#) Download additional content outside of app's lifecycle in an App extension.

Photos and Camera

- ☐ [Support external cameras in your iPadOS app](#) In iPadOS 17 use external cameras such as the Apple Studio Display or USB attached webcams that conform to UVC.
- ☐ [Embed the Photos Picker in your app](#) Picker still runs in a separate process. API for SwiftUI, UIKit and AppKit.
- ☐ [Support Cinematic mode videos in your app](#) New Cinematic API to add cinematic mode (iPhone 13 and 14).
- ☐ [Support HDR images in your app](#) Show HDR images with SwiftUI, UIKit, and AppKit. Apple working to publish an ISO technical specification for encoding HDR images.

WWDC 2023 Viewing Guide

Photos and Camera (continued)

- ☐ [Create a more responsive camera experience](#) Zero shutter lag uses a rolling buffer to capture image. Rebuild with iOS 17 to opt in. Video effects to adjust lighting and recognise pre-defined gestures.

Privacy and Security

- ☐ [What's new in privacy](#) Embedded Photo picker. Screen capture picker in macOS Sonoma. EventKitUI to create new calendar events. Oblivious HTTP to proxy through a relay provider. Sensitive Content Analysis. Safari stripping tracking parameters from URLs. Per-site permissions for Safari app extensions. Eye and hand data from visionOS not shared with app.
- ☐ [Get started with privacy manifests](#) Provide privacy manifests with SDKs. Xcode can generate a privacy report from the manifests to help you create your app's privacy details in App Store Connect. Some third-party SDKs have been identified as privacy impacting. When you include them you must provide a privacy manifest. Privacy impacting SDKs must be signed from Fall 2023.

System Frameworks

SwiftData

I didn't expect to see a new data persistence API.

- ☐ [Meet SwiftData](#) Define model schema from Swift code. Attributes inferred from properties, customized with `@Attribute`, `@Relationship`, and `@Transient`. Model container and context play a similar role to Core Data. New Swift types for predicate and fetch descriptors. Models are `@Observable`.
- ☐ [Build an app with SwiftData](#) Add the `@Model` macro to model types. Create a container, model context is added to the environment. Use `@Query` to fetch models and update views. In-memory container for preview data. Model context is autosaved.

WWDC 2023 Viewing Guide

System Frameworks (continued)

- ☐ [Model your schema with SwiftData](#) @Model macro generates a schema. @Attribute to customize properties. Unique constraints. Inserting can become an "upsert". Store large data externally. Relationships default to inverse and nullify on delete. Schemas are versioned. Lightweight migrations require no additional code.
- ☐ [Migrate to SwiftData](#) Generate SwiftData classes from Core Data model. SwiftData autosaves on UI lifecycle events and on a timer for context changes. Use Query in place of fetch requests. Coexist SwiftData and Core Data stacks talking to same store.
- ☐ [Dive deeper into SwiftData](#) ModelContainer persists schema. ModelConfiguration controls where data is stored. In-memory, on disk, specific URL, read-only, CloudKit container (private only). ModelContext tracks changes. Supports rollback, reset, undo and autosave.

Core Data

- ☐ [What's new in Core Data](#) Composite attributes composed of built-in Core Data types. Alternative to transformable types. New staged migration API when lightweight migration cannot be used. Reduces a complex migration to a series of lightweight migrations. Defer schema cleanup work from a lightweight migration to a later time. Backwards compatible to iOS 14.

CKSyncEngine

- ☐ [Sync to iCloud with CKSyncEngine](#) Cloud Kit sync engine you can use with your own persistence layer. Used by many Apple apps including Freeform and keychain key value store. See sample code.

Maps and Location

- ☐ [Meet Core Location for spatial computing](#) On visionOS, app receives location updates if the user has granted permission and recently looked at the app.
- ☐ [Meet Core Location Monitor](#) CLMonitor API is a top level Swift actor. Add a condition to monitor, either a circular geographic area or beacon identity. Don't use in widgets.
- ☐ [Discover streamlined location updates](#) New Swift native CLLocationUpdate API to get location updates as an AsyncSequence.
- ☐ [What's new in Core Motion](#) Headphone motion manager for head tracking is coming to macOS 14. Water submersion manager on Apple Watch Ultra.

WWDC 2023 Viewing Guide

System Frameworks (continued)

Networking

- ☐ [Ready, set, relay: Protect app traffic with network relays](#) Alternative to VPN. Apps can use relays to hide client IP addresses to protect from tracking. MASQUE and Oblivious HTTP relays. New ProxyConfiguration class in Network framework, URLSession and WebKit.
- ☐ [Build robust and resumable file transfers](#) Using URLSession to create resumable file transfers. Depends on sever support, already implemented in SwiftNIO.
- ☐ [Reduce network delays with L4S](#) Low Latency, Low Loss, and Scalable throughput (L4S). Use for real-time audio and video in busy network conditions. Requires server support. URLSession and Network framework already support L4S with no code changes required. Turn on L4S on devices for testing, being rolled out to random set of iOS 17 and macOS Sonoma users.
- ☐ [Meet Swift OpenAPI Generator](#) Open source Swift Package plug-in for defining HTTP services in YAML or JSON and generating client and server code. Easy to mock APIProtocol for testing.

TipKit

- ☐ [Make features discoverable with TipKit](#) New framework to add tips to help user discover and learn app features.

Wallet

- ☐ [What's new in Wallet and Apple Pay](#) Apple Pay Later. Deferred payments. Transfer Funds. Tap to PresentID.

App Services

Widgets and App Clips

- ☐ [Bring widgets to new places](#) New locations for widgets on the mac desktop, iPad lock screen, StandBy mode on iPhone and Smart Stack on Apple Watch. Widget safe areas now replaced by content margins. containerBackground modifier. Vibrant rendering in StandBy Night mode.

WWDC 2023 Viewing Guide

App Services (continued)

- ☐ [Bring widgets to life](#) New Preview API makes it easier to provide a timeline to a widget preview. Interactive actions use App Intents. New API for button and toggle, no other controls are supported.
- ☐ [Design widgets for the Smart Stack on Apple Watch](#) Visual consistency with six design layouts to format widgets in the smart stack.
- ☐ [Build widgets for the Smart Stack on Apple Watch](#) Code-along session building a widget with an AppIntent.
- ☐ [Meet ActivityKit](#) Live activities to keep track of an event or task. In iOS 17 also appear in StandBy mode and iPad. Can be interactive.
- ☐ [Design dynamic Live Activities](#) Adjust height when you have less information to display. In Standby, layout is scaled up 200%, so check assets and images. In Dynamic Island keep things concentric with shape. In expanded views, hug the sensor.
- ☐ [Update Live Activities with push notifications](#) ActivityKit push notifications to update live activities.

App Intents and Shortcuts

- ☐ [Explore enhancement to App Intents](#) Widget configuration can use `AppIntentConfiguration` in code instead of separate Intent Definition file. Use Xcode to migrate. SwiftUI buttons and switches now support app intents for interactive widgets. `RelativeContext` APIs. `AppIntentsPackage` APIs. Define App Shortcuts in an App Intents extension so App doesn't have to launch. Apple Pay.
- ☐ [Spotlight your app with App Shortcuts](#) Add colors, thumbnail and title used when showing shortcut. Machine learning allows more flexible matching of similar phrases to trigger shortcuts. App Shortcuts on watchOS require app installed on the device.
- ☐ [Design Shortcuts for Spotlight](#) Shortcuts show up in Spotlight when searching for your app. Focus on shortcuts to perform your app's primary purpose. Make small App Shortcut icons recognizable. Keep titles short.

EventKit

- ☐ [Discover Calendar and EventKit](#) `EventKitUI` provides three view controllers for event editor, event details, and a calendar chooser. `EventKitUI` adds events without write access. In iOS 17, event UI runs in separate process. Virtual conference extensions.

WWDC 2023 Viewing Guide

App Services (continued)

Health and Fitness

- ☐ [Build custom workouts with WorkOutKit](#) New Swift framework to create custom workouts for Apple Watch.
- ☐ [Build a multi-device workout app](#) New Workout APIs to control and mirror workout sessions between Apple Watch and iPhone. New cycling metrics. HealthKit app now on iPad.

Push Notifications

- ☐ [Meet Push Notification Console](#) New tool for interacting with Apple Push Notifications. Send test notifications with custom payloads. Delivery log to see history. Generate and validate authentication tokens

CarPlay

- ☐ [Optimise CarPlay for vehicle systems](#) For automakers and developers of vehicle systems wanting to integrate CarPlay.

Design

- ☐ [Design with SwiftUI](#) Using SwiftUI as a design tool when designing the watchOS Maps app.

SF Symbols

- ☐ [What's new in SF Symbols 5](#) Animated SF Symbols. Animate appearance, bounce, scale, pulse, variable color, replace. Over 700 new symbols.
- ☐ [Animate symbols in your app](#) Symbols framework adds symbol effects. Preview and get the available effect names from the SF Symbols app. Start/stop indefinite animations, repeat discrete animations. In SwiftUI, `symbolEffectsRemoved` prevent propagation for a view.
- ☐ [Create animated symbols](#) Creating custom symbols that work with the new animation effects. Symbol components combine with a custom symbol to add badges and containers.

WWDC 2023 Viewing Guide

Safari and Web

- ☐ [What's new in web apps](#) Add to Home Screen now available in Safari View Controller. Web apps now on Mac.
- ☐ [Rediscover Safari developer features](#) Summary of the developer tools in Safari. iPhone, iPad and xrOS simulators are now available in Safari for previewing web content (needs Xcode installed). New feature flag settings to enable experimental features.
- ☐ [What's new in CSS](#) What's new and coming soon to Safari. Masonry layout in Safari Technology Preview. Margin trim property to remove margins from elements that push against the container leading to extra padding. Define color in the P3 color gamut. New line-height units. Font size adjust to make fonts visually consistent.
- ☐ [Explore media formats for the web](#) Support for JPEG-XL, AVIF and HEIC. Use the picture element to fallback to older formats. Managed media source API combines flexibility of MSE with efficiency of HLS.
- ☐ [What's new in Safari extensions](#) Content blockers now support :has() selectors. Declarative Net Request to modify request headers. Create a single SVG icon for an extension. New Safari profiles have separate history, cookie and website data.
- ☐ [What's new in Web Inspector](#) New typography inspector with sliders to vary font properties. Visual hint for unwanted scroll.

Audio and Video

Spatial Audio

- ☐ [Deliver video content for spatial experiences](#) Working with HLS to deliver 2D and 3D content with spatial audio using Multiview HEVC (MV-HEVC).
- ☐ [Create a great spatial playback experience](#) AVFoundation supports 3D video. AVPlayerViewController support for visionOS and RealityKit. Wrap in UIViewControllerRepresentable to use with SwiftUI. See sample code.

WWDC 2023 Viewing Guide

Audio and Video (continued)

Speech

- ☐ [Extend Speech Synthesis with personal and custom voices](#) Personal Voice lets people record and recreate their voice. Generated on device. Use in place of system voices and Live Speech type-to-talk.
- ☐ [Customise on-device speech recognition](#) Customize SFSpeechRecognizer's language model. Provide phrases you want to weight more heavily, Add domain specific vocabulary.

SharePlay

- ☐ [Add SharePlay to your app](#) Use AirDrop to start a SharePlay session.
- ☐ [Share files with SharePlay](#) GroupSessionJournal, new in iOS 17, can share attachments up to 100MB.
- ☐ [Design spatial SharePlay experiences](#) Spatial Persona templates. Use less UI. Be permissive with interactions. Keep shared context consistent.
- ☐ [Build spatial SharePlay experiences](#) SharePlay on visionOS. Use the system coordinator and scene association for visual consistency.

Other Audio and Video Sessions

- ☐ [Create a great ShazamKit experience](#) New SHManagedSession API to replace SHSession. Remove start/stop and setup code. New SHLibrary class replaces SHMediaLibrary.
- ☐ [Enhance your app's audio experience with AirPods](#) For macOS apps that play media register for Now Playing to work with AirPods automatic switching. Register for mute gesture notifications to know when mute state changes.
- ☐ [Tune up your AirPlay audio experience](#) AirPlay enhanced audio buffering to stream faster and reduce interruptions.
- ☐ [Integrate your media app with HomePod](#) Add a media intent to allow users to play media from your app over AirPlay to the HomePod.
- ☐ [What's new in voice processing](#) Voice processing API available for tvOS. New API for audio ducking. Muted talker detection now added to macOS 14 and tvOS 17.

WWDC 2023 Viewing Guide

Spatial Computing

Spatial computing is Apple's term that brings together ARKit, RealityKit, Reality Composer Pro, Unity, Metal and Compositor on a new platform - visionOS. There's a lot here! Don't miss the sessions on running existing apps on the platform.

Design

- ☐ [Design for spatial user interfaces](#) Three layer app icons. No light/dark mode, glass material and UI adapt to light and dark backgrounds. Font weights are heavier for legibility. Vibrancy indicates hierarchy. Center content to reduce neck's range of motion. Tab bar is vertical, floating on the side. Ornaments present toolbars at bottom. Modal sheets push window back.
- ☐ [Principles of spatial design](#) Windows use a glass background, can be resized and moved. Design with points. Human centered. Place important content in the center and require minimal movement. Take advantage of space and subtle depth. Nearby elements can be smaller. Guide focus, create atmosphere with sound. Do more with less.
- ☐ [Design for spatial input](#) Eyes and hands are the new spatial inputs. Interfaces react to where you look. More comfortable to look at the center. Keep content at the same depth. Avoid shapes with sharp edges. Make eye target area at least 60 points. Hover effects provide feedback. Direct touch can cause fatigue.
- ☐ [Design considerations for vision and motion](#) Incorrect or missing depth cues cause viewer discomfort. Use image cues such as color, blur, shadows and relative size. Keep text at comfortable distance and size. Minimize eye effort. Keep perception of world stationary to avoid motion sickness. Avoid head-locked content and oscillations.
- ☐ [Explore immersive sound design](#) Spatial audio examples (wear headphones).

Getting Started

- ☐ [Get started with building apps for spatial computing](#) Apps launch into the shared space by default and can switch to immersive. Windows contain SwiftUI scenes showing 2D or 3D content. Volumes show 3D content. Dedicated Full Space can passthrough the surroundings or be fully immersive. Adding xrOS target to existing app to preview how it will look on the new platform.

WWDC 2023 Viewing Guide

Spatial Computing (continued)

- ❑ [Develop your first immersive app](#) Using Reality Composer Pro with Xcode. xrOS app template for window or volume (3D) scene types. Immersive Space runs app in full space hiding other apps. Template loads RealityKit content. Most SwiftUI controls work as expected. Simulator and Xcode preview. Reality Composer Pro to create spatial content.
- ❑ [Build great games for spatial computing](#) Shared space vs full space vs fully immersive. Dynamic foveation renders higher resolution where the player is looking. Use RealityKit for spacial audio. Hand tracking as an input can be tiring. Unity or RealityKit to create 3D content.

Running Existing Apps

- ❑ [Run your iPad and iPhone apps in the Shared Space](#) Most iOS apps will run unmodified on visionOS. Uses the iPad variant, light mode, in landscape if available. Existing ARViews and ARSessions won't work. SpriteKit and Storyboards are only supported in the xrOS (Designed for iPad) target. ARKit and RealityKit are only available when building for xrOS.
- ❑ [Enhance your iPad and iPhone apps in the Shared Space](#) Optimize apps for the Shared Space. Add hover effects to custom controls. Don't make assumptions about availability of microphones and cameras.

SwiftUI

- ❑ [Meet SwiftUI for spatial computing](#) Getting started with SwiftUI for visionOS. New scene APIs for windows, volumes and full spaces. Glass background (no light or dark mode). Ornaments. Hand action gestures. Hover effects. Model3D or RealityView to show RealityKit models. Integrate ARKit for hand tracking.
- ❑ [Take SwiftUI to the next dimension](#) Volumes as a container for 3D content. RealityKit adds Model3D to allow SwiftUI to load 3D content. Attachments API lets you add SwiftUI content to RealityView content. Hand and eye gestures to interact with content.
- ❑ [Go beyond the window with SwiftUI](#) Only your app is visible in an immersive space scene. User is at the origin of the space. Use RealityView to load content asynchronously.
- ❑ [Elevate your windowed app for spatial computing](#) Updating a SwiftUI app for visionOS. Fixed scale bitmap images can look blurry with dynamic content scaling. Prefer vector assets, vibrancy, and semantic styles. Add hover effects to custom controls. Tab view to switch top-level views. Toolbar ornaments.

WWDC 2023 Viewing Guide

Spatial Computing (continued)

UIKit

- ❑ [Meet UIKit for spatial computing](#) visionOS does not support API deprecated prior to iOS 14. Some APIs such as `UIDeviceOrientation` and `UIScreen` are unavailable. Add a new layered App icon. Prefer semantic colors, materials, and font styles. Add hover style to custom controls. Move controls to an ornament. Use `UIHostingController` to host a `RealityView` to show 3D content.

RealityKit

- ❑ [Build spatial experiences with RealityKit](#) Create 3D content with RealityKit. New SwiftUI `RealityView` to contain RealityKit entities, convert between coordinate spaces, and subscribe to RealityKit events. Volumetric window style for 3D content. New `ImmersiveSpace` scene type.
- ❑ [Enhance your spatial computing app with RealityKit](#) SwiftUI support for RealityView. Attachments embed SwiftUI content into RealityKit content. Video player component in RealityKit for embedding video content in a 3D scene. Portals create a window into a different world. Particle emitters add visual effects such as sparks. Anchors place content on walls, floors or relative to your head or hand.
- ❑ [Meet RealityKit Trace](#) Instruments has a RealityKit Trace template to profile RealityKit content.
- ❑ [Optimize app power and performance for spatial computing](#) System is always rendering content. Use Instruments and Xcode gauges. RealityKit Trace template helps show when your app is causing power and performance issues. MetricKit to get data from users.
- ❑ [Explore rendering for spatial computing](#) RealityKit applies realistic rendering to content. Image Based Lighting (IBL). Grounding shadow helps show relative position of content. New shader graph material. Dynamic content scaling increases detail where the eye is looking.

ARKit

- ❑ [Meet ARKit for spatial computing](#) New design for visionOS based on lessons learned from iOS. ARKit for spatial computing available in Swift and C. Sensor data is never sent to client. Hand tracking provides anchors with skeletal data for each hand.
- ❑ [Evolve your ARKit app for spatial computing](#) Updating an existing AR app to take advantages of the updated ARKit and RealityKit frameworks.

WWDC 2023 Viewing Guide

Spatial Computing (continued)

Reality Composer Pro

- ☐ [Meet Reality Composer Pro](#) Walk through of Reality Composer Pro.
- ☐ [Explore materials in Reality Composer Pro](#) Physically based rendering (PBR). ShaderGraphMaterial based on MaterialX from ILM. Creating materials.
- ☐ [Work with Reality Composer Pro content in Xcode](#) Follow on from previous sessions to show how to make the content interactive.

Unity

- ☐ [Bring your Unity VR app to a fully immersive space](#) Build fully immersive apps with Unity for visionOS. Upgrade to Unity 2022 and use the Universal render pipeline.
- ☐ [Create immersive Unity apps](#) Creating immersive or shared space unity apps.

Safari

- ☐ [Meet Safari for spatial computing](#) Running Safari to view web content on visionOS. WebXR, based on WebGL, a W3C standard for immersive scenes on the web.

Quick Look

- ☐ [Discover Quick Look for spacial computing](#) Quick Look on visionOS to view USDZ files. Windowed Quick Look present previews outside your app.
- ☐ [Create 3D models for Quick Look spatial experiences](#) Quick Look shows a 3D model in a volume window. Many ways to create 3D models including object capture and RoomPlan. Keep under 200 mesh parts and 100k vertices.

RoomPlan

- ☐ [Explore enhancements to RoomPlan](#) ARSession combines RoomPlan with ARKit image capture. MultiRoom support. Pause session between scans. Works best for single-floor houses up to 2,000 square feet. VoiceOver support.

USD

- ☐ [Meet Object Capture for iOS](#) Using object capture for iOS to create 3D USDZ models. See sample code. Now uses LIDAR for low texture objects. Still best to avoid reflective or transparent objects.
- ☐ [Explore the USD ecosystem](#) Survey of what's available from Apple and third-parties to build USD 3D content for visionOS.

WWDC 2023 Viewing Guide

Machine Learning

- ☐ [Discover machine learning enhancements in Create ML](#) Text classifier using BERT embedding model is multilingual and more accurate. New multi-label image classifier template. Improving models with augmented training data.
- ☐ [Explore Natural Language multilingual models](#) Transformer based contextual embedding models - Bidirectional Encoded Representations from Transformers (BERT). Trained on sentences with one word masked out and asked to suggest the word. Can be trained on multiple languages at once. Currently supports 27 different languages.
- ☐ [Improve Core ML integration with async prediction](#) New optimize submodule for post-training model compression. New API for runtime inspection of compute device availability. Core ML optimizes and caches the model for the device when loading. New async prediction API.
- ☐ [Use Core ML Tools for machine learning model compression](#) Compressing model reduces the size but can impact accuracy. Core ML Tools 7 adds training time compression.

Vision

- ☐ [What's new in VisionKit](#) Create sticker from lifted subject. Visual Look Up adds food, products, signs, symbols. Optical flow tracking when recognizing text. Monetary values, Thai and Vietnamese, tables, context aware data detectors, mac Catalyst.
- ☐ [Detect animal poses in Vision](#) Real-time detection of joints and poses in cats and dogs.
- ☐ [Explore 3D body pose and person segmentation in Vision](#) Vision now captures body structure in 3D. Depth is now an input with an image. Vision can now select more than one person in an image.
- ☐ [Integrate with motorised iPhone stands using DockKit](#) DockKit allows iPhone to control motorized camera stands to track subjects in camera apps.
- ☐ [Lift subjects from images in your app](#) VisionKit subject lifting in a few lines of code. Vision is a lower level framework with more flexibility. Use CoreImage for image processing.

WWDC 2023 Viewing Guide

Metal and Games

- ☐ [Your guide to Metal ray tracing](#) How to use Metal ray tracing.
- ☐ [Discover Metal for immersive apps](#) Use CompositorServices and Metal to render content. Add interactivity with ARKit.
- ☐ [Optimise GPU renderers with Metal](#) Shader performance with function specialization, asynchronous compilation. Dynamic linking. Metal compiler options.

Three sessions on porting games to Mac and iPad using the Game Porting Toolkit.

- ☐ [Bring your game to Mac, Part 1: Make a game plan](#)
- ☐ [Bring your game to Mac, Part 2: Compile your shaders](#)
- ☐ [Bring your game to Mac, Part 3: Render with Metal](#)

One More Thing

Already a subscriber?

Are you an iOS developer interested in learning what's new but struggling to keep up? Maybe you're also a little tired of watching all these WWDC videos?

I write regular articles covering what's new in iOS and Swift. Find out what changed in the latest release of Xcode. What new features you need to support? What new bugs will waste your time!

Sign up to my free iOS newsletter and I'll send the full text of each new article direct to your inbox so you never miss a post!

Find out more and subscribe



useyourloaf.com/newsletter