Documentation

Language: Swift

Updates / WWDC24

WWDC24

Highlights of new technologies introduced at WWDC24.

Overview

Browse a selection of documentation for new technologies and frameworks introduced at WWDC24. Many existing frameworks have added significant functionality, and you’ll find new ways to enhance your apps targeting the latest platform release.

For a comprehensive list of downloadable sample code projects, see WWDC24 Sample Code. For the latest design guidance localized in multiple languages, see Human Interface Guidelines > What’s New.

Topics

Accessibility and inclusion

{} Enhancing the accessibility of your SwiftUI app

Support advancements in SwiftUI accessibility to make your app accessible to everyone.

🗎 Optimizing your app for Assistive Access

Adjust your app’s UI to make sure it works well for people who use Assistive Access.

🗎 Music Haptics

Play haptic tracks along with known music tracks.

App services

{} Building a guessing game for visionOS

Create a team-based guessing game for visionOS using Group Activities.

📄 Discovering HID devices from Terminal

Identify devices connected to your Mac from the command line.

📄 Creating virtual devices

Use and interact with a virtual human interface device for testing and development.

Audio and video

{} Creating a MIDI device driver

Implement a configurable virtual MIDI driver as a driver extension that runs in user space in macOS and iPadOS.

App Store distribution and marketing

{} Implementing Wallet Extensions

Support adding an issued card to Apple Pay from directly within Apple Wallet using Wallet Extensions.

📄 Loading the latest version of the Apple Pay JS SDK

Link to the most recent autoupdating version of the Apple Pay JS SDK or a version of your choice.

📄 Testing win-back offers in Xcode

Validate your app’s handling of win-back offers that you configure for the testing environment.

Developer tools

📄 Configuring your app icon using an asset catalog

Add app icon variations to an asset catalog that represents your app in places such as the App Store, the Home Screen, Settings, and search results.

📄 Determining how much code your tests cover

Use code coverage to focus new test development on areas that lack adequate testing.

📄 Adding tests to your Xcode project

Include test targets that build code to test the logic in your functions, check for integration issues, automate UI workflows, and measure performance.

📄 Updating your existing codebase to accommodate unit tests

Remove coupling between components to increase test coverage and reliability.

📄 Building your project with explicit module dependencies

Reduce compile times by eliminating unnecessary module variants using the Xcode build system.

Graphics and games

📄 Adding touch controls to games that support game controllers in iOS

Use touch input and virtual controllers to make your game available to players without controllers.

📄 Improving the player experience for games with large downloads

Provide ample content in your base installation and then use on-demand resources and the Background Assets API to handle additional content.

📄 Improving your game's graphics performance and settings

Fix performance glitches and develop default settings for smooth experiences on Apple platforms using the powerful suite of Metal development tools.

📄 Adapting your game interface for smaller screens

Make text legible on all devices the player chooses to run your game on.

📄 Personalizing spatial audio in your app

Enhance the realism of spatial audio output by tracking a person's head movement and accounting for their personal spatial audio profile.

Health and fitness

{} Visualizing HealthKit State of Mind in visionOS

Incorporate HealthKit State of Mind into your app and visualize the data in visionOS.

📄 Authorizing access to health data

Request permission to read and share data in your app.

Maps and location

📄 Creating a Maps token

Generate your token to access MapKit services with proper authorization.

Identifying unique locations with Place IDs

Obtain information about a point of interest that persists over its lifetime.

Displaying place information using the Maps Embed API

Show place information on a map using a URL.

Monitoring the user’s proximity to geographic regions

Use condition monitoring to determine when the user enters or leaves a geographic region.

ML and Vision

Core ML

Integrate machine learning models into your app.

Vision

Apply computer vision algorithms to perform a variety of tasks on input images and videos.

Photos and camera

Writing spatial photos

Create spatial photos for visionOS by packaging a pair of left- and right-eye images as a stereo HEIC file with related spatial metadata.

Creating spatial photos and videos with spatial metadata

Add spatial metadata to stereo photos and videos to create spatial media for viewing on Apple Vision Pro.

Spatial computing

Interacting with your app in the visionOS simulator

Use your Mac to navigate spaces and control interactions with your visionOS apps in Simulator.

Understanding the visionOS render pipeline

Compare how visionOS handles events and manages its rendering loop differently from other Apple platforms.

Swift

Creating a data visualization dashboard with Swift Charts

Visualize an entire data collection efficiently by instantiating a single vectorized plot in Swift Charts.

Traits

Annotate test functions and suites, and customize their behavior.

Running tests serially or in parallel

Control whether tests run serially or in parallel.

Testing asynchronous code

Validate whether your code causes expected events to happen.

Defining test functions

Define a test function to validate that code is working correctly.

Migrating a test from XCTest

Migrate an existing test method or test class written using XCTest.

SwiftUI and UI frameworks

Creating visual effects with SwiftUI

Add scroll effects, rich color treatments, custom transitions, and advanced effects using shaders and a text renderer.

Elevating your iPad app with a tab bar and sidebar

Provide a compact, ergonomic tab bar for quick access to key parts of your app, and a sidebar for in-depth navigation.

Customizing a document-based app’s launch experience

Add unique elements to your app’s document launch scene.

Collaborating and sharing copies of your data

Share data and collaborate with people from your app.

System services

Creating your first app intent

Create your first app intent that makes your app available in system experiences like Spotlight or the Shortcuts app.

Making actions and content discoverable and widely available

Adopt App Intents to make your app discoverable with Spotlight, controls, widgets, and the Action button.

Identifying and blocking calls

Create a Call Directory app extension to identify and block incoming callers by their phone number.

Getting up-to-date calling and blocking information for your app

Implement the Live Caller ID Lookup app extension to provide call-blocking and identity services.

Adding your app’s content to Spotlight indexes

Create a description for your app’s content and add it to a Spotlight index to make it searchable.

Building a search interface for your app

Add a search interface to your app to execute Spotlight queries and offer suggested text completions.

Searching for information in your app

Search for app-specific content and refine search results using predicates and filters.

Sending channel management requests to APNs

Manage channels that your application uses for broadcast push notifications.

Troubleshooting push notifications

Debug your server to send push notifications with device and broadcast push notifications.

Using iCloud with macOS virtual machines

Access iCloud from macOS guest virtual machines.

See Also

WWDC

WWDC25

Highlights of new technologies introduced at WWDC25.

WWDC23

Highlights of new technologies introduced at WWDC23.

WWDC22

Highlights of new technologies introduced at WWDC22.

WWDC21

Highlights of new technologies introduced at WWDC21.