

[Updates](#) / WWDC22

# WWDC22

Highlights of new technologies introduced at WWDC22.

## Overview

Browse a selection of documentation for new technologies, frameworks, and APIs introduced at WWDC22. 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 [WWDC22 Sample Code](#). For the latest design guidance, see [Human Interface Guidelines > What's New](#).

## Topics

### SwiftUI



#### SwiftUI

Declare the user interface and behavior for your app on every platform.



#### Swift Charts

Construct and customize charts on every Apple platform.



#### Food Truck: Building a SwiftUI multiplatform app

Create a single codebase and app target for Mac, iPad, and iPhone.



#### Core Transferable

Declare a transfer representation for your model types to participate in system sharing and data transfer operations.

`@frozen struct Image`

A view that displays an image.

## UIKit

`{}` Supporting desktop-class features in your iPad app

Enhance your iPad app by adding desktop-class features and document support.

 Building a desktop-class iPad app

Optimize your iPad app's user experience by adopting desktop-class enhancements for multitasking with Stage Manager, document interactions, text editing, search, and more.

`@MainActor class UIFindInteraction`


An interaction that provides text finding and replacing operations using a system find panel.

`@MainActor class UIEditMenuInteraction`

An interaction that provides edit operations using a menu.

`@MainActor var interactionActivityTrackingBaseName: String? { get set }`

The base name the view controller uses for logging signposts that annotate user interactions.

 Presenting content on a connected display

Fill connected displays with additional content from your app.

 TextKit

Manage text storage and perform custom layout of text-based content in your app's views.

`class UIImage`

An object that manages image data in your app.

`@MainActor class UIPageControl`

A control that displays a horizontal series of dots, each of which corresponds to a page in the app's document or other data-model entity.

`@MainActor class UICalendarView`

A view that displays a calendar with date-specific decorations, and provides for user selection of a single date or multiple dates.

## AppKit

 AppKit

Construct and manage a graphical, event-driven user interface for your macOS app.

`@MainActor class NSComboButton`

A button with a pull-down menu and a default action.

`protocol NSPreviewRepresentableActivityItem : NSObjectProtocol`

An interface you adopt in custom objects that you want to share using the macOS share sheet.


`class NSImage`

A high-level interface for manipulating image data.

 TextKit

Manage text storage and perform custom layout of text-based content in your app's views.

## Xcode and Developer Tools

 Configuring a multiplatform app

Share project settings and code across platforms in a single app target.

 Configuring requirements for merging a pull request

Protect stable branches by requiring a successful Xcode Cloud build or action before it's possible to merge a pull request.

 Documenting apps, frameworks, and packages

Create developer documentation from in-source comments, add articles with code snippets, and add tutorials for a guided learning experience.

 Enabling Developer Mode on a device

Grant or deny permission for locally installed apps to run on iOS, iPadOS, visionOS, and watchOS devices.

 Xcode Cloud

Automatically build, test, and distribute your apps with Xcode Cloud to verify changes and create high-quality apps.

## Swift Programming Language

 Swift

Build apps using a powerful open language.

# Widgets and Complications

## Creating accessory widgets and watch complications

Support accessory widgets that appear on the Lock Screen and as complications on Apple Watch.

# App Intents

## App Intents

Make your app's content and actions discoverable with system experiences like Spotlight, widgets, and 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.

## Integrating custom data types into your intents

Provide the system with information about the types your app uses to model its data so that your intents can use those types as parameters.

## App intents

Define the custom actions your app exposes to the system, and incorporate support for existing SiriKit intents.

## Focus

Adjust your app's behavior and filter incoming notifications when the current Focus changes.

# Shared with You

## Shared with You

Surface shared content and collaborate in your app.

```
@MainActor class SWAttributionView
```

A view that displays the sender who shares a highlight and provides related actions.

```
class SWHighlight
```

An object that represents a universal link to share by any number of contacts in one or more conversations.

```
class SWHighlightCenter
```

An object that contains a priority-ordered list of universal links to share with the current user.

```
protocol SWHighlightCenterDelegate : NSObjectProtocol
```

The protocol you use to notify the delegate when the list or rank order of surfaced highlights changes.

## Extensions



ExtensionKit

Make custom UI from an app extension available in a host app, and manage the list of enabled and disabled app extensions.



ExtensionFoundation

Create executable bundles to extend the functionality of other apps.

## Augmented Reality



RoomPlan

Create a 3D model of a room by interactively guiding people to scan their physical environment using a device's camera.

```
class ARPlaneExtent
```

The size and y-axis rotation of a detected plane.

```
class ARGeoTrackingConfiguration
```

A configuration that tracks locations with GPS, map data, and a device's compass.

```
var isCameraAssistanceEnabled: Bool { get set }
```

A Boolean value that combines the spatial awareness of ARKit with Nearby Interaction to improve the accuracy of a nearby object's position.

```
var isCameraAssistanceEnabled: Bool { get set }
```

A Boolean value that combines the spatial awareness of ARKit with Nearby Interaction to improve the accuracy of a nearby object's position.

## Metal



MetalFX

Boost your Metal app's performance by upscaling lower-resolution content to save GPU time.



Resource loading

Load assets in your games and apps quickly by running a dedicated input/output queue alongside your GPU tasks.

`protocol MTLDevice : NSObjectProtocol, Sendable`

The main Metal interface to a GPU that apps use to draw graphics and run computations in parallel.

~~`func makeIOHandle(url: URL, compressionMethod: MTLIOCompressionMethod) throws -> any MTLIOFileHandle`~~

Creates an input/output file handle instance that represents a compressed file at a URL.

Deprecated

~~`func makeIOHandle(url: URL) throws -> any MTLIOFileHandle`~~

Creates an input/output file handle instance that represents a file at a URL.

Deprecated

`func makeIOCommandQueue(descriptor: MTLIOCommandQueueDescriptor) throws -> any MTLIOCommandQueue`

Creates an input/output command queue you use to submit commands that load assets from the file system into GPU resources or system memory.

## Audio, Video, and Media



### ShazamKit

Find information about a specific audio recording when a segment of it's part of captured sound in the Shazam catalog or your custom catalog.



### AVKit

Create user interfaces for media playback, complete with transport controls, chapter navigation, picture-in-picture support, and display of subtitles and closed captions.



### AVFoundation

Work with audiovisual assets, control device cameras, process audio, and configure system audio interactions.



### Creating images from a video asset

Display images for specific times within the media timeline by generating images from a video's frames.



### Loading media data asynchronously

Build responsive apps by using language-level concurrency features to efficiently load media data.



## Encoding and decoding audio

Convert audio formats to efficiently manage data and quality.

### `class AVAudioSequencer`

An object that plays audio from a collection of MIDI events the system organizes into music tracks.



#### Creating a camera extension with Core Media I/O

Build high-performance camera drivers that are secure and simple to deploy.



#### Overriding the default USB video class extension

Create a simple DriverKit extension to override the default driver-matching behavior for USB devices.



## Capturing screen content in macOS

Stream desktop content like displays, apps, and windows by adopting screen capture in your app.

## WatchKit

### `class WKBluetoothAlertRefreshBackgroundTask`

A task for handling timely Bluetooth alerts in the background.



#### Using background tasks

Handle scheduled update tasks in the background, and respond to background system interactions including Siri intents and incoming Bluetooth messages.

## Web and Safari



#### Syncing Safari web extensions across devices and platforms

Let users install your extension on one device and then use and manage the extension on all their other iOS and macOS devices.



#### Sending web push notifications in web apps and browsers

Update your web server and website to send push notifications that work in Safari, other browsers, and web apps, following cross-browser standards.

## Spotlight Search



### Core Spotlight

Add search capabilities to your app, and index your content so people can find it from Spotlight and Safari.

## Weather



### WeatherKit

Deliver weather conditions and alerts to your users.

## Live Text



### Enabling Live Text interactions with images

Add a Live Text interface that enables users to perform actions with text and QR codes that appear in images.



### Scanning data with the camera

Enable Live Text data scanning of text and codes that appear in the camera's viewfinder.

## Apple Maps



### MapKit

Display map or satellite imagery within your app, call out points of interest, and determine placemark information for map coordinates.



### MapKit JS

Embed interactive Apple Maps on your website, annotate points of interest, and perform georelated searches.



### Apple Maps Server API

Reduce API calls and conserve device power by streamlining your app's georelated searches.



### Interacting with nearby points of interest

Provide automatic search completions for a partial search query, search the map for relevant locations nearby, and retrieve details for selected points of interest.



### Explore a location with a highly detailed map and Look Around

Display a richly detailed map, and use Look Around to experience an interactive view of landmarks.

## Apple Pay and Wallet





## Wallet

Manage tickets, boarding passes, payment cards and other passes in the Wallet app.



## PassKit (Apple Pay and Wallet)

Process Apple Pay payments in your app, and create and distribute passes for the Wallet app.



## Wallet Orders

Create, distribute, and update orders in Wallet.



## Apple Pay Merchant Token Management API

Retrieve and manage payment life-cycle events for your Apple Pay merchant tokens.



## Apple Pay on the Web

Support Apple Pay on your website with JavaScript-based APIs.



## Payment token format reference

Verify an Apple Pay payment token and validate a transaction.

# App Store and Distribution



## StoreKit

Support In-App Purchases and interactions with the App Store.



## App Store Server API

Manage your customers' App Store transactions from your server.



## App Store Server Notifications

Monitor In-App Purchase events in real time and learn of unreported external purchase tokens, with server notifications from the App Store.



## Notary API

Submit your macOS software for notarization through a web interface.

# Security and Privacy



## Supporting passkeys

Eliminate passwords for your users when they sign in to apps and websites.



## Connecting to a service with passkeys

Allow users to sign in to a service without typing a password.



## Public-Private Key Authentication

Register and authenticate users with passkeys and security keys, without using passwords.



## Service Management

Manage startup items, launch agents, and launch daemons from within an app.

```
class LARight
```

A grouped set of requirements that gate access to a resource or operation.

```
class LAPersistedRight
```

A right that gates access to a key and a secret.

```
@MainActor @preconcurrency struct LocalAuthenticationView<Label> where  
Label : View
```

A SwiftUI view that displays an authentication interface.

```
class LARightStore
```

A container for data protected by a right.

# Machine Learning



## Create ML Components

Create more customizable machine learning models in your app.

# Communication



## Push to Talk

Display the system user interface for your app's Push to Talk services.

# Performance Analysis



## MetricKit

Aggregate and analyze per-device reports on exception and crash diagnostics and on power and performance metrics.

```
class MXMetricManager
```

The shared object that registers you to receive metrics, creates logs for custom metrics, and gives access to past reports.


```
class MXAppLaunchDiagnostic
```


A diagnostic subclass that encapsulates app launch diagnostic reports.


```
class MXAppLaunchMetric
```

An object representing metrics about app launch time.

## Hardware and Virtual Machines

 DriverKit  
Develop device drivers that run in user space.

 SCSIPeripheralsDriverKit  
Develop drivers for peripherals that use SCSI Block Command and Multimedia Command protocols.

 DeviceDiscoveryExtension  
Stream media to a third-party device that a user selects in a system menu.

 Virtualization  
Create virtual machines and run macOS and Linux-based operating systems.

## Photos

```
class PHPhotoLibrary
```

An object that manages access and changes to the user's photo library.

```
@MainActor @preconcurrency struct PhotosPicker<Label> where Label :  
View
```

A view that displays a Photos picker for choosing assets from the photo library.

**{}** Supporting Continuity Camera in your macOS app  
Enable high-quality photo and video capture by using an iPhone camera as an external capture device.

## Foundation

```
struct Components
```

A type that represents the components of a locale, for use when creating a locale with specific overrides.

```
struct LocalizedStringResource
```

A reference to a localizable string, accessible from another process.

## Backgrounds Assets



### Background Assets

Improve or eliminate the time people wait while your app downloads assets.

## CarKey



### CarKey

Access the remote keyless features of configured vehicles in the Wallet app.

## Apple School Manager



### Roster API

Read information about people and classes from an Apple School Manager organization.

---

## See Also

### WWDC



#### WWDC25

Highlights of new technologies introduced at WWDC25.



#### WWDC24

Highlights of new technologies introduced at WWDC24.



#### WWDC23

Highlights of new technologies introduced at WWDC23.



#### WWDC21

Highlights of new technologies introduced at WWDC21.