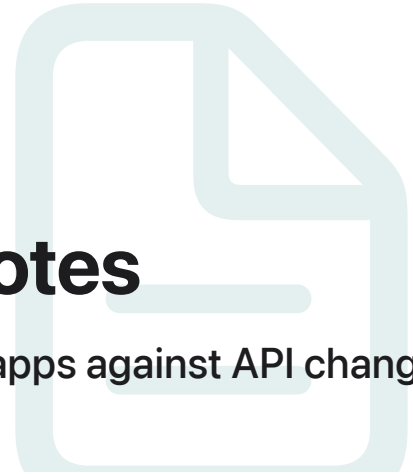


Article

iOS & iPadOS 26 Release Notes

Update your apps to use new features, and test your apps against API changes.



Overview

The iOS & iPadOS 26 SDK provides support to develop apps for iPhone and iPad running iOS & iPadOS 26. The SDK comes bundled with Xcode 26, available from the Mac App Store. For information on the compatibility requirements for Xcode 26, see [Xcode 26 Release Notes](#).

General

New Features

- Recovery Assistant is a new way to recover your device if it doesn't start up normally. It can look for problems and attempt to resolve them if found. (151856202)

Resolved Issues

- Fixed: Some iPhone 15 and iPhone 16 models might show a low battery symbol and be unable to start up after updating to iOS 26 beta 1.

Follow the steps below to restore the affected iPhone using the updated version of iOS 26 beta 1, or to downgrade to the currently-released version of iOS.

1. Disconnect the iPhone from all USB-C cables.

2. Connect a USB-C cable to a computer, then open Finder if using Mac, or the Apple Devices app if using Windows.
3. When the iPhone's screen is black, connect the USB-C cable to the iPhone while simultaneously pressing and holding both the side button and volume down button.
4. After 3 seconds, release the side button and continue holding the volume down button until you see the iPhone in Finder or Apple Devices (this may take up to 15 seconds).
5. Restore the iPhone using the updated version of iOS 26 beta 1 or the version of iOS that is currently available.
6. Restore the iPhone from your backup. (153071880)

AdAttributionKit

New Features

- Advertisers of apps can now specify attribution windows based on ad network and ad interaction type in the Info.plist. This information is taken into account when deciding which ad impression drives the conversion. (136554905)
- Advertisers of apps can now specify cooldown periods during which ad conversions will not be attributed for their app. This cooldown can be specified for app installs and reengagement attribution. (136728405)
- AdAttributionKit now provides country code in the postback when crowd anonymity thresholds are met. (136850837)

AirPlay

Known Issues

- AutoMix song transitions might not work as expected when streaming with AirPlay. (155925891)

Alternate App Distribution

Known Issues

- In-development web browsers, when tested on a device that is not eligible for alternate app distribution features, will not be able to successfully install web-distributed apps. The app installation will begin but will fail before completion. (152209203)

App Store

New Features

- A new Accessibility section has been added to the App Store product pages that highlights accessibility features within apps and games. These Accessibility Nutrition Labels give users a new way to learn if an app will be accessible to them before they download it, and give developers the opportunity to better inform and educate their users on features their app supports. (138344118)

AppIntents

Resolved Issues

- Fixed: Code that references `@AppIntent(schema: .visualIntelligence.semanticContentSearch)` fails to compile. (152321182)

Apple Intelligence

New Features

- The Foundation Models framework provides you with direct access to the on-device large language model at the core of Apple Intelligence. For release notes about the framework, see the Foundation Models framework section below. (139996377)

Resolved Issues

- Fixed: In Image Playground and Genmoji, the new modifiers to customize appearance do not work for Japanese-language users. Image creation is blocked when attempting to add modifiers to its appearance. (151833204)
- Fixed: Model quality output degrades after extended, repeated inferences of the same adapter. (152468267)

Apple TV app

Resolved Issues

- Fixed: Certain text on Apple TV app content cards might be hard to read. (149376362)
- Fixed: Certain line items in Apple TV app settings appear twice or disappear unexpectedly. (152248009)
- Fixed: Certain text and images in the “Key Plays” content tab in live sports playback might not display correctly. (152516659)

Apps

Resolved Issues

- Fixed: On iPhone 11, if you have organized apps into folders on your Home Screen, you might only see the leftmost column of the grid of apps when opening a folder. The other two columns in the folder do not display the app icons, preventing users from launching those apps. (156425266)

Known Issues

- On some iPads, buttons and symbols might show visual corruption during app launch. (152442679)

ARKit

Resolved Issues

- Fixed: Usage of ARBodyTrackingConfiguration causes a crash on certain devices. (152417820)

AssistantSchemas

Resolved Issues

- Fixed: If you have adopted any of the following email AssistantSchemas, you will experience a compilation error due to a parameter type change: createDraft, updateDraft, reply Mail, forwardMail, message, and draft. (148633307)

AuthenticationServices

New Features

- ASAuthorizationControllerRequestOptions.preferImmediatelyAvailable Credentials now works for passkey registration requests. This request only shows UI when the device is immediately able to create a passkey; otherwise, no UI is shown. (150688929)

AVFoundation

Resolved Issues

- Fixed: AVPlayerLayer does not ensure a valid video frame is always displayed during item replacement, as the isReadyForDisplay property does not remain true during transitions. (151902458)

Background Assets

Resolved Issues

- Fixed: The system might not update downloaded asset packs as frequently as is expected for apps that internal testers install from TestFlight. (143281558)
- Fixed: Asset pack downloads might fail unexpectedly or stall indefinitely, including across reboots. (151498902)
- Fixed: The system might not deliver status updates to your app for ongoing asset pack downloads. (151647839)
- Fixed: Pausing and resuming an app installation or update while the system is downloading essential asset packs might cause the installation or update to stall indefinitely. (151942388)
- Fixed: The installation of large asset packs might fail. (153128086)
- Fixed: The URL Override setting doesn't save an entered URL's port number. The device always attempts to communicate with the mock server over port 443. (156113742)

(FB18913475)

Bluetooth

Resolved Issues

- Fixed: Certain Non-Connectivity Car Consortium (CCC) Digital Car Key pairings might fail. (154703309)

Books

Resolved Issues

- Fixed: Highlight menu color palette in EPUB is black and white. (152321612)

Camera

Resolved Issues

- Fixed: In some cases, the mode selection picker remains expanded after launch. (154589210)

Known Issues

- Sharing a photo immediately after capture might take longer than expected. (154875558)

CarPlay

Resolved Issues

- Fixed: CarPlay template apps that use the Now Playing screen might occasionally not load artwork on full-screen Now Playing. (151807323)

Cellular Data Usage

Known Issues

- On a Dual SIM iPhone with multiple lines enabled, cellular data usage might not update after changing the default line for cellular data. (152608995)

Workaround: Restart the device after switching the default cellular data line.

CloudKit

Resolved Issues

- Fixed: CloudKit sharing URLs do not launch third-party apps. (151778655)
- Fixed: The request access APIs, such as CKShareRequestAccessOperation, are available in the SDK but are currently nonfunctional. (151878020)

Communication Safety

Known Issues

- Communication Safety might cause hangs in FaceTime when triggered to show an intervention. (151729870)

Workaround: Reboot the device.

Contacts

Known Issues

- Characters for Monogram Avatars in some non-English languages might show up as unexpected symbols. (154824077)

CoreData

Resolved Issues

- Fixed: In beta 5 SDK, CoreData changed several Sendable annotations to resolve compatibility issues with Swift 6's new MainActor default isolation feature. These changes include marking NSManagedObject as NS_SWIFT_NONISOLATED NS_SWIFT_NONSENDABLE, marking NSManagedObjectContext as NS_SWIFT_NONISOLATED NS

`_SWIFT_SENDABLE`, and requiring `NS_SWIFT_SENDABLE` closures for the family of `perform`, `performBlock`, `performBlockAndWait` and similar methods. These changes are ABI compatible with past releases but might introduce new warnings while building source code that violates the longstanding CoreData concurrency guidelines.

`NSManagedObject` are mutable reference types inextricably related to others in a graph and cannot be made `Sendable`. They are expected to be isolated to the scope of the `NSManagedObjectContext` that creates or fetches them. `NSManagedObjectContext` is a style of actor which encapsulates its own dispatch queue. While it's impermissible to use many methods on `NSManagedObjectContext` from other threads, it is permissible to pass references around to invoke the `performBlock` family of methods, for the purpose of routing a `Sendable` closure to its managed dispatch queue. CoreData supports a user default `-com.apple.CoreData.ConcurrencyDebug 1` which can be used to enable additional assertions. (153848710) (FB18216198)

Deprecations

- This release removes support for the following deprecated `NSPersistentStore` option keys:
 - `NSPersistentStoreUbiquitousContentNameKey`
 - `NSPersistentStoreUbiquitousContentURLKey`
 - `NSPersistentStoreUbiquitousPeerTokenOption`
 - `NSPersistentStoreRemoveUbiquitousMetadataOption`
 - `NSPersistentStoreUbiquitousContainerIdentifierKey`
 - `NSPersistentStoreRebuildFromUbiquitousContentOption`

These keys have been deprecated for more than a decade. Affected apps rebuilt with the iOS or macOS 26 SDK will get errors. Apps built against older SDKs will log warnings when opening the store.

The on-device persistent store will be fully usable after these keys are removed from options dictionary, albeit without syncing. For data syncing, these APIs were superseded by `NSPersistentCloudKitContainer` in iOS 13 and `SwiftData` in iOS 17. If absolutely necessary, the iCloud content remains accessible in the containers with the last baseline as a .zip archive file. (157297746) (FB19286235)

DeclaredAgeRange API

Known Issues

- If requestAgeRange triggered the onboarding flow for a parent, selecting Always or Never will lead to the API not responding. (152187255)

Workaround: Try calling requestAgeRange again to get a valid response.

- On iOS, requestAgeRange might incorrectly return notAvailable. (152194790)

Workaround: Please try signing out and back in with your Apple Account or wait 24 hours.

- On macOS, if onboarding leads to the Ask First state, the user is not prompted to share or not share their age range. (152327536)

Workaround: Try calling requestAgeRange again to get a valid response.

Documents

Known Issues

- EPUB files are shown with an app icon rather than a document icon. Other document types might also be affected. The icon shown is the icon of the app set to open the document. The document opens normally. (154800649)

Find My App

Resolved Issues

- Fixed: Play Sound fails when invoked from Share Item Location links, and "Not Reachable" is displayed. (152029301)

Foundation

New Features

- Interpolating non-localized types into a LocalizedString
Resource/String(localized:)/AttributedString(localized:) value will now display a deprecation warning instead of potentially falling back to a fully-unlocalized string.

Provide a localized value to interpolate into the text instead, or silence the warning by wrapping the unlocalized value in a call to `String(describing:)`. (126876158)

- `ISO8601FormatStyle` now allows fractional seconds, regardless of the setting of `includingFractionalSeconds`. Additionally, `ISO8601FormatStyle` now allows hours-only time zone offsets. (136950769)

Resolved Issues

- Fixed: Functions that initialize a string from a C-string pointer do not validate that the provided encoding is a valid encoding for C-strings. (143756086) (FB16417968)

Foundation Models framework

New Features

- `LanguageModelSession.prewarm()` now caches the instructions and prefix of your prompt, in addition to loading system resources. The best place to use `prewarm()` is when your app is waiting for user interactions that might trigger response generation. This will reduce the time to the first-generated token. (152381043)
- `#Playground` in Xcode now allows for filing feedback for Foundation Models framework responses. (153770707)
- The `.contentTagging` use case now supports non-English languages. You can query the supported languages using `SystemLanguageModel(useCase: .contentTagging).supportedLanguages`. Tags will be generated based on the language of the prompt. (155801948)
- When you use guided generation, the framework now supports programmatically detecting when and why the model refuses to respond when the content is not unsafe. In previous beta releases, this manifested as a guardrail violation. Now you can catch a specific error case `LanguageModelSession.GenerationError.refusal` and obtain a model-generated explanation for why it refused to respond. Depending on your use case, you might be able to display the explanation in your UI. (156086748)
- In `GeneratedContent`, you can use the `isComplete` property to check whether the content was fully generated by the model. (156109416)
- When using guided generation, you can now access the underlying weakly typed `GeneratedContent` by accessing the `rawContent` property on `Response` or `ResponseStream`.

(156351123)

- New guardrails mode `Guardrails.permissiveContentTransformations` allows transformations of content that might otherwise violate the default guardrails. Use this mode for text-to-text tasks, such as summarization and rewrite. (156721060)

Resolved Issues

- Fixed: When you pass `includeSchemaInPrompt: false` to `respond` or `streamResponse`, it is not respected. (151926006)
- Fixed: Custom `@Generable` types named `Number` or `Boolean` might not function reliably in guided generation. (152280144)
- Fixed: Tool calling might not function properly if primitive types such as `Int`, `String`, or `Bool` are used as the argument. (152318534)
- Fixed: In an Xcode Playground, requests made to the model might receive a "rate limit exceeded" error. (152325506)
- Fixed: Model requests in macOS command line tools might experience rate limiting. (152681332)
- Fixed: Generable types with a recursive definition lead to a `SchemaError` when generating a response. (153147722) (FB17962270)
- Fixed: A prompt containing Chinese might lead to error "Unsupported language zh-hans detected". (153151710) (FB17963656)
- Fixed: Generable types cannot be made public due to a bug in the `Generable` macro. (153216183) (FB17990794)
- Fixed: Requests to the model might experience rate limiting, even when the device is connected to power. (153216632)
- Fixed: The Foundation Models framework cannot be imported when building for Mac Catalyst. (153255533) (FB18004324)
- Fixed: Creating a `LanguageModelSession` with tools that have duplicate names leads to a fatal error. (153426645) (FB18074984)
- Fixed: When using guided generation and tool calling together, some requests might lead to `LanguageModelSession.GenerationError.decodingFailure` due to a bug in constrained decoding. (153773704) (FB18190120)

- Fixed: When creating a session with a transcript, any new tools you attach to the session might not be used. Additionally, if your transcript contains tools from a previous session, those tools must be passed as the `tools :` argument in the new session in order to function properly. (154904647)
- Fixed: Some prompts erroneously return an error indicating a guardrail violation (`FoundationModels.LanguageModelSession.GenerationError error 2`). (155273863)
- Fixed: When using tool calling with guided generation, tools might not get called. (155313086) (FB18691470)
- Fixed: Generating an enum with associated values through a dynamic schema crashes with a `SchemaError`. (155957346) (FB18878026)
- Fixed: In some cases, requests you make to the model erroneously throw a `guardrail Violation` for all prompts due to model assets not fully downloaded. (156223847) (FB18944619)

Known Issues

- Some instructions and prompts to the model might not lead to expected responses. Use Feedback Assistant to report satisfactory or unsatisfactory cases. When applicable, you can use `LanguageModelSession.logFeedbackAttachment()` to serialize the feedback to a JSON file and attach it to Feedback Assistant. (152318091)

Workaround: Refine your instructions and prompts using `#Playground` in Xcode. When applying guided generation with a `@Generable` type, add `@Guide` with a custom description on properties to steer the model's responses.

- If `Argument` type is an enum with associated type, tools will encounter a `ToolCallError` with an underlying `decodingFailure` when the model attempts to call it. (156723065)

Workaround: Wrap the enum in a struct.

Game Center

Known Issues

- User cannot start a Challenge for a leaderboard that has not been published Live via App Store Connect. (151928236)

Workaround: Test leaderboard locally using Game Progress Manager in Xcode, or publish a leaderboard as Live before creating an associated challenge.

Game Controller

New Features

- For supported game controllers, pressing the Home button once opens the Game Overlay. Set `preferredSystemGestureState` to receive additional Home button press events. (137780853)
- Pair multiple Apple devices to your DualSense or DualSense Edge controller and easily switch between them directly using the controller or Bluetooth settings. See [this PlayStation blog post](#) for details. (137782227)

Game Mode

Resolved Issues

- Fixed: The `LSSupportsGameMode Info.plist` key is currently ignored. (151814151)

HealthKit

New Features

- A user's medications and their dose events in the Health app are now available for reading via a new API. Use `HKUserAnnotatedMedicationQuery` to get a list of `HKUserAnnotatedMedication` objects that encapsulate the users medications. A new `HKSample` subclass, `HKMedicationDoseEvent`, tracks the medication doses logged by the user. (114279172)
- Workout session APIs are now available on iOS and iPadOS. This includes `HKWorkoutSession` and `HKLiveWorkoutBuilder`, which can be used in your app to track workouts. (125746390)

Resolved Issues

- Fixed: An app that was granted Read-only authorization for certain data types (e.g. workout routes) might lose its ability to read the data type without the user explicitly turning it off in

Settings > Health. (149024236)

- Fixed: The authorization screen prompted by the Medications Authorization API cannot be dismissed if the user has no available medications needing authorization. (152094574)
- Fixed: On an iPhone or iPad without a passcode, Health data might become inaccessible when you reach the Power Off slider. (155576088)

Image Playground

Resolved Issues

- Fixed: The Create Image action fails to appear in Shortcuts app and Spotlight. (153235442)

iPad Multitasking

Resolved Issues

- Fixed: The Multitasking mode picker is missing from Control Center on some iPad models. (152457491)

Known Issues

- Apps launched on external displays default to the same size and position. (143158703)
- Switching apps by swiping left or right with one finger along the bottom edge, or with four/five fingers anywhere on screen, does not work in the new Windowed App Multitasking mode. (151293681)

Journaling Suggestions

New Features

- Journaling Suggestions on iPhone now sync over iCloud securely to iPads. Users can enjoy the suggestions created on iPhone on their iPad Journal app or other journaling apps that adopt the API. (152322897)
- Smart notifications now let users know when it's the best time to journal, based on their routine and location. You can also adopt smart notifications in your journaling app.

(152469176)

- Suggestions have been revamped to show scene classification and inferences such as Holidays and Celebrations, new groupings based on patterns, and more. (152469363)

Resolved Issues

- Fixed: Journaling suggestions might not download. (151995727)

Known Issues

- iCloud Storage for Journaling Suggestions returns after "Delete Data From iCloud". (152102605)

Workaround: Disable Journaling Suggestions, in Settings > Apple Account > iCloud > Saved to iCloud > Journaling Suggestions, before invoking "Delete Data From iCloud".

Kernel

Deprecations

- @Description: Calls to `sem_open` or `sem_unlink` from a process signed with a Team ID entitlement no longer observe any semaphores created by processes signed with different Team IDs. Named semaphores are now scoped to a single development team. (123476459)

Keyboards

Resolved Issues

- Fixed: Keyboards are sometimes missing keys in first-party apps. The keys are not visible, but users are still able to type over the missing key to use it. (152375527)
- Fixed: Users with Korean 10-key keyboard might experience missing and blank keys on iPhone. (154158767)
- Fixed: Tapping the Return key on Japanese and Chinese 10-key keyboards is unresponsive after you rotate the device from Landscape to Portrait. (154163977)

Landscape Mode

Known Issues

- Safe area, search bar alignment, and locations might be incorrect in landscape mode. (152365383)

Lock Screen

Resolved Issues

- Fixed: Adjusting the glass tint color for Lock Screen time might not be easily accessible. (151240707)
- Fixed: When editing a photo wallpaper the widget area incorrectly overlaps with the time. (151418602)
- Fixed: If you create a wallpaper using the first-row circles at the top of the Wallpaper Gallery, you might not be able to place widgets directly below the time. (151709613)
- Fixed: Wallpapers and posters with adaptive clock lose depth if you upgrade from iOS 26 beta 1 to iOS 26 beta 2. (153005914)
- Fixed: Time might appear black in color. (154773881)

Known Issues

- If device is locked and an app is running on the Lock Screen, launching a camera app with Camera Control might not work. (151153912)

Workaround: Reboot the device.

- Lock Screen controls might be hard to see over lighter wallpaper content. (151324807)
- Clock, notifications, and other elements might not display. (152192129)

Workaround: Swipe down from the status bar to re-present the Lock Screen, or reboot the device.

- Devices might not auto sleep while the passcode UI is visible on the Lock Screen. (152604427)

Workaround: Dismiss passcode UI or manually lock the device.

Mail

Resolved Issues

- Fixed: While in Dark Mode, Mail controls may not adjust to content with a light background. (156225857)

Known Issues

- When opening the search field on iPadOS, a blank column will display instead of the message list if the user does not have recent searches or link/document suggestions. (149789435)

Workaround: Once the user starts typing in the search field or dismisses the search, the message list will re-appear.

- The Select button is missing in Search to perform bulk edit actions. (151313732)

Workaround: Users can still get into the same state by swiping down with two fingers to select then long-pressing to choose an action.

Maps

Known Issues

- Pinned places, including Home and Work, might be missing icons or display the incorrect icons. (147263981)

Workaround: Quit and relaunch Maps.

- Curated guides do not scroll on iOS, iPadOS, or visionOS when the iPhone is in landscape mode. (152123749)

Workaround: On iPhone, view curated guides in portrait mode.

- Vehicles with a touchpad interface might not be able to start navigation in CarPlay. (152272727)

Workaround: Navigation can be started from another available input device, such as iPhone.

- Custom tap gesture callbacks registered using `onTapGesture` might not work as expected on a SwiftUI Map. (157612948) (FB19394663)

Workaround: Use `simultaneousGesture(TapGestureRecognizer().onEnded { })` to register the callback.

Memory Tools

Known Issues

- Leaks might be falsely reported by memory analysis tools when a target has instances of types that use Obj-C properties implemented in Swift using bridged types with `@objc @implementation`. Memory analysis tools include the `leaks` CLI tool, Leaks instrument, and Xcode memory graph debugger. (157798911)

Menu Bar

Resolved Issues

- Fixed: On iPad, the gesture to invoke the Menu Bar does not work in right-to-left text layouts. (152447825)

Menu Customization

Known Issues

- Menu customization in apps like Notes and Mail might have some visual glitches. (148472167)

Messages

Resolved Issues

- Fixed: Sent translated messages do not get re-translated after editing. (149401758)
- Fixed: In regions where Screen Unknown Senders is on by default, notifications for message categories are erroneously off by default. (149450560)
- Fixed: In one-to-one conversations, background changes might be attributed incorrectly after quitting and re-opening the Messages app. (150548773)

- Fixed: Transaction or Promotion messages filtered by a Message Filter App Extension might be badged as "Unknown" in the conversation list rather than "Transaction" or "Promotion". (151869409)
- Fixed: Generative backgrounds do not work with Japanese keyboard. (155186919)

Known Issues

- Users on older devices won't see compatibility messages for polls, so they might be unaware a poll was sent. (148545742)
- Expanding Conversation Details causes the list of conversations to collapse. (149436051)

Workaround: Closing Conversation Details will bring back the list.

- Deleting and reporting as spam a message that was filtered by a Message Filter App extension does not send the spam report to the extension. (150832702)
- Devices with "Filter Unknown Senders" on before update might have "Time Sensitive" on by default under the Allow Notifications setting in the Unknown Senders section of Messages Settings. (150856051)

Workaround: Enable "Time Sensitive" manually.

- Messages fail to translate when only the recipient has translation enabled. (157779997)

Workaround: Both participants in the conversation must enable translation.

Metal

New Features

- Metal 4 is now supported. See [Metal](#) for additional details. (113781091)

Resolved Issues

- Fixed: If you're using Metal 4 command encoders, you should add render and compute pipelines that support indirect command buffers to your residency sets. The Metal device driver currently does not enforce this requirement. (145066238)
- Fixed: Metal Shader Validation might not work with shaders that use Metal Performance Primitives. (149263281)

Known Issues

- Metal Shader Validation might not work with Metal 4 ray tracing pipelines. (152520367)

Workaround: Selectively disable Shader Validation for pipelines using ray tracing. See [documentation](#).

MetalFX

Resolved Issues

- Fixed: Denoised temporal upscaling for MTL4CommandBuffer's MTL4FXTemporalDenoisedScaler does not work. (146436460)
- Fixed: Temporal upscaling for MTL4CommandBuffer's MTL4FXTemporalScaler does not work. (146436741)

MetricKit

Resolved Issues

- Fixed: MetricKit might fail to deliver MXDiskSpaceUsageMetrics on iOS 26 Beta. (151712405)

NetworkExtension

Deprecations

- Algorithms DES, 3DES, SHA1-96 and SHA1-160 as well as Diffie-Hellman groups less than 14 are no longer supported for IKEv2 VPNs. (148767790)

Notifications

Known Issues

- Notification icons do not automatically follow the system appearance when configured to do so. (152191515)

NSLog

New Features

- Dynamic string data in format arguments for NSLog will be redacted to `\<private\>` in the Unified Logging System. This specifically targets data that enters the Unified Logging System via NSLog, and will not impact the Xcode console or NSLog's `stdout` output. If you wish to log un-redacted data to the Unified Logging System please use the "os_log" or "Logger" interfaces. (137129180)

Object Capture

New Features

- A new algorithm significantly improves PhotogrammetrySession reconstruction quality of low-texture objects not captured with the ObjectCaptureSession front end. It will be downloaded and cached once in the background when the PhotogrammetrySession is used at runtime. If network isn't available at that time, the old low quality model will be used until the new one can be downloaded. There is no code change needed to get this improved model. (145220451)

Order Tracking

New Features

- Wallet now finds emails sent by merchants or delivery carriers and conveniently tracks order progress by pulling them together in one place. This feature is available with beta on Apple Intelligence enabled devices. (152090735)

Known Issues

- Users might experience a spinner issue when clicking the 'Track' button on the email banner. (152329353)

Workaround: Charge the device overnight and try again afterward.

Passcode

Resolved Issues

- Fixed: Users with alphanumeric passcodes might be presented a passcode entry screen that accepts a single digit, preventing correct entry. (156070293)

Phone

Known Issues

- A user on multiple calls on an iPhone without a Dynamic Island will see two Swap buttons, and the one next to the Contact's name will not be functional. (152258825)

Workaround: Use the functional Swap button between the Audio and Mute buttons.

Photos

Known Issues

- Existing Social Groups might disappear from the Photos app when manually creating a new one. (152248578)

Workaround: Quit and relaunch the Photos app.

Photos / Photos Picker

Known Issues

- Invoking search in the Photos picker causes the picker to crash. Or invoking Search in a collection in Photos causes Search to crash. (152403781)

Workaround: Use the Photos app to search for the content. Once you've found it, you can share or copy it to continue with the workflow. Use Search in the Library tab to find the content.

Push to Talk

Deprecations

- The legacy Push to Talk Entitlement `com.apple.developer.pushkit.unrestricted-voip.ptt` is no longer supported on apps built with the iOS 26 SDK or later. Push to Talk apps must be updated to use the Push to Talk framework introduced in iOS 16. (147700313)

Reality Composer

Resolved Issues

- Fixed: When you create or open a project in RealityComposer on devices with LiDAR, the screen freezes and shakes after entering AR mode and pressing the Play button. (152324341)

RealityKit

Resolved Issues

- Fixed: Entities with a `PortalComponent` ignore any `ModelSortGroupComponent` and instead use a fixed rendering order.

In case of sorting rendering issues with portal surface, explicitly add a `ModelSortGroupComponent` using a `.planarUIAlwaysBehind` sort group. (149899345)

- Fixed: `ParticleEmitterComponent` does not render properly on iOS, macOS, and tvOS. (152201501)
- Fixed: The `animate` functions on `Entity` and `RealityViewContent` do not start animations. (152456435)

Known Issues

- Some properties and components do not update SwiftUI Views when accessed through the `.observable` property on `Entity`. (147063698)

Search

Resolved Issues

- Fixed: Some applications might quit unexpectedly when using search. The issue will be resolved automatically with an over-the-air update when connected to Wi-Fi. (157464670)

Security

Deprecations

- For apps linked on or after iOS 26 and macOS 26, the default minimum TLS version of `NSURLSession` and `Network` frameworks has changed from 1.0 to 1.2. If your process connects only to your servers, verify that they support TLS 1.3 or TLS 1.2. If your process needs to connect to 3rd-party servers which cannot be updated to support newer versions of TLS, restore the previous behavior through `NSURLSession` and `Security` framework APIs: `NSURLSessionConfiguration.tlsMinimumSupportedProtocolVersion` and `sec_protocol_options_set_min_tls_protocol_version`. (135996267)

Setup Assistant

Known Issues

- The cursive Welcome text might be missing at the end of Setup Assistant. (152601626)

Workaround: Swipe up or tap Get Started to complete setup.

Shortcuts

Resolved Issues

- Fixed: Titles for some Messages actions and filter properties display incorrectly. (153740390)

Simulator

Resolved Issues

- Fixed: Siri does not accept voice input, making it unusable. (152738556)
- Fixed: Messages app crashes when launched. (153672262)
- Fixed: Preview app crashes whenever it is launched. (155906732)

Siri

Resolved Issues

- Fixed: Knowledge requests that are handled by ChatGPT might produce results that are based on stale data. (154889929)

Software Update

New Features

- Depending on the amount of free space available, iOS might dynamically reserve update space for Automatic Updates to download and install successfully. (137003545)

Resolved Issues

- Fixed: Software Updates might reserve more space than necessary. (151413019)

StoreKit

New Features

- There is a new option for the `Transaction.Offer.PaymentMode` API called `oneTime`. This new case supports the method of payment for In-App Purchase offer codes. (142501142)
- Subscription promotional offers can now be signed using JWS and attached to a purchase using the new `PurchaseOption.promotionalOffer(_:compactJWS:)` API. There are also new corresponding SwiftUI APIs in StoreKit to attach a signed promotional offer or a signed introductory offer override to a view. (143395736)
- `SubscriptionOfferView` is a new SwiftUI view to merchandise auto-renewable subscriptions. It is designed to capture customers' attention about the app's service. (145251635)

Resolved Issues

- Fixed: Subscription status updates might not be reported correctly if the subscription went into billing retry in StoreKit Testing in Xcode. (133799135) (FB14789854)
- Fixed: Price of offers is not displayed in the payment sheet when making a purchase to a subscription with a higher level of service in StoreKit Testing in Xcode. (140635780) (FB15980635)

- Fixed: Renewal transactions might be created regardless of the Ask to Buy status of the purchase request in StoreKit Testing in Xcode. (145242611)
- Resolved an issue with the `Identifiable` conformance of the `PurchaseIntent` API. Conformance to this protocol now begins starting with iOS 18.0 and macOS 15.0. (148751460) (FB17151889)
- Fixed: `SKProduct` from the original StoreKit API fails to decode products when using StoreKit Testing in Xcode. (150851879)
- Resolved an issue where the `id` member of the `PurchaseIntent` API was only available starting with iOS 18.0 and macOS 15.0, and no longer available for Mac Catalyst. It is now available starting with iOS 16.4, macOS 14.4, and Mac Catalyst 16.4. The `PurchaseIntent` conformance to `Identifiable` remains unchanged. (152858281) (FB17829716)
- Fixed: Transactions might not finish, resulting in subsequent purchases of the same product failing. (155449267)

Swift Charts

New Features

- Available in iOS 26, macOS 26, and visionOS 26, `Chart3D` allows you to visualize your data and mathematical surfaces in 3D, powered by RealityKit. (148361385)

Resolved Issues

- Fixed: Annotations on a scrollable chart might be clipped. (109164195)

Swift Compiler

Resolved Issues

- Fixed: The Swift compiler might crash when building a project that initializes a `UISymbolContentTransition`. (150858005)

Swift Standard Library

Resolved Issues

- Fixed: The span properties of `InlineArray` and `CollectionOfOne` trap at runtime. (147500528)
- Fixed: mutating members of `MutableSpan` and `MutableRawSpan` are unavailable. (152467655)

Known Issues

- `lengthOfBytes(using: .utf16)` and `-lengthOfBytesUsingEncoding: NSUTF16StringEncoding/NSUnicodeStringEncoding` might produce an incorrect result when used on Swift Strings, including `NSStrings` formed by bridging Swift Strings to Objective-C. (156675395)

SwiftUI

New Features

- `ControlSize` now conforms to `Comparable`, and `View/controlSize(_:)` can now be used to clamp the environment's `controlSize` to a given range. (99633360) (FB11465757)
- In apps built with the macOS 26 SDK, `Section` footers within a `Form` of the `GroupedForm` `Style` now have leading alignment, default font, and foreground styles. Use the `sectionActions(content:)` view modifier on your `Section` to supply section actions, which maintain a trailing placement in macOS. In iOS and iPadOS, each section action displays as its own form row. (129868475)
- `Text`, `TextEditor`, and `TextField` now by default use string contents to determine the appropriate base writing direction for each paragraph, instead of relying on layout directionality. To specify the writing direction explicitly on a per-paragraph basis, use Foundation's `AttributedString.writingDirection` attribute. To make the base writing direction follow the layout direction for an entire view, apply the view modifier `.writingDirection(strategy: .layoutBased)`. (134821288)
- In apps built with the macOS 26 and iOS 26 SDKs, a `Picker` view of a style that produces a button-like control now has a fitted sizing behavior by default. If needed, use the `buttonSizing(_:)` view modifier to make the `Picker` flexible and fill the available width of its container. (136649748)

- When linking news SDKs, `NavigationLinks` produce a single view, rather than a list of views in view list contexts. This change improves performance of many `NavigationLinks` in lazy containers like `List`. However, if you are relying on `ContainerValues` propagating out of the `label` view of a `NavLink`, or similarly relying on `ContainerValues` of a `ButtonStyle` used to style a link, the `containerValue(_:_:)` modifier should be moved outside of the link. Below is a minimal example that demonstrates the behavior difference:

```
import SwiftUI

struct ContentView: View {
    @State private var presentPopover = false
    var body: some View {
        NavLink("Custom Link", value: 84)
            .buttonStyle(MyButtonStyle(containerValue: "Eighty-four"))
    }
}

struct ParentView: View {
    var body: some View {
        Group(subviews: ContentView()) { subviews in
            ForEach(subviews) { subview in
                Text(subview.containerValues.myCustomValue)
            }
        }
        .frame(minWidth: 100, minHeight: 100)
    }
}

struct MyButtonStyle: PrimitiveButtonStyle {
    var containerValue: String
    func makeBody(configuration: Configuration) -> some View {
        Button(configuration)
            .buttonBorderShape(.circle)
            .containerValue(\.myCustomValue, containerValue)
    }
}
```

```

private struct MyContainerValueKey: ContainerValueKey {
    static let defaultValue: String = "Default value"
}

extension ContainerValues {
    var myCustomValue: String {
        get { self[MyContainerValueKey.self] }
        set { self[MyContainerValueKey.self] = newValue }
    }
}

#Preview {
    ParentView()
}

```

(140283584)

- List no longer ignores the vertical insets of rows with a height close to the default minimum height on iOS and visionOS. Use `listRowInsets(_:):` to change the vertical row insets. (141160852)
- In `NavigationSplitView` and `TabViews` configured as `sidebarAdaptable`, the view trailing the sidebar's safe area is inset in the width of the sidebar. It can display content outside its safe area, underneath the sidebar. (141222137)
- On macOS, after linking new SDKs, the style of search fields with `SearchFieldPlacement.sidebar` is now fixed to the toolbar. Previously the search field would scroll as the first element in the list. (143546967)
- In iOS and iPadOS apps that adopt the new design, a picker of `.segmented` style now supports `.large` and `.extraLarge` control sizes. (144417732)
- In apps that adopt the new design, the `buttonBorderShape(_:)` view modifier can be used to customize the shape of bordered buttons. Previously this modifier only affected buttons in `Widgets` in macOS. (145773436)
- The `buttonSizing(_:)` view modifier specifies the sizing behavior of `Button`, `Picker`, `Menu`, and other button-producing controls. If you are using `Spacer` views or an infinite-width frame in your `Button` label to create a flexible button, apply `buttonSizing(.flexible)` to the `Button` instead. (146327046)

- On iPadOS, sidebars and inspector titles now default to inline in the regular size class. You can use `navigationTitleDisplayMode(:_)` to specify a different title display mode. (150891824)
- On iPadOS, navigation link chevrons are not visible by default in the content column of `NavigationSplitViews` in the regular size class. You can use `navigationLinkIndicatorVisibility(:_)` to specify a different indicator visibility. (151646790)

Resolved Issues

- Fixed: `TextEditor`'s undo stack might contain invalid operations for applications with multiple `TextEditor` views. (83650197) (FB9662463)
- Fixed: The menu content of some `Menu` views does not update as the result of a state change. (106878937)
- Fixed: `View.onHover(perform:)` and `View.onContinuousHover(coordinateSpace:perform:)` affect hit-testing for gestures. Gestures are not received if one of these view modifiers is used within an overlay. (108560020)
- Fixed: Section views without `isExpanded` binding are collapsible in a sidebar-styled `List` on macOS. (115797465) (FB13192271)
- Fixed: `listSectionSpacing` does not work correctly for sections that have headers and footers. For more control over the list section layout, use the `listSectionMargins` modifier. (140929163)
- Fixed: On iOS, `navigationSplitViewColumnWidth()` does not respect the specified minimum and maximum column widths. (143529326)
- Fixed: Multiple title views in sidebar list labels on iOS are not styled hierarchically as title and subtitle. (144253754)
- Fixed: In custom layouts that do not implement `explicitAlignment(of:in:proposal:subviews:cache:)`, alignment guides do not work correctly for a right-to-left layout direction. (145073832)
- Fixed: Applying a bold modifier to `Text` resets the weight configuration. The interaction of bold and weight is not consistent between `Text` and `Font`. (147270079)
- Fixed: Environment updates from outside of a popover might fail to propagate into the popover's content view. (147954025)

- Fixed: Gestures added using the `simultaneousGesture(_:isEnabled:)` view modifier are incorrectly simultaneous with ancestor gestures. (147970990)
- Fixed: On platforms supporting edge-attached and non-edge-attached sheets, non-edge-attached sheets present erroneously as full-screen covers when `.navigationTransition(.zoom...)` is specified. If you wish to keep the full-screen cover in compact size classes, configure a sheet with:

```
.sheet(...) {
    ContentView()
        .navigationTransition(.zoom...)
        .presentationSizing(.page)
        .presentationCompactAdaptation(.fullScreen)
```

(150455117)

- Fixed: On iOS and iPadOS, sheets applied to toolbar items sometimes do not display if the sheet is presented at the same time a popover is dismissed. (150764801)
- Fixed: On iOS and iPadOS, toolbar items in the navigation bar may become non-centered after a state change. (152168945)
- Fixed: If Default Actor Isolation is set to `MainActor`, `@Animatable` macro emits concurrency warnings in Swift 5 language mode and does not compile in Swift 6 language mode. (152524435)
- Fixed: On iOS and iPadOS, `.fullScreenCover` backgrounds are transparent when they should be opaque. (154232311)
- Fixed: Gestures do not have the same lower priority over the view's existing `UIGestureRecognizer` and `NSGestureRecognizer` as they do over the view's existing SwiftUI gestures.

In apps built with iOS 26, macOS 26, tvOS 26, and visionOS 26, use `highPriorityGesture(_:isEnabled:)` to make sure your gesture takes precedence over the view's existing gestures, or use `simultaneousGesture(_:isEnabled:)` to give your gesture the same priority as the view's existing gestures. (155581361)

- Fixed: On iOS, iPadOS, and Mac Catalyst, popovers that access environment objects declared outside of a popover's content view might crash when `isPresented = true`. (156906038)

Known Issues

- On iOS and iPadOS, inspector does not respect the width applied with `inspectorColumnWidth`. (145162377)
- `toolbarForegroundColorStyle` no longer tints toolbar button labels on watchOS. (151487439)
Workaround: Tint the button label directly, using `Text("foo").foregroundColor(...)`.
- On iOS and iPadOS, bordered prominent buttons in toolbars do not have the correct default padding or symbol metrics. (151792861)
- `.toolbarVisibility(_:for:)` does not hide the navigation bar on watchOS. (152326250)

Deprecations

- Text concatenation using the `+` operator is deprecated because it makes it hard to create localized strings that are correct across all languages. Use `Text` interpolation instead. See documentation on `Text` for more info on how to produce localized strings. (128144043)
- On iOS and iPadOS, inspector columns do not present if the hierarchy does not contain a `NavigationSplitView`.

@Heading: SwiftUI

TextKit

New Features

- iOS 26, tvOS 26, visionOS 26, watchOS 26, and macOS Tahoe 26 have two methods for resolving the natural alignment `NSTextAlignment.natural` and the last line of `NSTextAlignment.justified` into concrete alignments, `left` and `right`. The first approach utilizes the UI language, which is determined by passing `nil-language` to `NSParagraphStyle.defaultWritingDirection(forLanguage:)`. This behavior was employed prior to this release. The second method is new and dynamically utilizes the base writing direction for the paragraph. When the base writing direction is set to `NSWritingDirection.rightToLeft`, the text is aligned to `right`, and vice versa. The behavior is selected by API introduced in this major release: `NSTextLayoutManager.resolvesNaturalAlignmentWithBaseWritingDirection`, `NSStringDrawingOptionsResolvesNaturalAlignmentWithBaseWritingDirection`, `UITraitCollection.resolvesNaturalAlignmentWith`

BaseWritingDirection, and `NSTextField.resolvesNaturalAlignmentWithBaseWritingDirection`. (152045248)

Resolved Issues

- Fixed: The directionality of `NSParagraphStyle` indentation properties (`firstLineHeadIndent`, `headIndent`, and `tailIndent`) is determined by inconsistent hidden rules implemented by the TextKit typesetting engine.

With iOS 26, macOS 26, tvOS 26, watchOS 26, and visionOS 26, the TextKit 2 typesetting engine has been standardized to utilize the resolved writing direction for the paragraph when linked with the OS 26 versions of the corresponding SDKs. For binary compatibility, applications developed with prior SDK versions will use the UI language in a subset of API interfaces. Specifically, for `UILabel`, `NSTextField`, `SwiftUI.Text`, and `NSStringDrawing`, the indentation directionality is determined by the UI language queried by passing `nil` to `NSParagraphStyle.defaultWritingDirection()` in applications built with prior SDK versions. (155893102)

TextKit 2

New Features

- A new property, `includesTextListMarkers`, is introduced to `NSTextList`, `NSTextContentStorage`, and `NSWritingToolsCoordinator`. This property controls whether to include the text list marker string in the contents of `NSAttributedString` for paragraphs associated with `NSTextList`. TextKit 1 expects the marker string, while TextKit 2 does not. The TextKit 2 behavior was adopted by UIKit starting with iOS 18 (`includesTextListMarkers=NO`). AppKit is also adopting the TextKit 2 text list behavior starting with macOS 26. (128479184)

Translation

Known Issues

- Downloading languages for translation might display incomplete indicators on the download screen resulting in translation features being unavailable. (148923222)

Workaround: Reboot the device.

UIKit

New Features

- In TextKit 2, the `includesTextListMarkers` property has been introduced to `NSTextList`, `NSTextContentStorage` and `UIWritingToolsCoordinator`. For paragraphs associated with `NSTextList`, the property controls whether to include the text list marker string in the `NSAttributedString` contents. The classes within TextKit 1 expect the marker string, while the classes within TextKit 2 do not. (144903432)

Known Issues

- In apps using `UIToolbar`, the Search bar sometimes does not respond to taps and displays without the magnifying glass, dictation button, or placeholder text. (151126350)

Workaround: Quit the app then re-launch it.

- On iPhone only, the `searchTextField` property of a `UISearchBar` belonging to a `UISearchController` (i.e. `searchController.searchBar.searchTextField`) might not return the same instance when called at different times. (153550157)

Workaround: Look for `UISearchBar` or `UISearchController` API equivalent to the `UISearchTextField` API you're using. If none can be found, you can prevent the issue by setting the `UINavigationController.searchBarPlacementAllowsToolbarIntegration` property to `false` on the navigation item the search controller has been assigned to.

- In rare cases after launching an app on iPhone, unusual client code timing of assembling the view controller hierarchy might cause the search bar belonging to a view controller in a tab to disappear from the navigation bar when first switching to that tab. (156174227)

Workaround: Set the `searchBarPlacementAllowsToolbarIntegration` property of the `UINavigationController` to `false` at the same time you set the `searchController`.

- In rare cases, the layout calculation from a client's layout constraint for their search results might change when presenting a `UISearchController` directly (for example, using `presentViewController:`) after building with the new SDK. This is due to removal of internal constraints with the identifier `UIView-Encapsulated-Layout-Width` and `UIView-Encapsulated-Layout-Height`. (157208725)

Workaround: Examine the existing constraints and determine how to modify them to be stable independent of the presence of the encapsulation constraints. Compare the results of `po [<view> _autoLayoutTrace]` with an older SDK and look for views marked with a bullet.

If they don't have a bullet on the new SDK, their encapsulation constraints will have been removed. This will get you started on adjusting your constraints setup.

Deprecations

- `UIScreen.mainScreen` has been changed from `API_TO_BE_DEPRECATED` to deprecated in iOS 26, tvOS 26, and visionOS 26. (142902390)

Wallpapers

Known Issues

- Widgets might be missing on the previews of photo wallpapers in the gallery. (151464159)
- After install or reboot, the default wallpaper might show as pink when the display dims or is on rotation. (152119543)
- Some wallpapers specific to certain devices might be missing. (152284310)

Watch App

Known Issues

- Apple Watch apps with a `MinimumOSVersion` of watchOS 26 that have an iOS counterpart fail to install to the watch via the Watch app on iPhone. (153195856)

Workaround: Use Xcode to install the app directly to the watch.

Watch Faces

Known Issues

- Complications configured on watch faces might appear 'Off' on the Watch app. (152324791)

Workaround: Use Apple Watch directly to edit complications.

Weather

Resolved Issues

- Fixed: Users might see a blank white button on some tips for Weather features. The button will be operable. (152088799)

Web Apps

Resolved Issues

- Fixed the “Add to Home Screen” flow failing to load webpage data, preventing users from making new Home Screen web apps. (154655565)

WebKit API

New Features

- The load APIs on WebPage now return an AsyncSequence directly that can be used to track relevant navigation events. The `currentNavigationEvent` property has been removed in favor of the `navigations` property, which produces an indefinite sequence directly. (152414525)
- The WebPage API now supports directly loading URLs. Additionally, when loading an HTML string, there is now a default value for the `baseURL` parameter. (152904248) (FB17850359)

Resolved Issues

- Fixed: `webViewOnScrollGeometryChange` might report an incorrect content size. (146576790)

Known Issues

- `WKPreferences.isLookToScrollEnabled` is not available on non-visionOS platforms. (152106377)

Wi-Fi Aware

Resolved Issues

- Fixed: `transmitLatency` property of `WAPerformanceReport` might not report latency values. (151628308)
- Fixed: Network connection to a paired device with an already-established connection might not succeed. (151873702)
- Fixed: Connections might be terminated unexpectedly after 120 seconds. (152279075)
- Fixed: Connections attempted while browser is not running might fail. (152336071)
- Fixed: The `wifiAware` property for connection-related `NWError` objects will be `nil`. (153100140)
- Fixed: New connections to an endpoint after multiple connect and disconnect operations might not succeed. (153689457)

Widgets

Known Issues

- A select few widgets such as Batteries, Contacts, Files might have an incorrect UI treatment. (145492734)