Documentation

Article

macOS Tahoe 26 Release Notes

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

Overview

The macOS 26 SDK provides support to develop apps for Mac computers running Tahoe 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)

AGL

Deprecations

- AGL is no longer available in the macOS SDK. AGL was previously used to present OpenGL content in Carbon apps, and Carbon no longer exists in the SDK. AGL symbols now do nothing on 64-bit systems, including Intel x86_64 and Apple Silicon Macs. It is safe to remove any AGL usage and stop linking AGL. OpenGL still remains in the SDK. (153913819)

AirPlay

Known Issues

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

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)

Resolved Issues

- Fixed: Updating iOS or iPadOS apps on macOS from the App Store might hang with a spinning progress indicator or with the progress partially complete. (152878930)

AppKit

Resolved Issues

- Fixed: With TextKit, the `textRangeByIntersectingWithTextRange` method in `NSTextRange` does not return nil for non-intersecting ranges. (138067979)

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)

Known Issues

- Xcode features like Predictive Code Completion and the coding assistant might require Apple Intelligence to be enabled. (150889516)

  Workaround: Enable Apple Intelligence in System Settings.

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`, `replyMail`, `forwardMail`, `message`, and `draft`. (148633307)

AuthenticationServices

New Features

- `ASAuthorizationControllerRequestOptions.preferImmediatelyAvailableCredentials` 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: An iOS or iPadOS app on an Apple silicon Mac might appear openable in TestFlight even while its essential assets are still being downloaded. (151709449)
- 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)

Catalyst

New Features

- For Catalyst apps built with the latest SDK, `UINavigationItem.title` is the window title rather than the window subtitle. (142423319)
- For Catalyst apps built with the latest SDK, `UIBarButtonSystemItem.fixedSpace` and `.flexibleSpace` are now bridged to `NSToolbar`. The system no longer ignores fixed and flexible spaces when `UIBarButtonItems` are automatically converted to the window toolbar in the Mac idiom. However, the width specified via `fixedSpaceItem(of Width:)` is ignored. (145262754)

CloudKit

Resolved Issues

- Fixed: CloudKit sharing URLs do not launch third-party apps. (151778655)

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)

Disk Images

New Features

- macOS now supports the Apple Sparse Image Format (ASIF). These space-efficient images can be created with the `diskutil image` command-line tool or the Disk Utility application and are suitable for various uses, including as a backing store for virtual machines storage via the Virtualization framework. See `VZDiskImageStorageDeviceAttachment`. (152040832)

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)

Finder

Resolved Issues

- Fixed: Finder does not display Dark Mode app icons or tinted folder colors when the Folder Color setting in System Settings > Appearance is set to Automatic. (152193702)

Known Issues

- Users who enable path bar or status bar in Finder and use list view might be unable to access the last item in the list. (151917092)

  Workaround: Disable path bar or status bar temporarily.

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: .content Tagging).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 `stream Response`, 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 `guardrailViolation` 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.

Full Screen

Resolved Issues

- Fixed: You might experience layout issues when going full screen with certain apps on a Mac. (151266898)

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 on macOS. (153125166)

Known Issues

- Game Mode will not activate for application binaries spawned directly from Terminal. (153127050)

  Workaround: Use the `open` command to launch your game from Terminal. You can pass arguments, change environment variables, and redirect standard output/error when using this command, such as `open MyGame.app --stdout /tmp/mygame.out --stderr /tmp/mygame.err --env MTL_HUD_ENABLED=1 --args -MyGameArgument -AnotherArgument`. See `man open` for more information.

Image Playground

Resolved Issues

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

Intel Macs

Resolved Issues

- Fixed: Safe and Recovery modes on Intel Macs have performance and graphical issues, though the modes are still functionally usable. Various elements on the screen, such as

menus, alerts, and Control Center, might partially display. Scrolling and other operations in some apps cause visual glitches. (149419875)

Keyboard

New Features

- In the “ABC – India” keyboard layout, the ₹ (rupee) symbol has replaced the backtick symbol. To type the backtick symbol that’s commonly used in Markdown and some programming contexts, you can use the ⌥ (option) modifier in conjunction with the same key or you can use an alternative layout like “ABC”. (149026394)

Maps

Resolved Issues

- Fixed: Users cannot plan a route that leaves at or arrives by a future time. (150947515)
- Fixed: If you tap to expand the “Recents” section and there are more recent places than can fit in the view, the Terms & Conditions link will disappear, making it inaccessible. (152197565)

Known Issues

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

  Workaround: Use `simultaneousGesture(TapGesture().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 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)

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.
- Scrolling through a Messages conversation might be sluggish. (152453655)

  Workaround: Quit and relaunch Messages.

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)

Music

Resolved Issues

- Fixed: You might experience audio issues when using the Music app and changing the underlying audio device sample rate. (152465491)

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

Resolved Issues

- Fixed: The icons for Calendar notifications might appear washed out. (151658533)

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)

NSTextView

New Features

- NSTextView supports sound files, such as QuickTime Audio, attached via NSTextAttachment, by utilizing AVPlayer for playback inline. (140224296)

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)

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.

Photos Picker

Resolved Issues

- Fixed: On macOS, the Options menu is missing in the Photos picker. Ability to exclude Location and Captions information cannot be set when using the Photos picker. (152336867)

Quick Look

Resolved Issues

- Fixed: Variants menu does not open in Quick Look while trying to view different configurations of a USDZ asset made with Variants. (154817946)

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)

Rosetta

New Features

- You can test that your apps no longer depend on Rosetta by setting the boot-arg `nox86exec=1`. When this boot-arg is set, any process that would normally run through Rosetta will immediately crash on launch and generate a crash report. This can by set by

running `sudo nvram boot-args="nox86exec=1"` and then rebooting the system. (136764433)

RTL

Known Issues

- You might experience layout issues with RTL languages. (151009428)

Safari

Resolved Issues

- Fixed: macOS 26 users might experience Safari page loading issues on facebook.com and potentially other websites, due to a new race in the network stack.

  Affected users should update to macOS 26 beta 5 and clear related cache by visiting Safari > Settings... > Privacy > Manage Website Data..., then deleting entries for facebook.

  - and fbcdn.net resources. Restart Safari. Users will need to re-login to the website. Users experiencing issues on other websites can consider clearing cache for those websites or clearing all website data. (153547547)

Known Issues

- Black lines appear at the bottom of inactive tabs in Safari. (153681371)

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)

Secure Text Fields

Resolved Issues

- Fixed: On a Mac with a Touch Bar, a secure text field swallows keyboard events in some contexts. For example, this might occur when creating a new user in the Users & Groups

settings pane. (151268030)

Security

Deprecations

- For apps linked on or after iOS 26 and macOS 26, the default minimum TLS version of `URLSession` 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 `URLSession` and Security framework APIs: `URLSessionConfiguration.tlsMinimumSupportedProtocolVersion` and `sec_protocol_options_set_min_tls_protocol_version`. (135996267)

Settings

Known Issues

- Users who enable Reduce Transparency might encounter flickering in background windows or the Dock. (152060485)
- If System Settings > Desktop & Dock > Displays have separate Spaces is disabled, WindowServer will crash at login time. (153570422)

  Workaround: Boot into Recovery, then mount the Data volume on your partition. Launch Terminal and run `rm /Volumes/<Partition Name "Macintosh HD">/Users/<user name>/Library/Preferences/com.apple.Spaces.plist`.

Setup Assistant

Resolved Issues

- Fixed: 802.1X Wi-Fi networks are not saved during initial device setup. (147787689)
- Fixed: On Intel Macs, the background on the Hello or Welcome screens might be black. (152107967)
- Fixed: Visual pairing for Quickly Set Up Mac does not work. (152326903)
- Fixed: On the 13″ MacBook Air, Hello might be offset from the wallpaper during initial device setup. (152447100)

Shortcuts

Resolved Issues

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

Siri

Resolved Issues

- Fixed: Siri Visual Responses might be illegible behind certain backgrounds. (151682699)
- Fixed: Knowledge requests that are handled by ChatGPT might produce results that are based on stale data. (154889929)

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)

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

- On macOS, an animated SwiftUI.Transaction that changes a `Window`'s size animates the window's frame, alongside the frame of the hosting view. (61158194)
- You can now use `View/findNavigator(isPresented:)`, `View/find Disabled(_:)`, and `View/replaceDisabled(_:)` to control the presentation of the Find Bar in `TextEditor` on macOS 26. (85308161)
- `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 `Grouped FormStyle` 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 `button Sizing(_:)` view modifier to make the `Picker` flexible and fill the available width of its container. (136649748)
- The default label style for macOS menu content is now `.titleAndIcon`. (137306701)
- The implementation of some macOS buttons no longer uses `NSButton`. (139105246)

- 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 `NavigationLink`, 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 {
        NavigationLink("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)
- In macOS, a `Form` of the `.grouped` style now has a more compact appearance when placed within a sidebar or inspector. (141534926)
- Reuse existing AppKit gesture recognizers in SwiftUI using `NSGestureRecognizerRepresentable`, and refer to them by name using `name`. (142918018)
- 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 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)
- In macOS apps that adopt the new design, buttons of the `.bordered` style can be tinted with the `tint(_:)` view modifier. (150127133)

Resolved Issues

- Fixed: `TextEditor`'s undo stack might contain invalid operations for applications with multiple `TextEditor` views. (83650197) (FB9662463)
- Fixed: In macOS, `Button` inherits its environment's font instead of using a default font appropriate for its `controlSize`. Apply view modifiers to the button's label to customize its font. (92434021)
- Fixed: In macOS, the value label of `MenuPickerStyle` does not truncate when it is long enough to cause the button to be wider than its container. (93843377)
- 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: 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: 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: Gestures do not have the same lower priority over the view's existing `UIGestureRecognizers` and `NSGestureRecognizers` 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)

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)

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`, `NSStringDrawingOptions.ResolvesNaturalAlignmentWithBaseWritingDirection`, `UITraitCollection.resolvesNaturalAlignmentWithBaseWritingDirection`, 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

- For Catalyst apps using `.translationTask()`, buttons in the download approval UI might not work. (151313115)

  Workaround: Either manually download the languages you need beforehand from System Settings, or build your Catalyst app with the “Optimize for Mac” setting.

UIKit

New Features

- In TextKit 2, the `includesTextListMarkers` property has been introduced to `NSTextList`, `NSTextContentStorage` and `NSWritingToolsCoordinator`. 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. (144903293)

VideoToolbox

Resolved Issues

- Fixed: You might experience decoding performance issues for high-bitrate HEVC, resulting in stuttering and lag. (153243806)

Virtual Machines

Resolved Issues

- Fixed: Virtual machine networking fails if you start a bridge mode VM while a shared or host mode VM is running, and vice versa. The networking of the existing VM is not affected. (151477625)

Weather

Resolved Issues

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

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)

Xcode

Resolved Issues

- Fixed: Xcode Previews usage frequently panics on macOS 26.0 Tahoe beta 4. (141641869)

See Also

macOS 26

- macOS Tahoe 26.3 Beta 3 Release Notes
  Update your apps to use new features, and test your apps against API changes.
- macOS Tahoe 26.2 Release Notes
  Update your apps to use new features, and test your apps against API changes.
- macOS Tahoe 26.1 Release Notes
  Update your apps to use new features, and test your apps against API changes.