Documentation Language: Swift

Updates / Foundation updates

Article

Foundation updates

Learn about important changes to Foundation.

Overview

Browse notable changes in Foundation.

June 2025

- Post and observe concurrency-safe notifications with new features in `NotificationCenter` that support Swift-friendly “message” types. Use `NotificationCenter.MainActorMessage` for notification messages bound to the `MainActor` and `NotificationCenter.AsyncMessage` for messages that are `Sendable`. Add an observer to one of these notification message types with the various `addObserver(of:for:using:)` methods in `NotificationCenter`. Many standard notifications in Foundation, UIKit, and AppKit now offer a message-based API, associating the message with an existing `Notification.Name`. You can also associate a message with a new name for Swift-only notifications.

- Simplify usage of `UndoManager` on the main actor, now that the undo manager is marked with `@MainActor`. The undo manager also adds a `setActionName(_:)` method that takes a `LocalizedStringResource`. Use this with the App Intents framework, which introduces a new `UndoableIntent`.

- Access your app, app extension, or framework’s bundle with the `#bundle` macro. Using the macro is more performant and convenient than using `.self` or a bundle identifier, particularly when loading localized strings with initializers that take a `bundle:` parameter. The macro back-deploys, so you can use it with projects whose deployment targets specify earlier versions of the operating system.

- Get UTF-8 and UTF-16 views of an `AttributedString` with the `utf8` and `utf16` methods of `AttributedStringProtocol`.

- Access multiple ranges of an `AttributedString` with the new `DiscontiguousAttributedSubstring` type. You can get this type from an attributed string with either a `RangeSet` of indices, or with the new `AttributedTextSelection` type.

June 2024

Predicates

- Use `Regex` regular expressions in predicates to perform pattern matching. You can use either the Swift regex domain specific language or traditional regex literals.
- Use the new `#Expression` macro when you want behavior similar to `#Predicate` but you need to return a type other than `Bool`.
- Create compound predicates by writing a `Predicate` that evaluates another `Predicate`.

Calendars

- Perform `Calendar` searches by calling `dates(byMatching:startingAt:in:matchingPolicy:repeatedTimePolicy:direction:)` with `DateComponent` values specifying what to search for, and receiving a `Sequence` of matching dates. You can also repeatedly add `DateComponent` values to a date with `dates(byAdding:startingAt:in:wrappingComponents:)` and receive a sequence. To add `Calendar.Component` values instead, use `dates(byAdding:value:startingAt:in:wrappingComponents:)`.

Format styles

- Use the `DiscreteFormatStyle` protocol to format values that constantly change, but don’t necessarily change the formatted output on every update, such as time displays that truncate the seconds field. The following format style types conform to the new protocol: `Duration.UnitsFormatStyle`, `Duration.TimeFormatStyle`, `Date.FormatStyle`, `Date.FormatStyle.Attributed`, `Date.VerbatimFormatStyle`, `Date.VerbatimFormatStyle.Attributed`, `Date.ISO8601FormatStyle`, `Duration.UnitsFormatStyle.Attributed`, and `Duration.TimeFormatStyle.Attributed`.
- Set new configuration options on `FormatStyle` to omit specific symbols from the output and suppress grouping of large time values (for example, in order to produce `10000:00` rather than `10,000:00`).
- Use the `notation` modifier on currency format styles to specify a notation such as `compactName`, similar to how `notation` already works on numeric format styles.

Undo manager

- Provide custom information for undo actions by setting user info on `UndoManager`. The info can include things like a timestamp of the action's creation or an icon that complements the action name.
- Reveal the size of an undo manager's current undo or redo stack with the properties `undoCount` and `redoCount`.

Key-Value observing

- Add a collection of shared observers to `Observable` objects that you can then add to multiple instance of the observable.
- Observables now use ARC-style weak references to their observers, preventing crashes when observers deallocate without properly removing themselves.

Collections

- You can initialize an `IndexSet` from a Swift `RangeSet`, and vice versa.

See Also

Technology updates

Accelerate updates

Learn about important changes to Accelerate.

Accessibility updates

Learn about important changes to Accessibility.

ActivityKit updates

Learn about important changes in ActivityKit.

AdAttributionKit Updates

Learn about important changes to AdAttributionKit.

App Clips updates

Learn about important changes in App Clips.

App Intents updates

Learn about important changes in App Intents.

AppKit updates

Learn about important changes to AppKit.

Apple Intelligence updates

Learn about important changes to Apple Intelligence.

AppleMapsServerAPI Updates

Learn about important changes to AppleMapsServerAPI.

Apple Pencil updates

Learn about important changes to Apple Pencil.

ARKit updates

Learn about important changes to ARKit.

Audio Toolbox updates

Learn about important changes to Audio Toolbox.

AuthenticationServices updates

Learn about important changes to AuthenticationServices.

AVFAudio updates

Learn about important changes to AVFAudio.

AVFoundation updates

Learn about important changes to AVFoundation.