WWDC 2024

Legend *(of interest)*

- ☗ Mega Evil Corp
- ✺ Mini Evil Corp
- ◉ hot-n-GUI, LLC
- ❖ LV iOS Meetup

GENERAL Thoughts and Comments

1. I see a big push to reintroduce AppIntents to developers - they've been ignored by many - for the purpose of making the new Siri+AI more powerful
2. Lots of Core ML related videos, most seem like a rehash with some new features - again probably to improve Siri+AI functionality
3. Am confused by what the difference is between Swift.org and Swift Programming Language (https://github.com/apple/swift) ☗
4. TVML deprecated as of tvOS 18 - replaced by SwiftUI features

Bring context to today's weather

- New binary data transfer, 90% smaller than JSON - FlatBuffers ◉ ❖

Bring your app to Siri

- Introduces Siri Domains (photos, mail, system) which one can utilize for their own content
- Takes AppIntents to the next level, providing more context
- Right now, can't really use these new in-app actions in Siri but can use Shortcuts to test them. Siri coming later
- SiriKit vs App Intents - still unclear to me the difference or dividing line

Bring your app's core features to uses with App Intents

- Rehash of existing features
- Showed some new features to make it more useful across more mechanisms (shortcuts, Siri, etc.

Bring your machine learning and AI models to Apple Silicon

- Introduces new techniques / features for compressing models so they more easily fit on device

Elevate your tab and sidebar experience in iPadOS

- Tab bar on the bottom is replaced with new floating, more compact tab bar at the top of the screen 🔔

Meet Swift Testing ◉ ❖

- Supports all Swift platforms including Linux and Windows
- Tests can be marked as @MainActor 🔔
- Easier to organize tests using Tags
- Supports being executed on real devices such as phones and watches
- Supports VS Code for those folks who prefer that over Xcode
- It's open source

Migrate your app to Swift 6

- Swift 6 mode is opt-in for existing and new code
- Should use complete concurrency checking first in Swift 5 mode, then migrate to Swift 6 language mode

What's new in Swift 6

- Fully static Linux SDK for Swift allows cross-compilation from MacOS to any Linux server 🔔
- Swift Testing ◉ ❖
- Explicitly built modules - improves debugger start-up time, more reliability of builds, opt-in flag in Xcode Settings
- Mutex *(import Synchronization, let cache = Mutex<[String: Resource]>([:]), cache.withLock { })*

What's new in SwiftData ◉ ❖

- New "compound constraints" called #unique causes updates instead of duplication of entries 🔔
- Supports custom data stores
- Can now access data history
- New support for better "Preview(traits: ...)" using SwiftData
- New "@Previewable" macro to generate a query inside the #Preview 🔔
- New #Expression macro to create complex queries 🔔
- New "#Index" macro adds metadata to models to increase query performance 🔔

What's new in SwiftUI ❖

- iPadOS tab bars moved to top of screen and float
- Sheet sizing - `.presentationSizing(.form)` - for more automatic sheet

sizing
- New navigationLink feature to support cooler transitions - `.matchedTransitionSource(id:,in:)` and `.navigationTransition(.zoom(sourceID:,in:)`
- New Charts features - function plotting, vectorized plots
- New MeshGradient for custom color combos
- New symbol effects - wiggle, breathe, rotate ◉
- Pencil support for “squeeze” and “double tap” actions - available in iPadOS 17.5
- Custom Containers
- `@Entry` macro to simply access to things like environment values

What’s new in UIKit

- iPadOS tab bars moved to top of screen and float 🔔
- New customization feature for side-bar and tab bars
- New zoom transitions
- UIKit has access to SwiftUI animations
- Automatic Trait tracking - no longer need to “registerForTraitChanges([])” as just access a trait will trigger updates later when that trait changes
- New UIUpdateLink (similar to CADisplayLink) for better performance of some UI animations
- New symbol effects - wiggle, breathe, rotate
- UITextView automatically gets support for the new “Writing Tools” such as proofreading, changing the voice, etc.
- Pencil support for “squeeze” and “double tap” actions - available in iPadOS 17.5

What’s new in Xcode 16

- Improved code completion
- Swift 6 language mode - opt-in
- Preview build improvements - explicit build modules 🔔
- New Universal Backtrace (next to view hierarchy debugger
- Swift testing
- Instruments - new “blame” graph shows what calls are taking the most time
- Predictive Code Completion
- Swift Assist - can generate code and such based on your request

Catch up on accessibility in SwiftUI

- `.accessibilityLabel` now has an `.isEnabled` parameter that can be used to conditionally override the default label

- Can now add accessibility actions that are “intents” to widgets

Create a custom data store with SwiftData

- New protocols: DataStoreConfiguration, DataStoreSnapshot, DataStore
- You can handle filtering/sorting or let the ModelContext handle that for you - less efficient, but much easier to code

Enhance you UI animations and transitions

- New zoom transition- what we’ve all done in the past, tap on a cell and it animates next view/vc from the bounds of that cell instead of a typical slide
- Can now use SwiftUI builtin animations in UIKit (UIView.animate)

Evolve your document launch experience

- New features for document based apps on iPadOS mainly

Extend your app’s controls across the system

- “ControlWidget” - yet another kind of widget
- “ControlValueProvider” — async way to get control state remotely
- “.promptsForUserConfiguration” — immediately shows widget configuration when it makes sense for them to make an initial choice when creating a widget (control only?)

Squeeze the most out of Apple Pencil ◉

- Can now specify which tools are available on the Picker tool
- Pencil Pro has barrel roll angle support

What’s new in App Intents ◉

- New support for Apple Intelligence and Control Widgets
- IndexedEntity - new way to make app info available to Spotlight
- CSSearchableItem.associateAppEntity() -> adds feature to existing searchable items created such as PDP donations ✺
- Transferable, TransferRepresentation -> convert entity to PDF, PNG, RTF for easy use by Siri, etc.
- URLRepresentableEntity, URLRepresentableIntent - provides way to export your entities, as universal links, to things like Siri and Shortcuts
- Can now define AppIntents in a framework and use them in the main app and extensions

Build multilingual-ready apps ❖

- Lots of new enhancements to better improve user experience when using

non-English for input and output
- Personalize text that addresses the user
- New languages added to the grammar engine

Demystify explicitly built modules

- Xcode 16 changes how modules are built
- Three phases of compilation: Scan, Build Modules, Build Source
- Speeds up the Debugger since it now uses the same built modules
- Explicit Modules needs to be enabled for Swift Modules in project settings

Demystify SwiftUI Containers

- Lots of new API's to make custom containers easier to define
- Existing containers are things like List, Form, etc...

Meet the Translation API

- Easy to use Translation presentation for SwiftUI
- Flexible translation API - for more custom presentations within your app
- Supports same language as the Translate app
- Must use real devices for development/testing - not supported on simulators - what about unit testing?
- Batch requests for translation should be in the same source language

Run, Break, and Inspect: Explore effective debugging in LLDB

- Debugging in Xcode/LLDB 101, good primer for novices
- Use @DebugDescription macro instead of CustomDebugStringConvertible protocol

Support semantic search with Core Spotlight

- New API to make donations to spotlight more searchable
- Shows how to use your spotlight donations to search in your own app

Unlock the power of places with MapKit

- Place ID Lookup
- New Token creation for MapKitJS
- Place Card API - can be used to select any map feature and display an accessory (callout)
- Additional new search filters - lots of new ways to search for cities, for rivers, etc...

What's new in location authorization

- CLLocationUpdate.liveUpdates() new last year
- CLMonitor new last year
- NEW CLServiceSession(authorization:) to streamline authorization process - declarative API instead of the old procedural one

What's new in privacy

- New pickers - app does not have access to the values held within
- Image Playground - only provides final image back to the app
- AccessorySetupKit - provides easier to use approach to pairing and accessing accessing devices without having to give app access to all your Bluetooth devices
- MAC addresses will now change for different networks; will rotate to new values on same network about~2 weeks
- Contact Access Button - provides easier access to create contacts
- Can "lock" or "hide" any app - starting requires FaceID, etc.
- Passkeys can be creating automatically when user signs in with other means

Analyze heap memory 🔔

- Excellent tutorial on how to debug a bunch of types of memory issues

Add personality to your app through UX writing

- App's Voice
- App's Tone
- Takes the viewer through a couple of exercises related to how to find your app's voice and tone ❖

Capture HDR content with ScreenCaptureKit

- Not so into the streaming aspect, but worth looking into the framework to find how to do good screenshots ❖

Consume non copyable types in Swift

- Data types that cannot be copied, hence not modifiable
- Everything is copyable by default
- Add a "~" to make non copyable - i.e. *struct FloppyDisk: ~Copyable {}*
- New keyword "consume" (which is normally implied) -> moves contents of the object to the new variable leaving original one uninitialized so cannot be accessed anymore
- Passing non-copyable objects as parameters requires ownership to be specified - "consuming" or "borrowing" or "inout"

- Can mark methods as “consuming”

Create custom visual effects with SwiftUI

- Scroll effects: scrollTransitions modifier, visualEffect modifier ❖
- Color treatments: mesh gradients (+animation) ❖
- View transitions: custom transitions ❖
- Text transitions: new TextRenderer
- Metal shaders: ShaderLibrary.ColorMultiply?,

Explore Swift performance

- Low level tutorial on how Swift handles allocation of the different kinds of memory (global, stack, heap)
- Showed how async functions are handled as a special case with their own stacks
- Showed how escaping vs non-escaping closures have to use heap vs stack memory
- Showed how using Generics “<Model: DataModel>” is much better and performant than this “models: [any DataModel]”
- Uses C representations throughout to show how the Swift would be represented in C
- One take away: heavy use of protocol types has a performance cost
- Not for novices

Get started with Dynamic Type 🔔 ✺ ❖

- Shows the basics of how to use Dynamic Type
- Use system styles and fonts
- Shows how to use *@Environment(\.dynamicTypeSize)* to modify layout based on text size (while not using viewThatFits)
- Adjusting images/symbols as appropriate- don’t always scale if they are purely decorative
- `@ScaledMetric` can be used to scale a regular image to match text size
- Large Content Viewer - can be used to temporarily expand size of items, such as tab item

Meet the Contact Access Button ❖

- Contact authorization come in different flavors now: Generic alert leads to “Select Contacts” and “Allow Full Access”
- NEW Contact Access Picker can be embedded in your app’s screen
- Contact Access Button - limited access only
- If this new button is tapped before permission has been given then a simplified permission alert is displayed to the user

- Button display is customizable to some extent to match your app's look

Track model changes with SwiftData history ◉ ❖

- Tracks changes to the data store, not the schema
- Uses tokens as markers
- History can be deleted
- You can create "tombstone attributes" to preserve model details when deleted

Using CloudKit Console to monitor and optimize database activity ◉

- New notifications that alert the admin to issues related to selected databases
- Can have web based (default) and/or email based notifications