


[Swift](#) /  / [Concurrency](#) / Updating an app to use strict concurrency

Sample Code

Updating an app to use strict concurrency

Use this code to follow along with a guide to migrating your code to take advantage of the full concurrency protection that the Swift 6 language mode provides.

Download

iOS 15.0+ | iPadOS 15.0+ | watchOS 8.0+ | Xcode 16.0+

Overview

Note

This sample code project is associated with WWDC24 session 10169: [Migrate your app to Swift 6](#). It's based on a similar session from WWDC21 showing how to adopt Swift concurrency in an app, [Swift concurrency: Update a sample app](#).

This sample provides two separate versions of the app:

- The original version uses Swift concurrency features but contains a number of issues that are detected by enabling Swift complete concurrency checking and that need to be resolved before enabling the Swift 6 language mode.
- The updated version resolves these issues and has enabled Swift 6. It also adds new features that record the location of the user when they log that they drank coffee.

Watch the session to see the process step by step, and then compare the two projects to see the differences.

Configure the sample code project

To add the complication to an active watch face, start by building and running the sample code project in the simulator, and follow these steps:

1. Click the Digital Crown to exit the app and return to the watch face.
2. Using the trackpad, firmly press the watch face to put the face in edit mode, and then click Customize.
3. Swipe left until the configuration screen highlights the complications. Select the complication to modify.
4. Scroll to the complication for the sample app, and then click the Digital Crown again to save your changes.
5. Click the newly added complication to go to the app.

For more information on setting up watch faces, see [Change the watch face on your Apple Watch](#).

After configuring and running the app, you can test the background updates. Make sure that the complication appears on the active watch face. Then build and run the app in Simulator, and follow these steps:

1. Add one or more drinks using the app's main view.
2. Click the Digital Crown to send the app to the background.
3. Open Settings, and scroll down to Health > Health Data > Nutrition > Caffeine to see all of the drinks you added to the app.
4. Click Delete Caffeine Data to clear all of the caffeine samples from HealthKit.
5. Navigate back to the watch face.
6. The complication will update within 15 minutes; however, the update may be delayed based on the system's current state.

Adopt strict concurrency checking in Swift

To see the issues highlighted in the original project, follow these steps:

1. In Xcode, navigate to the project settings.
 2. For each target, change the Strict Concurrency Checking below Build Settings to Complete.
 3. Each target generates warnings that must be resolved.
 4. After the warnings are resolved, change the Swift Language Version in Build settings to Swift 6.
- Compare the original project to the updated project to see how it resolves the concurrency issues.

See Also

Essentials

{ } Code-along: Elevating an app with Swift concurrency

Code along with the WWDC presenter to elevate a SwiftUI app with Swift concurrency.

{ } Updating an App to Use Swift Concurrency

Improve your app's performance by refactoring your code to take advantage of asynchronous functions in Swift.