

Apple releases iOS 17.4.1 for iPhone users, more

Apple has officially released iOS 17.4.1 for iPhone users today. The update comes two weeks after Apple released iOS 17.4 with major changes in the European Union.

You can update your iPhone to iOS 17.4.1 by heading to the Settings app, choosing General, then choosing Software Update. The build number for today's update is 21E236. It's available for any iPhone capable of running iOS 17, which includes the iPhone XR/XS and newer.

Apple's release notes for iOS 17.4.1 simply say: "This update provides important bug fixes and security updates and is recommended for all users."

iOS 17.4 was released on March 5 with a number of new features, including alternative app store support in the European Union. The update also included transcripts in Apple Podcasts, improvements to Stolen Device Protection and Siri, new emoji, and more.

In addition to iOS 17.4.1, Apple has released the following updates (all of which contain important bug fixes and security updates):

- visionOS 1.1.1
- iPadOS 17.4.1

Find anything new in iOS 17.4.1? Let us know down in the comments.

Bundling resources with a Swift package

Overview

If you declare a Swift tools version of 5.3 or later in your package manifest, you can bundle resources with your source code as Swift packages. For example, Swift packages can contain asset catalogs, storyboards, and so on.

Add resource files

Similar to source code, Xcode scopes resources to a target. Place resource files into the folder that corresponds with the target they belong to. For example, any resources for the MyLibrary target need to reside in Sources/MyLibrary. However, consider using a subfolder for resources to distinguish them from source files. For example, put all resource files into a directory named Resources, resulting in all of your resource files residing at Sources/MyLibrary/Resources.

To add resources to a Swift package, do any of the following:

- Drag them into the Project navigator in Xcode.
- From the File menu in Xcode, choose Add Files to [packageName].
- Use Finder or the Terminal app.

When you add a resource to your Swift package, Xcode detects common resource types for Apple platforms and treats them as a resource automatically. For example, you don't need to make changes to your package manifest for the following resources:

- Interface Builder files; for example, XIB files and storyboards
- Core Data files; for example, xcdatamodeld files
- Asset catalogs
- .lproj folders you use to provide localized resources
- If you add a resource file that Xcode doesn't treat as a resource by default, you must configure it in your package manifest, as described in the next section.

Explicitly declare or exclude resources

To add a resource that Xcode can't handle automatically, explicitly declare it as a resource in your package manifest. The following example assumes that `text.txt` resides in `Sources/MyLibrary` and you want to include it in the `MyLibrary` target. To explicitly declare it as a package resource, you pass its file name to the target's initializer in your package manifest:

```
targets: [  
    .target(  
        name: "MyLibrary",  
        resources: [  
            .process("text.txt")  
        ],  
    ),  
]
```