Sample Code

Configuring your app to use alternate app icons

Add alternate app icons to your app, and let people choose which icon to display.

Download

iOS 18.0+ | iPadOS 18.0+ | Xcode 26.0+

Overview

This sample code project demonstrates how to configure your app so that people can change the icon that appears on the Home screen, in Spotlight, and elsewhere in the system. People select an alternate icon in the app interface from a collection that you provide.

Add icon files for the alternate icons

For each alternate app icon, the project requires an Icon Composer file. Create your app icon using Icon Composer, then add it to the Project navigator in your Xcode project. For more information, see Creating your app icon using Icon Composer. For design guidance, see App icons.

Configure the asset catalog compiler

The system gathers information about the app's icons from the app's `Info.plist` file under the top-level key `CFBundleIcons`. Xcode adds entries to this file for the icons the project specifies through build settings under Asset Catalog Compiler - Options.

For each icon file that the project specifies by name in the build setting Alternate App Icon Sets, Xcode adds an entry under the key `CFBundleAlternateIcons`.

Xcode enters the name of the primary app icon specified in the build setting Primary App Icon Set Name under the key `CFBundlePrimaryIcon`. This setting is also available through the App Icons and Launch Screen section of the General pane. For more information about build settings, see Build settings reference.

> **Important**
>
> To change the values of `CFBundleIcons`, `CFBundleAlternateIcons`, and `CFBundlePrimaryIcon`, modify their related build settings. Don't edit or remove these keys manually from the `Info.plist` file.

Alternatively, you can use a build configuration file to specify alternate app icons. Add a build configuration file to your Xcode project, then add the `ASSETCATALOG_COMPILER_ALTERNATE_APPICON_NAMES` build setting to the configuration file. Set `ASSETCATALOG_COMPILER_ALTERNATE_APPICON_NAMES` to a list of strings matching the alternate app icons' names.

```
ASSETCATALOG_COMPILER_ALTERNATE_APPICON_NAMES = AppIcon-Green AppIcon-Blue AppIcon-C
```

For more information about build configuration files, see Adding a build configuration file to your project.

Change the app's icon

When people select an alternate icon in the app interface, the app calls `setAlternateIconName(_:completionHandler:)` with the name of the new icon. This tells the system to display the new icon for this app. The system automatically displays an alert notifying people of the change. Passing `nil` displays the app's primary icon.

```
UIApplication.shared.setAlternateIconName(iconName) { error in
    if let error {
        self.logger.error("Failed request to update the app's icon: \(error)")
    }
}
```

The current icon's name is available through the property `alternateIconName`.

See Also

App icons and launch screen

- Creating your app icon using Icon Composer

  Use Icon Composer to stylize your app icon for different platforms and appearances.

- Configuring your app icon using an asset catalog

  Add app icon variations to an asset catalog that represents your app in places such as the App Store, the Home Screen, Settings, and search results.

- Specifying your app's launch screen

  Make your iOS app launch experience faster and more responsive by customizing a launch screen.