

[Updates](#) / Default apps updates

Article

Default apps updates

Learn about the latest changes to enabling your app to be the system default.

Overview

Build apps and extensions that people can configure as the default app for many common tasks. Browse [entitlements](#) that enable your apps to declare that they can be set as the system default.

- Use [openSettingsURLString](#) to link directly to your app's settings, including the Default App option, where applicable. The new [openDefaultApplicationsSettingsURLString](#) option in `UIKit` opens the global Default Apps settings panel.

June 2025

Dialing apps

- New API in [LiveCommunicationKit](#) lets users choose your app as the default for initiating cellular carrier conversations. Read [Preparing your app to be the default dialer app](#) to see how to set this up using the [Default Dialer App](#) entitlement. Your app has access to conversation history that happened since it became the default, and no longer requires user confirmation to initiate a connection.

SMS, RCS, and MMS messaging apps

- The new [TelephonyMessagingKit](#) framework enables your app send SMS, RCS, and MMS messages over the cellular carrier network. Use the [Default Carrier Messaging App](#) entitlement to declare your app as the default handler for these carrier messages.

2025

Navigation apps

- Read [Preparing your app to be the default navigation app](#) to see how to register your app to be the default responder to navigation requests for users in the European Union. This article explains how to use the `geo-navigation://` URL scheme to handle navigation queries the user initiates from other apps.

Translation apps

- Learn about [Preparing your app to be the default translation app](#) so that your translation app can respond to user requests to perform text translation.

2024

Alternative app marketplaces

Alternative app marketplace apps for iOS or iPadOS enable users to install other third-party apps in the European Union. Developers can distribute their marketplace app on the web, and users can then select the alternative marketplace as their default, if desired. Apple provides the [MarketplaceKit](#) framework that facilitates the secure installation of apps that your marketplace distributes. Read [Creating an alternative app marketplace](#) to learn how to build your own marketplace.

- Learn about the [com.apple.developer.marketplace.app-installation](#) entitlement used by alternative app marketplaces.
- Read [Distributing your app from your website](#) to learn how to ship the alternative app marketplace.

Calling apps

In iOS and iPadOS 18.2 and later, a user may select an app other than the Phone or FaceTime apps to place calls. If your app places phone calls, for instance using services such as Voice over IP (VoIP), and you wish to optionally become the default calling app, see [Preparing your app to be the default calling app](#).

Contactless NFC and SE platform apps

iOS 18.1 introduced APIs that support secure contactless transactions within compatible iOS apps using the NFC & SE Platform for in-store payments, car keys, closed-loop transit, corporate badges, student IDs, home keys, hotel keys, merchant loyalty and rewards, and event tickets, with government IDs to be available at a later date.

The NFC & SE Platform is a secure solution developed by Apple that enables authorized developers to provide capabilities, such as securely adding, storing, and presenting a contactless card for NFC use cases, from within their iOS app. Supported NFC and SE platform apps can be selected by users as their default handler for these transactions.

- To learn more about the NFC and SE Platform see <https://developer.apple.com/support/nfc-se-platform/>.
- Manage and employ Secure Element credentials with contactless transaction capabilities using [`SecureElementCredential`](#)
- Enable your app to be the default app for contactless NFC and SE Platform transactions with the [`com.apple.developer.secure-element-credential.default-contactless-app`](#) entitlement.

HCE-based contactless transactions for apps

iOS 17.4 introduced APIs that support contactless transactions for in-store payments, car keys, tickets, and more uses from within compatible iOS apps using host card emulation (HCE) in the European Economic Area (EEA).

The [`CardSession`](#) API in the CoreNFC framework enables authorized developers to perform contactless transactions from within their app. Supported `CardSession` apps can be selected by users as their default handler for these transactions.

- Learn about host card emulation apps by reading <https://developer.apple.com/support/hce-transactions-in-apps/>.
- Use [`CardSession`](#) to enable host card emulation (HCE) transactions in your app.
- Enable your app to be the default app for HCE-based contactless NFC with the [`com.apple.developer.nfc.hce.default-contactless-app`](#) entitlement.

Messaging apps

In iOS and iPadOS 18.2 and later, a user may select an app other than the Messages app to send instant messages. The system launches the default messaging app to handle when a user taps an `im:` link from another app. [Preparing your app to be the default messaging app](#) describes how to enable your app to optionally be selected as the default.

2023 and earlier

Call Directory app extensions

Build a Call Directory app extension so a user's device can automatically use your app to look up incoming callers, present useful caller ID information, or block unwanted callers. Read [Identifying and blocking calls](#) for more information on creating these app extensions.

- You can specify that your Call Directory app extension adds identification and blocks phone numbers in its implementation of `beginRequest(with:)`.
- To block incoming calls for a specific phone number, use the `addBlockingEntry(withNextSequentialPhoneNumber:)` method in the implementation of `beginRequest(with:)`.

Keyboard apps

Use custom keyboard apps and extensions to replace the system keyboard for users that want different text-entry capabilities, such as a novel input method. People can choose to have this custom keyboard available systemwide, and select the default keyboard used in text fields. Read [Creating a custom keyboard](#) for information on how to build and configure your custom keyboard app and extension project.

- For more information on handling expected system behaviors in your custom keyboard, see [Configuring a custom keyboard interface](#)

Mail apps

The system launches the default mail client whenever a user opens a `mailto:` link. Signal your app's intent to be available as a default mail client by using the `com.apple.developer.mail-client` entitlement.

Password, credential, and verification code apps

Password managers, verification code providers, and other secure credential apps can include a Password AutoFill app extension to enable their app to automatically fill in a name and password within Safari and other apps.

- Register the `otpauth://` or `otpauth-migration://` URL scheme within your app to enable setup of verification codes.
- Use Xcode to add a new extension target of type `AutoFill Credential Provider` to your app's project to enable AutoFill for secure credentials throughout the system.

- For your app and app extension, use [AutoFill Credential Provider Entitlement](#) to ask someone for permission to fill in the credentials.


Web browser apps


Users can select an app to be their default web browser. To make your app available as the default browser app, confirm that your app meets the requirements below, then request a managed entitlement. See [Preparing your app to be the default web browser](#) to learn more.


- In iOS 18.2 and iPadOS 18.2, the [isDefault\(_ :\)](#) API allows a browser app to check if it is currently the default browser app. To reduce the likelihood that users will face continuous requests to set a browser as their default, this API will only tell the browser app if it is the default once per year.
 - See [Importing data exported from Safari](#) to see how your browser app can import data the user exported from Safari.
 - Use the [com.apple.developer.web-browser](#) entitlement to enable your app to be the default web browser.
-


See Also


Technology updates


 [Accelerate updates](#)
Learn about important changes to Accelerate.

 [Accessibility updates](#)
Learn about important changes to Accessibility.


 [UIKit updates](#)
Learn about important changes in UIKit.


 [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.