

Technology

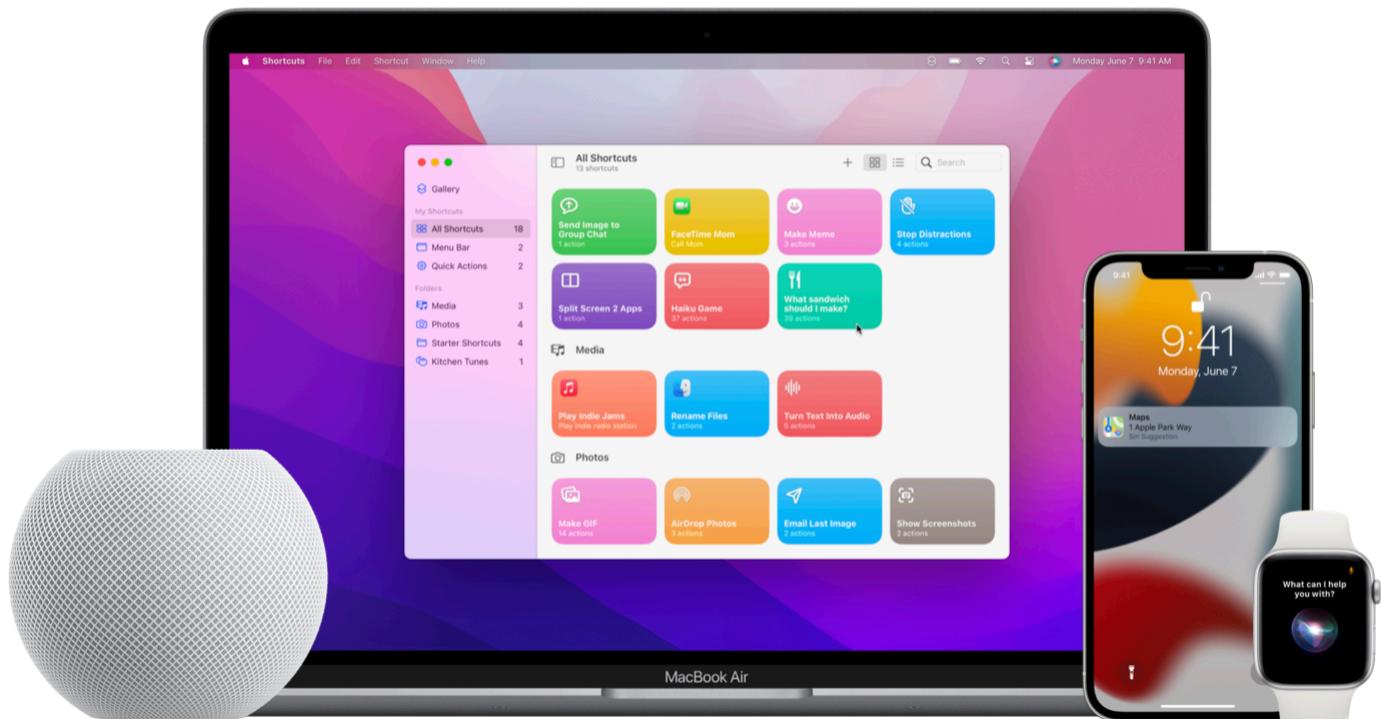
SiriKit

Empower users to interact with their devices through voice, intelligent suggestions, and personalized workflows.

iOS 10.0+ | iPadOS 10.0+ | Mac Catalyst 13.0+ | macOS 12.0+ | tvOS 14.0+ | visionOS 1.0+ | watchOS 3.2+

Overview

The Intents and IntentsUI frameworks drive interactions that start with “Hey Siri...”, Shortcuts actions, and widget configuration. The system also incorporates intents and user activities your app donates into contextual suggestions in Maps, Calendar, Watch complications, widgets, and search results.



Use the standard intents that the system provides to empower actions users already ask Siri to do, such as playing music or sending a text message. You can also offer your app's unique capabilities

throughout the system by designing custom intents. For more details about defining custom intents, see [Adding User Interactivity with Siri Shortcuts and the Shortcuts App](#).

You can process intents directly in your app, or in an Intents app extension. For guidance on setting up an app extension and sharing information between your app and extension, see [Structuring Your Code to Support App Extensions](#).

To display branding or other customized content in Siri and Maps after you fulfill a user request, create a custom view controller in an IntentsUI app extension. See [Creating an Intents UI Extension](#) for more details.

Important

With a person's permission, an installed health research app that uses [SensorKit](#) entitlements may collect Face Metrics data while your SiriKit app is in use. To prevent SensorKit from collecting Face Metrics data while your app is in use, you can set the [SRResearchDataGeneration](#) information property list key to NO.

Topics

Frameworks

Reference the API that compose SiriKit

Intents

Empower people to customize interactions for your app on their device.

IntentsUI

Customize content in the interface for Siri and Maps.

Sample code

Browse sample code that walks through specific Intents and IntentsUI workflows.

{ } Adding Shortcuts for Wind Down

Reveal your app's shortcuts inside the Health app.

{ } Booking Rides with SiriKit

Add Intents extensions to your app to handle requests to book rides using Siri and Maps.

{ } Handling Payment Requests with SiriKit

Add an Intent Extension to your app to handle money transfer requests with Siri.

{ } Handling Workout Requests with SiriKit

Add an Intent Extension to your app that handles requests to control workouts with Siri.

{ } Integrating Your App with Siri Event Suggestions

Donate reservations and provide quick access to event details throughout the system.

{ } Managing Audio with SiriKit

Control audio playback and handle requests to add media using SiriKit Media Intents.

{ } Providing Hands-Free App Control with Intents

Resolve, confirm, and handle intents without an extension.

{ } Soup Chef: Accelerating App Interactions with Shortcuts

Make it easy for people to use Siri with your app by providing shortcuts to your app's actions.

{ } Soup Chef with App Intents: Migrating custom intents

Integrating App Intents to provide your app's actions to Siri and Shortcuts.

Articles

Browse articles that cover high-level Intents and IntentsUI tasks.

 Adding User Interactivity with Siri Shortcuts and the Shortcuts App

Add custom intents and parameters to help users interact more quickly and effectively with Siri and the Shortcuts app.

 Defining Relevant Shortcuts for the Siri Watch Face

Inform Siri when your app's shortcuts may be useful to the user.

 Deleting Donated Shortcuts

Remove your donations from Siri.

 Dispatching intents to handlers

Provide SiriKit with an intent handler capable of handling a specific intent.

 Improving Siri Media Interactions and App Selection

Fine-tune voice controls and improve Siri Suggestions by sharing app capabilities, customized names, and listening habits with the system.

 Improving interactions between Siri and your messaging app

Donate app-specific content, use Siri's contact suggestions, and adopt the latest platform features to create a more consistent messaging experience.

☰ Registering Custom Vocabulary with SiriKit

Register your app's custom terminology, and provide sample phrases for how to use your app with Siri.

📄 Confirming the Details of an Intent

Perform final validation of the intent parameters and verify that your services are ready to fulfill the intent.

📄 Handling an Intent

Fulfill the intent and provide feedback to SiriKit about what you did.

📄 Resolving the Parameters of an Intent

Validate the parameters of an intent and make sure that you have the information you need to continue.

📄 Generating a List of Ride Options

Generate ride options for Maps to display to the user.

☰ Handling the Ride-Booking Intents

Support the different intent-handling sequences for booking rides with Shortcuts or Maps.

📄 Donating Reservations

Inform Siri of reservations made from your app.

📄 Specifying Synonyms for Your App Name

Provide alternative names for your app that are more familiar or easier for users to speak.

📄 Intent Phrases

The keys that you include in your global vocabulary file to show how users engage your app from Siri.

📄 Localizing Your Vocabulary for Chinese Dialects

Apply emphasis markers to your pronunciation tips to assist Siri with Chinese dialects.

📄 Parameter Vocabularies

The keys you include in your global vocabulary file to describe app-specific terms.

📄 Offering Actions in the Shortcuts App

Suggest shortcuts users may want to add to Siri or combine with other actions in their own shortcuts.

Creating an Intents App Extension

Add and configure an Intents app extension in your Xcode project.

Requesting Authorization to Use Siri

Request permission from the user for Siri and Maps to communicate with your app or Intents app extension.

Structuring Your Code to Support App Extensions

Move your back-end services to a private framework so your app and app extensions can use them.

Providing Live Status Updates

Provide regular updates to Maps about the status of a booked ride.

Donating Shortcuts

Tell Siri about shortcuts to actions that the user performed in your app.

Configuring the View Controller for Your Custom Interface

Configure your view controller to replace or augment the default interface in Siri or Maps.

Configuring Your Intents UI App Extension Target

Configure your Xcode project to include an Intents UI app extension that you use to customize the Siri and Maps interfaces.