API Collection

Controls

Offer controls that people place in Control Center, on the Lock Screen, and on the Action button to quickly perform an action from your app.

Overview

Controls act as a button that initiates an action from your app or opens your app to a specific view, or as a toggle. A toggle can turn a light on and off, or open and close a garage door. People place controls in prominent locations to personalize how they use their devices:

- On devices with an Action button, people can place a controls action in the Action button.
- On iPhone and iPad, people place controls in Control Center and on the Lock Screen.
- On Mac, people place controls from your macOS app in Control Center or as menu bar items.
- On Apple Watch, people place controls from your watchOS app or from a paired iPhone in Control Center or the Smart Stack. If you offer watchOS app in addition to an iPhone app and both apps offer controls, people can only place controls from your watchOS app in Control Center or the Smart Stack.

To offer a control for your app, use WidgetKit API to configure and update the control, create its layout with SwiftUI, and perform its action using App Intents. For more information about the App Intents framework, refer to Making actions and content discoverable and widely available.

Topics

Essentials

- Developing a WidgetKit strategy
  Explore features, tasks, related frameworks, and constraints as you make a plan to implement widgets, controls, watch complications, and Live Activities.
- Creating a widget extension
  Display your app's content in a convenient, informative widget on various devices.
- {} Emoji Rangers: Supporting Live Activities, interactivity, and animations
  Offer Live Activities, controls, animate data updates, and add interactivity to widgets.

Setup and configuration

- Creating controls to perform actions across the system
  Perform your app's actions from Control Center, the Lock Screen, and the Action button.
- Adding refinements and configuration to controls
  Customize the way controls display across the system and offer people the ability to configure them.

`struct StaticControlConfiguration`
The description of a control that has no user-configurable options.

`struct AppIntentControlConfiguration`
The description of a control that uses a custom app intent to provide user-configurable options.

`class ControlCenter`
An object you use to access configuration information for controls and reload them.

`struct ControlInfo`
A structure that contains information about user-configured controls.

`struct ControlWidgetButton`
A control template representing a button.

`struct ControlWidgetToggle`
A control template representing a toggle.

Updates

- Updating controls locally and remotely

Update and reload controls from your app or using push notifications.

`protocol` ControlPushHandler

A type that can receive push information about user-configured controls.

`struct` ControlPushInfo

A structure that contains information about the push token of a user-configured control.

Previews

`protocol` ControlValueProvider

A type that provides a value to a control template.

`protocol` AppIntentControlValueProvider

A type that uses a custom intent to provide a value to a control template.

See Also

System experiences

Widgets and watch complications

Allow people to personalize their devices, view relevant information, and perform interactions with widgets and watch complications.

Live Activities

Let people track updates from your app with Live Activities.