

[WidgetKit](#) / [Widgets and watch complications](#) / Accessing location information in widgets

Article

Accessing location information in widgets

Incorporate location information into your widget presentation to make it more relevant and contextual.

Overview

Widgets can present relevant and useful information by taking a user's location into account. Because your widget's extension doesn't run continually, using Location Services in your widget requires you to take a few additional steps.

Enable Location Services in your widget

To use Location Services in your widget:

- Add the `NSWidgetWantsLocation` key to your widget extension's `Info.plist` file.
- Add the relevant purpose strings to the `Info.plist` file of the app that contains the widget.

Important

Before your widget can receive location information, your app must request a person's permission to access their location. For most widgets, requesting When In Use authorization is the right choice, because of its better privacy and battery life implications. For more information, see [Requesting authorization to use location services](#).

Access location information from your widget

When the user adds a widget that uses location, the system asks whether they want to extend the app's location authorization to the widget. Use `isAuthorizedForWidgetUpdates` to

determine whether the widget is eligible to receive location updates. Users can change their approval choice at any time in Settings > Privacy > Location Services.

After the user approves the widget's use of location information, the system considers it "in use" for a short period of time after the widget is visible. If the system refreshes the widget's view during this in-use period, the widget has access to location information. For example, immediately after a user adds a widget to their Home Screen and approves extending the app's location authorization, the user's location is available to the widget.

When the widget hasn't been visible for a period of time, the system no longer considers it in use and stops delivering location updates. When `WidgetKit` reloads the widget's view, if `authorizedForWidgetUpdates` is `true` but the widget doesn't get location updates, it's a good practice to indicate that the user's location is not currently available. Note, this is different from when `authorizedForWidgetUpdates` is `false`, which indicates that the user has not approved the widget to receive location updates.

Test your widget in real-world scenarios

Due to the timeline-based updates of widgets, and the variability of a widget's in-use status, it's important to test widgets that use location in real-world scenarios. For example, create test scenarios that:

- Extend the app's location authorization when adding the widget.
- Don't extend the app's location authorization when adding the widget.
- Change the app's authorization in Settings > Privacy > Location Services after the widget is added.
- Change the widget's approval of location authorization in Settings > Privacy > Location Services after the widget is added.
- Add the widget to Home Screen pages that are both frequently and infrequently viewed.

Because widgets receive a limited number of refreshes every day, test your widget over multiple days.

Isolate location usage with multiple widget extensions

If your app provides multiple widgets and only some of the widgets use location, separate your widgets into multiple extensions. Add the `NSWidgetWantsLocation` key to the extension that contains widgets that use location. This allows the system to only prompt the user for the widgets that use location information, and makes it more contextually relevant for users.

For more information, see the Declare Multiple Widgets in Your App Extension section in [Creating a widget extension](#).

See Also

Capabilities

 Making network requests in a widget extension

Update your widget with new information you fetch with a network request.