Requesting authorization to use location services | Apple Developer Documentation

By Make authorization requests and respond to status changes

Archived: 2026-04-05 17:20:50 UTC

Overview

Location data is sensitive information, and the use of location data has privacy implications for the people who use your app. To ensure that people maintain control over their own information, the system prevents apps from using location data until they obtain authorization to do so. This authorization process involves a one-time interruption, during which the system prompts the device owner to grant or deny your app's request for location data. After the initial interruption, the system stores your app's authorization status and doesn't prompt again.

To help people understand why you need location data, make authorization requests only when someone engages a part of your app that requires that data. Making the request immediately before it's needed increases the likelihood of the person granting the request. If you make a request immediately at app launch, or in a part of your app that doesn't clearly use location data, the person might misinterpret your intent and deny the request.

Choose the access level you need

Before you place an authorization request, choose the level of access your app needs. Core Location supports two authorization levels:

- **When in Use** authorization makes location updates available only when someone uses your app. This authorization is the preferred choice, because it has better privacy and battery life implications.
- **Always** authorization makes location updates available at any time, and lets the system launch your app quietly to handle some updates. Request this access level only when necessary on other platforms. For example, request it if your app delivers time-sensitive responses to location changes automatically, or implements a location push service app extension. This access level isn't available when running in visionOS.

The definition of when an app is in use depends on the platform:

- On iOS, an app is in use when it's in the foreground and for a short time when it transitions from the foreground to the background. If you enable background location updates, an app with When in Use authorization continues to run in the background when location services are active; if location services aren't running, the normal suspension rules apply. If the system terminates the app or the app isn't running, the system doesn't launch an app with When in Use authorization to deliver new updates; it does launch an app with Always authorization for some types of location updates.

- On macOS, When in Use and Always authorizations are functionally equivalent. Because macOS apps continue to run in the background after their initial launch, they are always in use. If you create your Mac app using Mac Catalyst, request authorization based on the needs of your iOS app.
- On watchOS, complications can receive location updates, but the watchOS app must run at least once so it can request authorization to access location data. If an app's complication is on the current watch face, the system treats that complication as if it's in use and delivers location updates to it. The system doesn't launch watchOS apps, even if they have Always access.
- On visionOS, an app is in use when someone is looking at it, and for a short time after the person stops looking at it.

Regardless of which access level you choose, you can start any location services available on the current device and achieve the same results. Access levels primarily determine how your app receives updates when it isn't running. The following table summarizes the differences between access levels.

Capability	**When in Use**	**Always**
Supported platforms	All	All platforms except tvOS and visionOS
Supported location services	All	All
Launches a terminated app automatically	No. The user must launch the app.	Yes for significant location change, visits, and region monitoring services; no for others

For information about how to handle location updates in the background, see Handling location updates in the background.

Provide descriptions of how you use location services

The first time you make an authorization request, the system displays an alert asking the person to grant or deny the request. The alert includes a usage description string that explains why you want access to location data. You provide this string in your app's Information Property List and use it to inform people about how your app uses location data.

Core Location supports different usage strings for each access level. You must include a usage description string for When in Use access. If your app supports Always access, provide an additional string explaining why you want the elevated privileges. The following table lists the keys to include in your app's Information Property List and when to include them.

Add all usage description keys to your app's Information Property List before you make any authorization requests. Authorization requests fail immediately if the required keys aren't present.

Before you start any location services, check your app's current authorization status and place an authorization request if needed. You can get your app's current authorization from the `authorizationStatus` property of your location-manager object. However, a newly configured `CLLocationManager` object also reports your app's current authorization status to its delegate's `locationManagerDidChangeAuthorization(_:)` method automatically. You might use that method to place an authorization request when the current status is `CLAuthorizationStatus.notDetermined`. In the following example, the delegate method enables or disables location features when the status is known and requests authorization when the status is undetermined.

```
func locationManagerDidChangeAuthorization(_ manager: CLLocationManager) {
   switch manager.authorizationStatus {
   case .authorizedWhenInUse:  // Location services are available.
       enableLocationFeatures()
       break

   case .restricted, .denied:  // Location services currently unavailable.
       disableLocationFeatures()
       break

   case .notDetermined:        // Authorization not determined yet.
      manager.requestWhenInUseAuthorization()
       break

   default:
       break
   }
}
```

The `locationManagerDidChangeAuthorization(_:)` method offers a central place to process any authorization-related changes. People can change your app's authorization status at any time in system settings. If your app is running when the change happens, each of your app's `CLLocationManager` objects reports the change to that delegate method. The location manager also reports your app's current authorization at other times. For example, the location manager also calls the method when a suspended iOS app starts running again.
