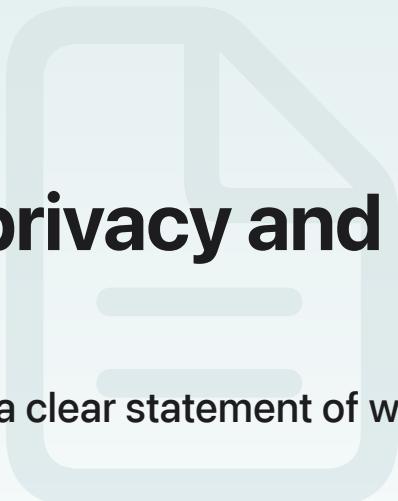


[visionOS](#) / Adopting best practices for privacy and user preferences

Article

Adopting best practices for privacy and user preferences

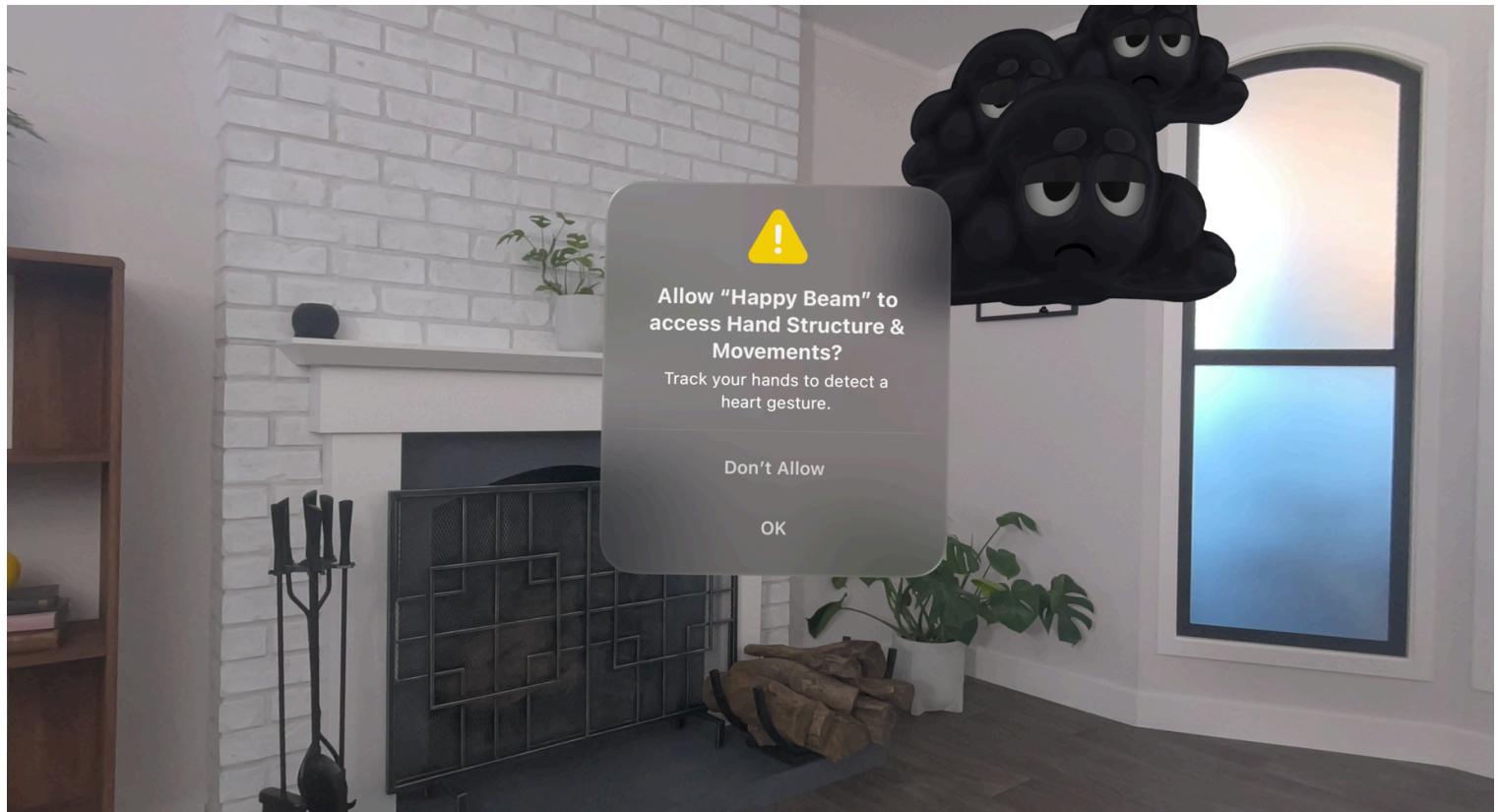
Minimize your use of sensitive information and provide a clear statement of what information you do use and how you use it.



Overview

To protect user privacy, the system handles camera and sensor inputs without passing the information to apps directly. Instead, the system enables your app to seamlessly interact with a user's surroundings and to automatically receive input from the user. For example, the system handles the eye- and hand-position data needed to detect interactions with your app's content. Similarly, the system provides a way to automatically alter a view's appearance when someone looks at it, without your app ever knowing what the user is looking at.

In the few cases where you actually need access to hand position or information about the user's surroundings, the system requires you to obtain authorization from the user first.



Important

It's your responsibility to protect any data your app collects, and to use it in responsible and privacy-preserving ways. Don't ask for data that you don't need, be transparent about how you use the data you acquire, and respect the choices of the person whose data it is.

For information about how to specify the privacy data your app uses, see [Describing data use in privacy manifests](#). For general information about privacy, see [Protecting the User's Privacy](#).

Adopt the system-provided input mechanisms

On Apple Vision Pro, people use their eyes and hands to interact with the items they see in front of them. Where they look determines where the system applies focus, and a tap gesture with either hand generates a touch event on that focused item. The system can also detect when someone's fingers interact with virtual items in the person's field of vision. When you adopt the standard UIKit and SwiftUI event-handling mechanisms, you get all of these interactions automatically.

For most apps, the system-provided gesture recognizers are sufficient for responding to interactions. Although you can get the position of someone's hands with ARKit, doing so isn't necessary for most apps. Collect hand-position data only when the system doesn't offer what you need. For example, you might use hand-position data to attach 3D content to the person's hands. Some other things to remember about hand-position data:

- People can deny your request for access to hand-position data. Be prepared to handle situations where the data isn't available.

- You must present an immersive space to access hand data. When you open an immersive space, the system hides other apps.

For information about how to handle the standard-system events, see the [SwiftUI](#) and [UIKit](#) documentation.

Provide clear messaging around privacy-sensitive features

The following ARKit features require you to provide a usage description string in your app's `Info.plist` file:

- World-tracking data
- Hand-tracking data

Other privacy-sensitive technologies in visionOS also require you to supply usage description strings. For example, you provide usage descriptions for the Core Location features you adopt. These strings communicate why your app needs the data, and how you plan to use the data to help the person using your app. The first time you request authorization to use the technology, the system prompts the person to grant or deny access to your app. The system includes your usage-description string in the dialog it displays.

For information about requesting access to ARKit data, see [ARKit](#). For guidance on how to craft good messages around privacy-friendly features, see [Human Interface Guidelines](#).

See Also

Design

Designing for visionOS

When people wear Apple Vision Pro, they enter an infinite 3D space where they can engage with your app or game while staying connected to their surroundings.

Improving accessibility support in your visionOS app

Update your code to ensure everyone can access your app's content in visionOS.