

[Apple silicon](#) / Providing touch gesture equivalents using Touch Alternatives

Sample Code

Providing touch gesture equivalents using Touch Alternatives

Enable Touch Alternatives to provide keyboard, mouse, and trackpad equivalents to your iOS app when it runs on a Mac with Apple silicon.

Download

iOS 15.5+ | iPadOS 15.5+ | Xcode 13.4+

Overview

Touch Alternatives is a feature that provides keyboard, mouse, and trackpad equivalents of certain touch gestures to iOS apps running on a Mac with Apple silicon. An app can enable Touch Alternatives by default, and a person can enable and disable the feature from the app's Settings window (prior to macOS 13, the Settings window is known as the Preferences window).

This sample app displays a circle that you can move around using touch gestures. You can use the app to explore Touch Alternatives on your Mac with Apple silicon.

Configure the sample code project

To run the sample app:

1. Download the sample and open the project in Xcode.
2. Select a development team in the Signing & Capabilities tab of the target settings.
3. Select My Mac (Designed for iPad) as the destination.
4. Build and run the app.

Important

The My Mac (Designed for iPad) destination option is available only on a Mac with Apple silicon.

When you launch the sample app for the first time, it enables Touch Alternatives. You can press the arrow keys to move the circle around the window. Press Space to return the circle to the center of the window. Use a mouse or trackpad to drag the circle around the window. You can also disable and enable the alternatives from the Touch Alternatives tab in the app's Settings window.

Enable Touch Alternatives

In macOS 11.3 and later, iOS apps that run on a Mac with Apple silicon can enable Touch Alternatives by default. To enable Touch Alternatives as the default for the sample app, the sample code project includes the property list file `com.apple.uikit.inputalternatives.plist` in the app bundle, and sets the `defaultEnablement` key in the file to `true`.

The file can contain the following keys:

- `defaultEnablement`. A Boolean value that indicates whether to enable Touch Alternatives by default when the app launches for the first time.
- `requiredOnboarding`. An array of strings that indicate which alternatives to highlight in the Touch Alternatives tab of the app's Settings window.

The `requiredOnboarding` array can contain any of the following values:

- `Tap`. Press Space to tap the center of the app's window.
- `Arrow Swipe`. Press the arrow keys to swipe from the center of the app's window toward the direction of the arrow.
- `Scroll Drag`. Scroll with a mouse or trackpad to drag from the pointer.
- `Trackpad Capture`. Press and hold Option to use a trackpad as a virtual touch screen.
- `Tilt`. Press the W, A, S, and D keys to simulate tilting the device.

The values in the array indicate which alternatives to highlight in the Touch Alternatives tab of the app's Settings window. The array should include only those values that make sense for the app. For example, the sample app doesn't detect the device's motion, so it doesn't include `Tilt` in the array.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
```

```
<dict>
  <key>defaultEnablement</key>
  <true/>
  <key>requiredOnboarding</key>
  <array>
    <string>Tap</string>
    <string>Arrow Swipe</string>
    <string>Scroll Drag</string>
    <string>Trackpad Capture</string>
  </array>
</dict>
</plist>
```

Starting with Xcode 14, developers can add the property list file `com.apple.uikit.inputalternatives.plist` to a project using the Touch Alternatives template, available in the Resource section of the iOS file templates. With earlier versions of Xcode, developers can use the Property List template.

See Also

iOS apps on Mac



Running your iOS apps in macOS

Modernize the iOS apps you choose to run on a Mac with Apple silicon, or opt out of running on a Mac altogether.



Adapting iOS code to run in the macOS environment

Support modern iOS features that result in a better user experience when running on Apple silicon.



Providing an edge-to-edge, full-screen experience in your iPad app running on a Mac

Take advantage of the true native resolution of a Mac display when running your iPad app in full-screen mode on a Mac.