Documentation

Xcode / Devices and Simulator / Running your app in Simulator or on a device

Article

Running your app in Simulator or on a device

Launch your app in a simulated iOS, iPadOS, tvOS, visionOS, or watchOS device, or on a device connected to a Mac.

Overview

To test your app, build and run it on a simulated or real device. Use simulated devices to debug your app on a variety of hardware to which you don’t have immediate access. The tradeoff is that simulated devices run within the Simulator app on your Mac and don’t replicate the performance or features of an actual device. To verify that your app runs exactly as intended, run it on one or more real devices. You can connect a real device to your Mac using a cable, or connect it over Wi-Fi after you pair it with Xcode.

Xcode previews let you see your app’s interface without building and running your app. For more information, see Previewing your app’s interface in Xcode.

Select a build scheme and run destination

Before you build and run your app, select a build scheme that includes the target for your app. A *scheme* is a collection of project details and settings that tell Xcode how to build and run a product from your project. Xcode determines where the resulting product can run based on the scheme you select, and populates the run destination menu in the toolbar with the list of available devices. For example, if the scheme contains a tvOS app, Xcode includes only tvOS simulators and devices as potential run destinations.

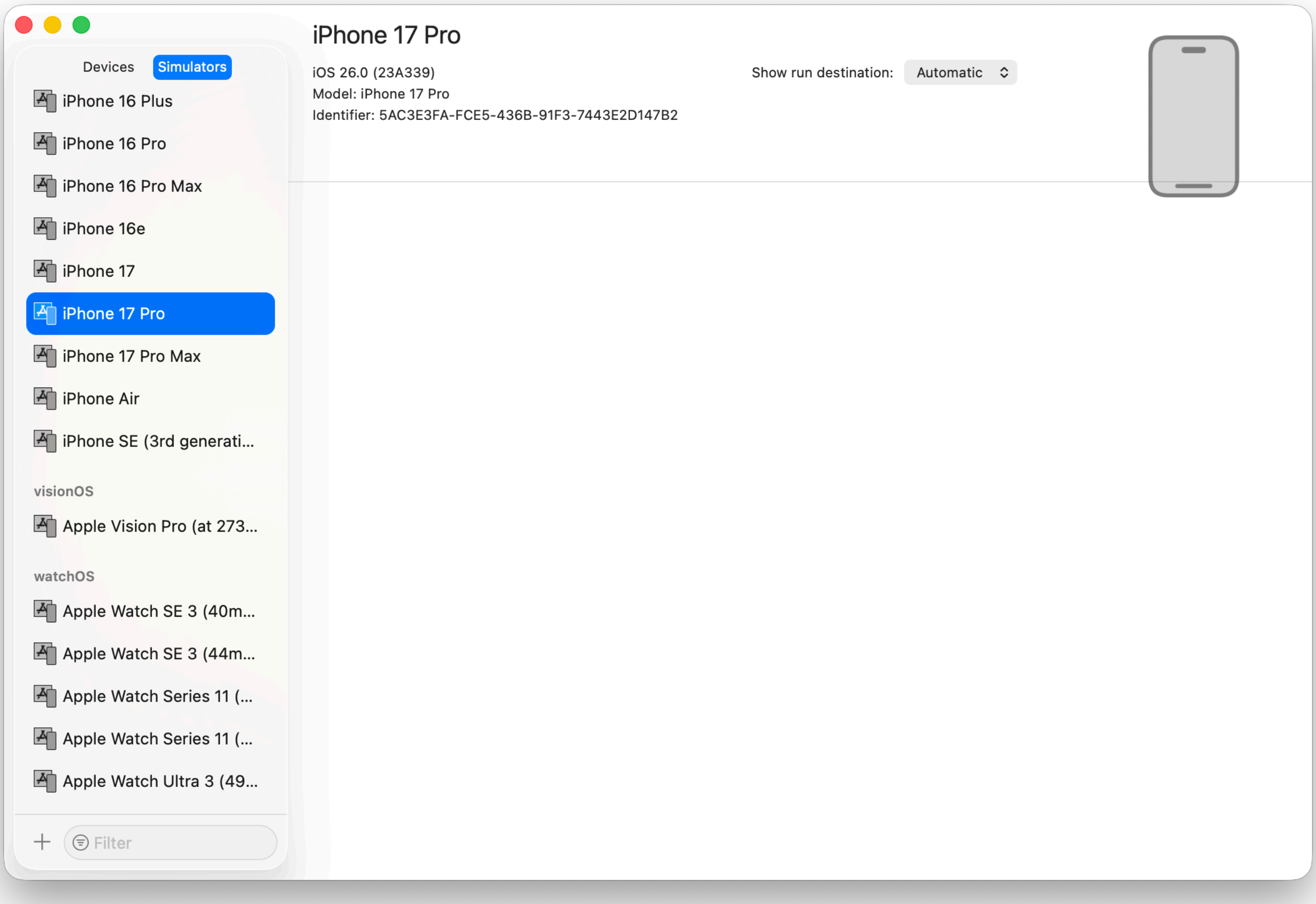
If you don’t have platform support installed for your target, you can’t build and run your app on a device. To install platform support, click the Get button that appears next to the `Any [Platform] Device` run destination. Alternatively, manage your downloads later in Components settings (see Downloading and installing additional Xcode components).

To learn more about schemes, see Customizing the build schemes for a project.

> **Important**
>
> When running apps in Simulator, some hardware-specific features might not be available. Frameworks that provide access to device-specific features also provide API to tell you when those features are available. Call those APIs and handle the case where a feature isn't available. To test the feature itself, run your code on a real device.

Configure the list of simulated devices

Manage real and simulated devices in the Devices and Simulators window in Xcode. To view this window, choose Window > Devices and Simulators. To view the available simulated devices, click the Simulators tab in the sidebar.



iPhone 17 Pro

iOS 26.0 (23A339)
Model: iPhone 17 Pro
Identifier: 5AC3E3FA-FCE5-436B-91F3-7443E2D147B2

Show run destination: Automatic

To add a new simulated device, click the Add button (+) at the bottom of the list of simulators and specify the configuration you want. You can add new simulators to specify a different device type or operating system version than the default set. To remove a simulator from the list, select it and press Delete.

Note

Xcode requires the Simulator runtime for each platform and system version for which you build and run Simulator. If Xcode doesn’t display device types for a platform, you might need to install that platform’s Simulator runtime. For more information, see Downloading and installing additional Xcode components.

Connect real devices to your Mac

To manage connections to your real devices, choose the Devices tab in the Devices and Simulators window in Xcode. The Devices tab shows the currently connected and disconnected devices and can help you diagnose problems that might occur. For example, Xcode might show a device as unavailable if it’s not running an operating system version your app supports. It also shows new devices available for pairing with your Xcode installation. Pair a device with Xcode to include them in the list of run destinations for your projects.

To pair a device with a physical connection, connect the device to your Mac using an appropriate cable. Unlock the device and follow any instructions that appear in Xcode or on the device.

To pair Apple Vision Pro or Apple TV without a physical connection:

1. Ensure that both your Mac and the device to connect are on the same Wi-Fi network. The Wi-Fi network must be compatible with Bonjour.

2. Broadcast the device to the target Mac over the local network. To do this for a visionOS device, choose Settings > General > Remote Devices and for a tvOS device, choose Settings > Remotes and Devices > Remote App and Devices.

3. Select the device from the list in the Devices and Simulators window in Xcode and click the pairing button which triggers a code to appear on the target device.

4. Enter the code on the Mac to complete the pairing process.

After pairing is complete, the device shows up under connected devices in the Devices and Simulators window in Xcode. You don’t need to keep a paired device physically connected to your Mac to install and run apps. If your device is connected to Wi-Fi on the same network as your Mac, Xcode can use that connection to install and run your app.

To pair an Apple Watch to a Mac, connect its companion iPhone to the Mac with a cable, and ensure that the iPhone is paired for development. After this step, follow any instructions on the Apple Watch to trust the Mac. When paired through an iPhone running iOS 17 or later, Xcode connects to the Apple Watch over Wi-Fi. Series 5 and older models of Apple Watch additionally require the Apple Watch and Mac to be associated with the same Bonjour-compatible Wi-Fi network. When paired through an iPhone running older versions of iOS, Xcode requires the iPhone to remain connected to the Mac in order to develop on any model of Apple Watch.

Before installing your app, perform a few additional steps:

- Add your Apple Account to the Accounts settings in Xcode.
- Choose a valid team in your project's Signing & Capabilities pane.
- Code sign your macOS app if it includes capabilities that require code signing; see Adding capabilities to your app.
- Register the device with your team if you belong to the Apple Developer Program.
- Enable Developer Mode on an iOS, iPadOS, visionOS, or watchOS device, as described in Enabling Developer Mode on a device.

> Note
>
> You don't need to configure a Mac device to run your macOS apps. Similarly, to run the macOS version of an iPad app, choose My Mac (the Mac running Xcode) as the device.

Run the app

To build and run the app on the selected simulated or real device, click the Run button in the toolbar or choose Product > Run. View the status of the build in the activity area of the toolbar.

If the build is successful, Xcode runs the app and opens a debugging session in the debug area. Use the controls in the debug area to step through your code, inspect variables, and interact with the debugger. To run the app without the debugger, disable the Debug executable option in the Info tab of the scheme editor.

If the build is unsuccessful, click the indicators in the activity area to read the error or warning messages in the Issue navigator. Alternatively, choose View > Navigators > Issues to view the messages.

When you're done testing the app, click the Stop button in the toolbar.

Interact with the simulated environment

If you choose a simulated device as the run destination, Simulator launches and displays a window that corresponds to the simulated environment. For some devices, Simulator surrounds the screen content with a shell that resembles the target device. In visionOS, it displays a synthetic space to mimic the experience someone would have when they wear the device.

Each device shell and space has specific controls to support interactions. For device-specific details, see the reference on interactions.

- Interacting with your app in the iOS and iPadOS simulator
- Interacting with your app in the tvOS simulator
- Interacting with your app in the watchOS simulator
- Interacting with your app in the visionOS simulator

See Also

Essentials

Enabling Developer Mode on a device

Grant or deny permission for locally installed apps to run on iOS, iPadOS, visionOS, and watchOS devices.