Devices and Simulator

Configure and manage devices connected to your Mac or devices in Simulator and use them to run your app.

Overview

As you build your app, run it on real or simulated devices to test new features and evaluate your progress.

Run your app on a real device to confirm that you can interact with it as you intend, and that your app's performance on a real device meets your expectations. To run your app on a real device, first enable Developer Mode on your iOS, iPadOS, visionOS, or watchOS device, then connect your device to your Mac and run your app from Xcode.

Run your app on Simulator to quickly evaluate new features and bug fixes, and to see how your user interface works on devices that you don't have access to. Use Simulator's features to streamline your testing, but be aware that there are some scenarios that require testing on hardware instead of Simulator.

Topics

Essentials

📄 Enabling Developer Mode on a device

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

📄 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.

Simulator management

📄 Downloading and installing additional Xcode components

Add more Simulator runtimes, optional features, and support for additional platforms.

📄 Installing your app in many Simulator platforms and versions

Set up your app in multiple Simulator platforms and versions without the build-and-run cycle.

Simulator interactions

📄 Interacting with your app in the iOS and iPadOS simulator

Use your Mac to control interactions with your iOS and iPadOS apps in Simulator.

📄 Interacting with your app in the tvOS simulator

Use your Mac to control interactions with your tvOS apps in Simulator.

📄 Interacting with your app in the watchOS simulator

Use your Mac to control interactions with your watchOS apps in Simulator.

📄 Interacting with your app in the visionOS simulator

Use your Mac to navigate spaces and control interactions with your visionOS apps in Simulator.

📄 Configuring Simulator for your working environment

Adjust Simulator settings for window or screen size, the light or dark appearance, and audio settings, and restart or reset a simulated device.

📄 Simulating an external display or CarPlay

Test how your app handles an external display or CarPlay from Simulator.

📄 Capturing screenshots and videos from Simulator

Record and share test results, or prepare for App Store distribution with screenshots and videos of your app from Simulator.

Simulator testing considerations

📄 Testing in Simulator versus testing on hardware devices

Review the differences between Simulator and hardware devices to determine which you should choose to test a scenario.

📄 Sharing data with Simulator

Enter text directly in Simulator, or share location data, images, web addresses, files, or data from the clipboard with Simulator.

📄 Testing complex hardware device scenarios in Simulator

Test hardware device-specific scenarios, such as Face ID or Touch ID authentication, fall detection, getting a memory warning, or location changes.

📄 Identifying graphics and animations issues in Simulator

Reveal performance and display issues in your views with color overlays, and slow down animations to debug and improve them.

Simulator troubleshooting

📄 Troubleshooting Simulator launch or animation issues

Diagnose and resolve issues launching a simulator, or with slow scrolling or animations in Simulator.

See Also

Tuning and debugging

☰ Debugging

Identify and address issues in your app using the Xcode debugger, Xcode Organizer, Metal debugger, and Instruments.

☰ Performance and metrics

Measure, investigate, and address the use of system resources and issues impacting performance using Instruments and Xcode Organizer.

☰ Testing

Develop and run tests to detect logic failures, UI problems, and performance regressions.