Performance and metrics

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

Topics

Essentials

Improving your app’s performance

Model, measure, and boost the performance of your app by using a continuous-improvement cycle.

Profiling apps using Instruments

Use Instruments to analyze the performance, resource usage, and behavior of your apps. Learn how to improve responsiveness, reduce memory usage, and analyze complex behavior over time.

Analyzing the performance of your shipping app

View power and performance metrics for apps you distribute through the App Store.

Creating a performance plan for your visionOS app

Identify your app’s performance and power goals and create a plan to measure and assess them.

Responsiveness

Analyzing responsiveness issues in your shipping app

Identify responsiveness issues your users encounter, and use the hang and hitch data in Xcode Organizer to determine which issues are most important to fix.

Improving app responsiveness

Create a user experience that feels responsive by removing hangs and hitches from your app.

Understanding user interface responsiveness

Make your app more responsive by examining the event-handling and rendering loop.

Understanding and improving SwiftUI performance

Identify and address long-running view updates, and reduce the frequency of updates.

Understanding hangs in your app

Determine the cause for delays in user interactions by examining the main thread and the main run loop.

Understanding hitches in your app

Determine the cause of interruptions in motion by examining the render loop.

Diagnosing performance issues early

Diagnose potential performance issues in your app during development and testing with the Thread Performance Checker tool in Xcode.

Reducing your app's launch time

Create a more responsive experience with your app by minimizing time spent in startup.

Reducing terminations in your app

Minimize how frequently the system stops your app by addressing common termination reasons.

Processor usage

Addressing CPU bottlenecks

Locate and fix pipeline stalls, cache misses, and other performance issues.

Analyzing CPU usage with the Processor Trace instrument

Identify code where your app uses the CPU inefficiently.

Memory and size

Reducing your app's memory use

Improve your app's performance by analyzing memory-use metrics and making changes to maximize memory efficiency.

Reducing your app’s size

Measure your app’s size, optimize its assets and settings, and adopt technologies that help streamline installation over a mobile internet connection.

Graphics

Analyzing the performance of your Metal app

Ensure consistent, smooth rendering by profiling your app’s frame time.

Analyzing the memory usage of your Metal app

Keep your app alive in the background by managing its memory footprint.

Power

Analyzing your app’s battery use

Increase the available use time for your app on a single battery charge by reducing your app’s power consumption.

Measuring your app’s power use with Power Profiler

Profile your app’s power impact whether or not your device is connected to Xcode.

Reducing your app’s battery use

Adopt design principles and recommended APIs to consume less power.

Disk usage

Reducing disk writes

Improve your app’s responsiveness by optimizing how it writes data to permanent storage.

Reducing your app’s disk usage

Measure and minimize the space your app uses to store its files.

Network

Analyzing HTTP traffic with Instruments

Measure HTTP-based network performance and usage of your apps.

Custom instruments

Creating custom modelers for intelligent instruments

Create Custom Modelers with the CLIPS language and learn how the embedded rules engine works.

See Also

Tuning and debugging

Devices and Simulator

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

Debugging

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

Testing

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