


[Xcode](#) / Debugging

# Debugging





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

## Topics

### Essentials

-  Diagnosing and resolving bugs in your running app  
Inspect your app to isolate bugs, locate crashes, identify excess system-resource usage, visualize memory bugs, and investigate problems in its appearance.

### Debugging strategies

-  Diagnosing issues in the appearance of a running app  
Inspect your running app to investigate issues in the appearance and placement of the content it displays.
-  Diagnosing memory, thread, and crash issues early  
Identify runtime crashes and undefined behaviors in your app during testing using Xcode's sanitizer tools.
-  Analyzing HTTP traffic with Instruments  
Measure HTTP-based network performance and usage of your apps.
-  Detecting when your app contacts domains that may be profiling users  
Use Instruments to assess whether your app or its third-party SDKs connect to domains that may profile users.

## Graphics

- ☰ Metal developer workflows  
Locate and fix issues related to your app's use of the Metal API and GPU functions.
- ☰ Metal debugger  
Debug and profile your Metal workload with a GPU trace.

## Breakpoints and variables

- 📄 Setting breakpoints to pause your running app  
Specify where your app pauses when running the debugger to investigate bugs.
- 📄 Stepping through code and inspecting variables to isolate bugs  
Find the cause of your bugs by watching variables change as you step through your source code in the debugger.

## Reports

- 📄 Building your app to include debugging information  
Configure Xcode to produce the symbol information for debugging and crash reports.
- ☰ Diagnosing issues using crash reports and device logs  
Use crash reports and device logs to debug app issues.

## Entitlements

- 📄 Diagnosing Issues with Entitlements  
Verify your app's entitlements at every stage of development to track down errors during distribution.

## 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.

### ☰ 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.