Standard Memory Debugging Tools

This thread has been locked by a moderator.

11k

In many cases I'll be talking to folks with a memory management problem and I'll say "You should investigate this with the **standard memory debugging tools**." They then turn around and ask me "What *are* those tools?" Well, this is what I mean:

- Zombies — This lets you quickly detect when an object is used after it's been deallocated. Learn more about it in the Finding zombies section of the *Instruments Help* and in Investigating crashes for zombie objects. There was also an *excellent* WWDC video about this, namely, WWDC 2010 Session 311 *Advanced Memory Analysis with Instruments*. This is no longer available in the video archive but if you can find a copy it's well worth a watch.
- Address Sanitizer — This is a lower-level tool that finds a variety of common memory management issues, including use after free and buffer overruns. Learn more about this in Diagnosing memory, thread, and crash issues early and the various articles it links to. There's also a good discussion of this tool, and other Xcode runtime diagnostic tools, in WWDC 2017 Session 406 *Finding Bugs Using Xcode Runtime Tools* (also no longer available from Apple).
- Older tools — There are a variety of older tools that might be useful in some specific circumstances. See the Enabling the Malloc Debugging Features section of the *Memory Usage Performance Guidelines* for more information about these. Of specific interest is `libgmalloc`, which is documented in a UNIX man page.

For some practical examples of how to identity a memory management crash report and then investigate that crash with these tools, take a look at WWDC 2018 Session 414 Understanding Crashes and Crash Logs.

Share and Enjoy
—
Quinn "The Eskimo!" @ Developer Technical Support @ Apple
`let myEmail = "eskimo" + "1" + "@" + "apple.com"`

Revision History:

- **2023-05-09** Added a link to Investigating crashes for zombie objects.
- **2023-03-22** Removed another WWDC session video link. Made minor editorial changes.
- **2020-10-23** Fixed some formatting errors.
- **2019-10-30** Removed the link to WWDC 2010 Session 311 *Advanced Memory Analysis with Instruments* because it's not long available in the archive. Refreshed all the other links.
- **2019-01-22** Fixed the link to `libgmalloc`.
- **2018-11-02** Updated to include a reference to WWDC 2018 Session 414 Understanding Crashes and Crash Logs.
- **2017-11-16** First posted.

Debugging

Reply

Posted 5 years ago by eskimo

Add a Comment

This site contains user submitted content, comments and opinions and is for informational purposes only. Apple disclaims any and all liability for the acts, omissions and conduct of any third parties in connection with or related to your use of the site. All postings and use of the content on this site are subject to the Apple Developer Forums Participation Agreement.

Developer › Forums

Platforms
iOS
iPadOS
macOS
tvOS
watchOS

Tools
Swift
SwiftUI
SF Symbols
Swift Playgrounds
TestFlight
Xcode
Xcode Cloud

Topics & Technologies
Accessibility
Accessories
App Extensions
App Store
Audio & Video
Augmented Reality
Business
Design
Distribution
Education
Fonts
Games
Health & Fitness
In-App Purchase
Localization
Maps & Location
Machine Learning
Security
Safari & Web

Resources
Documentation
Curriculum
Downloads
Forums
Videos

Support
Support Articles
Contact Us
Bug Reporting
System Status

Account
Apple Developer
App Store Connect
Certificates, IDs, & Profiles
Feedback Assistant

Programs
Apple Developer Program
Apple Developer Enterprise Program
App Store Small Business Program
MFi Program
News Partner Program
Video Partner Program
Security Bounty Program
Security Research Device Program

Events
App Accelerators
App Store Awards
Apple Design Awards
Apple Developer Academies
Entrepreneur Camp
Tech Talks
WWDC

To view the latest developer news, visit News and Updates.