Standard Memory Debugging Tools

This thread has been locked. Questions are automatically locked after two months of inactivity, or sooner if deemed necessary by a moderator.

6.8k

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 has been deallocated. You can learn more about it in the Finding zombies section of the *Instruments Help*. There was also an *excellent* WWDC video about this, namely, WWDC 2010 Session 311 *Advanced Memory Analysis with Instruments*. Unfortunately this is no long 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. You can learn more about this in the Address Sanitizer article. 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.
- Older tools — There's 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!"
Apple Developer Relations, Developer Technical Support, Core OS/Hardware
`let myEmail = "eskimo" + "1" + "@apple.com"`

Updates:

- 16 Nov 2017 — First posted.
- 2 Nov 2018 — Updated to include a reference to WWDC 2018 Session 414 Understanding Crashes and Crash Logs.
- 22 Jan 2019 — Fixed the link to `libgmalloc`.
- 30 Oct 2019 — 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.
- 23 Oct 2020 — Fixed some formatting errors.

Debugging

Asked 3 years ago by eskimo

Reply to this question

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 > Apple Developer Forums

Discover
macOS
iOS
watchOS
tvOS
Safari and Web
Games
Business
Education
WWDC

Design
Human Interface Guidelines
Resources
Videos
Apple Design Awards
Fonts
Accessibility
Internationalization
Accessories

Develop
Xcode
Swift
Swift Playgrounds
TestFlight
Documentation
Videos
Downloads

Distribute
Developer Program
App Store
App Review
Mac Software
Apps for Business
Safari Extensions
Marketing Resources
Trademark Licensing

Support
Articles
Developer Forums
Feedback & Bug Reporting
System Status
Contact Us

Account
Certificates, Identifiers & Profiles
App Store Connect

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