

ADVANCED DEBUGGING & BEST PRACTICES

with **XCode 4+ & Friends**

Instructor:

Kendall Helmstetter Gelner

@kendaldevdiary

Kendall.Gelner@KiGiSoftware.com

materials found:

<https://github.com/KiGi/AdvancedDebuggingCode>

NOW WITH FILES!

WHO AM I

- Full-Time iPhone developer since release of SDK.
- Independent consultant for iOS development.
- Over two decades programming experience from corporate to end user.
- Over 40000 Reputation on StackOverflow

THINGS OF NOTE

- All files included on GitHub:

<https://github.com/KiGi/AdvancedDebuggingCode>

- XCode tips, tricks and discoveries on twitter

@kendaldevdiary

- Also make sure to watch WWDC videos 413 (Advance Debugging With LLDB) & 404 (Debugging With XCode 5) - this talk builds on and adds to many ideas there.

ON DEBUGGING

"most people who are good at debugging are also good programmers, but a minority of good programmers are good at debugging."

--<http://programmingliterate.tumblr.com>

"It is difficult not to wonder whether that combination of elements which produces a machine for labor does not create also a soul of sorts, a dull resentful metallic will, which can rebel at times."

--Pearl S. Buck

"Once a new technology rolls over you, if you're not part of the steamroller, you're part of the road."

--Stewart Brand

THE PHILOSOPHY

Debugging is about giving you many paths of visibility into your application that gather enough information for inspiration to strike.

SCHEDULE

- Morning: Performance Instruments, LLDB
- Afternoon: Analysis Instruments, other tools, CoreData

**Demos for each tool, and Hands-on labs
between sections!**

PERFORMANCE INSTRUMENTS & OVERVIEW

SIMULATOR FEATURES

- The simulator has some helpful debugging features added over time:

Slow Animation

Graphics Blending

Location

SIMULATING LOCATION

- You can set simulated location in the simulator
- You can have XCode run through a GPX file simulating location.
- XCode can also help you make a GPX file.
- You can use the Automator instrument to more accurately simulate location.

INSTRUMENT: TIME PROFILER

- Great way to check why an app is busy - or why it's not.
- Helps you focus on worst performance first.
- Many helpful ways to look at results.
- XCode 5 has a simpler view you see my default, can help you see when it may be good to run Time Profiler.

LAB TIME

20 MIN

- Explore Core Animation effects on your favorite app
- Use time profiler and explore time used when scrolling debugging app (or your own)!
- Change location in simulator directly and from XCode.

ADVANCED LOGGING

BETTER LOGGING

- NSLog always outputs, even for customers.
- Evaluating data for NSLog can slow things down.
- Instead - consider NSLog, defining away for AppStore builds and altering where debugging goes.
- debugDescription can also be implemented for extra data.
- Consider using blocks for expensive log calculations

LLDB

LLDB - USEFUL COMMANDS

- po “recursiveDescription” works on any view to dump view hierarchy.
- if you crash is in objc_msgsend, you can at least find the method name called with:

p (char *)\$ecx (simulator)

p (char *)\$r1 (device)

LLDB - EXPRESSIONS

- Expressions compile real code to run in app.
- Access with “expr” command (GDB used “p” or “po” to evaluate expressions, so can LLDB)
- Define debugger global variables with “\$” prefix
- Use with continued breakpoints for real-time patching!
- thread return <EXPR> to return from method with expression.

DATA FORMATTERS

- Use to provide better view on your local variables.
- Can define in XCode variable window:

```
value={{(NSString*)[$var size]}:s
```

- Also possible to use python summaries. Not same format as XCode.
- Summary Details: **<http://lldb.lvm.org/varformats.html>**
- If LLDB does not “see” a class, try using `NSClassFromString`.

DATA FORMATTERS

- Can find examples of system formatters (in XCode 4.5+):

XCode.app/Contents/SharedFrameworks/
LLDB.framework/

Resources/Python/lldb

/formatters/objc

DATA FORMATTERS

- Can create custom python summaries
- Easier than it sounds
- Examples at

[http://llvm.org/svn/llvm-project/lldb/trunk/
examples/customization/bin-utils/](http://llvm.org/svn/llvm-project/lldb/trunk/examples/customization/bin-utils/)

Demo!

LAB TIME

20 MIN

- Add NSLog to DebuggingApp from extra files, convert uses of NSLog.
- Use EXPR when paused to try modifying an existing value (like a cell label).
- Try setting a debugger variable in one breakpoint and viewing it a different breakpoint.
- Add a custom formatter in XCode for KG FlickrItem and examine values as the cells are formatted

BREAKPOINTS

CONDITIONAL

- For any expression used in a breakpoint condition, make sure the LLDB command line accepts it first.
- Conditional breakpoints can use any expression.
- With an auto-continue breakpoint, you can add small blocks of expr code to try adding logic on the fly.
- Note you may have to cast return values for every call.
- Can add condition on LLDB defined variable.
- If the condition does not work it will stop every time.

DEBUGGER AS AUDIO PROBE

- Covered in older WWDC video, but there is a better why...
- Understand relations between calls in an API, or the flow of data in your app.
- Also, speech (numbers only...)!
- You can use your own sounds placed in ~/Library/Sounds

LAB TIME

20 MIN

- Use EXPR on a continue breakpoint in cellForRow to strip out extra text on flickr user name output.
- Add an audio breakpoint to both cellForRow & heightForRow in any table, and hear how they are called.
- Successfully have the debugger stop at a cell when the tag count is greater than 10, and speak the index.

LUNCH - BACK AT 1:00PM

INSTRUMENTS FOR ANALYSIS

ARC CONVERSION ISSUES

- If you are using CF objects, try `__bridge` cast first to fix.
- Replace autorelease pools with `@autoreleasepool`
- For anything really odd (or third party libraries), you can opt not to convert per file.

LEAKS VS CLOGS

- ARC prevents common memory issues, but not all!
- Leaks are like a murder mystery.
- You can also “clog” memory with memory you didn’t know you were still using.
- Use Leaks first, then Object Alloc instrument with Heapshot.

INSTRUMENT REGIONS

- You can flag points of interest in Instruments.
- How to automatically flag points of interest?
- Custom instruments with DTrace!

ZOMBIE HUNT

- Use the Zombie instrument to find dead objects that cause crashes.

LAB TIME

20 MIN

- Create a custom instrument to track `awakeFromNib`, try various controls.
- Use Object Alloc and try clicking on Flickr items then traversing other users.
- Do a leak check when opening Flickr items and closing.
- Run the “Modern ObjC Converter” (right by Arc convert) and see what it wants to do.

RISE OF THE (HELPFUL) MACHINES

REVEAL

- Included in your “Extra” folder
- Find at RevealApp.com, beta only for now (but free!).
- Shows breakout of views
- Just add framework (embedded in app bundle) and run

PONY DEBUGGER

- Included in your “Extra” folder
- Add project : libraries : linker flag : header path. (PonyAdd.txt)
- Install XCode command line tools.
- Install & run the server (PonyDebuggerQuickstart.txt)
- Then add code to attach to Pony server.
- Experiment, some things may not work.

NETWORK LINK CONDITIONER

- Now download through XCode:

XCode > Open Developer Tool > More Developer Tools

Look for “Hardware IO Tools” package (added to “Extra”)

- Affects all the apps on the system!
- Existing settings are OK, but you can make a better version of bad networks...

CHARLES

- Also in “Extra” folder.
- This one costs money, but is worth it.
- Free trial for quite a while.
- Great reporting of all the traffic details you could wish.
- Set HTTPS hosts to proxy traffic (install Charles local cert)
- Make sure a “curl” or browser call works before you try to code against something or look for bugs.

IPHONE CONFIGURATION UTILITY

- Download from web (Google for it), also in “Extras” folder
- Can be used to quickly switch to proxy server for traffic examination.
- Can have multiple profiles installed, with only some active.

APPCODE

- Download from web at <http://www.jetbrains.com/objc>
- No detail here, but worth a look.
- Can switch to XCode and back again.
- Not sure about XCode 5 compatibility.

LAB TIME

20 MIN

- Install PonyDebugger in the sample app, and try browsing the Flickr entries there.
- Install Charles and examine traffic when Flickr is called.
- Install Reveal framework in project and experiment in simulator (or on open network if you dare!).
- Try installing a profile on your device to go through the Charles proxy (your IP / port 8888 by default), or set up HTTP proxy on device WiFi connection.

BEST PRACTICES FOR A HEALTHY APP

COMMON APPROACHES

- Carefully set up app structure first.
- Make good use of real folders to keep file separation.
- Keep a common naming scheme.
- Make sure to use refactoring to rename classes and properties.

CODING THINGS

- Make sure to consider where app data is stored, use no-backup attribute as needed:

```
const char* attrName = "com.apple.MobileBackup";  
u_int8_t attrValue = 1;  
  
int result = setxattr(filePath, attrName, &attrValue, sizeof(attrValue), 0, 0);
```

- Look for libraries before coding something complex (AFNetworking).
- Keep things simple first, then use Time Profiler to find what needs to be more complex.

QUESTION BREAK
15 MIN

CORE DATA DEBUGGING

- Instruments can help track saves and other operations.
- Also just before a save, look at `ManagedObjectContext` `updatedObjects` or `insertedObjects` to see what will be saved.
- Output from Core Data objects in debugger truncates strings! Your full string is there.
- Use `valueForKey:` to get properties out of a Core Data object in the debugger. `willAccessValueForKey:` triggers fetch, as does “`allObjects`” on relationships.
- See WWDC2013 session “Core Data Performance Optimiztion and Debugging” (Session 211)

CORE DATA STRUCTURE

- Be careful of deletes, very dangerous.
- isDeleted does not really mean isDeleted.
- Inheritance means flat tables.
- Make everything you can optional, validations dangerous.
- Make use of helper libraries, but don't get too far removed.
- Always create model with more than one version to start.
- Create new model version for every app store release (if needed).

CORE DATA FILES

- Keep in caches.
- Auto migrate, destroy if that doesn't work.
- iCloud key/value pretty stable, Core Data documents fiddly.
- You can pre-load databases, copy them into writable directory on app start.
- You can copy DB from the simulator directories to keep different cases around (as long as you have the same version) or transplant databases from device.

MORE CORE DATA?

Come see my talk:

“The Ten Terrible Tribulations of Core Data”

Tuesday 1:40pm - Lone Tree Room

OPEN LAB

THANKS FOR ATTENDING!

Instructor:

Kendall Helmstetter Gelner

@kendaldevdiary

Kendall.Gelner@KiGiSoftware.com

materials found:

<https://github.com/KiGi/AdvancedDebuggingCode>