

Unified Logging

AI Wold

Logging methods

- NSLog
- asl_log
- syslog
- print, debugPrint
- Third party libraries (Lumberjack, etc)
- os_log (Unified logging)

Unified logging

- iOS 10+
- Uses a combination of memory/disk to store logs with compression and other fancy stuff
- New tools (Console, log in macOS Sierra)
- Ties in with Activity Tracing
- Easy to filter
- Not very well documented - WWDC videos have lots of tips
- C API, recently became available in Swift

Categorization

- Subsystem (e.g. com.apple.networkstatistics)
- Category (e.g. NetworkStatistics)
- Type
 - Fault - always on
 - Error - always on
 - Default - always on
 - Info
 - Debug - off by default

Tools

- New Console.app in macOS Sierra
 - Great filtering
 - Activities
 - Load log archives
- log command line tool
 - Streaming, filtering
 - Saving log archives
 - Change settings

Activity Tracing

- Log messages get associated with an "activity"
- Activity crosses threads, processes, etc. so all log messages associated with it are grouped
- UIKit automatically starts an Activity in IBActions
- You can create additional Activities manually
- The API is C based and hard to use from Swift

Questions

- Is there a good way to use `os_log` in an app that still targets iOS 9?
- How can you use the `log` command on an iOS device? On simulator you can run it with `simctl...`
- `sysdiagnose` tool
- How to customize log profiles

Resources

- WWDC 2016 Session 721 - Unified Logging and Activity Tracing
- WWDC 2014 Session 714 - Activity Tracing
- <https://www.objc.io/issues/19-debugging/activity-tracing/>
- Swift Activity wrapper:
<https://gist.github.com/zwaldowski/49f61292757f86d7d036a529f2d04f0c>
- `man 1 log`