

A Look at Command Line Swift

Intro

- Me
 - I work at GMO Pepabo on "Minne"
- GitHub Contributions
 - Danger-swift
 - vim-xcode
 - swiftformat-script
 - LattnerBot

What is a script

- A program without a GUI that has a only a few set options
 - Not a Mac app (has a GUI)
 - Not grep (too many options)
- Examples
 - Release.swift
 - Format.swift

When to use a shell script

- When covered with common Unix programs

When not to use a shell script

- External packages/libraries
- Long
- Frequent changes

Comparison of Swift with Ruby and Python

- Type safety
- Concurrency
 - Python has it
 - Swift relies on GCD or lower level APIs
 - Go seems like a good choice
- Documentation

Comparison continued

- Portability
 - Pyenv
 - Rbenv
 - Swiftenv
- Tooling
 - Reliance on Xcode
- Package management

SPM

- Strengths
 - Similarity to Bundler and pip
- Weaknesses
 - Commands are long
 - `swift package generate-xcodeproj`
 - Fixable w/ alias
 - `Package.swift` is very verbose

Example of Package.swift file

```
// swift-tools-version:4.0
import PackageDescription

let package = Package(
    name: "LattnerBot",
    products: [
        .library(
            name: "LattnerBot",
            targets: ["LattnerBot"]),
    ],
    dependencies: [
        .package(url: "https://github.com/JohnSundell/ShellOut.git", from: "2.0.0"), // Shell commands
        .package(url: "https://github.com/kylef/Commander.git", from: "0.8.0"), // CLI tool
        .package(url: "https://github.com/SlackKit/SlackKit.git", from: "4.1.0") // Slack interaction
    ],
    targets: [
        .target(
            name: "LattnerBot",
            dependencies: ["Commander", "ClocWrapper", "Output"]),
        // More targets
        .testTarget(
            name: "OutputTests",
            dependencies: ["Output"]),
    ]
)
```


General good fits for Swift cli

- iOS development team
 - Shared language
 - No whiplash from switching languages
- One person devs
 - Don't work with a scripting language often

Bad fits for Swift cli

- Supporting a non-Swift codebase
 - Shared flow w/ non-Swift team
- Lots of concurrency
- Cross platform portability
 - No Windows support
 - Unclear, undocumented Linux support
 - SPM
- Lots of string manipulation

How to make a *Swift* script or cli app

Executable

```
#!/usr/bin/swift
```

```
import Foundation
```

```
// ...
```

Tasks/process

```
@discardableResult
func shell(_ command: String) -> Int32 {
    let args = command.components(separatedBy: " ")

    let process = Process()
    process.launchPath = "/usr/bin/env"
    process.arguments = args
    process.launch()
    process.waitUntilExit()

    return process.terminationStatus
}
```

Arguments

```
var filePath: String?
var configFile: String?

let args = ProcessInfo.processInfo.arguments
args.enumerated().forEach { index, arg in
    switch arg {
    case "--path":
        filePath = args[index + 1]
    case "-p":
        filePath = args[index + 1]

    case "--config":
        configFile = args[index + 1]
    case "-c":
        configFile = args[index + 1]

    default:
        break
    }
}
```

Putting it all together

```
let disabledRules: [String] = [
    "consecutiveSpaces",
    "trailingSpace",
]

guard let filePath = filePath else {
    print("Missing --path/-p flag"); exit(1)
}

print("Running Swiftformat")
shell("swiftformat --disable \(disabledRules.joined(separator: ",")) \(filePath)")

guard let configFile = configFile else {
    print("Missing --config/-c flag"); exit(1)
}

print("Running Swiftlint Autocorrect")
shell("fd . -0 --full-path \(filePath) -e swift -x swiftlint autocorrect --config \(configFile) --path")
```

Marathon

— Demo

SPM

- Demo of LattnerBot
- Discussion of convenient libraries

Predictions for the future

- A general shift from Ruby to Swift for iOS tools
 - Cocoapods vs Carthage
 - Fastlane
 - Danger
- A community-led approach without much focus by Apple
 - Perhaps more documentation
- A potentially significant influx of community support

Swift features

- Async-await
- Native regex
- Python and Ruby interop
- Better SPM
- Better string subscripting (never)

Resources

- Apple Docs
- Stack Overflow
- Blogs

Tools

- Swiftenv
- Marathon

Vim-related

- Sample Vimrc
- ALE
- Deoplete
- vim-xcode
- <https://github.com/keith/swift.vim>

Nice libraries

- <https://github.com/Draveness/RbSwift>
- <https://github.com/kylef/Commander>
- <https://github.com/jakeheis/SwiftCLI>
- <https://github.com/JohnSundell/ShellOut>
- <https://github.com/JohnSundell/Files>
- <https://github.com/kareman/SwiftShell>

Questions for audience

- Do you use Swift for anything other than iOS, what?
- Do you prefer a language other than Swift for scripting or cli apps, which one?

Questions from audience