

Preparing macOS command-line tools for distribution

CocoaHeads Nantes — October 2024

Frank Lefebvre — @franklefebvre@mastodon.social

Creation

```
swift package init --type executable
```

```
swift package init --type tool
```

```
import ArgumentParser
```

```
@main
```

```
struct MyAwesomeTool: ParsableCommand {
```

```
    mutating func run() throws {
```

```
        /* your code here */
```

```
    }
```

```
}
```

Local Execution

```
swift run
```

```
swift package experimental-install
```

```
~/.swiftpm/bin/MyAwesomeTool
```

```
swift build --configuration release --arch arm64 --arch x86_64
```

```
./.build/apple/Products/Release/MyAwesomeTool
```

Distribution

- Code Signing
- Notarization

Code Signing

- Use “Developer ID Application” certificate
- A command-line tool is not a bundle: no embedded Info.plist file
- Options
 - Create a linker section containing Info.plist
 - Provide identifier explicitly when signing

Code Signing

with `__info_plist` section

Either modify `Package.swift` or add options to `swift build`

```
.target(  
    name: "MyAwesomeTool",  
    dependencies: [...],  
    linkerSettings: [  
        .unsafeFlags( ["-Xlinker", "-sectcreate",  
                        "-Xlinker", "__TEXT",  
                        "-Xlinker", "__info_plist",  
                        "-Xlinker", "path/to/Info.plist"] )  
    ]),
```

```
swift build -c release -Xlinker -sectcreate -Xlinker __TEXT -Xlinker __info_plist -Xlinker path/to/Info.plist
```

Code Signing

with explicit identifier

```
codesign --force  
    --entitlements /path/to/entitlements/file  
    --options runtime  
    --sign identity_or_certificate_hash  
    --identifier com.mydomain.MyAwesomeTool  
    /path/to/MyAwesomeTool
```

Notarization

- notarytool input
 - disk image
 - installer package
 - zipped application

Installer package

- Needs to be signed
 - “Developer ID Installer” certificate
 - Can reuse identifier
- Create filesystem path
 - `installer_root/usr/local/bin/MyAwesomeTool`

Installer package

pkgbuild

`--identifier com.mydomain.MyAwesomeTool`

`--version 1.0`

`--root installer_root`

`--sign identity_or_certificate_hash`

`MyAwesomeTool.pkg`

Notarization

Create Keychain Profile

- Generate app-specific password (<https://appleid.apple.com>)
- Store in keychain
 - `xcrun notarytool store-credentials MyNotaryProfile`
- Can be reused across projects

Notarization

Submit to notary service and staple receipt

- `xcrun notarytool submit MyAwesomeTool.pkg
--keychain-profile MyNotaryProfile --wait`
- `xcrun stapler staple -v MyAwesomeTool.pkg`

Summary

- swift build
- codesign
- pkgbuild
- notarytool submit
- notarytool staple

swift-export.pkg

<https://github.com/franklefebvre/swift-export>

```
cd /path/to/your/project  
swift export
```

