

CRUNCHY GIF!

SHOW & TELL

SHIP IT SHIP IT SHIP IT

JOSH HOLTZ 

WHO IS JOSH HOLTZ?

- ▶ fastlane - 2015 to present
 - ▶ Core contributor of fastlane since early 2015
 - ▶ Lead maintainer of fastlane since March 2018
- ▶ RokkinCat - 2011 to present
 - ▶ Software consulting agency focused on new product development

ANOTHER THING ABOUT ME...

I do have a stutter 🙄

I have a fear of public speaking 🙊



PROBLEM

- ▶ Screen recording for GitHub PRs
 - ▶ Only takes images
 - ▶ Under 10MB
- ▶ I'll upload 1 to 4ish GIFs per PR

PREVIOUSLY





SLOW



DIDN'T SCALE



ANNOYING

SO...



LET'S MAKE A MACOS APP!

JOSH'S "SHIP ASAP" PLAN

1. Do **what you** don't know 🤔
2. **Then** "ship it" 🚢
3. **Add** features ✨
4. **Actually** ship it 🚢🚢

WHY SHIP SO SOON?

- ▶ Proof/bookmark of things working
- ▶ Discover any problems while signing and packaging
- ▶ Remaining work is now just code and UX
- ▶ Shipping is now low effort

THE PLAN

1. How to convert video to GIF? 🤔
2. Figure how file input
3. "Ship it" 🚢

RESEARCH .mov TO .gif

- ▶ CocoaPods or Carthage dependency?
- ▶ CLI tool?
- ▶ Anything Stack Overflow could tell me

ffmpeg OUTSIDE OF APP

I did what I know 😊

```
$ ffmpeg -i input.mov -pix_fmt rgb24 output.gif
```

Now... how to use FFMPEG from within the app?

ffmpeg INSIDE OF APP

► Add ffmpeg executable into bundle

```
static func createFFMPEGProcess(arguments: [String], callback: @escaping (Bool) -> Void) -> (Process, DispatchWorkItem)? {  
  
    guard let launchPath = Bundle.main.path(forResource: "ffmpeg", ofType: "") else {  
        print("Cannot find ffmpeg")  
        return nil  
    }  
    let process = Process()  
    let task = DispatchWorkItem {  
        process.launchPath = launchPath  
        process.arguments = arguments  
        process.standardInput = FileHandle.nullDevice  
        process.launch()  
        process.terminationHandler = { process in  
            callback(process.terminationStatus == 0)  
        }  
    }  
    DispatchQueue.global(qos: .userInitiated).async(execute: task)  
  
    return (process, task)  
}
```


CALLING createFFMPEGProcess()

```
let arguments = [  
    "-i",  
    pathIn.absoluteString,  
    "-pix_fmt",  
    "rgb24",  
    pathOut.absoluteString  
]  
  
processMeta = GifTools.createFFMPEGProcess(arguments: arguments) { (terminated) in  
    print("terminated: \(terminated)")  
    done()  
}
```

FILE INPUT

- ▶ **Its terrifying on macOS**
- ▶ **Sandboxed apps require permissions on Catalina**
 - ▶ **Desktop, Downloads, Documents, File System**
- ▶ **Un-sandbox app?**
- ▶ **Wanted easy solution with something user's could trust**

FILE INPUT - DRAG & DROP

- ▶ Don't need to ask permission
- ▶ Can keep app sandboxed
- ▶ Actually makes app friendlier to use



NOW WE SHIP IT

WE HAD ISSUES



ffmpeg **NEEDS ITS OWN ENTITLEMENTS**



WTF, MATE?

ENTITLEMENTS

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>com.apple.security.app-sandbox</key>
    <true/>
    <key>com.apple.security.inherit</key>
    <true/>
</dict>
</plist>
```

MORE ISSUES



ffmpeg EXECUTABLE NEEDS TO BE SIGNED

SIGNING IN BUILD PHASE RUN SCRIPT

```
OPTIONS=""
if [ "${ENABLE_HARDENED_RUNTIME}" == "YES" ]; then
    OPTIONS="--options runtime "
fi

echo "OPTIONS: $OPTIONS"

codesign $OPTIONS-f -s "${EXPANDED_CODE_SIGN_IDENTITY}" \
--entitlements \
"${BUILT_PRODUCTS_DIR}/${EXECUTABLE_NAME}.app/Contents/Resources/ffmpeg.entitlements" \
"${BUILT_PRODUCTS_DIR}/${EXECUTABLE_NAME}.app/Contents/Resources/ffmpeg"
```



ALL "DONE"!

NOW ADD ALL THE FEATURES!

- ▶ Actual UI
- ▶ Drag & Drop out
- ▶ Progress indicator
- ▶ Save FPS and size settings
- ▶ Delete converted GIFs



WE SHIP AGAIN!

DEMO TIME

INTERNET THINGS

- ▶ Josh

- ▶ **Twitter:** @joshdholtz

- ▶ CrunchyGIF

- ▶ **Twitter:** @CrunchyGif

- ▶ **GitHub:** github.com/joshdholtz/crunchygif