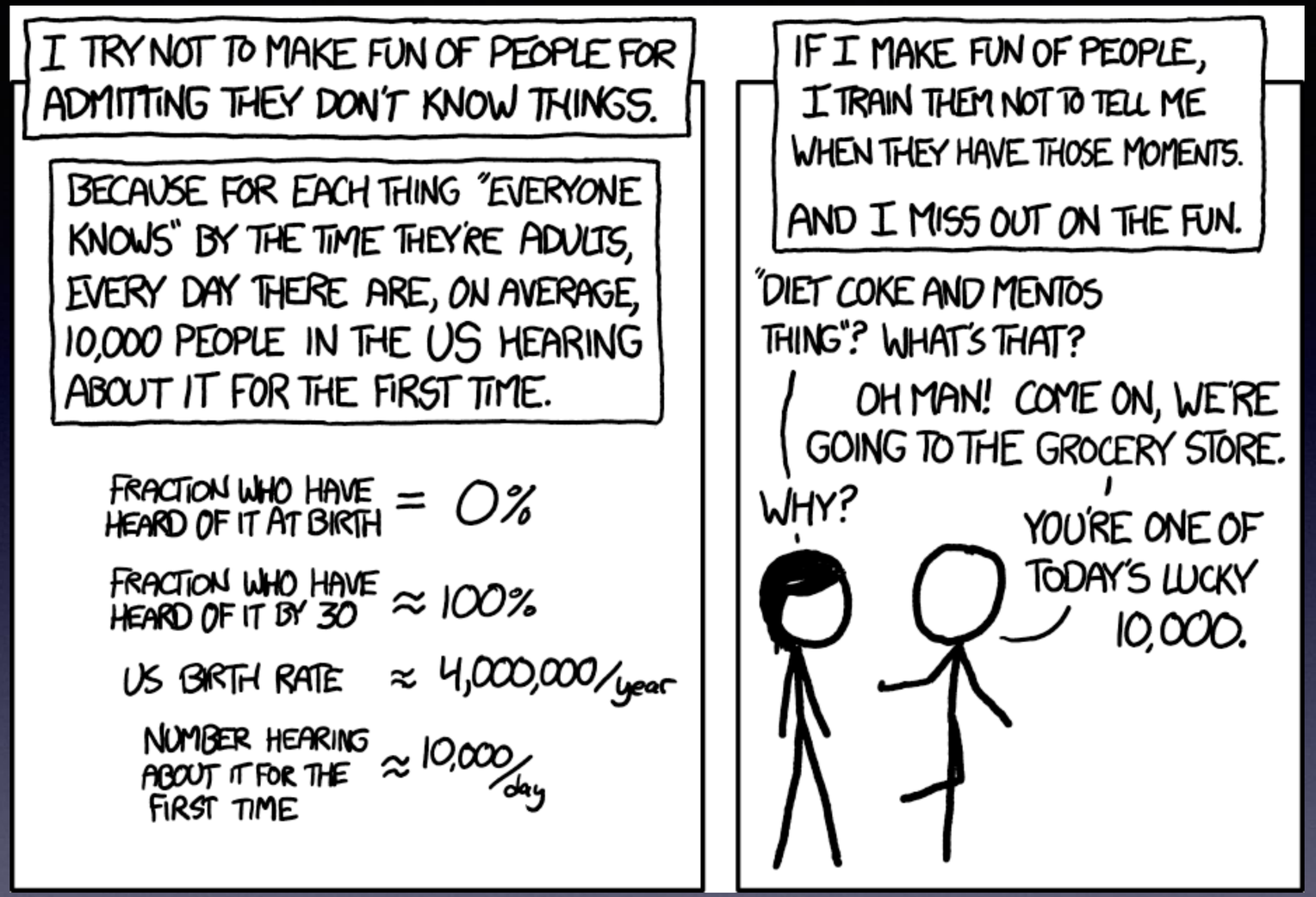


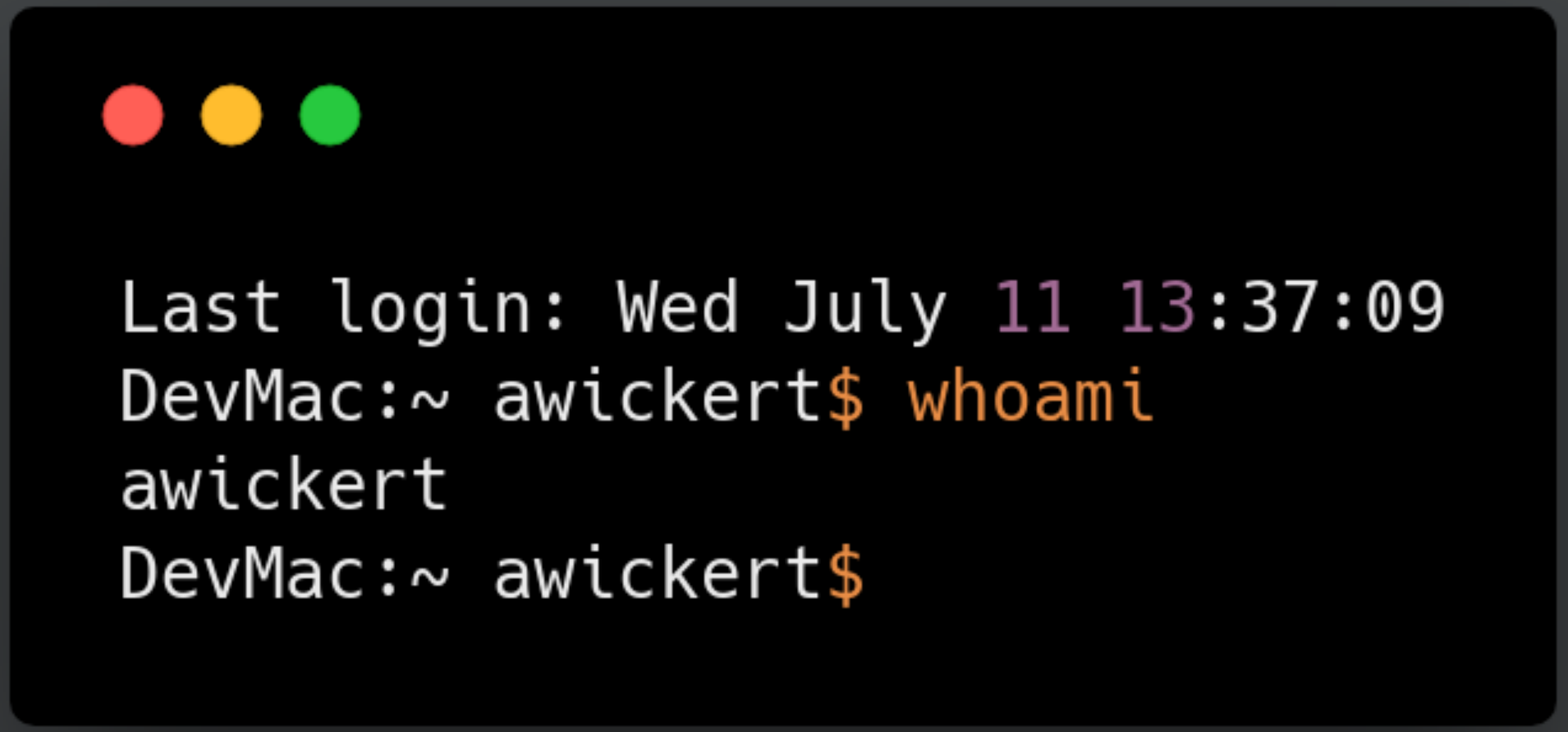
https://www.xkcd.com/1053/

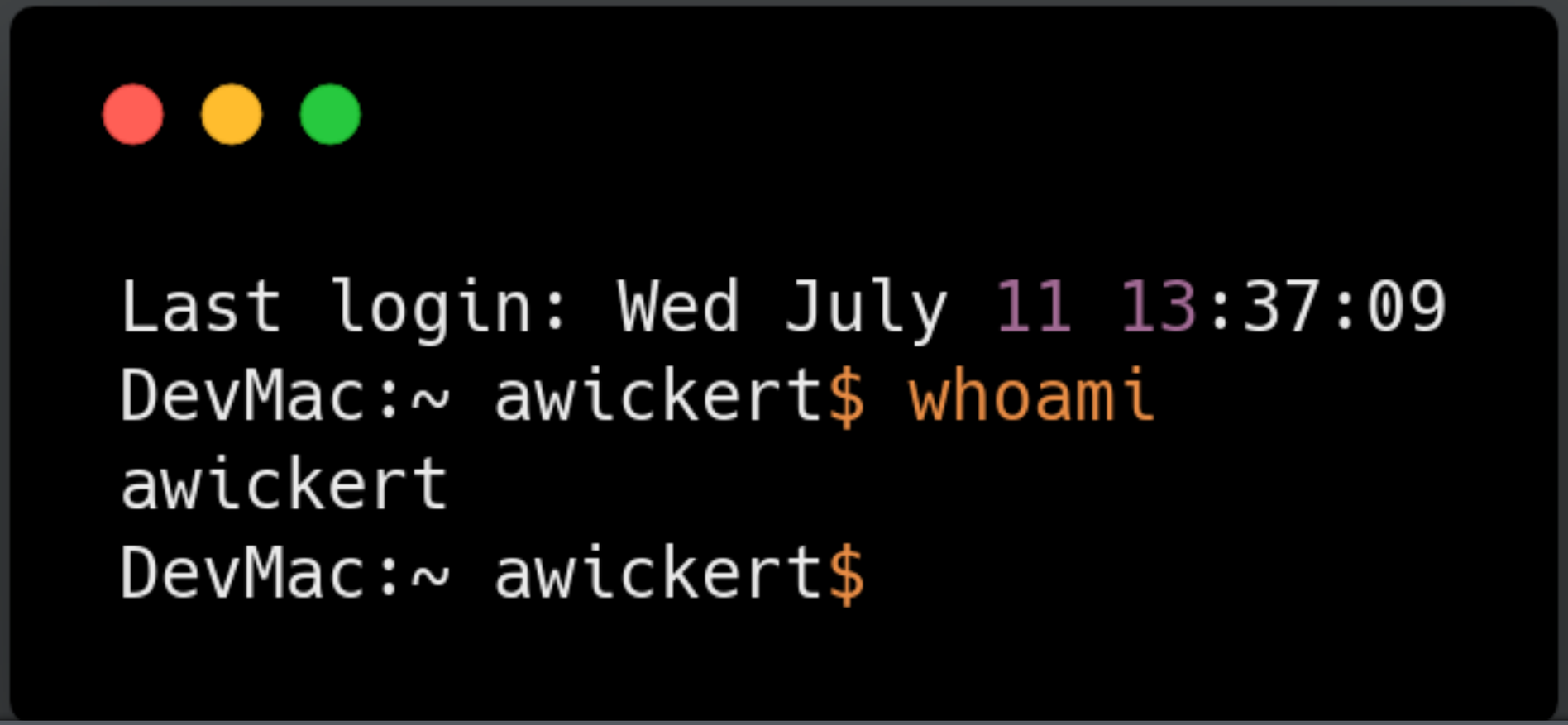
FROM IDEA TO SCRIPT TO APP

Adam Wickert, Desktop Systems Administrator
Hamilton College

A BEGINNER'S JOURNEY IN APP DEVELOPMENT

Adam Wickert, Desktop Systems Administrator
Hamilton College


Last login: Wed July 11 13:37:09
DevMac:~ awickert$ whoami
awickert
DevMac:~ awickert$


Last login: Wed July 11 13:37:09
DevMac:~ awickert$ whoami
awickert
DevMac:~ awickert$

WHO AM I?

WHAT THIS IS NOT

WHAT THIS IS

- The path I took to learning app development

- The path I took to learning app development
- Helpful workflows and tools

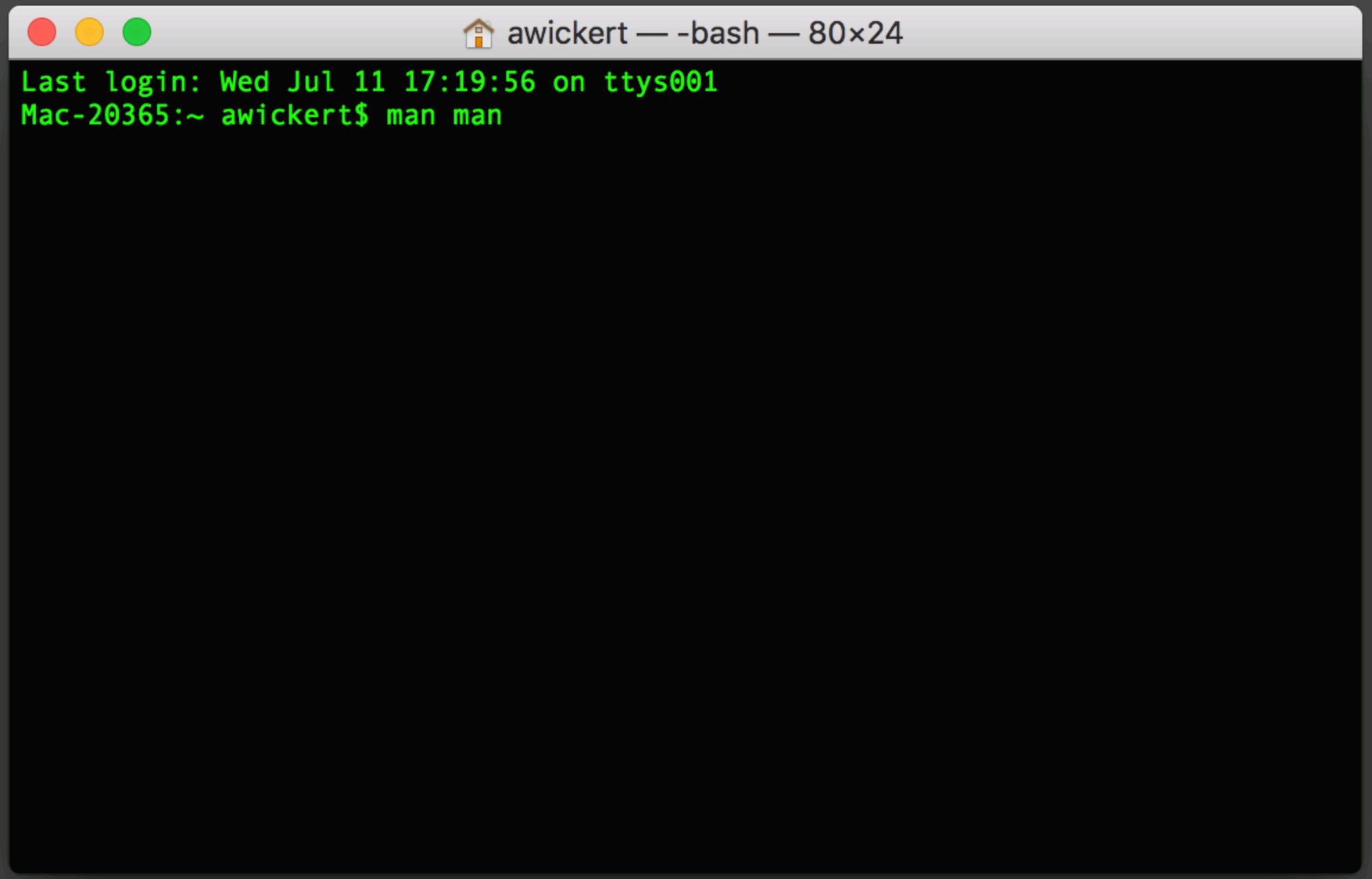
- The path I took to learning app development
- Helpful workflows and tools
- Resources I found helpful

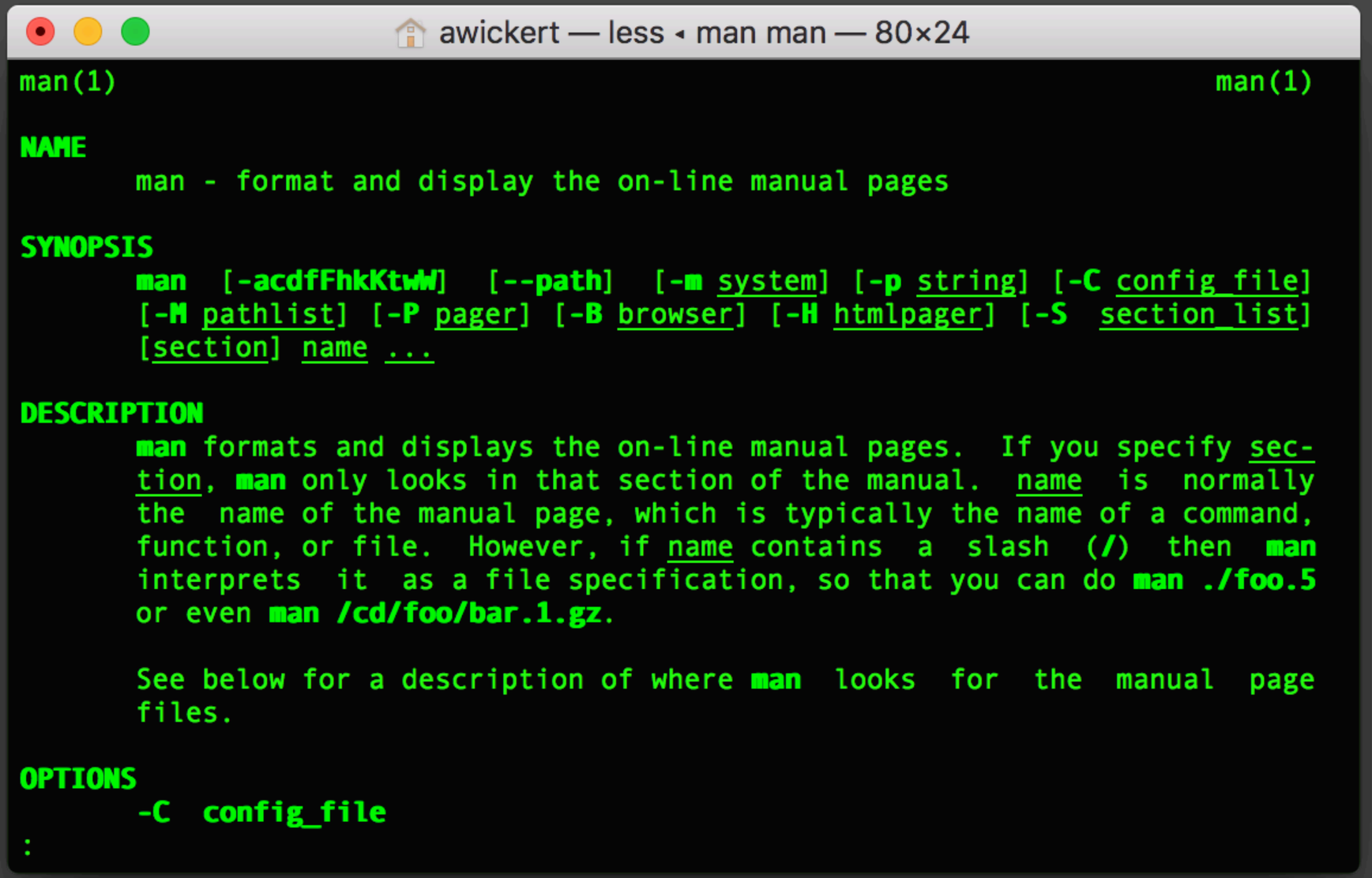
- The path I took to learning app development
- Helpful workflows and tools
- Resources I found helpful
- Some discussion on ideas and methods

WHAT I DO

WHAT I DO

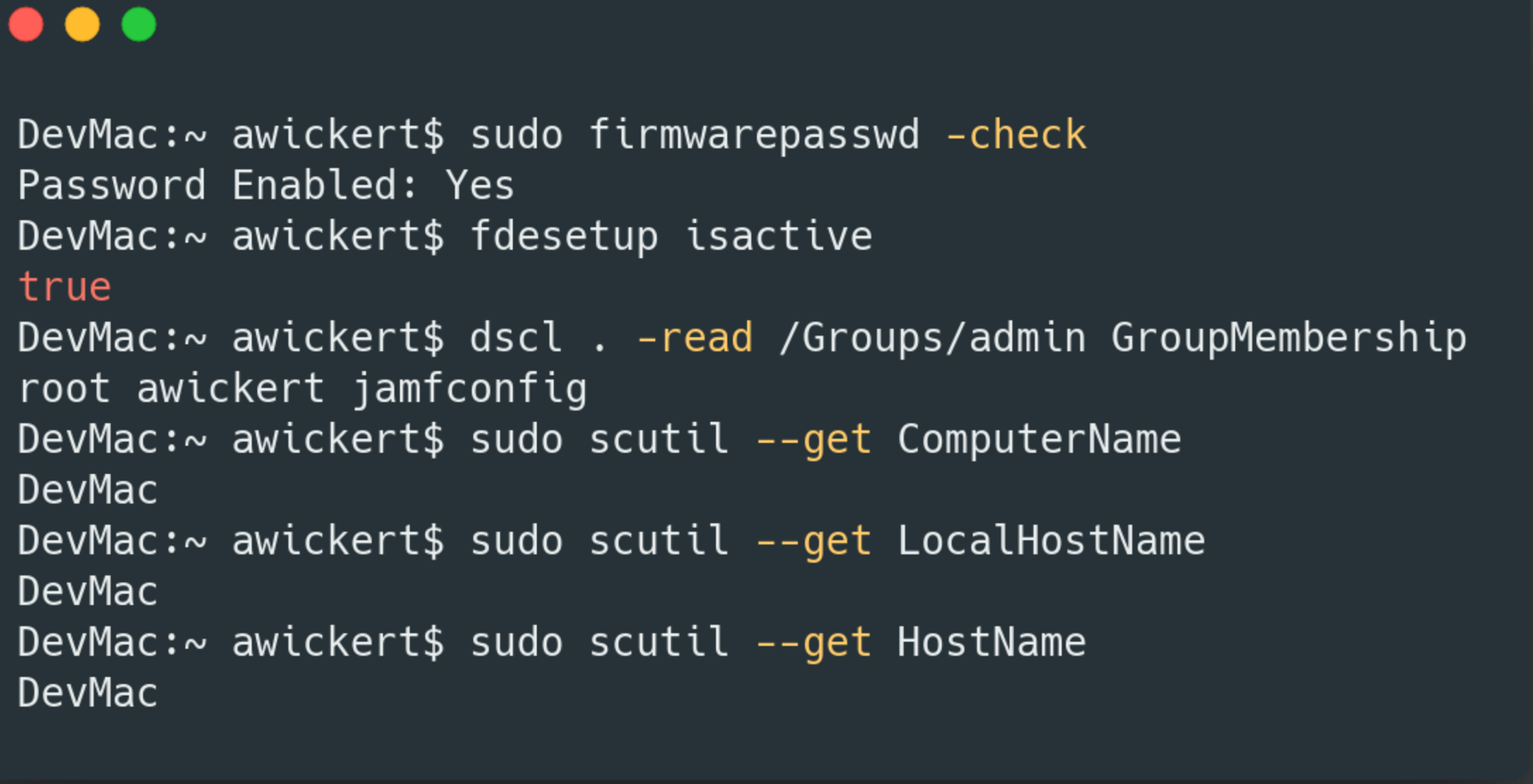
- Configure new systems
  - Package software
  - Write scripts
  - Create profiles
  - Manage deployments
  - Transfer data
- Lab Management
- Inventory
- Documentation
- Software Installation/Management


awickert — -bash — 80×24
Last login: Wed Jul 11 17:19:56 on ttys001
Mac-20365:~ awickert$ man man


awickert — less ◂ man man — 80×24
man(1) man(1)
NAME
man - format and display the on-line manual pages
SYNOPSIS
man [-acdfFhkKtwW] [--path] [-m system] [-p string] [-C config_file]
[-M pathlist] [-P pager] [-B browser] [-H htmlpager] [-S section_list]
[section] name ...
DESCRIPTION
man formats and displays the on-line manual pages. If you specify sec-
tion, man only looks in that section of the manual. name is normally
the name of the manual page, which is typically the name of a command,
function, or file. However, if name contains a slash (/) then man
interprets it as a file specification, so that you can do man ./foo.5
or even man /cd/foo/bar.1.gz.
See below for a description of where man looks for the manual page
files.
OPTIONS
-C config_file
:

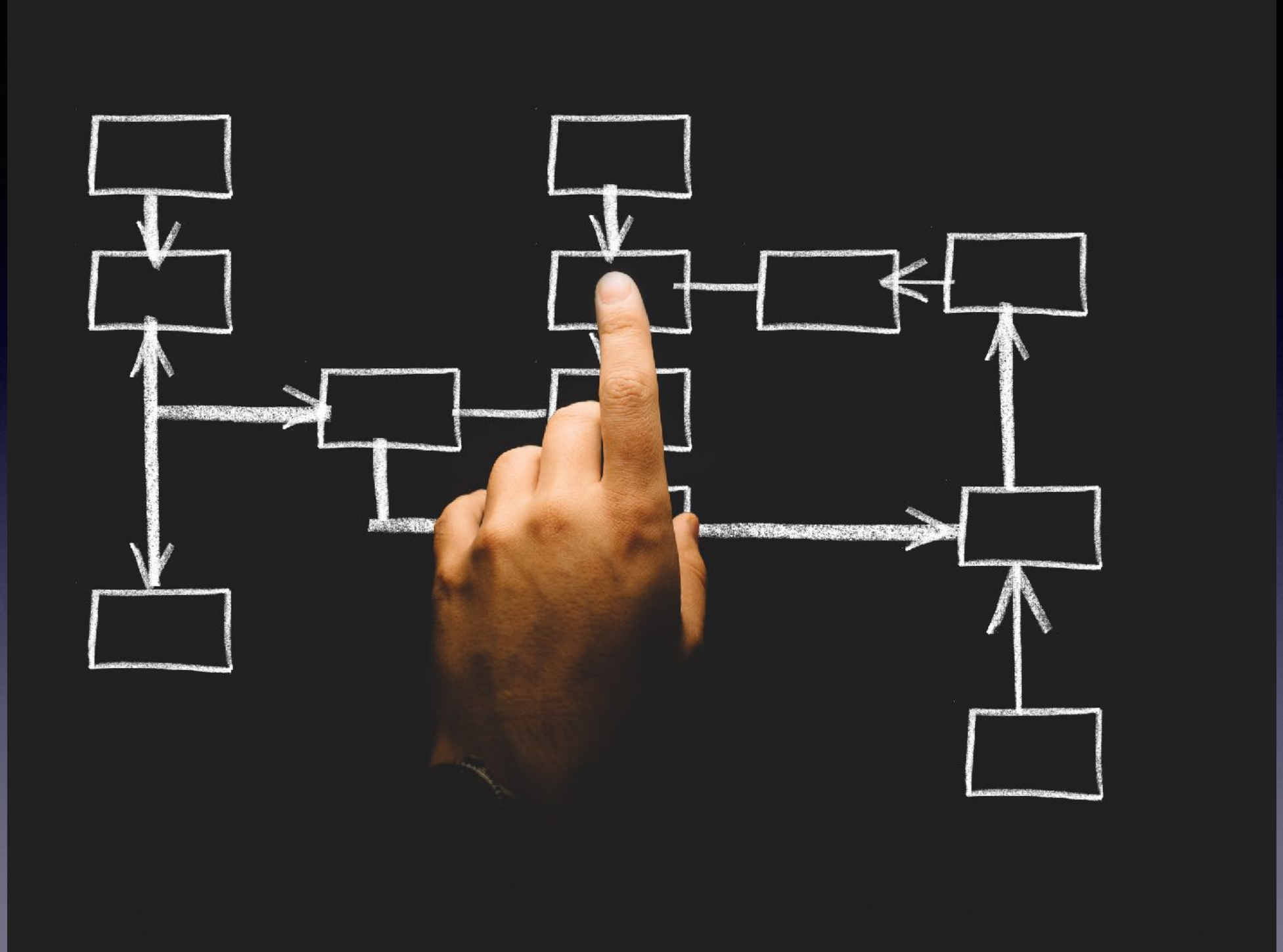
IDEA

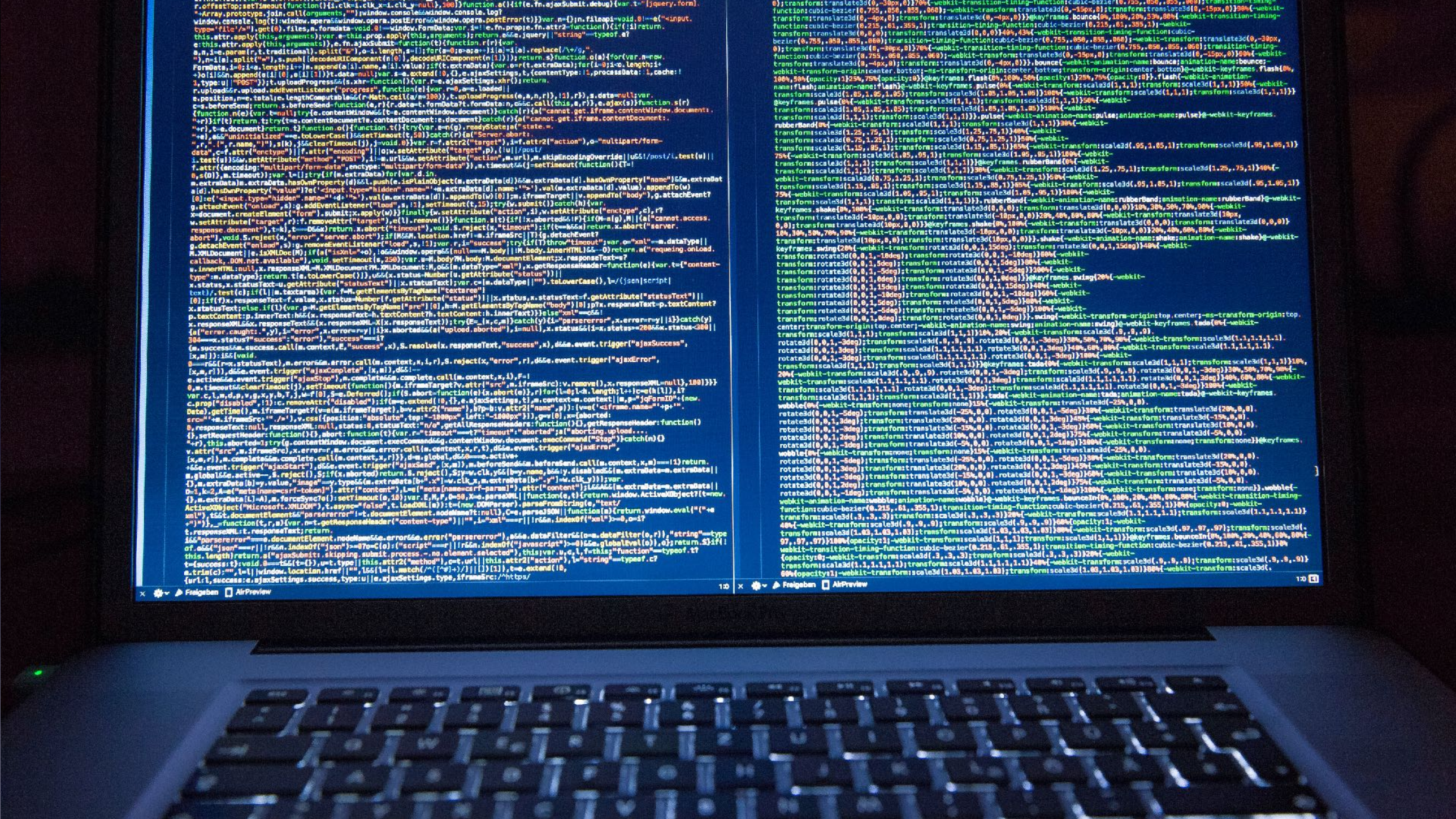
DevMac:~ awickert$ sudo firmwarepasswd -setpasswd
Enter new password:
Re-enter new password:
DevMac:~ awickert$ sudo fdesetup enable -user adminuser
DevMac:~ awickert$ dscl . -append /Groups/admin GroupMembership adminuser
DevMac:~ awickert$ sudo scutil --set ComputerName DevMac
DevMac:~ awickert$ sudo scutil --set LocalHostName DevMac
DevMac:~ awickert$ sudo scutil --set HostName DevMac

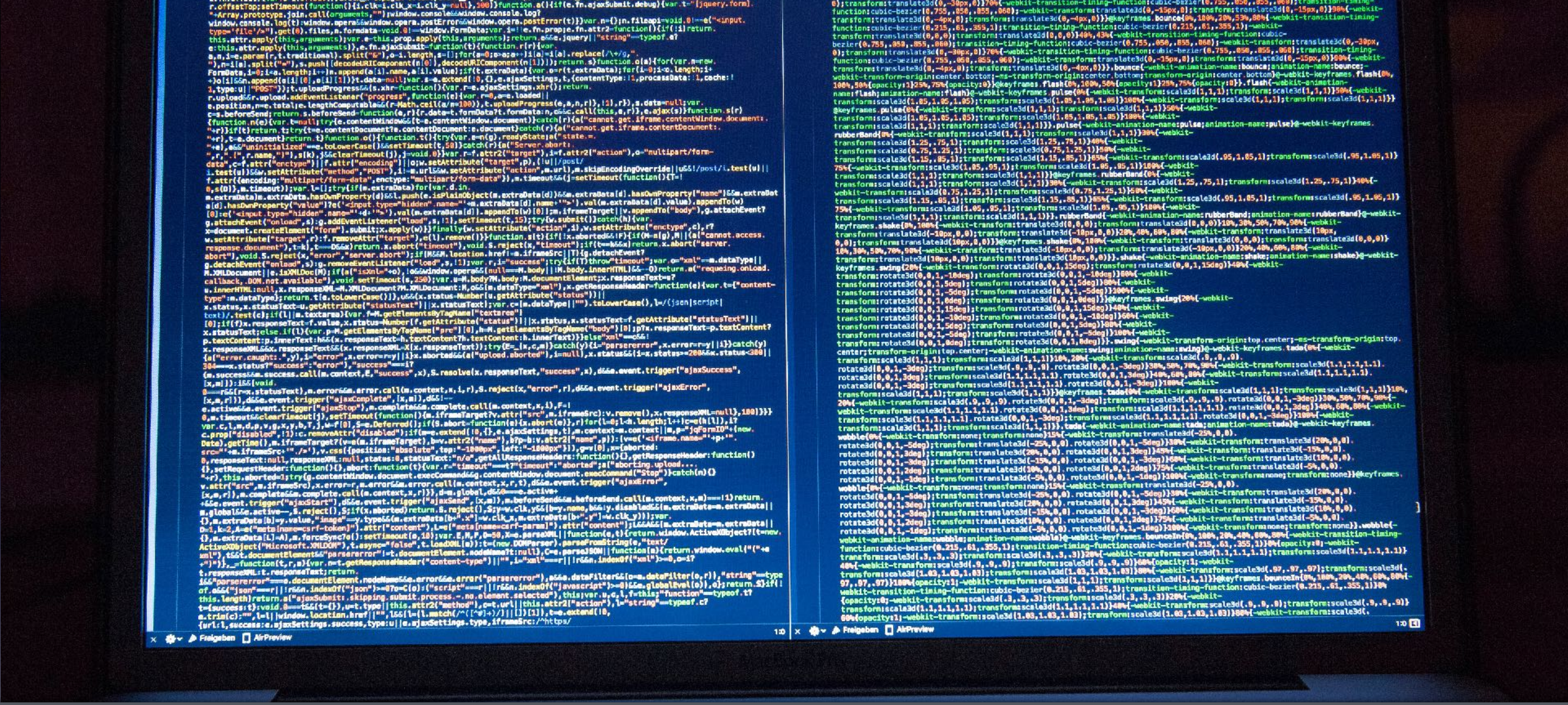

DevMac:~ awickert$ sudo firmwarepasswd -check
Password Enabled: Yes
DevMac:~ awickert$ fdesetup isactive
true
DevMac:~ awickert$ dscl . -read /Groups/admin GroupMembership
root awickert jamfconfig
DevMac:~ awickert$ sudo scutil --get ComputerName
DevMac
DevMac:~ awickert$ sudo scutil --get LocalHostName
DevMac
DevMac:~ awickert$ sudo scutil --get HostName
DevMac

THERE'S GOT TO BE A BETTER WAY

THERE'S GOT TO BE A BETTER WAY







SCRIPT

computersettings.sh — Free Mode

~/Desktop/Beginner Journey to App Development/computersettings.sh

```
#!/bin/bash
# 2018-07-11 awickert
# set basic system settings

#set local admin
dscl . - append /Groups/admin GroupMembership $adminuser

#set hostname
scutil --set ComputerName $computername
scutil --set HostName $computername
scutil --set LocalHostName $computername

#set firmware password
doesexist=`firmwarepasswd -check`

if [ "$doesexist" = "Password Enabled: No" ]; then

/usr/bin/expect <<- DONE
    spawn firmwarepasswd -setpasswd
    expect "Enter new password:"
    send "$firmware\r";
    expect "Re-enter new password:"
    send "$firmware\r";
    expect EOF
DONE

else
    echo "Firmware Password Already Exists"

fi
```

L: 30 C: 3 Unix Shell Script Unicode (UTF-8) Unix (LF) Saved: 7/7/18, 9:54:34 AM 611 / 77 / 30

computersettings.sh — Free Mode

~/Desktop/Beginner Journey to App Development/computersettings.sh

```
1  #!/bin/bash
2  # 2018-07-11 awickert
3  # set basic system settings
4
5  #set local admin
6  dscl . - append /Groups/admin GroupMembership $adminuser
7
8  #set hostname
9  scutil --set ComputerName $computername
10 scutil --set HostName $computername
11 scutil --set LocalHostName $computername
12
13 #set firmware password
14 doesexist=`firmwarepasswd -check`
15
16 if [ "$doesexist" = "Password Enabled: No" ]; then
17
18 /usr/bin/expect <<- DONE
19     spawn firmwarepasswd -setpasswd
20     expect "Enter new password:"
21     send "$firmware\r";
22     expect "Re-enter new password:"
23     send "$firmware\r";
24     expect EOF
25 DONE
26
27 else
28     echo "Firmware Password Already Exists"
29
30 fi
```

L: 30 C: 3 Unix Shell Script Unicode (UTF-8) Unix (LF) Saved: 7/7/18, 9:54:34 AM 611 / 77 / 30

computersettings.sh — Free Mode

~/Desktop/Beginner Journey to App Development/computersettings.sh

```
#!/bin/bash
# 2018-07-11 awickert
# set basic system settings

#set local admin
dscl . - append /Groups/admin GroupMembership $adminuser

#set hostname
scutil --set ComputerName $computername
scutil --set HostName $computername
scutil --set LocalHostName $computername

#set firmware password
doesexist=`firmwarepasswd -check`

if [ "$doesexist" = "Password Enabled: No" ]; then

/usr/bin/expect <<- DONE
    spawn firmwarepasswd -setpasswd
    expect "Enter new password:"
    send "$firmware\r";
    expect "Re-enter new password:"
    send "$firmware\r";
    expect EOF
DONE

else
    echo "Firmware Password Already Exists"

fi
```

L: 30 C: 3 Unix Shell Script Unicode (UTF-8) Unix (LF) Saved: 7/7/18, 9:54:34 AM 611 / 77 / 30

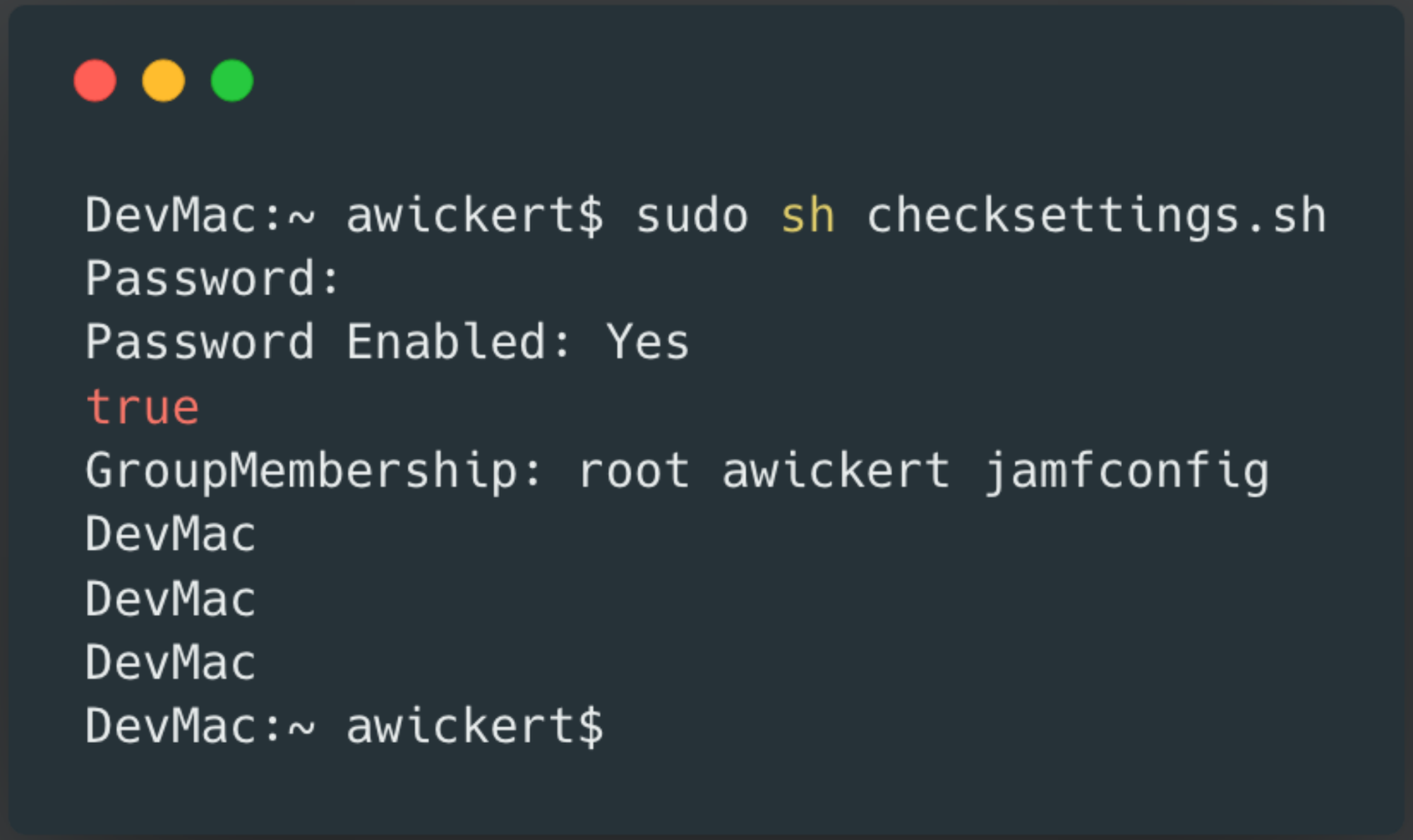
checksettings.sh

Free Mode

~/Desktop/Beginner Journey to App Development/checksettings.sh

```
1   #!/bin/bash
2   # 2018-07-11
3   # Script to check security settings and name
4
5   # check if firmware password is active
6   firmwarepasswd -check
7
8   # check to see if filevault is active
9   fdesetup isactive
10
11  # check admin users
12  dscl . -read /Groups/admin GroupMembership
13
14  #check computer name
15  scutil --get ComputerName
16  scutil --get LocalHostName
17  scutil --get HostName
```

L: 12 C: 5 Unix Shell Script Unicode (UTF-8) Unix (LF) Saved: 7/7/18, 5:35:12 PM 352 / 49 /...


DevMac:~ awickert$ sudo sh checksettings.sh
Password:
Password Enabled: Yes
true
GroupMembership: root awickert jamfconfig
DevMac
DevMac
DevMac
DevMac:~ awickert$

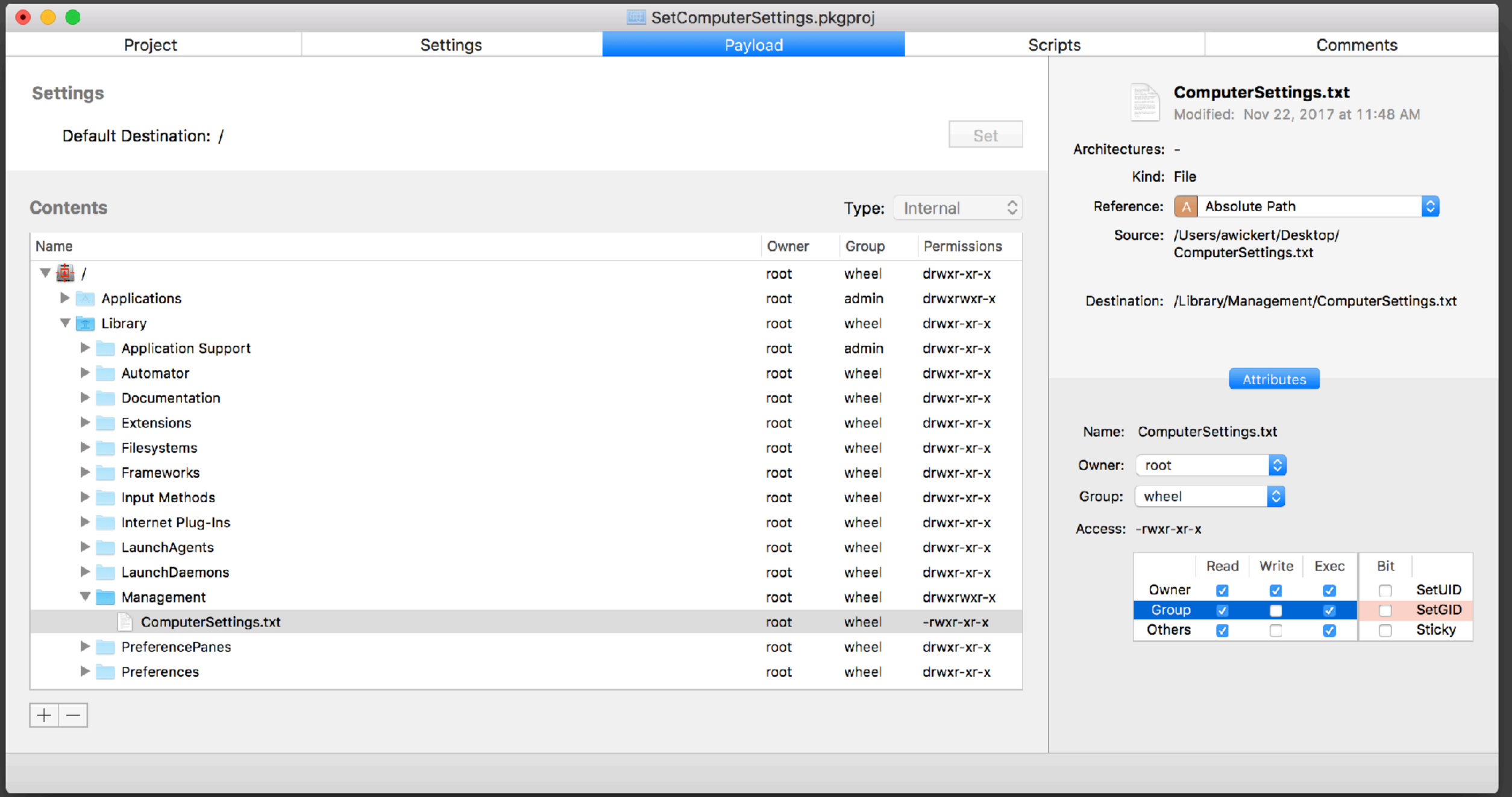
checksettings2.sh

Free Mode

~/Desktop/Beginner Journey to App Development/Project Files/checksettings2.sh master

```
#!/bin/bash
# 2018-07-11
# Script to check security settings and name

# check if firmware password is active
echo "Firmware" `firmwarepasswd -check`

# check to see if filevault is active
echo "FileVault Enabled:" `fdesetup isactive`

# check admin users
adminusers=`dscl . -read /Groups/admin GroupMembership`
echo "Admin Users:" ${adminusers:(+16)}

#check computer name
echo "Computer Name:" `scutil --get ComputerName`
echo "Local Host Name:" `scutil --get LocalHostName`
echo "Host Name:" `scutil --get HostName`

exit 0
```

L: 6 C: 1 Unix Shell Script Unicode (UTF-8) Unix (LF) Saved: 7/7/18, 6:24:03 PM 526 / 71 / 20 100%


SetComputerSettings.pkgproj
Project
Settings
Payload
Scripts
Comments
Settings
Default Destination: /
Set
Contents
Type: Internal
Name
Owner
Group
Permissions
/ root wheel drwxr-xr-x
Applications root admin drwxrwxr-x
Library root wheel drwxr-xr-x
Application Support root admin drwxr-xr-x
Automator root wheel drwxr-xr-x
Documentation root wheel drwxr-xr-x
Extensions root wheel drwxr-xr-x
Filesystems root wheel drwxr-xr-x
Frameworks root wheel drwxr-xr-x
Input Methods root wheel drwxr-xr-x
Internet Plug-Ins root wheel drwxr-xr-x
LaunchAgents root wheel drwxr-xr-x
LaunchDaemons root wheel drwxr-xr-x
Management root wheel drwxrwxr-x
ComputerSettings.txt root wheel -rwxr-xr-x
PreferencePanes root wheel drwxr-xr-x
Preferences root wheel drwxr-xr-x
ComputerSettings.txt
Modified: Nov 22, 2017 at 11:48 AM
Architectures: -
Kind: File
Reference: Absolute Path
Source: /Users/awickert/Desktop/ComputerSettings.txt
Destination: /Library/Management/ComputerSettings.txt
Attributes
Name: ComputerSettings.txt
Owner: root
Group: wheel
Access: -rwxr-xr-x
Read
Write
Exec
Bit
Owner
Group
Others
SetUID
SetGID
Sticky

SetComputerSettings.pkgproj

Project | Settings | Payload | Scripts | Comments

Pre-installation

No Script Set
Modified: -

A Choose...

Post-installation

computersettings.sh
Modified: Jul 7, 2018 at 9:54 AM

A Choose... Remove...

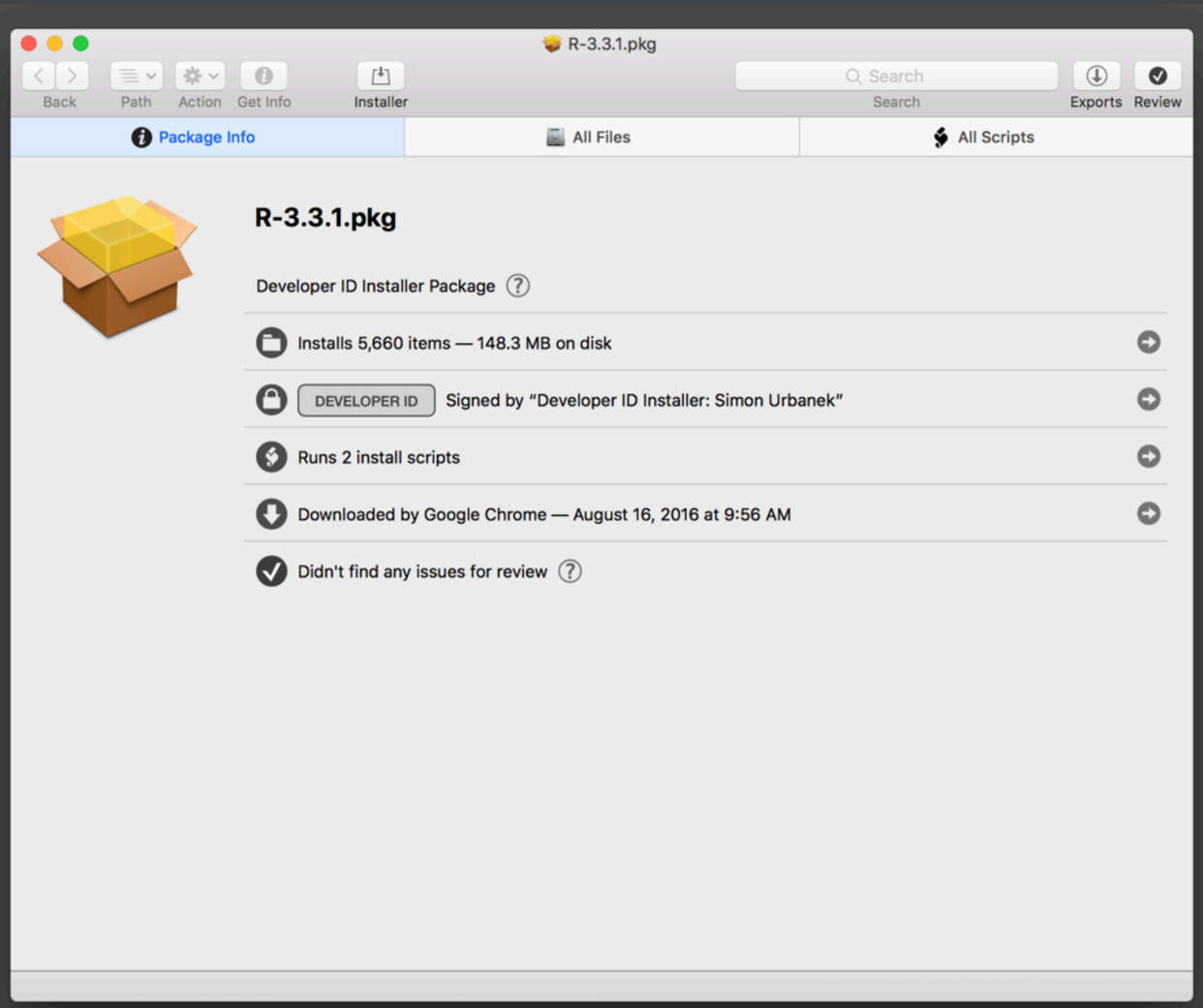
Additional Resources

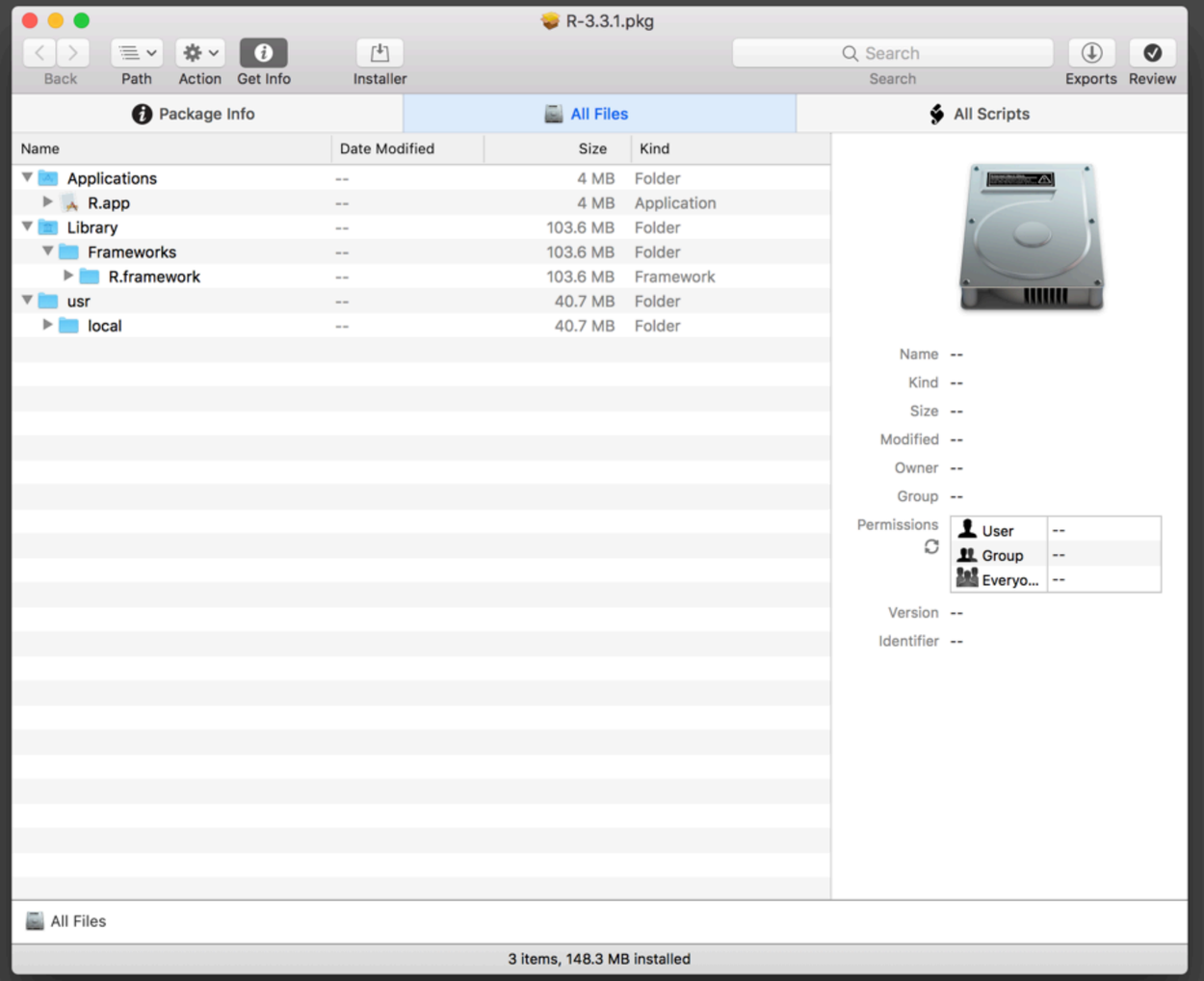
Name

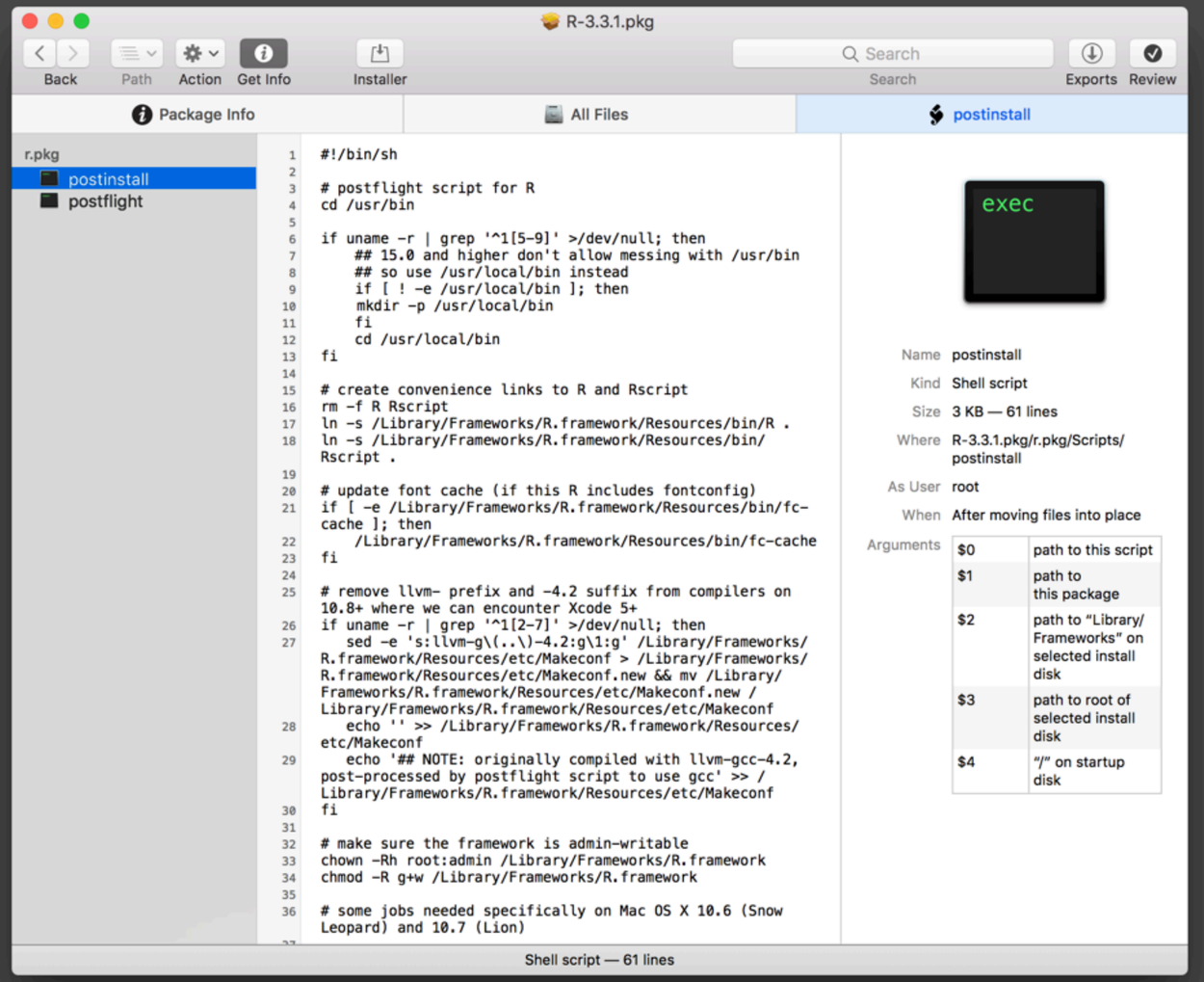
+ −

These resources can be used by the pre and post-installation scripts.

Empty Selection


R-3.3.1.pkg
Back
Path
Action
Get Info
Installer
Search
Search
Exports
Review
Package Info
All Files
All Scripts
R-3.3.1.pkg
Developer ID Installer Package
Installs 5,660 items — 148.3 MB on disk
DEVELOPER ID
Signed by "Developer ID Installer: Simon Urbanek"
Runs 2 install scripts
Downloaded by Google Chrome — August 16, 2016 at 9:56 AM
Didn't find any issues for review


R-3.3.1.pkg
Back
Path
Action
Get Info
Installer
Search
Search
Exports
Review
Package Info
All Files
All Scripts
Name
Date Modified
Size
Kind
Applications -- 4 MB Folder
R.app -- 4 MB Application
Library -- 103.6 MB Folder
Frameworks -- 103.6 MB Folder
R.framework -- 103.6 MB Framework
usr -- 40.7 MB Folder
local -- 40.7 MB Folder
Name --
Kind --
Size --
Modified --
Owner --
Group --
Permissions
User --
Group --
Everyo... --
Version --
Identifier --
All Files
3 items, 148.3 MB installed


R-3.3.1.pkg
Back
Path
Action
Get Info
Installer
Search
Search
Exports
Review
Package Info
All Files
postinstall
r.pkg
postinstall
postflight
#!/bin/sh
postflight script for R
cd /usr/bin
if uname -r | grep '^1[5-9]' >/dev/null; then
15.0 and higher don't allow messing with /usr/bin
so use /usr/local/bin instead
if [! -e /usr/local/bin]; then
mkdir -p /usr/local/bin
fi
cd /usr/local/bin
fi
create convenience links to R and Rscript
rm -f R Rscript
ln -s /Library/Frameworks/R.framework/Resources/bin/R .
ln -s /Library/Frameworks/R.framework/Resources/bin/Rscript .
# update font cache (if this R includes fontconfig)
if [-e /Library/Frameworks/R.framework/Resources/bin/fc-cache]; then
/Library/Frameworks/R.framework/Resources/bin/fc-cache
fi
# remove llvm- prefix and -4.2 suffix from compilers on 10.8+ where we can encounter Xcode 5+
if uname -r | grep '^1[2-7]' >/dev/null; then
sed -e 's:llvm-g\(..\)-4.2:g\1:g' /Library/Frameworks/R.framework/Resources/etc/Makeconf > /Library/Frameworks/R.framework/Resources/etc/Makeconf.new && mv /Library/Frameworks/R.framework/Resources/etc/Makeconf.new /Library/Frameworks/R.framework/Resources/etc/Makeconf
echo '' >> /Library/Frameworks/R.framework/Resources/etc/Makeconf
echo '## NOTE: originally compiled with llvm-gcc-4.2, post-processed by postflight script to use gcc' >> /Library/Frameworks/R.framework/Resources/etc/Makeconf
fi
make sure the framework is admin-writable
chown -Rh root:admin /Library/Frameworks/R.framework
chmod -R g+w /Library/Frameworks/R.framework
some jobs needed specifically on Mac OS X 10.6 (Snow Leopard) and 10.7 (Lion)
exec
Name postinstall
Kind Shell script
Size 3 KB — 61 lines
Where R-3.3.1.pkg/r.pkg/Scripts/postinstall
As User root
When After moving files into place
Arguments
$0 path to this script
$1 path to this package
$2 path to "Library/Frameworks" on selected install disk
$3 path to root of selected install disk
$4 "/" on startup disk
Shell script — 61 lines

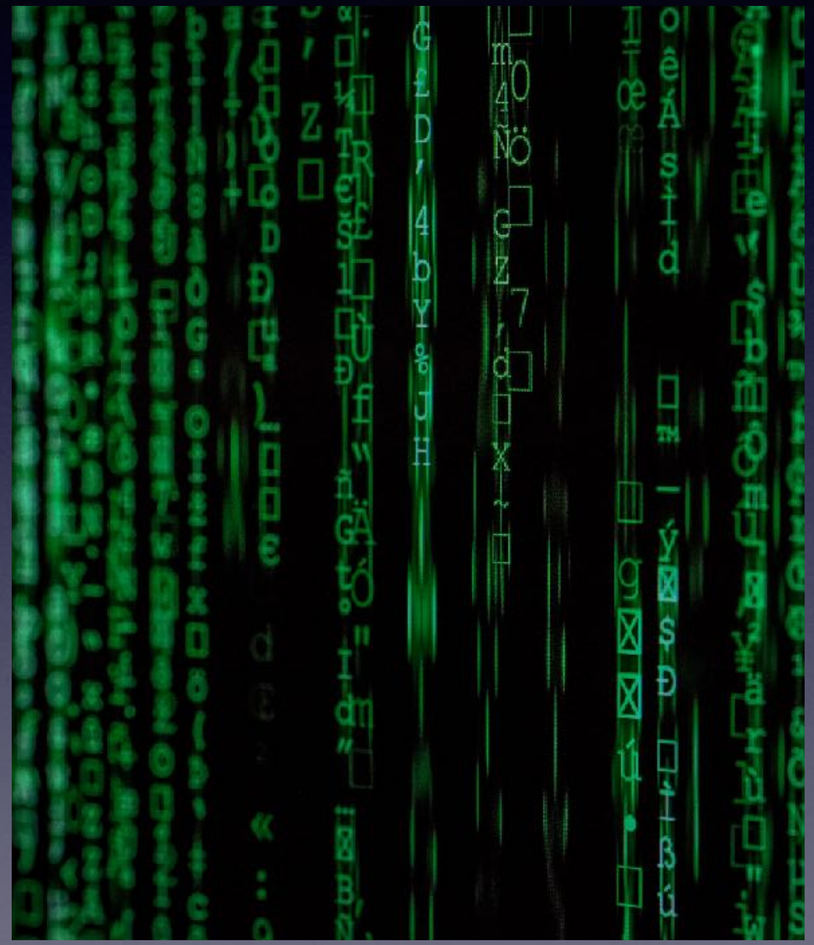
jamf | NATION

Join **MacAdmins** on Slack.

109 users online now of **18766** registered.

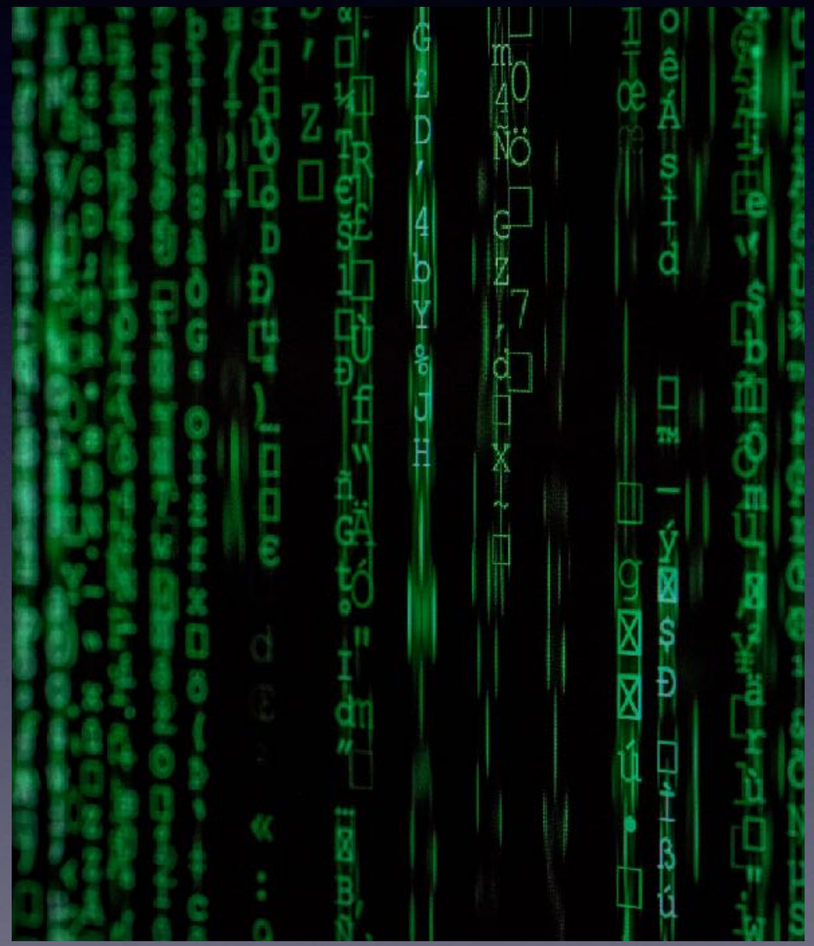
APP

LANGUAGES AND LINKS

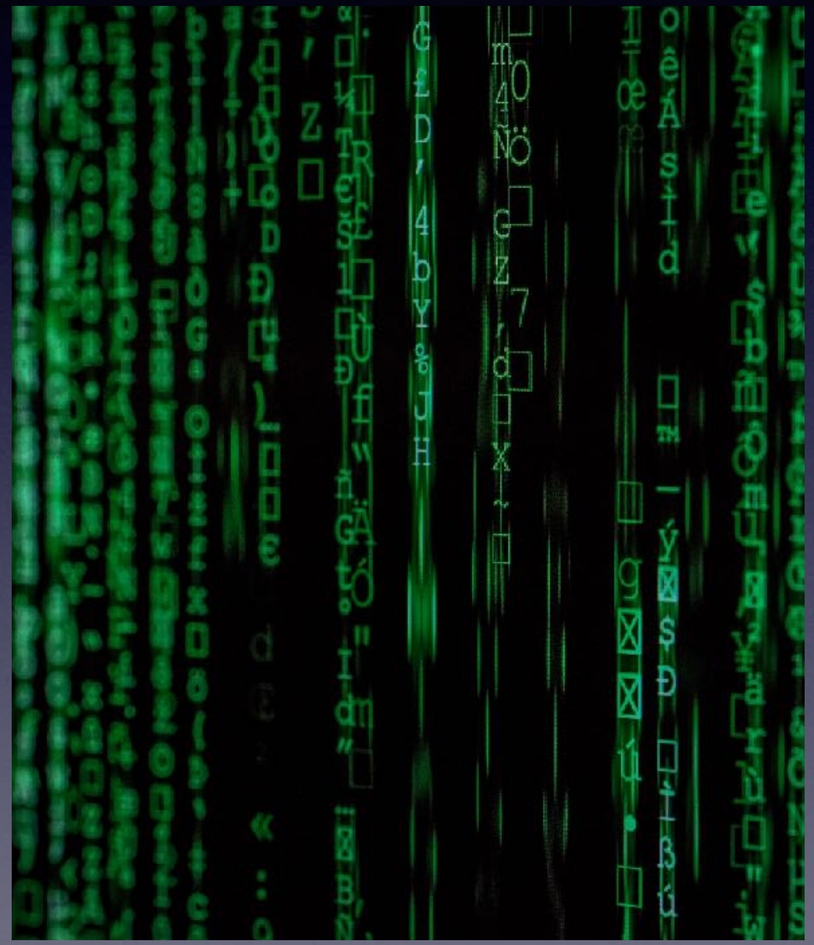


LANGUAGES AND LINKS

- Python
- Swift
- C/Objective C/C++
- Ruby/Ruby on Rails
- JavaScript
  - node.js
  - react.js
- Perl, PHP, C#, Java, SQL, and more

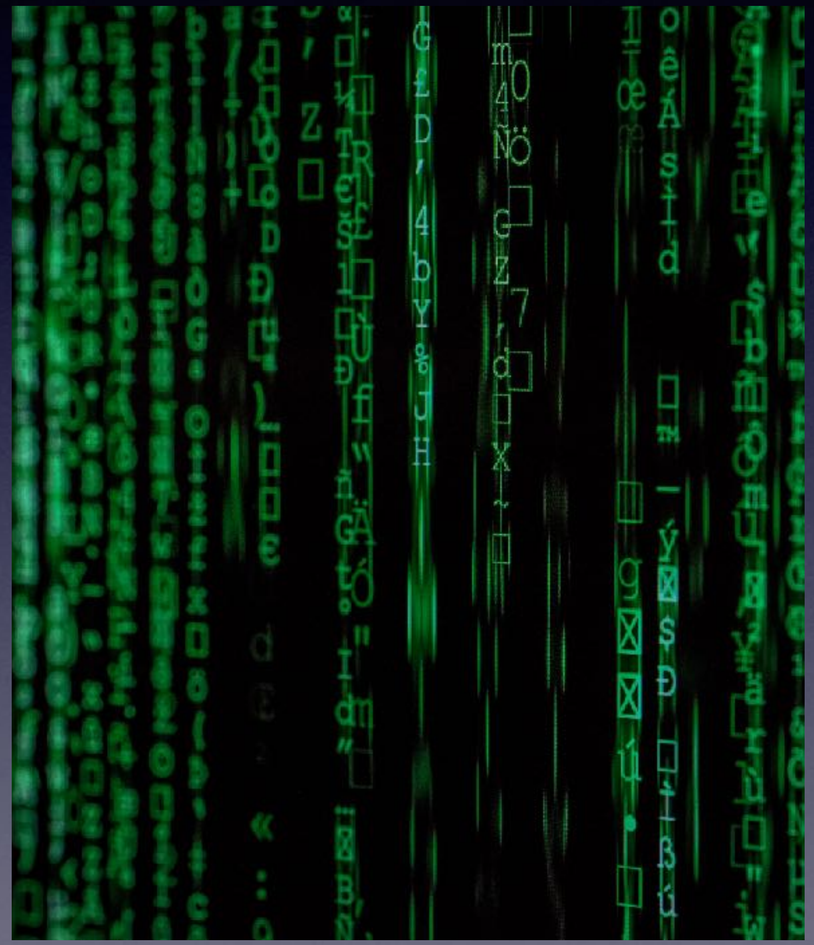


LANGUAGES AND LINKS



LANGUAGES AND LINKS

- edx.org
- codecademy.com
- hackingwithswift.com
- youtube.com/macadmins
- udemy.com
- learnpython.org
- swift playgrounds
- dev.tube and so many more



Craft Your Own GUIs with Python and Tkinter

Penn State MacAdmins 2016

Craft Your Own GUIs with Python and Tkinter

Penn State MacAdmins 2016

https://youtu.be/Wb1YFgHqUZ8

ONE-DAY RESOLUTIONS

codecademy

People learned a lot { Python } in 2017.

Python Software Foundation

Python | PSF | Docs | PyPI | Jobs | Community

python™

Search GO Socialize

About | Downloads | Documentation | Community | Success Stories | News | Events

Python is powerful... and fast;
plays well with others;
runs everywhere;
is friendly & easy to learn;
is Open.

These are some of the reasons people who use Python would rather not use anything else.

Getting Started

Python can be easy to pick up whether you're a first time programmer or you're experienced with other languages. The following pages are a useful first step to get on your way writing programs with Python!

- Beginner's Guide, Programmers
- Beginner's Guide, Non-Programmers
- Beginner's Guide, Download & Installation
- Code sample and snippets for Beginners

Friendly & Easy to Learn

The community hosts conferences and meetups, collaborates on code, and much more. Python's documentation will help you along the way, and the mailing lists will keep you in touch.

- Conferences and Workshops
- Python Documentation
- Mailing Lists and IRC channels

ONE-DAY RESOLUTIONS

codecademy

People learned a lot

{ Py }

in 2017.

GitHub, Inc.

Appify UI

Create Mac apps.
Use HTML5 for the UI.
Script it with anything.
Can not possibly be simpler.

What is this?

A Mac app is essentially just an executable file in a folder along with a config file. That's all that is required.

An Appify UI app is

1. A folder structure
    - That conforms to the Cocoa Application Bundle standard
2. A config file
3. A shell script
4. A compiled binary
    - To load the interface
5. An interface
    - A compiled nib file with a single WebKit WebView
6. A url

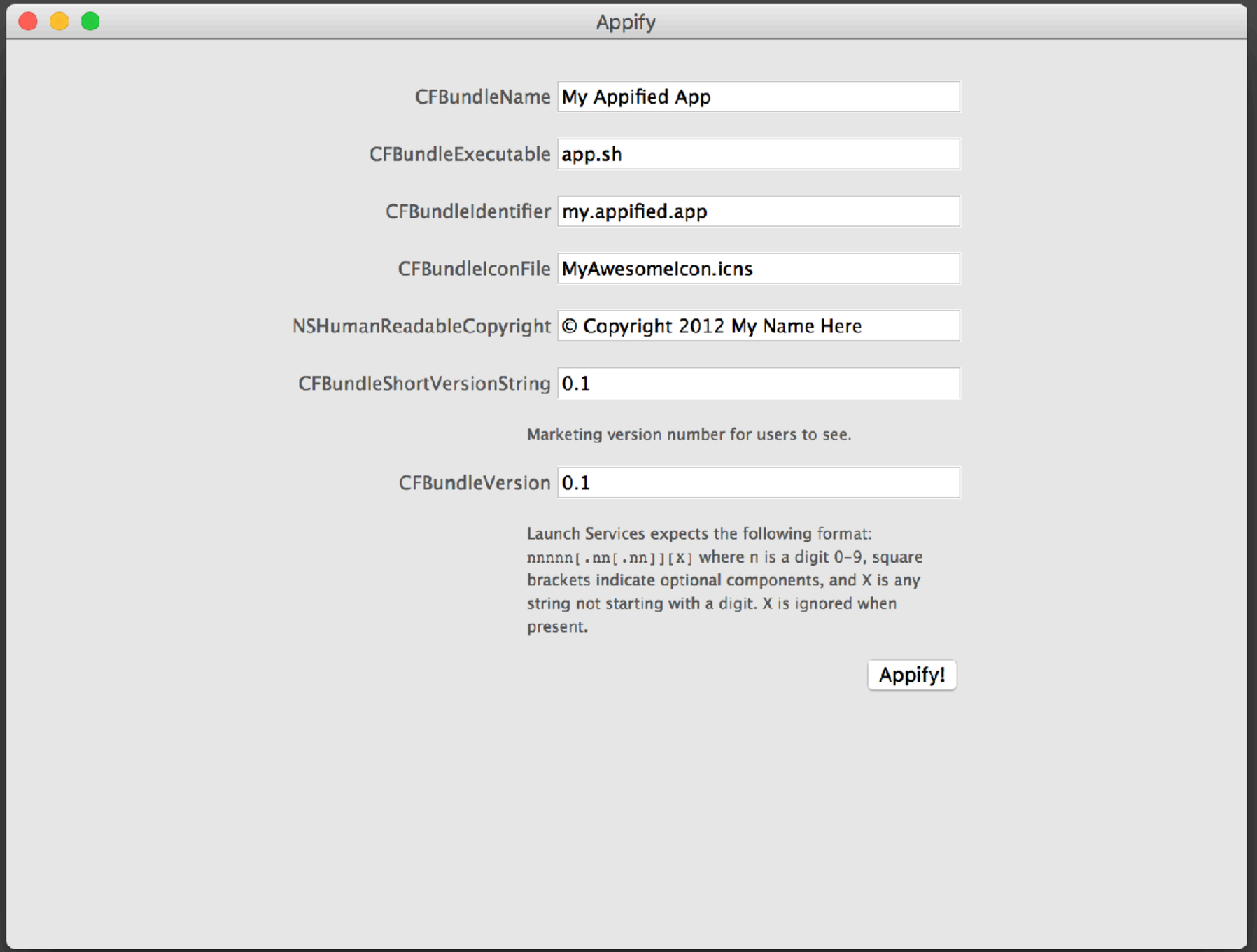
Appify UI.app

A shell script that accepts arguments from an HTML form.

It creates a new Appify UI app on your Desktop with the configuration you provide.

The UI could be a lot better. Pull requests eagerly accepted.

Appify UI Node Demo.app


Appify
CFBundleName
My Appified App
CFBundleExecutable
app.sh
CFBundleIdentifier
my.appified.app
CFBundleIconFile
MyAwesomeIcon.icns
NSHumanReadableCopyright
© Copyright 2012 My Name Here
CFBundleShortVersionString
0.1
Marketing version number for users to see.
CFBundleVersion
0.1
Launch Services expects the following format: nnnnn[.nn[.nn]][X] where n is a digit 0–9, square brackets indicate optional components, and X is any string not starting with a digit. X is ignored when present.
Appify!

4. Make sure that `My Awesome App.app/Contents/Resources/app/server.js` exports something with a `listen` method

How to package a Node.js Mac app for distribution

You could send it around as-is. By default it'll open their web browser and prompt them to install node.js if it's not already installed.

You could probly also package the `node` binary in the app. I haven't tried this, so please update this README once you do.

How to modify the `nib` and cocoa binary

The current version uses a heavily modified version of Apache Callback Mac (formerly PhoneGap-mac / MacGap). It's about as simple as you can get.

I may update this section later.

Similar Projects

https://github.com/maccman/macgap is probably what you should be using.

https://github.com/rogerwang/node-webkit is awesome, but large.

https://github.com/sveinbjornt/Platypus makes an app from terminal script, wraps it's output in the GUI window, handles `CMD` + `Q` etc

GitHub, Inc.

Home

jeff-h edited this page on May 31, 2014 · 17 revisions

The MacGap project provides HTML/JS/CSS developers an Xcode project for developing Native OSX Apps that run in OSX's WebView and take advantage of WebKit technologies.

The project exposes a JavaScript API for OS X integration, such as displaying native OS X 10.9 notifications. The MacGap project is extremely lightweight and nimble; a blank application is about 980KB.

Features:

- tiny compiled app sizes
- Mac App Store compatible
- access to many Mac OS X-specific features

For JavaScript developers.

- API reference and how-to's
- Switching on Developer Tools
- Roadmap
- MacGap 2 planning
- Offline Patterns
- Custom Build

Pages 24

API Reference

API changelog

Wiki Home

JavaScript developer hints

Help us improve MacGap

Clone this wiki locally

https://github.com/MacGap

Clone in Desktop

Home

DOCUMENTATION

Getting Started

App Configuration

Developing your app

Customising your app

Manipulating the Menu

MacGap Events

Contributing to MacGap

COMMAND REFERENCE

MacGap

Clipboard

Dock

Defaults

Dialog

Fonts

File

Menu

MenuItem

Notify

Getting Started

Step 1 - Clone the Repository

```
$ git clone https://github.com/MacGapProject/MacGap2.git
```

Step 2 - Open the Project in Xcode

```
$ open MacGap2/MG.xcodeproj/
```

Step 3 - Build your App by editing public/index.html in Xcode

Home

DOCUMENTATION

Getting Started

App Configuration

Developing your app

Customising your app

Manipulating the Menu

MacGap Events

Contributing to MacGap

COMMAND REFERENCE

MacGap

Clipboard

Dock

Defaults

Dialog

Fonts

File

Menu

MenuItem

Notify

Sound

App Configuration

Documentation for MacGap

Platypus

Platypus is a developer tool that creates native Mac applications from interpreted scripts such as shell scripts or Perl, Ruby and Python programs. This is done by wrapping the script in an application bundle along with a native executable binary that runs the script.

Platypus makes it easy for you to share your scripts with people unfamiliar with the command line interface, without any knowledge of the native Mac APIs — a few clicks and you will have your own Mac application. Creating installers, maintenance applications, login items, launchers, automations and droplets is very easy using Platypus.

Features

- Supports shell scripts, Perl, Python, PHP, Ruby, Swift, Expect, Tcl, AppleScript or any other user-specified interpreter
- Can execute scripts with root privileges via the OS X Security Framework
- Can display graphical feedback of script execution as progress bar, text window with script output, droplet, WebKit HTML rendering or status item menu
- Supports dragging and dropping files or text snippets, which are then passed to the script as arguments
- Can create applications that are handlers for URI schemes
- Can create applications that run in the background (LSUIElement)
- Set own application icon or select from presets

sveinbjorn.org

Platypus Documentation

For **version 5.2.** The latest version of this document can be found here.

Last updated on February 24th 2017, by Sveinbjorn Thordarson.

Index

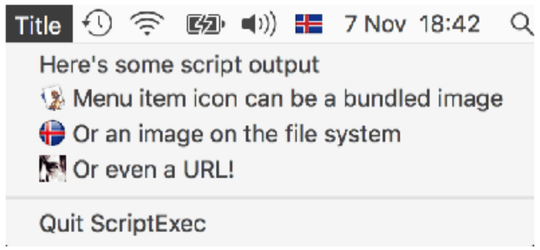
- **Introduction**
  - About Platypus
  - Credits
  - How does Platypus work?
  - What Platypus is NOT
  - System Requirements
- **The Basics**
  - Basic Interface
  - Interpreter
  - Interface
    - None
    - Progress Bar
    - Text Window
    - Web View
    - Status Menu
    - Droplet
  - Setting the Icon
  - Identifier, Author and Version
  - Special Options
  - Bundled Files
- **Advanced Options**
  - Accepting files and dragged items
  - Build-Time Options
  - Built-In Editor

sveinbjorn.org

Status Menu

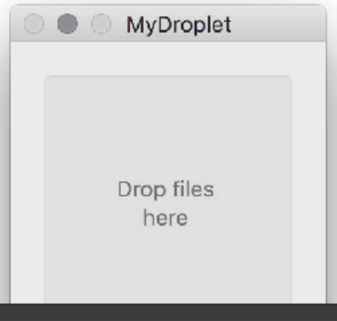
Application creates a Status Item in the menu bar when launched. Every time the status item is clicked, the script is executed and its text output is shown line for line in a menu. If a menu item is selected, the script is executed again with the title of the selected menu item passed as an argument to the script.

The properties of the Status Item (icon, title, etc.) can be configured by pressing the **Status Item Settings** button.



Droplet

Creates a square window instructing the user to drop files on it for processing. While processing, script output will be displayed line for line within the box along with an indeterminate circular progress indicator.



Platypus

Platypus Default

App Name: Checksettings

Script Type: Bash | /bin/bash | Args

Script Path: .../Beginner Journey to App Development/checksettings.sh

+ New | Reveal | Edit | Select Script...

Interface: Status Menu | Text Settings | Status Item Settings

Identifier: edu.HamiltonCollege.Checksettings

Author: Adam Wickert

Version: 1.0

Run with root privileges

Run in background

Accepts dropped items

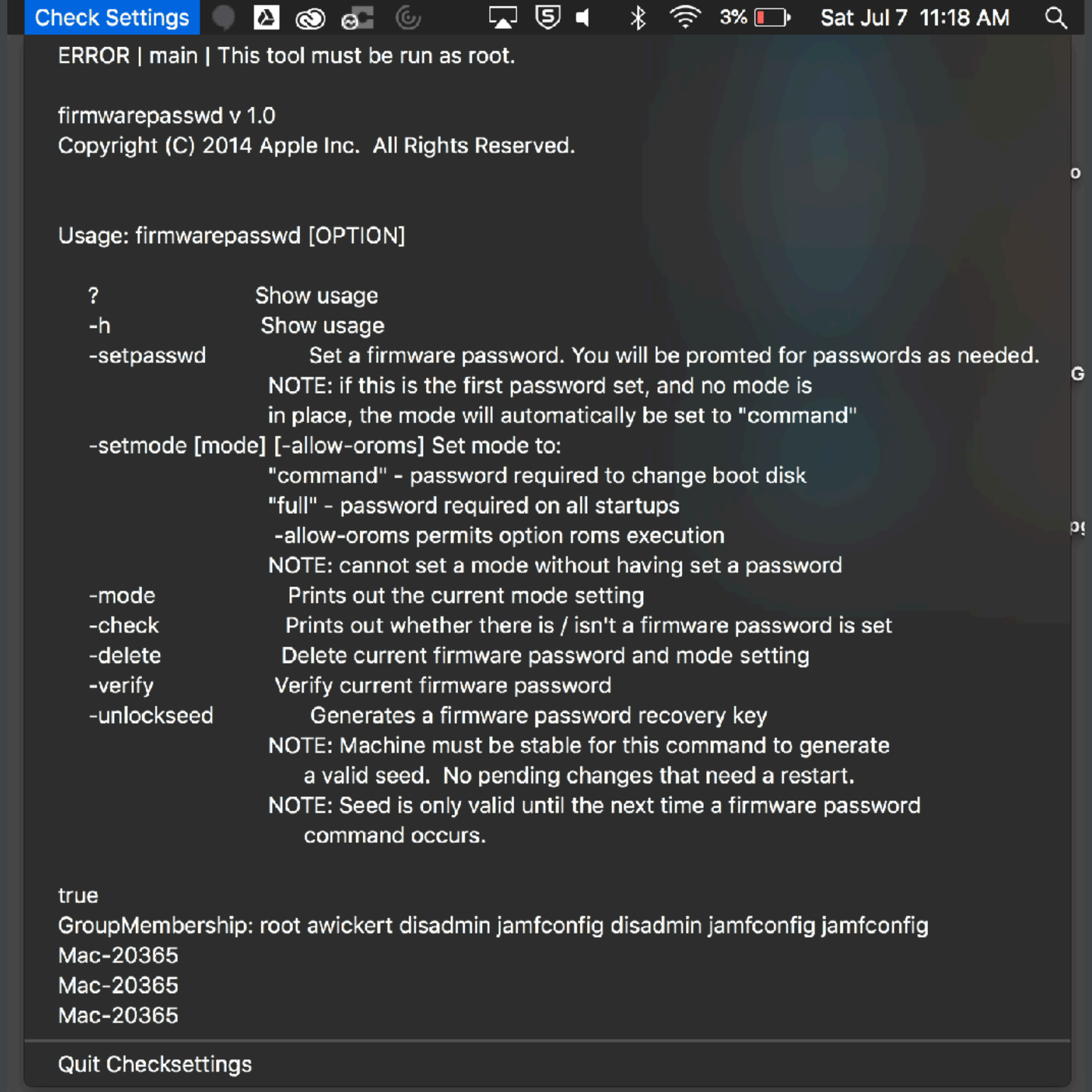
Remain running after execution

Files to copy into the application's Resources folder (default working directory for script)

Bundled Files

Estimated final app size: ~914 KB

Clear | Create App


Check Settings
3%
Sat Jul 7 11:18 AM
ERROR | main | This tool must be run as root.
firmwarepasswd v 1.0
Copyright (C) 2014 Apple Inc. All Rights Reserved.
Usage: firmwarepasswd [OPTION]
? Show usage
-h Show usage
-setpasswd Set a firmware password. You will be promted for passwords as needed.
NOTE: if this is the first password set, and no mode is
in place, the mode will automatically be set to "command"
-setmode [mode] [-allow-oroms] Set mode to:
"command" - password required to change boot disk
"full" - password required on all startups
-allow-oroms permits option roms execution
NOTE: cannot set a mode without having set a password
-mode Prints out the current mode setting
-check Prints out whether there is / isn't a firmware password is set
-delete Delete current firmware password and mode setting
-verify Verify current firmware password
-unlockseed Generates a firmware password recovery key
NOTE: Machine must be stable for this command to generate
a valid seed. No pending changes that need a restart.
NOTE: Seed is only valid until the next time a firmware password
command occurs.
true
GroupMembership: root awickert disadmin jamfconfig disadmin jamfconfig jamfconfig
Mac-20365
Mac-20365
Mac-20365
Quit Checksettings

Platypus

Platypus Default

App Name: Checksettings

Script Type: Bash | /bin/bash | Args

Script Path: .../Beginner Journey to App Development/checksettings.sh

+ New | Reveal | Edit | Select Script...

Interface: Status Menu | Text Settings | Status Item Settings

Identifier: edu.HamiltonCollege.Checksettings

Author: Adam Wickert

Version: 1.0

Run with root privileges

Run in background

Accepts dropped items

Remain running after execution

Files to copy into the application's Resources folder (default working directory for script)

Bundled Files

Estimated final app size: ~914 KB

Clear | Create App

Platypus

Platypus Default

App Name: Checksettings

Script Type: Bash | /bin/bash | Args

Script Path: .../Beginner Journey to App Development/checksettings.sh

+ New | Reveal | Edit | Select Script...

Interface: Text Window | Text Settings

Identifier: edu.HamiltonCollege.Checksettings

Author: Adam Wickert

Version: 1.0

Accepts dropped items

Run with root privileges

Run in background

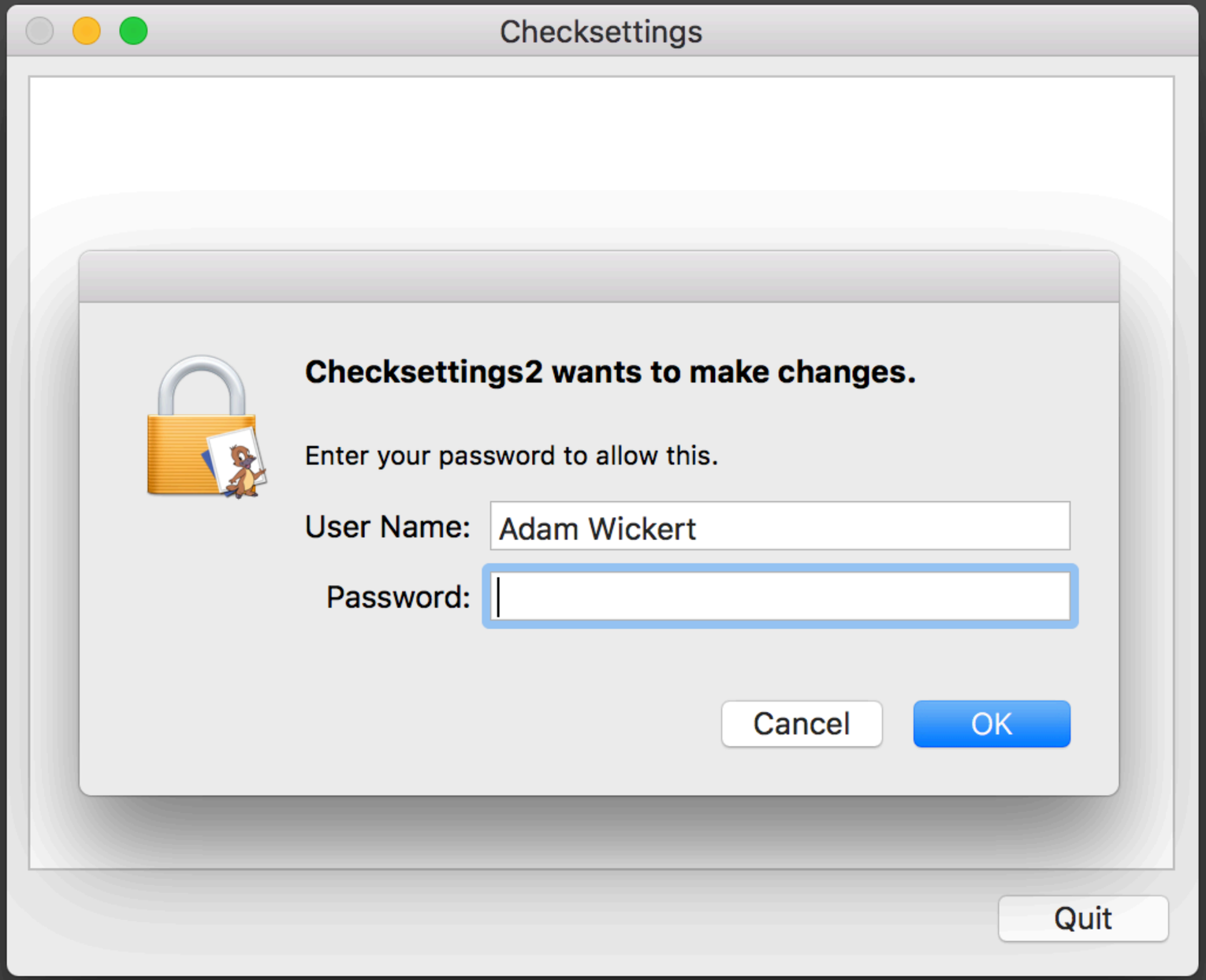
Remain running after execution

Files to copy into the application's Resources folder (default working directory for script)

Bundled Files

Estimated final app size: ~914 KB

Clear | Create App


Checksettings
Checksettings2 wants to make changes.
Enter your password to allow this.
User Name:
Adam Wickert
Password:
Cancel
OK
Quit

Checksettings

```
     -delete                         Delete current firmware
password and mode setting
     -verify                         Verify current firmware
password
     -unlockseed                     Generates a firmware password
recovery key
                                        NOTE: Machine must be
stable for this command to generate
                                              a valid seed.  No
pending changes that need a restart.
                                        NOTE: Seed is only valid
until the next time a firmware password
                                              command occurs.
true
GroupMembership: root awickert disadmin jamfconfig disadmin
jamfconfig jamfconfig
Mac-20365
Mac-20365
Mac-20365
```

Quit

checksettings2.sh — Free Mode

~/Desktop/Beginner Journey to App Development/checksettings2.sh

```
1   #!/bin/bash
2   # 2018-07-11
3   # Script to check security settings and name
4
5   # check if firmware password is active
6   # echo "Firmware" `firmwarepasswd -check`
7
8   # check to see if filevault is active
9   echo "FileVault Enabled:" `fdesetup isactive`
10
11  # check admin users
12  adminusers=`dscl . -read /Groups/admin GroupMembership`
13  echo "Admin Users:" ${adminusers:(+16)}
14
15  #check computer name
16  echo "Computer Name:" `scutil --get ComputerName`
17  echo "Local Host Name:" `scutil --get LocalHostName`
18  echo "Host Name:" `scutil --get HostName`
19
20  exit 0
```

L: 6 C: 3 Unix Shell Script Unicode (UTF-8) Unix (LF) Saved: 7/7/18, 6:19:47 PM 528 / 71 /...

Platypus

Platypus Default

App Name: Settings Check

Script Type: Bash | /bin/bash | Args

Script Path: ...Beginner Journey to App Development/checksettings2.sh

+ New | Reveal | Edit | Select Script...

Interface: Status Menu | Text Settings | Status Item Settings

Identifier: org.HamiltonCollege.SettingsCheck

Author: Adam Wickert

Version: 1.0

Accepts dropped items

Run with root privileges

Run in background

Remain running after execution

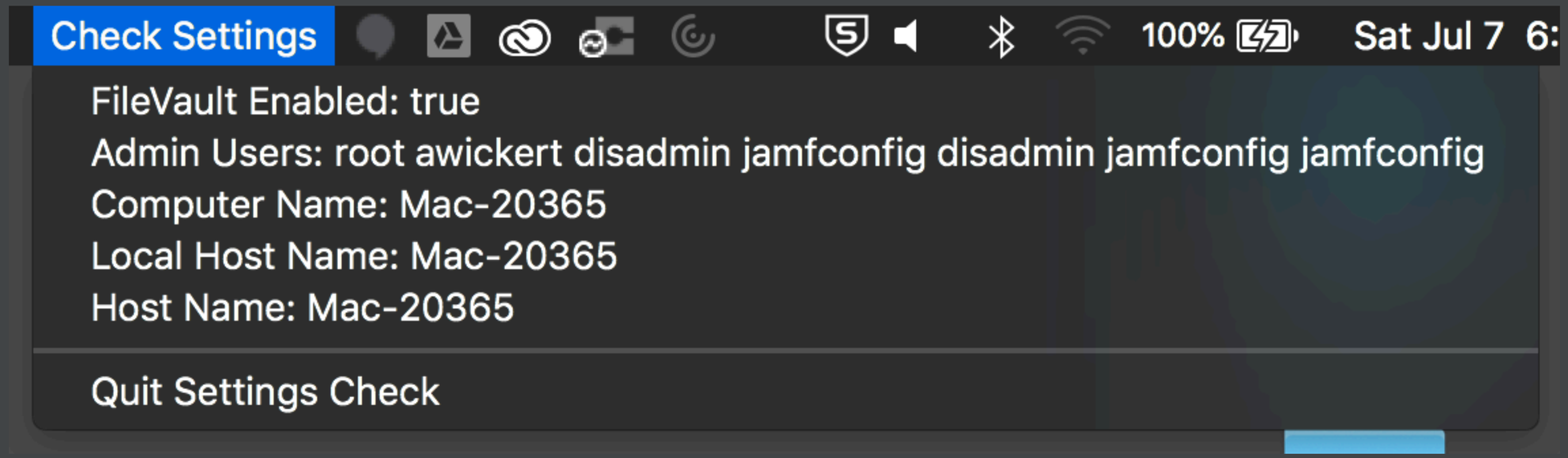
Files to copy into the application's Resources folder (default working directory for script)

Bundled Files

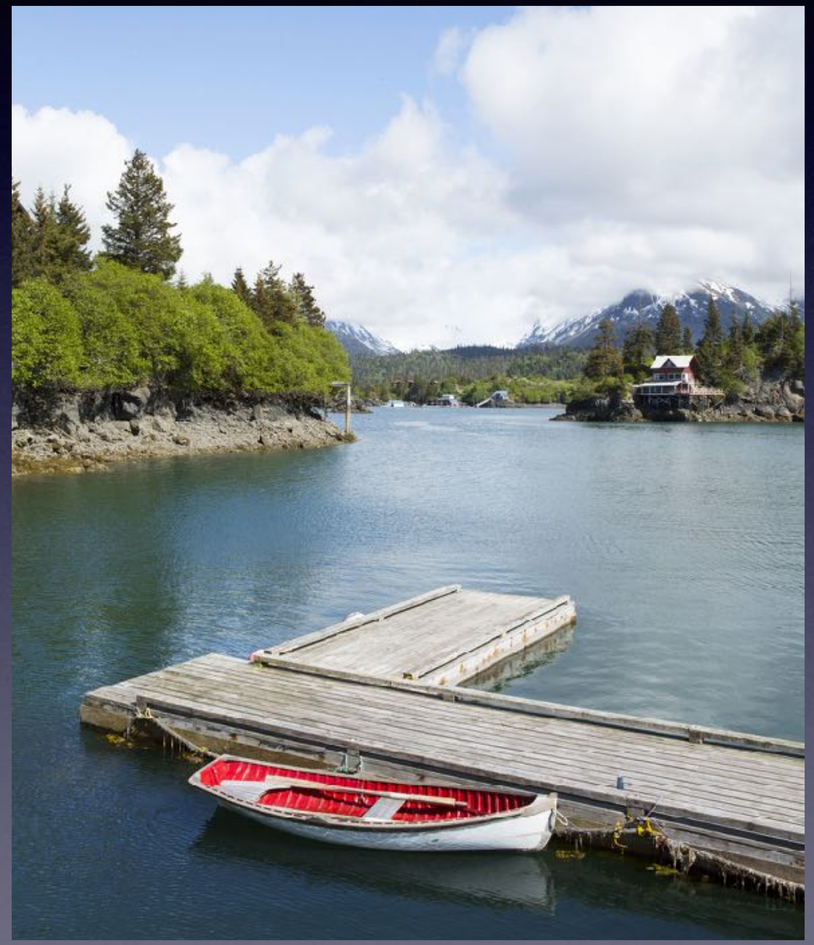
Estimated final app size: ~914 KB

Clear | Create App

Create Application

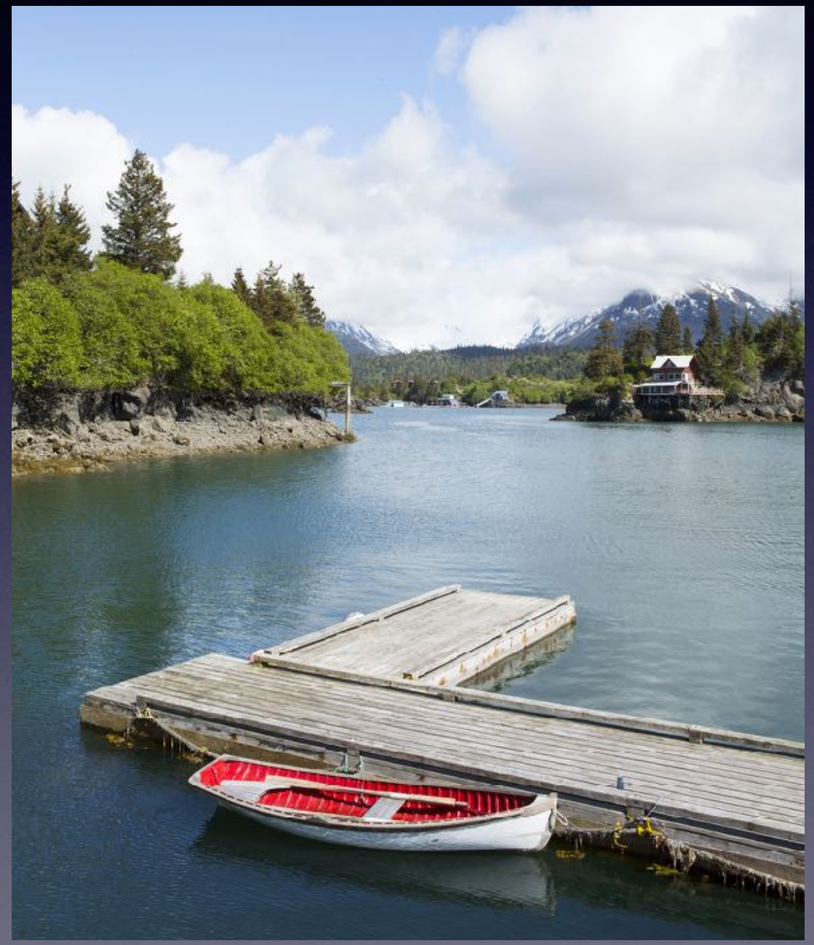

Check Settings
100%
Sat Jul 7 6:
FileVault Enabled: true
Admin Users: root awickert disadmin jamfconfig disadmin jamfconfig jamfconfig
Computer Name: Mac-20365
Local Host Name: Mac-20365
Host Name: Mac-20365
Quit Settings Check

POSSIBLE PATHS



POSSIBLE PATHS

- Learn a new programming language
- Code in a language you know and make a GUI
- Use a helper tool to build an app with a working script



KC Green, gunshowcomic.com

QUESTIONS?
COMMENTS?
DISCUSSION?

gitlab.com/awickert/SettingsCheck

awickert@hamilton.edu
@awickert on slack

THANK YOU!