

Leveraging DEPNotify and Jamf Pro for Device Deployment

Or: How I learned to stop imaging and use Automated Device Enrollment



John Mahlman

Lead Mac Engineer

Some Financial Company, Pennsylvania

- Over 10 years in Mac IT
- Write bad code
- Brew good beer
- Play Tabletop Games
- Love Philly sports teams

Find me: @jmahlman (slack, git, jamfnation)

Website: <https://yearofthegEEK.net>

UArts at a Glance

- Approximately 1,800 students
- 6 Academic buildings
- Over 200 “student facing” public Macs
- Offices, faculty/staff, Students (BYOD) — 97% Macs
- Computers range from 2009-2018 models
- On-Prem Jamf Pro since 2012
- Over 1,700 managed systems

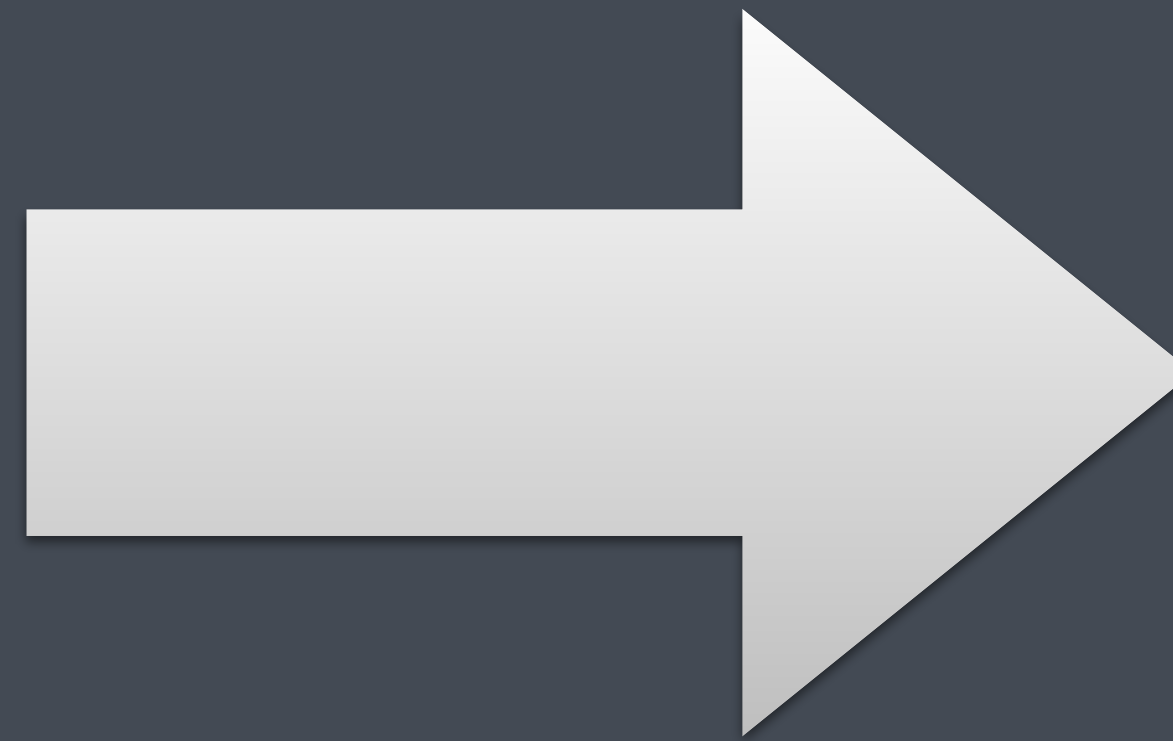
Where did we start?

Let's go back in time a few years..er..months?

Imaging... (not too long ago)

~\$ sudo bless --netboot --nextonly --server bsdnp://<server>

~\$ sudo shutdown -r now

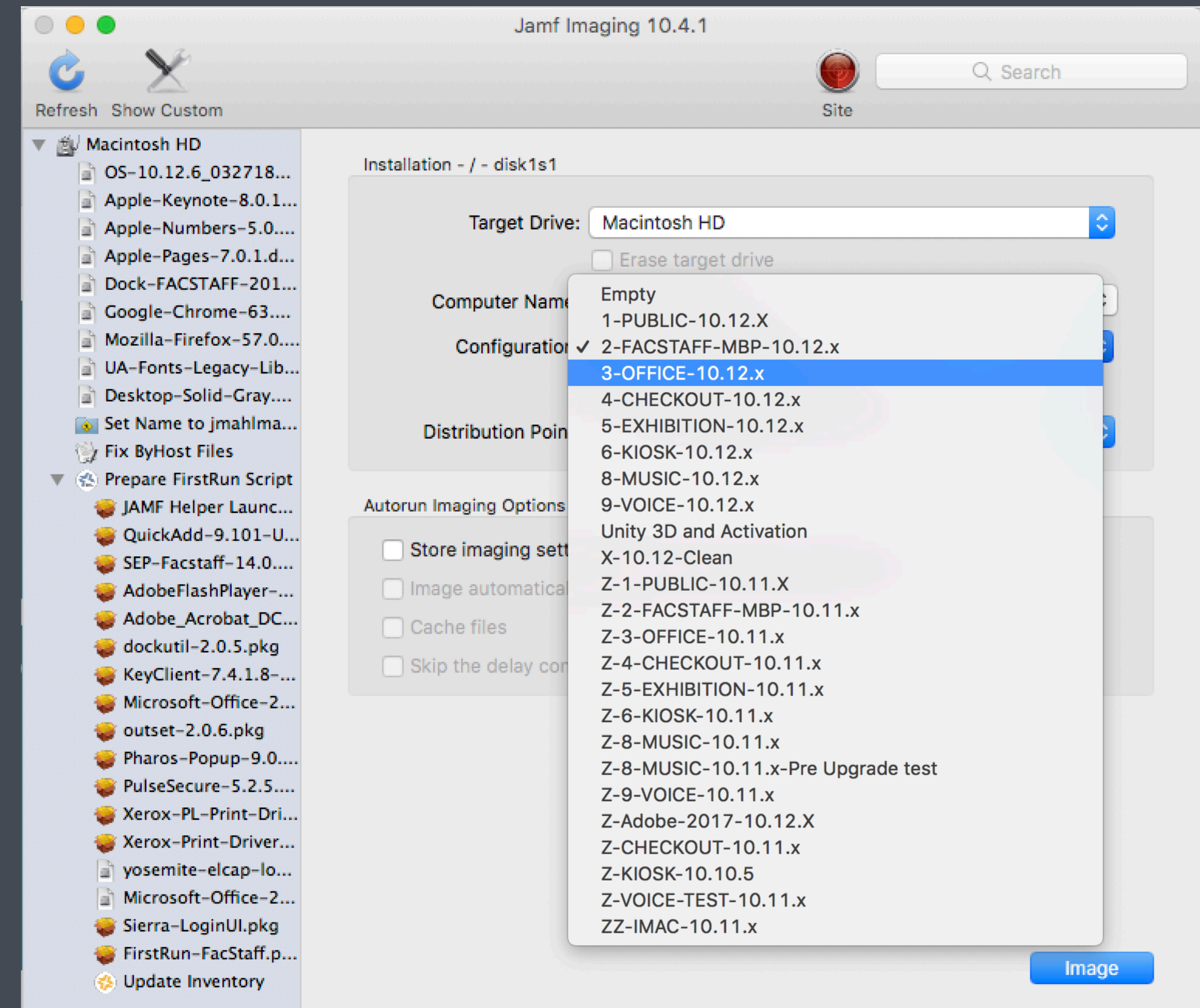


Imaging... (not too long ago)

Send command or
Set policy....

Go home...

Have beverage!



Imaging... (not too long ago)

Send command or
Set policy....

Go home...

Have beverage!



And then *it* happened...

You all know what I'm talking about...



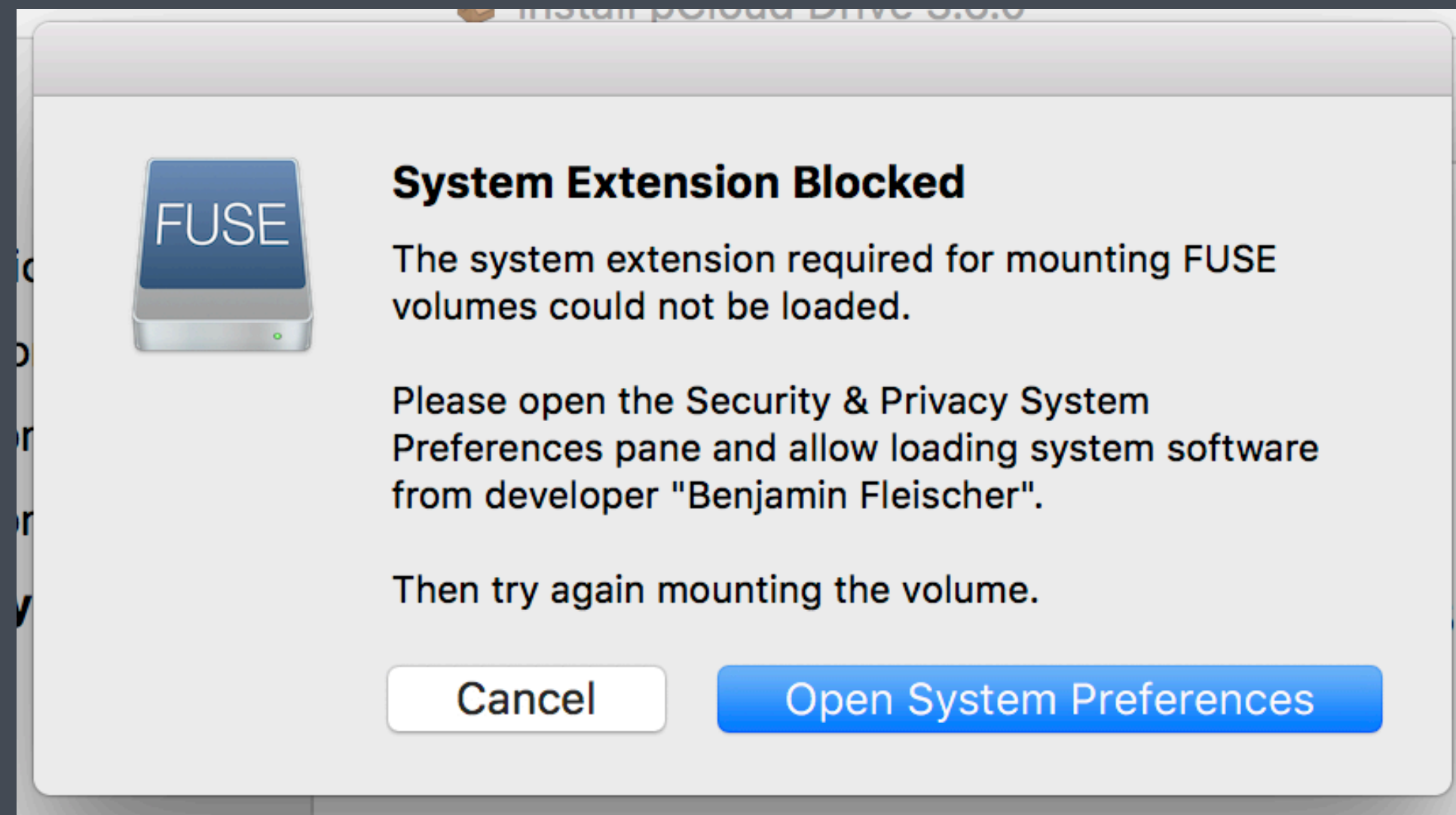
It happened

“Apple doesn't recommend or support monolithic system imaging as an installation method, because the system image might not include model-specific information such as firmware updates.”

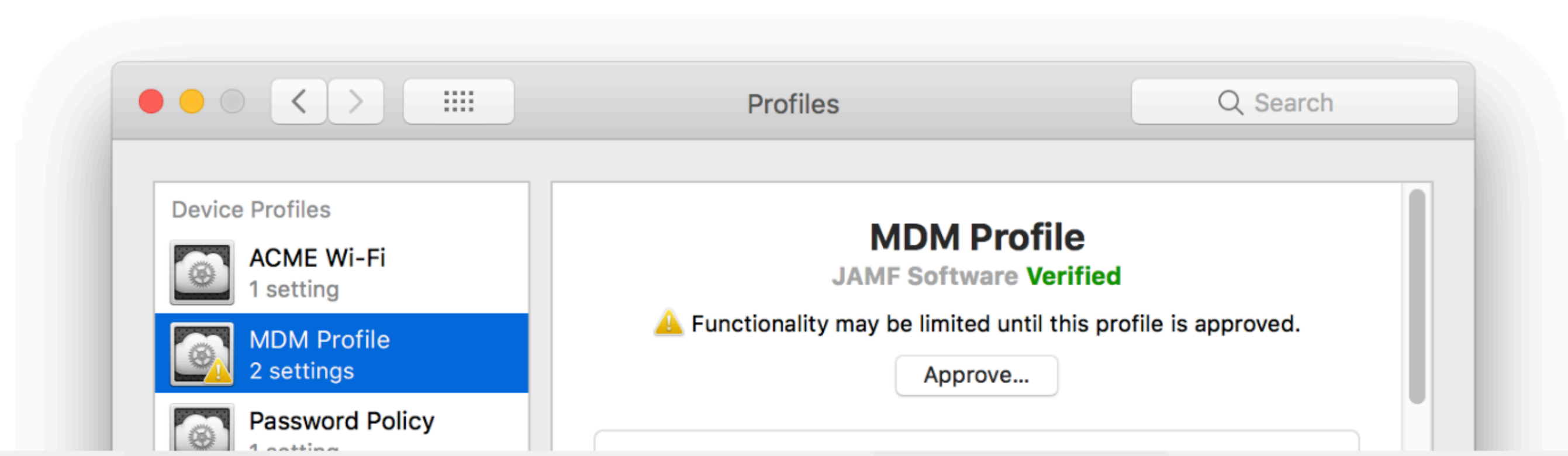
Apple, <https://support.apple.com/en-us/HT208020>

(Obtained 8/7/18)

User Approved Kernel Extension Loading (UAKEL, Ukulele)



User Approved MDM (UAMDM)



You must approve your organization's MDM Profile

Click the **Open System Preferences** button below and do the following:

1. Click the profile labeled "MDM Profile" in the Device Profiles column. It is identified by a yellow alert triangle.
2. Click the **Approve** button.

[Open System Preferences](#)

Apple T2 chip/Secure Boot

“Secure Boot offers three settings to make sure that your Mac always starts up from a legitimate, trusted Mac operating system...Full Security is the default Secure Boot setting...”

Apple, <https://support.apple.com/en-us/HT208330>

(Obtained 8/7/18)

Is imaging dead?



MOSTLY

IT'S DEAD, JIM.™



So, what are we going to do?

-Me, 2017

DEPNotify

- Reads input echoed into log file
- Input sets up UI and controls flow
- All UI aspects are controllable



```
echo "Command: MainTitle: New Mac Setup" >> $DNLOG
echo "Command: Image: /var/tmp/your-logo.png" >> $DNLOG
echo "Command: WindowStyle: NotMovable" >> $DNLOG
echo "Command: ContinueButtonRegister: Begin" >> $DNLOG
echo "Status: Please click the button below..." >> $DNLOG
```


The Process...

Preparation

- New machines get added to Automated Enrollment then assigned to jamf
- Old machines get wiped via internet recovery or policy

Deployment

- Boot machines to Setup Assistant
- Install Mobile Config
- Install software based on cohort (machine type)

Assignment

- Rename machine and assign to user
- Enter Asset Tag
- Give to user
- Enjoy a drink

The Process...

Preparation

- New machines get added to Automated Enrollment then assigned to jamf
- Old machines get wiped via internet recovery or policy

Deployment

- Boot machines to Setup Assistant
- Install Mobile Config
- Install software based on cohort (machine type)

Assignment

- Rename machine and assign to user
- Enter Asset Tag
- Give to user
- Enjoy a drink

Preparation...

New Machines

- Assign to MDM
- Setup Prestage
- Assign Devices to Prestage

General

DISPLAY NAME Display name for the PreStage enrollment

FACSTAFF-TEST-Deploy

DEVICE ENROLLMENT PROGRAM INSTANCE
DEP instance to associate with the PreStage enrollment. Devices associated with the selected DEP instance can be assigned the PreStage enrollment.

Uarts Devices ▾

☐ Automatically assign new devices
Automatically assign all new devices to this PreStage enrollment

☒ Use existing location information, if applicable

SUPPORT PHONE NUMBER Support phone number for the organization

215-717-6677

DEPARTMENT Department to associate with the PreStage enrollment

OTIS

☐ Require Authentication
Require the user to provide username and password on computers with macOS v10.10 or later

☒ Make MDM Profile Mandatory
Require the user to apply the MDM profile

☐ Allow MDM Profile Removal
Allow the user to remove the MDM profile

Setup Assistant Options Selected items are not displayed in the Setup Assistant during enrollment

None

Preparation...

New Machines

- Assign to MDM
- Setup Prestage
- Assign Devices to Prestage

Account Settings

Management Account Local administrator account to use for managing computers enrolled via user-initiated enrollment

ACCOUNT USERNAME

casperadmin

☒ **Create an additional local administrator account**
Additional local administrator account to create for computers enrolled via user-initiated enrollment

USERNAME

macadmin

PASSWORD

.....

VERIFY PASSWORD

.....

☐ **Hide Account**
Hide the local administrator account from users

Local User Account Type Type of user account to create during enrollment

☐ **Administrator Account**
Make the user an administrator for the computer

☐ **Standard Account**
Make the user a standard user on the computer

☒ **Skip Account Creation**
The user will not create a local user account

Preparation...



Existing Machines (APFS)

- Package Installer
- Script with
‘eraseinstall’ and
‘nointeraction’ flags
- Make Policy

```
#!/bin/bash
/Applications/Install\ macOS\ High\ Sierra.app/Contents/Resources/
startosinstall --applicationpath "/Applications/Install macOS High Sierra.app"
--rebootdelay 30 --nointeraction $4
```



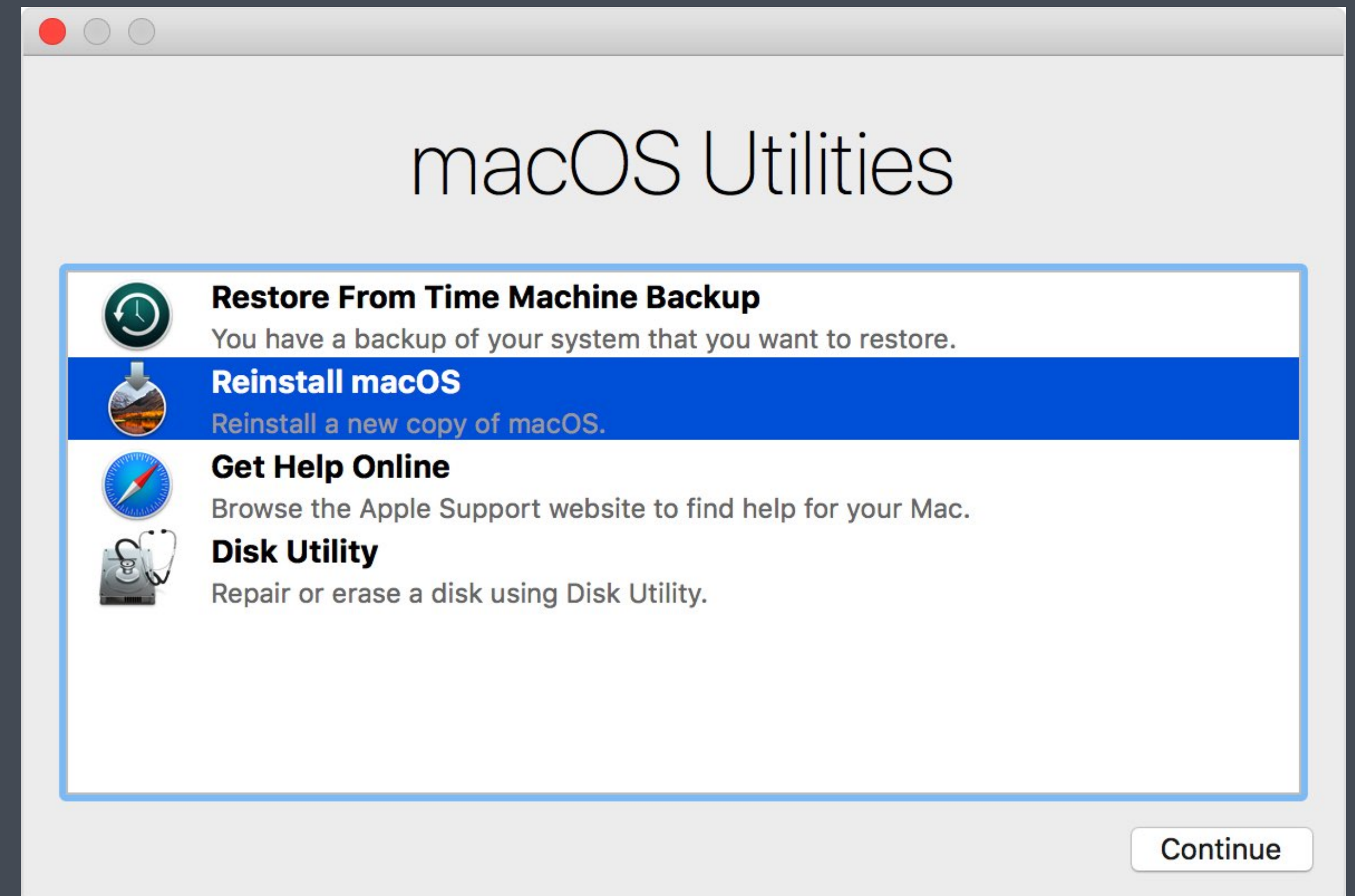
For additional flags!

- Install macOS High Sierra (In-Place) Ongoing install-high-sierra
- 1  Install macOS-High-Sierra-10.13.4-Installer.pkg
 - 2  Run Script macOS High Sierra Install

Preparation...

Existing Machines (HFS)

- Internet Recovery!



The Process...

Preparation

- New machines get added to Automated Enrollment then assigned to jamf
- Old machines get wiped via internet recovery or policy

Deployment

- Boot machines to Setup Assistant
- Install Mobile Config
- Install software based on cohort (machine type)

Assignment

- Rename machine and assign to user
- Enter Asset Tag
- Give to user
- Enjoy a drink

The Process...

Preparation

- New machines get added to Automated Enrollment then assigned to jamf
- Old machines get wiped via internet recovery or policy

Deployment

- Boot machines to Setup Assistant
- Install Mobile Config
- Install software based on cohort (machine type)

Assignment

- Rename machine and assign to user
- Enter Asset Tag
- Give to user
- Enjoy a drink

The Process...

Preparation

- New machines get added to Automated Enrollment then assigned to jamf
- Old machines get wiped via internet recovery or policy

Deploy and Assign

- Boot machines to Setup Assistant
- Install Mobile Config
- Install software based on cohort (machine type)
- Rename machine and assign to user
- Enter Asset Tag

The Process...

Enrollment Trigger

- Install DEPNotify
 - App Package
 - Logo
 - Provisioning Script

Run Script to do things!

- Install Software
- Assign computer to user in Jamf Pro
- Create local account
- Rename computer
- Install updates

But...we ran into issues...

- Ran behind the login window
 - Added a “wait for dock” loop
- Ran before user was completely logged in
 - Added timer
- Still was not running every time...
- Launch Daemon!

The Process...











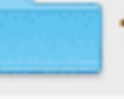



Enrollment Trigger

- Install DEPNotify
 - App Package
 - Logo
- Launch Daemon
- Deployment Script

Launch Daemon runs Script

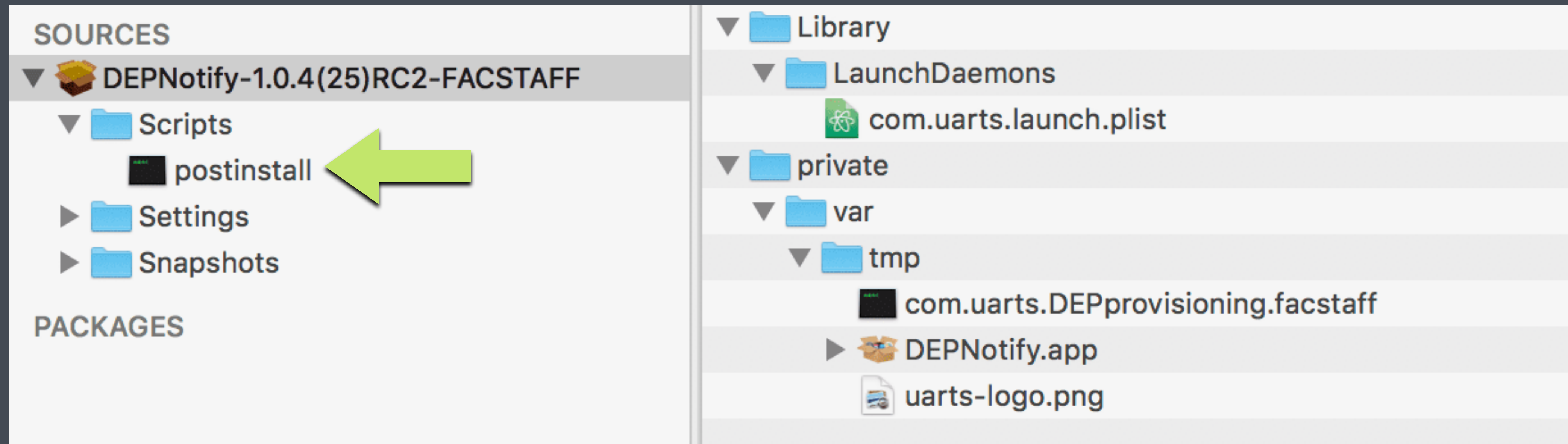
- Install Software
- Assign computer to user in jamf
- Create local account
- Rename computer
- Install updates

DEPNotify Package

SOURCES	
▼  DEPNotify-1.0.4(25)RC2-FACSTAFF	▼  Library
▼  Scripts	▼  LaunchDaemons
 postinstall	 com.uarts.launch.plist
▶  Settings	▼  private
▶  Snapshots	▼  var
PACKAGES	▼  tmp
	 com.uarts.DEProvisioning.facstaff
	▶  DEPNotify.app
	 uarts-logo.png

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
3 <plist version="1.0">
4 <dict>
5   <key>GroupName</key>
6   <string>wheel</string>
7   <key>InitGroups</key>
8   <false/>
9   <key>Label</key>
10  <string>com.uarte.launch</string>
11  <key>Program</key>
12  <string>/var/tmp/com.uarte.DEPprovisioning.facstaff</string>
13  <key>RunAtLoad</key>
14  <true/>
15  <key>StartInterval</key>
16  <integer>10</integer>
17  <key>UserName</key>
18  <string>root</string>
19  <key>StandardErrorPath</key>
20  <string>/var/tmp/depnotify.launch.err</string>
21  <key>StandardOutPath</key>
22  <string>/var/tmp/depnotify.launch.out</string>
23 </dict>
24 </plist>
25
```

DEPNotify Package



```
1  #!/bin/sh
2  ## postinstall
3
4  echo "disable auto updates ASAP" >> /var/log/jamf.log
5  /usr/sbin/softwareupdate --schedule off
6
7  # Disable diagnostic data
8  SUBMIT_DIAGNOSTIC_DATA_TO_APPLE=FALSE
9  SUBMIT_DIAGNOSTIC_DATA_TO_APP_DEVELOPERS=FALSE
10
11  ## Make the main script executable
12  echo "setting main script permissions" >> /var/log/jamf.log
13  chmod a+x /var/tmp/com.arts.DEProvisioning.facstaff.sh
14
15  ## Set permissions and ownership for launch daemon
16  echo "set LaunchDaemon permissions" >> /var/log/jamf.log
17  chmod 644 /Library/LaunchDaemons/com.arts.launch.plist
18  chown root:wheel /Library/LaunchDaemons/com.arts.launch.plist
19
20  ## Load launch daemon into the Launchd system
21  echo "load LaunchDaemon" >> /var/log/jamf.log
22  launchctl load /Library/LaunchDaemons/com.arts.launch.plist
23
24  exit 0    ## Success
25  exit 1    ## Failure
26
```


Enrollment Policy

General

DISPLAY NAME Display name for the policy

DEP-Install DEPNotify-FACSTAFF

☒ Enabled

SITE Site to add the policy to

None ▼

CATEGORY Category to add the policy to

DEP ▼

Trigger

Event(s) to use to initiate the policy

☐ **Startup**
When a computer starts up. A startup script that checks for policies must be configured in Jamf Pro for this to work

☐ **Login**
When a user logs in to a computer. A login hook that checks for policies must be configured in Jamf Pro for this to work

☐ **Logout**
When a user logs out of a computer. A logout hook that checks for policies must be configured in Jamf Pro for this to work

☐ **Network State Change**
When a computer's network state changes (e.g. when the network connection changes, when the computer name changes, when the IP address changes)

☒ **Enrollment Complete**
Immediately after a computer completes the enrollment process

☐ **Recurring Check-in**
At the recurring check-in frequency configured in Jamf Pro

☐ **Custom**
At a custom event

Enrollment Policy

Packages

DISTRIBUTION POINT Distribution point to download the package(s) from

Each computer's default distribution point ▼

DEPNotify-1.0.4(25)RC2-FACSTAFF.pkg

ACTION Action to take on computers

Install ▼

☐

Update Autorun data

Add or remove the package from each computer's Autorun data

The Process....

Enrollment Trigger

- Install DEPNotify
- App Package
- Logo
- Launch Daemon
- Deployment Script

Launch Daemon runs Script

- Install Software
- Assign computer to user in jamf *
- Create local account
- Rename computer *
- Install updates

Can I automate this for labs, etc..?

-Also also me, 2018

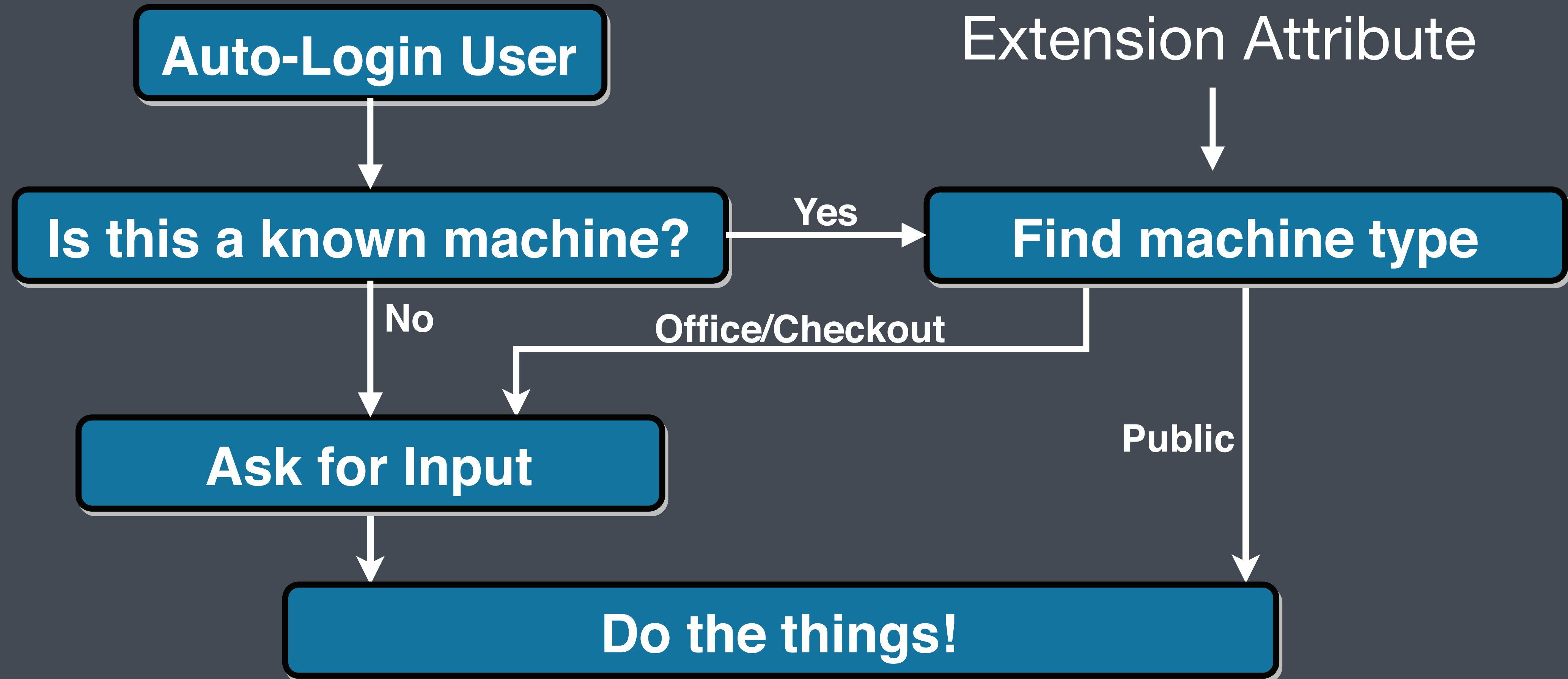
HECK YEAH!

-Me, 3 months ago

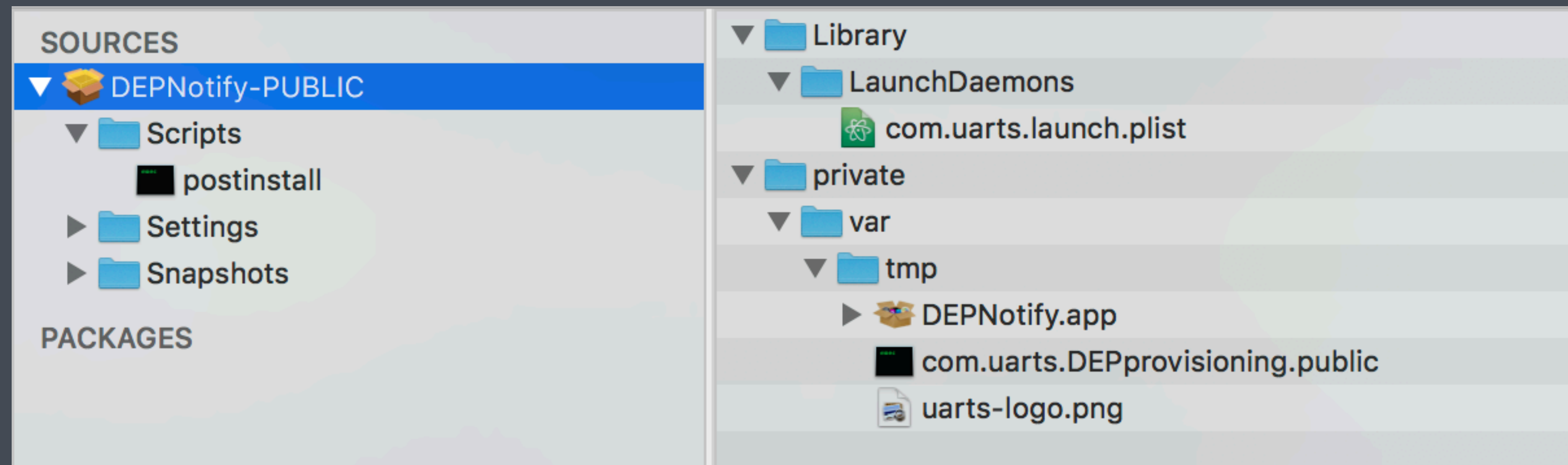
Neil Martin's JNRS presentation:

<https://github.com/neilmartin83/Jamf-Nation-Roadshow-London-2018>

Automate it, yo!



DEPNotify Package - Automated



```
## Load launch daemon into the Launchd system
echo "load LaunchDaemon" >> /var/log/jamf.log
launchctl load /Library/LaunchDaemons/com.arts.launch.plist
```




```
# We have to wait for the login window to show because the machine will reboot...
# so let's start this after the setup assistant is done.
```

```
CURRENTUSER=$(/usr/bin/python -c 'from SystemConfiguration import SCDynamicStoreCopyConsoleUser; \
import sys; username = (SCDynamicStoreCopyConsoleUser(None, None, None) or [None])[0]; \
username = [username,""][username in [u"loginwindow", None, u""]]; sys.stdout.write(username + "\n");')
while [[ "$CURRENTUSER" == "_mbsetupuser" ]]; do
    sleep 5
    CURRENTUSER=$(/usr/bin/python -c 'from SystemConfiguration import SCDynamicStoreCopyConsoleUser; \
import sys; username = (SCDynamicStoreCopyConsoleUser(None, None, None) or [None])[0]; \
username = [username,""][username in [u"loginwindow", None, u""]]; sys.stdout.write(username + "\n");')
done
```

```
echo "add auto-login user" >> /var/log/jamf.log
/usr/local/jamf/bin/jamf policy -event dep-autologin
```

```
exit 0    ## Success
exit 1    ## Failure
```

DEPNotify Package - Automated

●	DEP-Add AutoLogin User	Ongoing	dep-autologin	All computers
1	 Install autologin-uartstempdeploy.pkg			
2	 If rebooting, reboot to Current			
3	 Reboot if a package or SWU requires			

Thanks, MacUserGenerator! (<https://github.com/ninxsoft/MacUserGenerator>)

Welcome

In just a few steps, you can register and set up your Mac.



United States

Afghanistan

Åland Islands

Albania

Algeria

American Samoa

Andorra

Angola



Back




Continue

Do you need to hear instructions for setting up your Mac?
To learn how to use VoiceOver to set up your computer, press the Escape key now.

Select Your Keyboard



Choose a keyboard layout.

 U.S.

☐ Show All



Back



Continue

Remote Management

Remote management enables the administrator of "University Of Arts" to set up email and network accounts, install and configure apps, and manage this computer's settings.



"University Of Arts" can automatically configure your computer.

[Learn more about remote management](#)



Back



Continue

Remote Management

Remote management enables the administrator of "University Of Arts" to set up email and network accounts, install and configure apps, and manage this computer's settings.



"University Of Arts" can automatically configure your computer.

Contacting enrollment server and configuring your Mac.



Back



Continue

Data & Privacy

This icon appears when an Apple feature asks to use your personal information.



You won't see this with every feature since Apple collects this information only when needed to enable features, secure our services, or personalize your experience.

Apple believes privacy is a fundamental human right, so every Apple product is designed to minimize the collection and use of your data, use on-device processing whenever possible, and provide transparency and control over your information.

[Learn More...](#)



Back



Continue

Select Your Time Zone

To select a time zone, click the map near your location and choose a city from the Closest City menu.
You can also have the time zone change automatically, if possible, based on your current location.

☐ Set time zone automatically using current location



Time Zone: Pacific Daylight Time

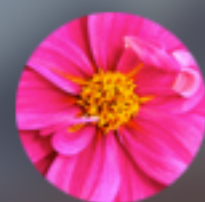
Closest City: Cupertino, CA - United States



Back



Continue



macadmin



Other...



Sleep



Restart



Shut Down


```
# Set basic variables
OSVERSION=$(sw_vers -productVersion)
serial=$(ioreg -rd1 -c IOPlatformExpertDevice | awk -F'"' '/IOPlatformSerialNumber/{print $4}')
eaxml=$(curl "$JSSURL"JSSResource/computers/serialnumber/"$serial"/subset/extension_attributes -u "$APIUSER":"$APIPASS" -H "A
jssMacName=$(echo "$eaxml" | xpath '//extension_attribute[name="New Computer Name"]' | awk -F'<value>|</value>' '{print $2}')
jssCohort=$(echo "$eaxml" | xpath '//extension_attribute[name="New Cohort"]' | awk -F'<value>|</value>' '{print $2}')
```

```
# If the computer is NOT in the jss or if it's an OFFICE machine
# we want to get user input because most likely this is being reprovisioned.

if [[ "$jssMacName" == "" ]] || [[ "$jssCohort" == "" ]] || [[ "$jssCohort" == "OFFICE" ]]; then
# Configure DEPNotify registration window

sudo -u "$CURRENTUSER" defaults write menu.nomad.DEPNotify PathToPlistFile /var/tmp/
sudo -u "$CURRENTUSER" defaults write menu.nomad.DEPNotify RegisterMainTitle "Setup..."
sudo -u "$CURRENTUSER" defaults write menu.nomad.DEPNotify RegisterButtonLabel Setup
sudo -u "$CURRENTUSER" defaults write menu.nomad.DEPNotify UITextFieldUpperPlaceholder "T1337-M01"
sudo -u "$CURRENTUSER" defaults write menu.nomad.DEPNotify UITextFieldUpperLabel "Computer Name"
sudo -u "$CURRENTUSER" defaults write menu.nomad.DEPNotify UITextFieldLowerPlaceholder "UA42DSK1337"
sudo -u "$CURRENTUSER" defaults write menu.nomad.DEPNotify UITextFieldLowerLabel "Asset Tag"
sudo -u "$CURRENTUSER" defaults write menu.nomad.DEPNotify UIPopupMenuUpper -array \
'Lab' 'Studio' 'Smart-Classroom' 'Office' 'Checkout' 'Kiosk'
sudo -u "$CURRENTUSER" defaults write menu.nomad.DEPNotify UIPopupMenuUpperLabel "Cohort"
```



New Mac Setup


Make sure this Mac is plugged into a wired network connection before beginning.

Please click the button below...

Begin

```
# If the computer is NOT in the jss or if it's an OFFICE machine
# we want to get user input because most likely this is being reprovisioned.

if [[ "$sissMacName" == "" ]] || [[ "$sissCohort" == "" ]] || [[ "$sissCohort" == "OFFICE" ]]; then
# Configure DEPNotify registration window
sudo -u "$CURRENTUSER" defaults write menu.nomad.DEPNotify PathToPlistFile /var/tmp/
sudo -u "$CURRENTUSER" defaults write menu.nomad.DEPNotify RegisterMainTitle "Setup..."
sudo -u "$CURRENTUSER" defaults write menu.nomad.DEPNotify RegisterButtonLabel Setup
sudo -u "$CURRENTUSER" defaults write menu.nomad.DEPNotify UITextFieldUpperPlaceholder "T1337-M01"
sudo -u "$CURRENTUSER" defaults write menu.nomad.DEPNotify UITextFieldUpperLabel "Computer Name"
sudo -u "$CURRENTUSER" defaults write menu.nomad.DEPNotify UITextFieldLowerPlaceholder "UA42DSK1337"
sudo -u "$CURRENTUSER" defaults write menu.nomad.DEPNotify UITextFieldLowerLabel "Asset Tag"
sudo -u "$CURRENTUSER" defaults write menu.nomad.DEPNotify UIPopupMenuUpper -array \
'Lab' 'Studio' 'Smart-Classroom' 'Office' 'Checkout' 'Kiosk'
sudo -u "$CURRENTUSER" defaults write menu.nomad.DEPNotify UIPopupMenuUpperLabel "Cohort"
```



Setup...

Computer Name

T1337-M01

Asset Tag

UA42DSK1337

Cohort


Lab

Setup

```
# Let's read the user data into some variables...
computerName=$(/usr/libexec/plistbuddy $DNPLIST -c "print 'Computer Name'" | tr [a-z] [A-Z])
cohort=$(/usr/libexec/plistbuddy $DNPLIST -c "print 'Cohort'" | tr [a-z] [A-Z])
ASSETTAG=$(/usr/libexec/plistbuddy $DNPLIST -c "print 'Asset Tag'" | tr [a-z] [A-Z])
# Update Asset Tag, Computer Name, and Cohort in JSS

cat << EOF > /var/tmp/tempInfo.xml
<computer>
  <general>
    <asset_tag>$ASSETTAG</asset_tag>
  </general>
  <extension_attributes>
    <extension_attribute>
      <name>New Cohort</name>
      <value>$cohort</value>
    </extension_attribute>
    <extension_attribute>
      <name>New Computer Name</name>
      <value>$computerName</value>
    </extension_attribute>
  </extension_attributes>
</computer>
EOF

## Upload the xml file
/usr/bin/curl -sfku "$APIUSER":"$APIPASS" "$JSSURL"JSSResource/computers/serialnumber/"$serial" -H \
"Content-type: text/xml" -T /var/tmp/tempInfo.xml -X PUT
```



Setup...

Computer Name

T802-M70

Asset Tag

UA42LAP1337

Cohort

✓ Lab

Studio

Smart-Classroom

Office

Checkout

Kiosk

Setup


```

echo "Status: Running FirstRun scripts and installing packages..." >> $DNLOG
echo "Command: DeterminateManualStep:" >> $DNLOG
if [[ $cohort == "CHECKOUT" ]]; then
    echo "Command: MainText: This Mac is installing all necessary software and \
running some installation scripts \n \n Cohort: $cohort \n \n macOS Version: $OSVERSION" >> $DNLOG
    $JAMFBIN policy -event install-CHECKOUT-software
    $JAMFBIN policy -event enroll-firstRunCHECKOUT-scripts
elif [[ $cohort == "OFFICE" ]]; then
    echo "Command: MainText: This Mac is installing all necessary software and \
running some installation scripts. \n \n Cohort: $cohort \n \n macOS Version: $OSVERSION" >> $DNLOG
    $JAMFBIN policy -event install-OFFICE-software
    $JAMFBIN policy -event enroll-firstRunOFFICE-scripts
else
    echo "Command: MainText: This Mac is installing all necessary software and \
running some installation scripts. \n \n Cohort: $cohort \n \n macOS Version: $OSVERSION" >> $DNLOG
    $JAMFBIN policy -event install-PUBLIC-software
    $JAMFBIN policy -event enroll-firstRunPUBLIC-scripts
fi

```

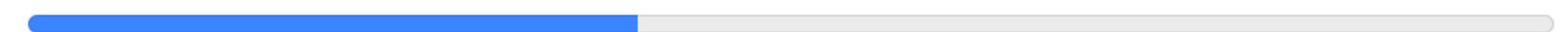


PHI 38 - MIN 7

This mac is installing all necessary software and running some installation scripts.

Cohort: CHECKOUT

macOS Version: 10.13.6



Running FirstRun scripts and installing packages...


```
echo "Status: Running FirstRun scripts and installing packages..." >> $DNLOG
echo "Command: DeterminateManualStep:" >> $DNLOG
if [[ $cohort == "CHECKOUT" ]]; then
    echo "Command: MainText: This Mac is installing all necessary software and \
running some installation scripts \n \n Cohort: $cohort \n \n macOS Version: $OSVERSION" >> $DNLOG
    $JAMFBIN policy -event install-CHECKOUT-software
    $JAMFBIN policy -event enroll-firstRunCHECKOUT-scripts
elif [[ $cohort == "OFFICE" ]]; then
    echo "Command: MainText: This Mac is installing all necessary software and \
running some installation scripts. \n \n Cohort: $cohort \n \n macOS Version: $OSVERSION" >> $DNLOG
    $JAMFBIN policy -event install-OFFICE-software
    $JAMFBIN policy -event enroll-firstRunOFFICE-scripts
else
    echo "Command: MainText: This Mac is installing all necessary software and \
running some installation scripts. \n \n Cohort: $cohort \n \n macOS Version: $OSVERSION" >> $DNLOG
    $JAMFBIN policy -event install-PUBLIC-software
    $JAMFBIN policy -event enroll-firstRunPUBLIC-scripts
fi
```

```
# Remove DEPNotify and the logs
/bin/rm -Rf /var/tmp/DEPNotify.app
/bin/rm -Rf /var/tmp/uarts-logo.png
/bin/rm -Rf $DNLOG
/bin/rm -Rf $DNPLIST

# Wait a few seconds
sleep 5

# Remove the autologin user password file so it doesn't login again
/bin/rm -Rf /etc/kcpassword

# Create a bom file that allow this script to stop launching DEPNotify after done
/usr/bin/touch /var/db/receipts/com.uarts.provisioning.done.bom

# Remove the Launch Daemon
/bin/rm -Rf /Library/LaunchDaemons/com.uarts.launch.plist

# Remove this script
/bin/rm -- "$0"
```



Almost done!

Please do not shutdown, reboot, or close your device, it will automatically reboot when complete.



Checking for and installing any OS updates...

Name

Enter Password

Login using your UARTS Username and Password



Sleep



Restart



Shut Down



Is this imaging?

I'm not into scripting...any ideas?

-You, right now...maybe?

jamf / DEPNotify-Starter

Watch

21

Unstar

99

Fork

17

<> Code

Issues7

Pull requests1

Security

Insights

Bash script to start DEPNotify and run policies during enrollment with Jamf Pro

<https://gitlab.com/Mactroll/DEPNotify>

75 commits

2 branches

7 releases

3 contributors

MIT

Branch: master

New pull request

Create new file

Upload files

Find File

Clone or download

kbareis

Merge pull request #19 from homebysix/patch-1

Latest commit 6d267c6 on Feb 11

github/ISSUE_TEMPLATE	Update issue templates	9 months ago
example-img	Documentation update	9 months ago
example-video	Documentation update	9 months ago
.gitignore	Additional cleanup and new options	9 months ago
CHANGELOG.md	Update CHANGELOG.md	6 months ago
CONTRIBUTORS.md	Documentation update	9 months ago
LICENSE	Initial commit	last year
README.md	Merge pull request #11 from jamf/develop-2.0.0-fgd-changes	7 months ago
RELEASES.md	Update RELEASES.md	6 months ago
depNotify.sh	Minor spelling correction	5 months ago
depNotifyReset.sh	Added depNotifyReset.sh	6 months ago

README.md

<https://github.com/jamf/DEPNotify-Starter>

\$CurrentCompany

- About 700 Macs
- Global
- Very high security
 - No Automated Deployment (GASP)
 - No APNS (FAINT!)

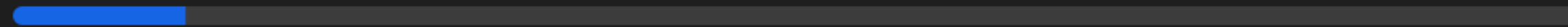
\$CurrentCompany Deployment



Macintosh Imaging Process

Welcome to Mac at [REDACTED] Make sure the computer name is added to active directory and the roles are assigned.. This process should take 30 to 45 minutes to complete.

If you need additional software after imaging, use Self Service afterwards.



Waiting on completion of Computer Information

Input Computer Information



\$CurrentCompany Deployment



Collect Macintosh Information

Computer Name *i*

T10646

Location *i*

Pennsylvania



Package *i*

Default



Input Computer Information

The hopeful future!

- Hope that Apple gives us a way to have 100% zero-touch
 - Skip Setup Assistant
 - --eraseinstall flag
 - Catalina R/O Filesystem + Data Container? Awesome!
- Perhaps use NoMAD Login instead?
 - NoMAD + DEPNotify

Neil Martin's MacADUK 2019 presentation:
<https://github.com/neilmartin83/MacADUK-2019>

Resources

- My GitHub
 - <https://github.com/jmahlman/DEPNotify-automated>
- DEPNotify
 - <https://gitlab.com/Mactroll/DEPNotify>
- Neil Martin's Presentation/Code from JNRS and MacADUK
 - <https://github.com/neilmartin83/Jamf-Nation-Roadshow-London-2018>
 - <https://github.com/neilmartin83/MacADUK-2019>
- Jamf Professional Services DEPNotify repo
 - <https://github.com/jamf/DEPNotify-Starter>



THANK YOU!