

# Create Workflow

## Corrections / Changes

After filming this video, I realized that you don't need to chroot your scripts and that Bootstrappr runs the script in a way that gives you the same results as chroot. As demonstrated, if you run these commands in recovery mode, it won't work, but it will work when Bootstrappr runs it. Learning and understanding chroot is useful not only for macOS but in Linux systems as well. Therefore, I didn't remove it from the content.

## Web Resources

### **bootstrappr**

<https://github.com/munki/bootstrappr>

### **Munki Kickstart**

<https://github.com/munki/munki/wiki/Bootstrapping-With-Munki>

### **SkipAppleSetupAssistant**

<https://github.com/MagerValp/SkipAppleSetupAssistant/releases>

### **Pycreateuserpkg**

<https://github.com/gregneagle/pycreateuserpkg>

*Continues on next page*

## Scripts

### 01\_rename.sh

```
#!/bin/bash

# Ask for Computer Name
read -p "Please Enter Computer Name:" newname

# Change Computer Names
chroot -u root /Volumes/Macintosh\ HD/ /bin/bash -c "scutil --set
ComputerName $newname"
echo "ComputerName changed to $newname"
chroot -u root /Volumes/Macintosh\ HD/ /bin/bash -c "scutil --set
LocalHostName $newname"
echo "LocalHostName changed to $newname"
chroot -u root /Volumes/Macintosh\ HD/ /bin/bash -c "scutil --set
HostName $newname"
echo "HostName changed to $newname"
```

*Continues on next page*

## 06\_munki\_manifest.sh

```
# Ask user to choose Munki Manifest from a list

# Create a list of manifests on your munki server

# Run this command on your Admin machine every time you create a new
manifest

# You can use cron to automate it

# manifestutil list-manifests > /Volumes/munki/files/
munki_manifests.list

# Import manifest list and write to a file
curl http://munki/munki_repo/files/munki_manifests.list -o /Volumes/
Macintosh\ HD/tmp/munki_manifests.list

# Add option to make a manifest after computername

# Get Computer Name and write it to the list
newname="$(chroot -u root /Volumes/Macintosh\ HD/ /bin/bash -c "scutil
--get ComputerName")"
echo $newname >> /Volumes/Macintosh\ HD/tmp/munki_manifests.list

# Add option to make a manifest after serialnumber and write it to the
list
serial="$(chroot -u root /Volumes/Macintosh\ HD/ /bin/bash -c
"system_profiler SPHardwareDataType | grep Serial" | awk '{print
$4}')"
echo $serial >> /Volumes/Macintosh\ HD/tmp/munki_manifests.list

# Ask User to choose Munki Manifest
echo "Which Munki Manifest will this machine use?"
```

```

nl /Volumes/Macintosh\ HD/tmp/munki_manifests.list
count="$(wc -l /Volumes/Macintosh\ HD/tmp/munki_manifests.list | awk
'{print $1}')"
n=""
while true; do
    read -p 'Select option: ' n
    # If $n is an integer between one and $count...
    if [ "$n" -eq "$n" ] && [ "$n" -gt 0 ] && [ "$n" -le "$count" ];
then
break
fi done

manifest="$(sed -n "${n}p" /Volumes/Macintosh\ HD/tmp/
munki_manifests.list)"

# Write Munki Preference File
echo "Writing Munki configuration file"
chroot -u root /Volumes/Macintosh\ HD/ /bin/bash -c "defaults write /
Volumes/Macintosh\ HD/Library/Preferences/ManagedInstalls
SoftwareRepoURL "http://munki/munki_repo""
chroot -u root /Volumes/Macintosh\ HD/ /bin/bash -c "defaults write /
Volumes/Macintosh\ HD/Library/Preferences/ManagedInstalls
ClientIdentifier "$manifest""
chroot -u root /Volumes/Macintosh\ HD/ /bin/bash -c "defaults write /
Volumes/Macintosh\ HD/Library/Preferences/ManagedInstalls
SuppressUserNotification -bool true"
echo "Writing configuration file complete."

# Cleanup
rm -rf /Volumes/Macintosh\ HD/tmp/munki_manifests.list

```

*Continues on next page*

## 07\_suppress\_diagnostics.sh

```
echo "Disabling Diagnostics & Usage dialog"
chroot -u root /Volumes/Macintosh\ HD/ /bin/bash -c "defaults write /
Volumes/Macintosh\ HD/Library/Application\ Support/CrashReporter/
DiagnosticMessagesHistory AutoSubmit -boolean FALSE"
chroot -u root /Volumes/Macintosh\ HD/ /bin/bash -c "defaults write /
Volumes/Macintosh\ HD/Library/Application\ Support/CrashReporter/
DiagnosticMessagesHistory AutoSubmitVersion -int 14"
chroot -u root /Volumes/Macintosh\ HD/ /bin/bash -c "defaults write /
Volumes/Macintosh\ HD/Library/Application\ Support/CrashReporter/
DiagnosticMessagesHistory ThirdPartyDataSubmit -boolean FALSE"
chroot -u root /Volumes/Macintosh\ HD/ /bin/bash -c "defaults write /
Volumes/Macintosh\ HD/Library/Application\ Support/CrashReporter/
DiagnosticMessagesHistory ThirdPartyDataSubmitVersion -int 14"
chroot -u root /Volumes/Macintosh\ HD/ /bin/bash -c "chmod 664 /
Volumes/Macintosh\ HD/Library/Application\ Support/CrashReporter/
DiagnosticMessagesHistory.plist"
chroot -u root /Volumes/Macintosh\ HD/ /bin/bash -c "chown
root:wheel /Volumes/Macintosh\ HD/Library/Application\ Support/
CrashReporter/DiagnosticMessagesHistory.plist"
echo "Diagnostics & Usage dialog disabled"
```