



# Intro to LaunchDaemons





# Rebecca Latimer

Hibbing Public Schools





# Intro to LaunchDaemons

- My Story
- What is a LaunchDaemon?
- Anatomy of a LaunchDaemon
- So You Want to Create One?
- Walkthrough
- More Resources



# My Story

- Not a developer, just a “tech lady”



What are they? (What is a daemon anyway?)





# LaunchDaemons and LaunchAgents

- Similar but not the same
- Daemons run as root even when nobody is logged in
- Agents run as as a user when someone is logged in



# launchd - One LaunchDaemon To Launch All LaunchDaemons

Process Name	Memory	Threads	Ports	PID	User
launchd	48.6 MB	3	6,971	1	root



# Daemons in Your Machine

- Daemons are running on your computer right now
- `sudo launchctl list`
- Jamf works through launchdaemons



```
<?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>Label</key>
  <string>com.jamfsoftware.task.Every 15 Minutes</string>
  <key>ProgramArguments</key>
  <array>
    <string>/usr/local/jamf/bin/jamf</string>
    <string>policy</string>
    <string>-randomDelaySeconds</string>
    <string>300</string>
  </array>
  <key>StartInterval</key>
  <integer>900</integer>
  <key>UserName</key>
  <string>root</string>
</dict>
</plist>
```

# Anatomy of a LaunchDaemon

# LaunchDaemons Are Just Plists



- XML file structure
- Usually run scripts or commands
- We will break down the one that lives in
- `/Library/LaunchDaemons/com.jamfsoftware.task.1.plist`

```
<?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>Label</key>
  <string>com.jamfsoftware.task.Every 15 Minutes</string>
  <key>ProgramArguments</key>
  <array>
    <string>/usr/local/jamf/bin/jamf</string>
    <string>policy</string>
    <string>-randomDelaySeconds</string>
    <string>300</string>
  </array>
  <key>StartInterval</key>
  <integer>900</integer>
  <key>UserName</key>
  <string>root</string>
</dict>
</plist>
```

## Key 1 - Label

```
<?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>Label</key>
  <string>com.jamfsoftware.task.Every 15 Minutes</string>
  <key>ProgramArguments</key>
  <array>
    <string>/usr/local/jamf/bin/jamf</string>
    <string>policy</string>
    <string>-randomDelaySeconds</string>
    <string>300</string>
  </array>
  <key>StartInterval</key>
  <integer>900</integer>
  <key>UserName</key>
  <string>root</string>
</dict>
</plist>
```

## Key 2 - Program Arguments

```
<?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>Label</key>
  <string>com.jamfsoftware.task.Every 15 Minutes</string>
  <key>ProgramArguments</key>
  <array>
    <string>/usr/local/jamf/bin/jamf</string>
    <string>policy</string>
    <string>-randomDelaySeconds</string>
    <string>300</string>
  </array>
  <key>StartInterval</key>
  <integer>900</integer>
  <key>UserName</key>
  <string>root</string>
</dict>
</plist>
```

## Key 3 - When to Start



# So You Want to Create One?

- Can be created as a text file
- Handy online generator:
- <https://zerolaunched.herokuapp.com>



# What Script Can You Use?

- Anything you want as long as:
  - It is executable
  - It runs as root
  - The computer can process the script





# When Will It Run?

- RunAtLoad - as soon as the computer boots
- StartInterval - every x seconds
- StartCalendarInterval - on a set schedule



```
<key>StartCalendarInterval</key>
<dict>
  <key>Hour</key>
  <integer>2</integer>
  <key>Minute</key>
  <integer>30</integer>
  <key>Weekday</key>
  <integer>2</integer>
</dict>
```

## StartCalendarInterval

- Similar to cron syntax
- Keys that aren't included are wildcards



# Where Do They Live?

- /System/Library/LaunchDaemons
- /System/Library/LaunchAgents
- /Library/LaunchDaemons
- /Library/LaunchAgents
- ~/Library/LaunchAgents



# Making Your Own LaunchDaemon Work

1. Create LaunchDaemon
2. Move LaunchDaemon to /Library/  
LaunchDaemons folder
3. Load with `launchctl load  
command`

```
rebecca.latimer -- zsh -- 80x24
Last login: Sun Sep 12 16:24:23 on ttys000
rebecca.latimer@Rebeccas-MacBook-Air ~ %
```

- AirDrop
- Recents
- Applications
- Desktop
- Documents
- Downloads
- iCloud
- iCloud Drive
- Locations
- Macintosh HD
- Network
- Tags
- Important

```
createFile.sh
1  #!/bin/sh
2
3  filename=/usr/local/newFile
4  date=$(date +"%Y-%m-%d %T")
5  if [ ! -f $filename ]
6  then
7      touch $filename
8      echo "$date $filename was created."
9  else
10     echo "$date $filename already exists."
11 fi
```

Screen Recording  
2021-09...PM.mov

- Calendar
- Contacts
- Messages
- Music
- Photos
- Reminders
- Safari
- System Preferences
- Terminal
- Utilities
- Weather
- App Store
- System Status
- System Settings
- System Information
- System Profiler
- System Log
- System Keychain
- System Preferences
- System Settings
- System Information
- System Profiler
- System Log
- System Keychain



# Troubleshooting

- Check logs
- Permissions
- Interval syntax
- Label





# More Resources



<https://gingerscripting.com/launchdaemons/>

Find Me:

**@GingerScripting** on Twitter  
**rebelati** on Slack





**Thank you for  
listening!**

