

The Software Lifecycle Whisperer

How Installomator Tames Install-Update-Automate Chaos





Who am I?

Assistant Head, Client Platform Services at Marriott Library

Background & Experience

- Assistant Head, Client Platform Services at University of Utah, Marriott Library
- 30+ years in IT enterprise management
- Conference Speaker at WWDC, JNUC, PSU MacAdmins Conference, and local groups (Mac-Mgrs & IT Pros)





Who am I?

Assistant Head, Client Platform Services at Marriott Library

• Background & Experience

- Assistant Head, Client Platform Services at University of Utah, Marriott Library
- 30+ years in IT enterprise management
- Conference Speaker at WWDC, JNUC, PSU MacAdmins Conference, and local groups (Mac-Mgrs & IT Pros)

Current Environment

- Scale: 3,000+ Mac and Windows devices, ~250 iPads, others
- Focus: Staff, student, and library technology services
- Challenges: High-volume device turnover, diverse software needs, rapid deployment requirements
- Solutions: Jamf Pro, Installomator, Microsoft Endpoint Manager, AppsAnywhere, and custom automation workflows for device lifecycle management







Who am I?

Assistant Head, Client Platform Services at Marriott Library

Background & Experience

- Assistant Head, Client Platform Services at University of Utah, Marriott Library
- 30+ years in IT enterprise management
- Conference Speaker at WWDC, JNUC, PSU MacAdmins Conference, and local groups (Mac-Mgrs & IT Pros)

Current Environment

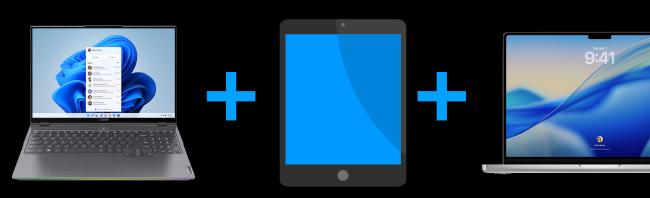
- Scale: 3,000+ Mac and Windows devices, ~250 iPads, others
- Focus: Staff, student, and library technology services
- Challenges: High-volume device turnover, diverse software needs, rapid deployment requirements
- Solutions: Jamf Pro, Installomator, Microsoft Endpoint Manager, AppsAnywhere, and custom automation workflows for device lifecycle management

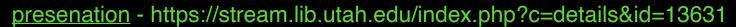
Student Checkout Device Program

- Device Management: Large-scale student laptop and tablet checkout programs
- Rapid Provisioning: Quick device onboarding and configuration processes
- Mixed Platforms: Windows, macOS, and iPadOS device support and management
- Volume Operations: High-throughput device imaging, deployment, and maintenance













PXE, WinPE, Microsoft Endpoint Configuration Manager

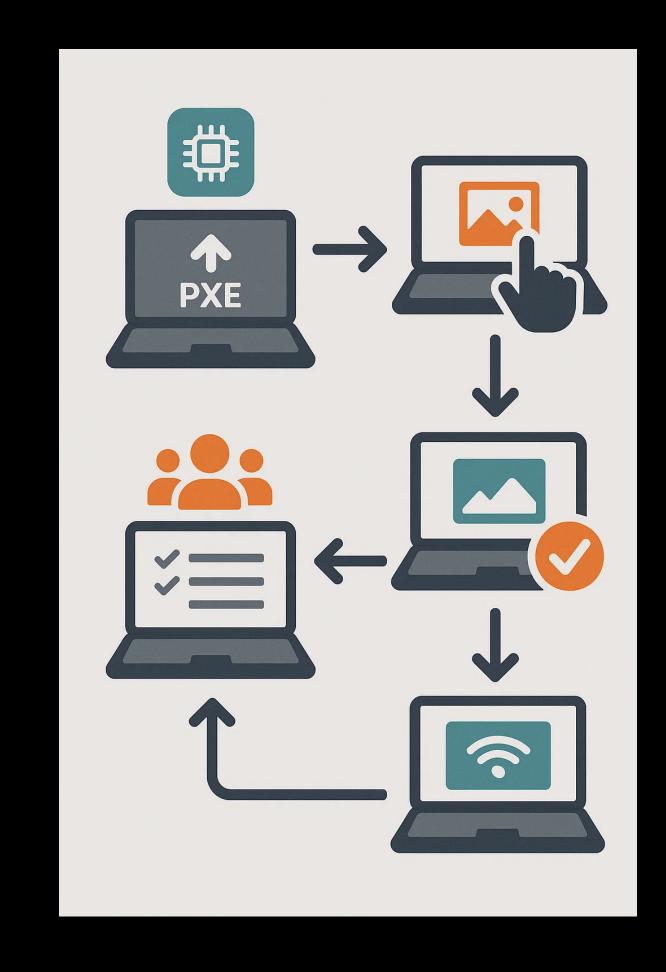
Automated Windows Imaging



- Automated Windows Imaging
 - Boot via PXE

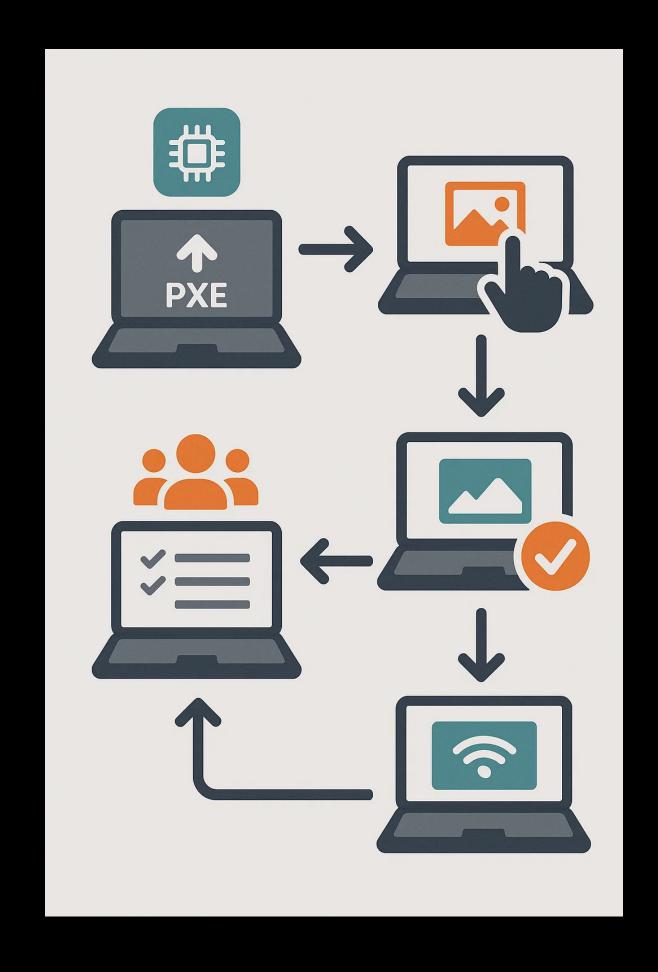
 Boot via PXE after BIOS changes and loads the imaging (WinPE) environment

 https://learn.microsoft.com/en-us/windows-hardware/manufacture/desktop/winpe-intro?view=windows-11





- Automated Windows Imaging
 - Boot via PXE
 Boot via PXE after BIOS changes and loads the imaging (WinPE) environment
 https://learn.microsoft.com/en-us/windows-hardware/manufacture/desktop/winpe-intro?view=windows-11
 - <u>Select Image</u>
 Select system image to image Windows System (i.e. student, staff, etc.)

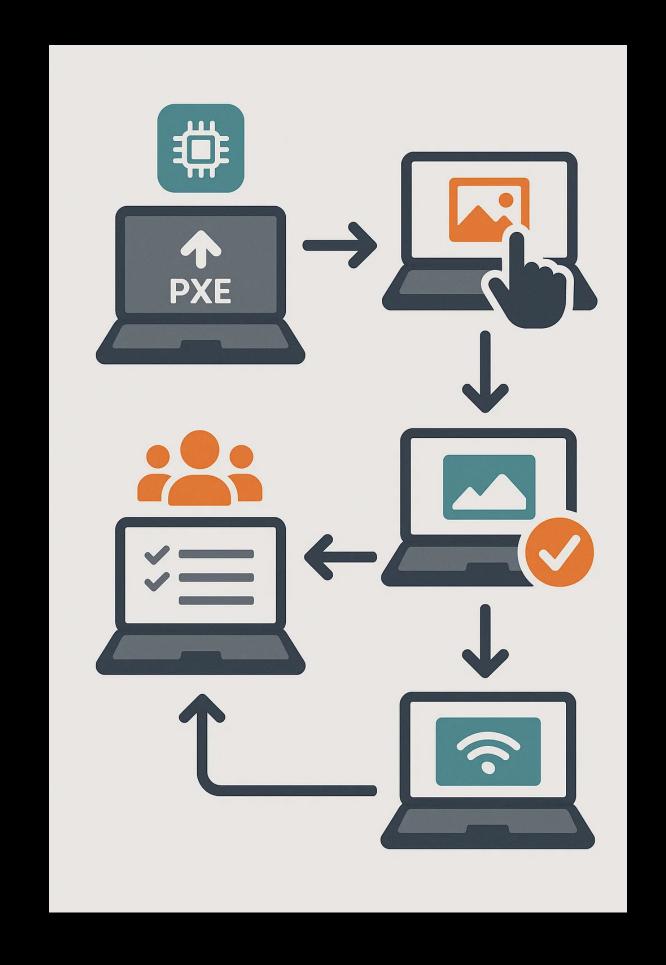




- Automated Windows Imaging
 - Boot via PXE
 Boot via PXE after BIOS changes and loads the imaging (WinPE) environment
 https://learn.microsoft.com/en-us/windows-hardware/manufacture/desktop/winpe-intro?view=windows-11
 - <u>Select Image</u>
 Select system image to image Windows System (i.e. student, staff, etc.)
 - Staging

 Approve target computer, add to device collections

 https://learn.microsoft.com/en-us/intune/configmgr/core/understand/introduction

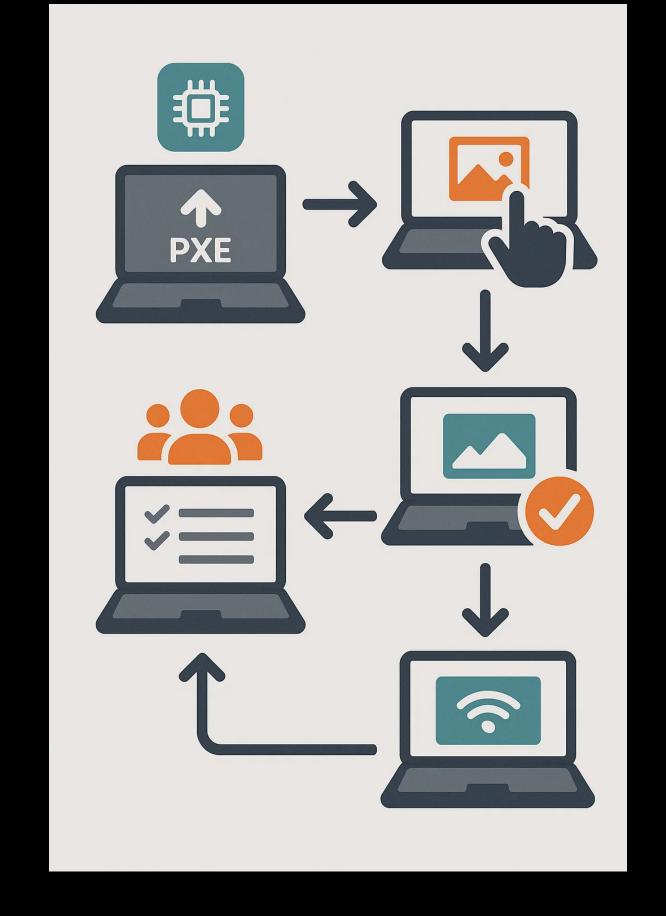




- Automated Windows Imaging
 - Boot via PXE
 Boot via PXE after BIOS changes and loads the imaging (WinPE) environment
 https://learn.microsoft.com/en-us/windows-hardware/manufacture/desktop/winpe-intro?view=windows-11
 - <u>Select Image</u>
 Select system image to image Windows System (i.e. student, staff, etc.)
 - Staging

 Approve target computer, add to device collections

 https://learn.microsoft.com/en-us/intune/configmgr/core/understand/introduction
 - Verification
 Check wallpaper presence, software availability, and WiFi connectivity





iMazing Configurator, Cambrionix Hubs

Easy Automated Imaging



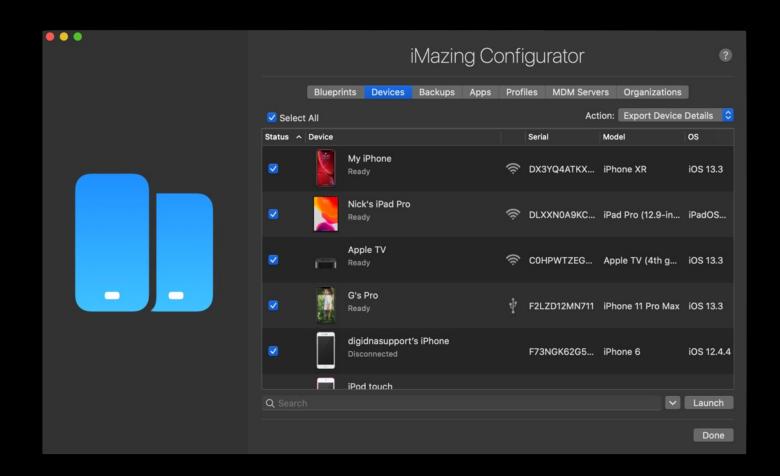
iMazing Configurator, Cambrionix Hubs

- Easy Automated Imaging
 - Quick Return to Service
 Connect the iPad to a USB Hub on the iMac with iMazing Configurator to start automated imaging.



iMazing Configurator, Cambrionix Hubs

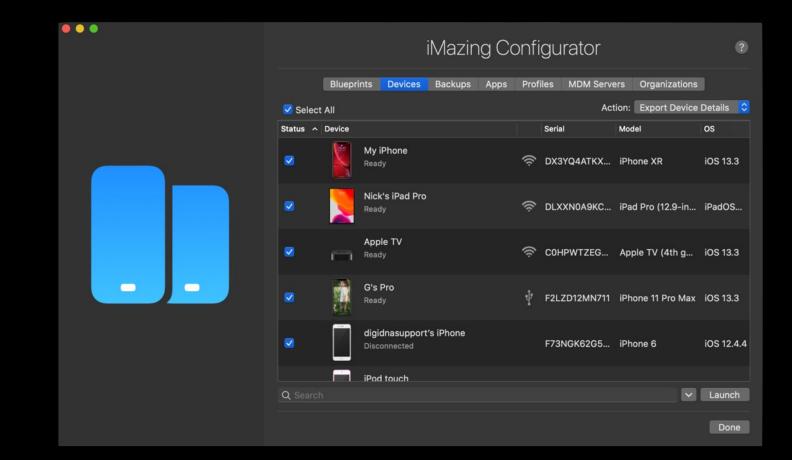
- Easy Automated Imaging
 - Quick Return to Service
 Connect the iPad to a USB Hub on the iMac with iMazing Configurator to start automated imaging.
 - <u>iMazing Configurator</u>
 Automates imaging, connects to campus wireless, and re-enrolls in Jamf Pro
 https://imazing.com/configurator





iMazing Configurator, Cambrionix Hubs

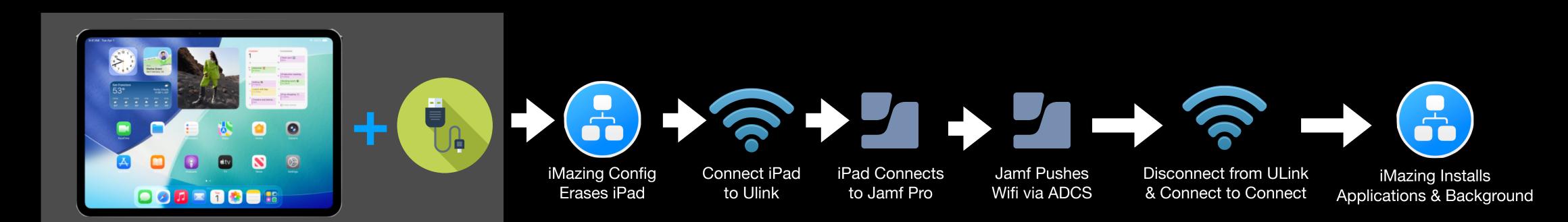
- Easy Automated Imaging
 - Quick Return to Service
 Connect the iPad to a USB Hub on the iMac with iMazing Configurator to start automated imaging.
 - <u>iMazing Configurator</u>
 Automates imaging, connects to campus wireless, and re-enrolls in Jamf Pro
 https://imazing.com/configurator
 - Cambrionix ThunderSync3-C10 USB Hub
 High-speed, multi-device iPad imaging with advanced control features
 https://www.cambrionix.com/products/thundersync3-c10







iPad Imaging Flow



Attach iPad to USB



DFU Blaster Pro, Acroname Hubs

Easy Automated Imaging



DFU Blaster Pro, Acroname Hubs

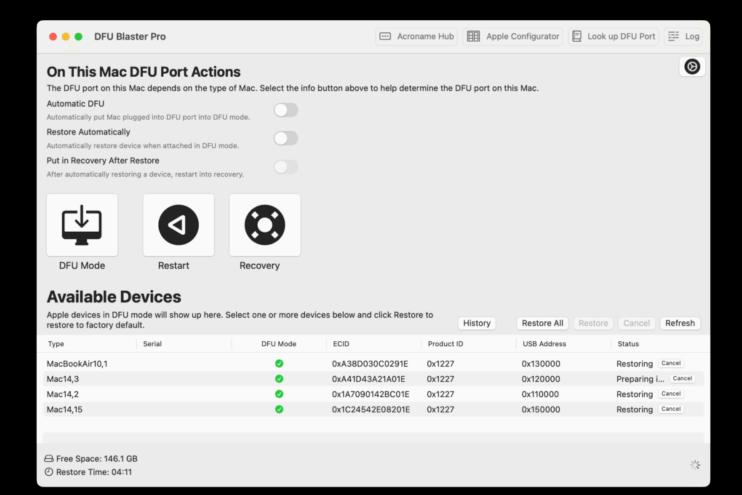
- Easy Automated Imaging
 - Quick Return to Service

 Connect the Laptop to the USB Hub with DFU Blaster Pro to start imaging



DFU Blaster Pro, Acroname Hubs

- Easy Automated Imaging
 - Quick Return to Service
 Connect the Laptop to the USB Hub with DFU Blaster Pro to start imaging
 - DFU Blaster Pro
 Automates imaging, DFU mode and re-enrolls in Jamf Pro https://twocanoes.com/products/mac/dfu-blaster/





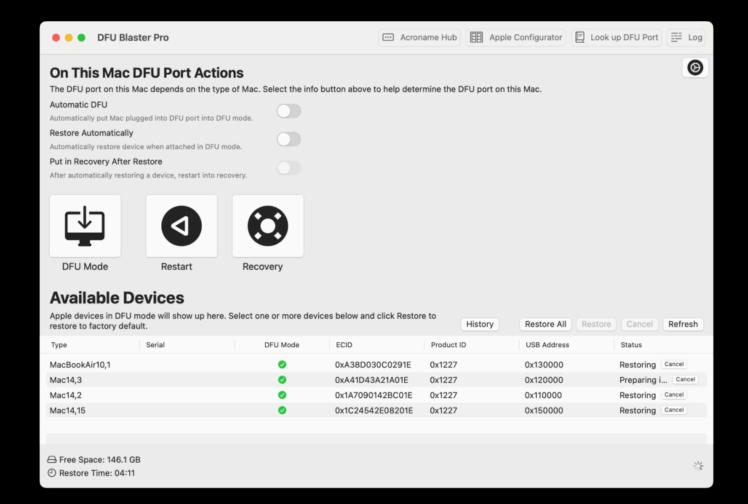
DFU Blaster Pro, Acroname Hubs

- Easy Automated Imaging
 - Quick Return to Service

 Connect the Laptop to the USB Hub with DFU Blaster Pro to start imaging
 - DFU Blaster Pro
 Automates imaging, DFU mode and re-enrolls in Jamf Pro
 https://twocanoes.com/products/mac/dfu-blaster/
 - Acroname USBHub3c Pro Edition Hub

 High-speed connectivity and programmable control for imaging multiple Macs

 https://acroname.com/store/programmable-industrial-power-delivery-hub-s99-usbhub-3c-pro



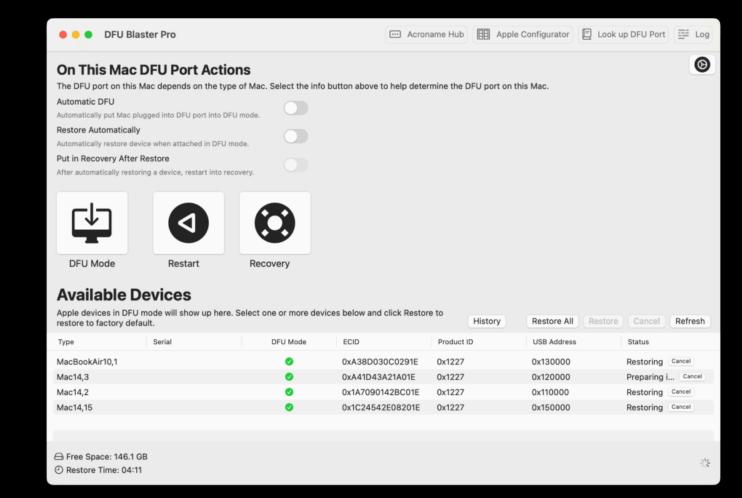


DFU Blaster Pro, Acroname Hubs

- Easy Automated Imaging
 - Quick Return to Service
 Connect the Laptop to the USB Hub with DFU Blaster Pro to start imaging
 - DFU Blaster Pro
 Automates imaging, DFU mode and re-enrolls in Jamf Pro
 https://twocanoes.com/products/mac/dfu-blaster/
 - Acroname USBHub3c Pro Edition Hub

 High-speed connectivity and programmable control for imaging multiple Macs

 https://acroname.com/store/programmable-industrial-power-delivery-hub-s99-usbhub-3c-pro
 - Satechi 200W USB-C 6-Port GaN Charger
 Provide a baseline battery charge to enable imaging of laptops with low battery https://satechi.net/products/6-port-gan-charger?variant=40197715492952

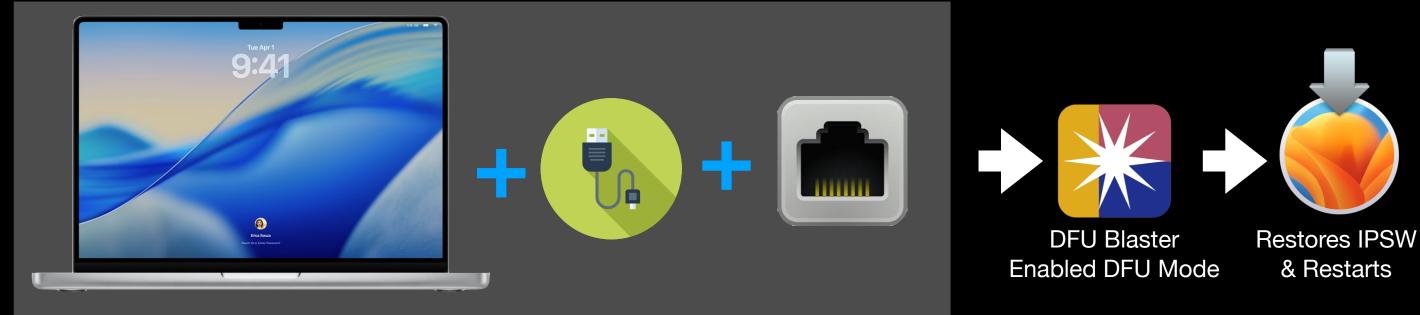


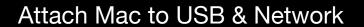






Mac Imaging Flow









S.U.P.E.R.M.A.N., swiftDialog, outset & custom scripts

Easy Automated Maintenance



S.U.P.E.R.M.A.N., swiftDialog, outset & custom scripts

- Easy Automated Maintenance
 - Quick Return to Service

Auto-run Jamf policies on network-connected laptops to remove old accounts, non-default software & update macOS



S.U.P.E.R.M.A.N., swiftDialog, outset & custom scripts

- Easy Automated Maintenance
 - Quick Return to Service
 Auto-run Jamf policies on network-connected laptops to remove old accounts, non-default software & update macOS
 - <u>S.U.P.E.R.M.A.N.</u>

Open source script that automates software updates, upgrades, and restart scheduling for enterprises https://github.com/Macjutsu/super



S.U.P.E.R.M.A.N., swiftDialog, outset & custom scripts

Easy Automated Maintenance

• Quick Return to Service

Auto-run Jamf policies on network-connected laptops to remove old accounts, non-default software & update macOS

• <u>S.U.P.E.R.M.A.N.</u>

Open source script that automates software updates, upgrades, and restart scheduling for enterprises https://github.com/Macjutsu/super

swiftDialog

Open Source utility that creates customizable dialog boxes and user interfaces for administrative scripts and workflows https://github.com/swiftDialog/swiftDialog/swiftDialog



S.U.P.E.R.M.A.N., swiftDialog, outset & custom scripts

Easy Automated Maintenance

Quick Return to Service

Auto-run Jamf policies on network-connected laptops to remove old accounts, non-default software & update macOS

• <u>S.U.P.E.R.M.A.N.</u>

Open source script that automates software updates, upgrades, and restart scheduling for enterprises https://github.com/Macjutsu/super

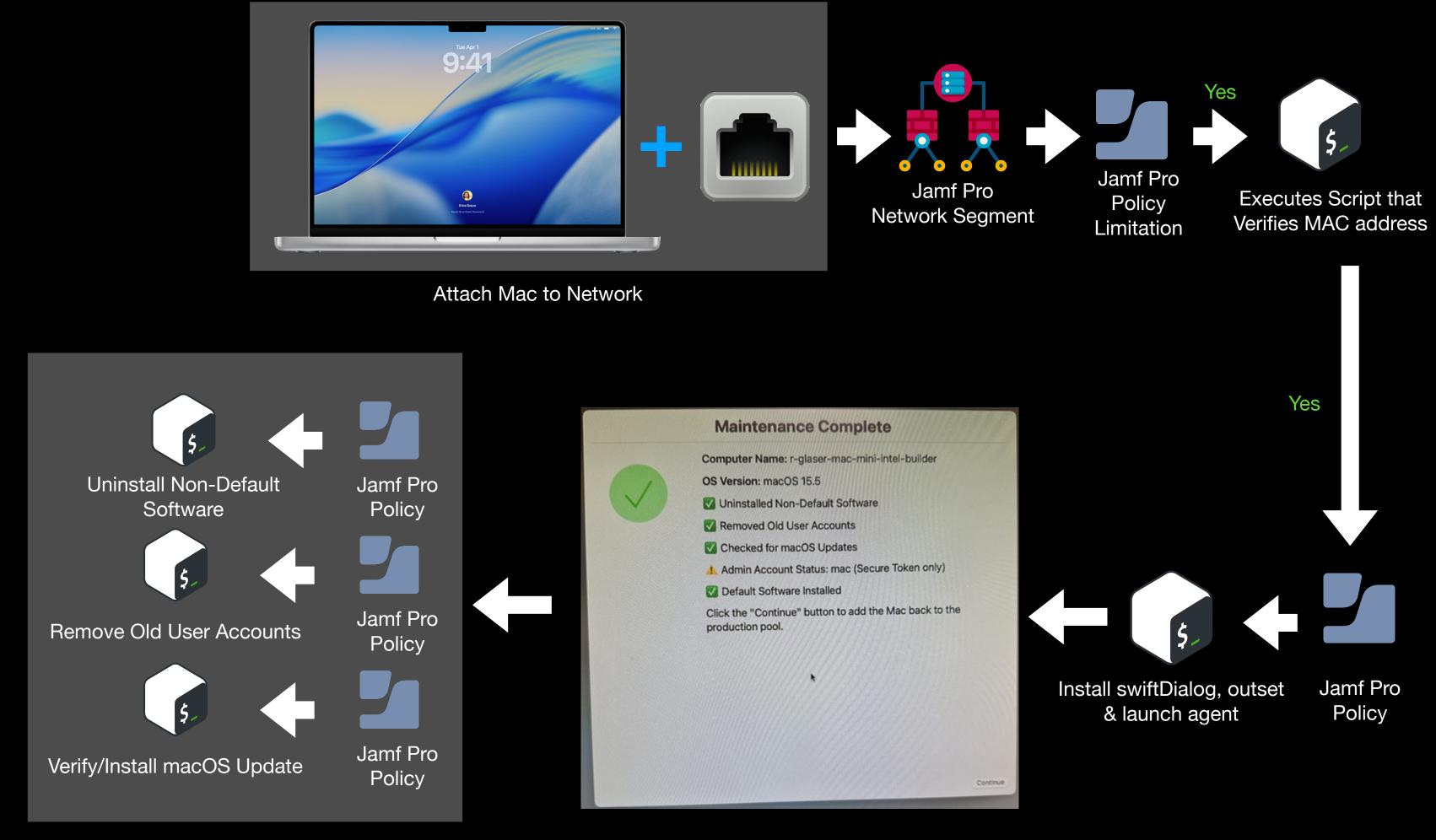
swiftDialog

Open Source utility that creates customizable dialog boxes and user interfaces for administrative scripts and workflows <a href="https://github.com/swiftDialog/sw

outset

Open Source tool that runs scripts/apps at startup, login, or on-demand based on folder organization https://github.com/macadmins/outset

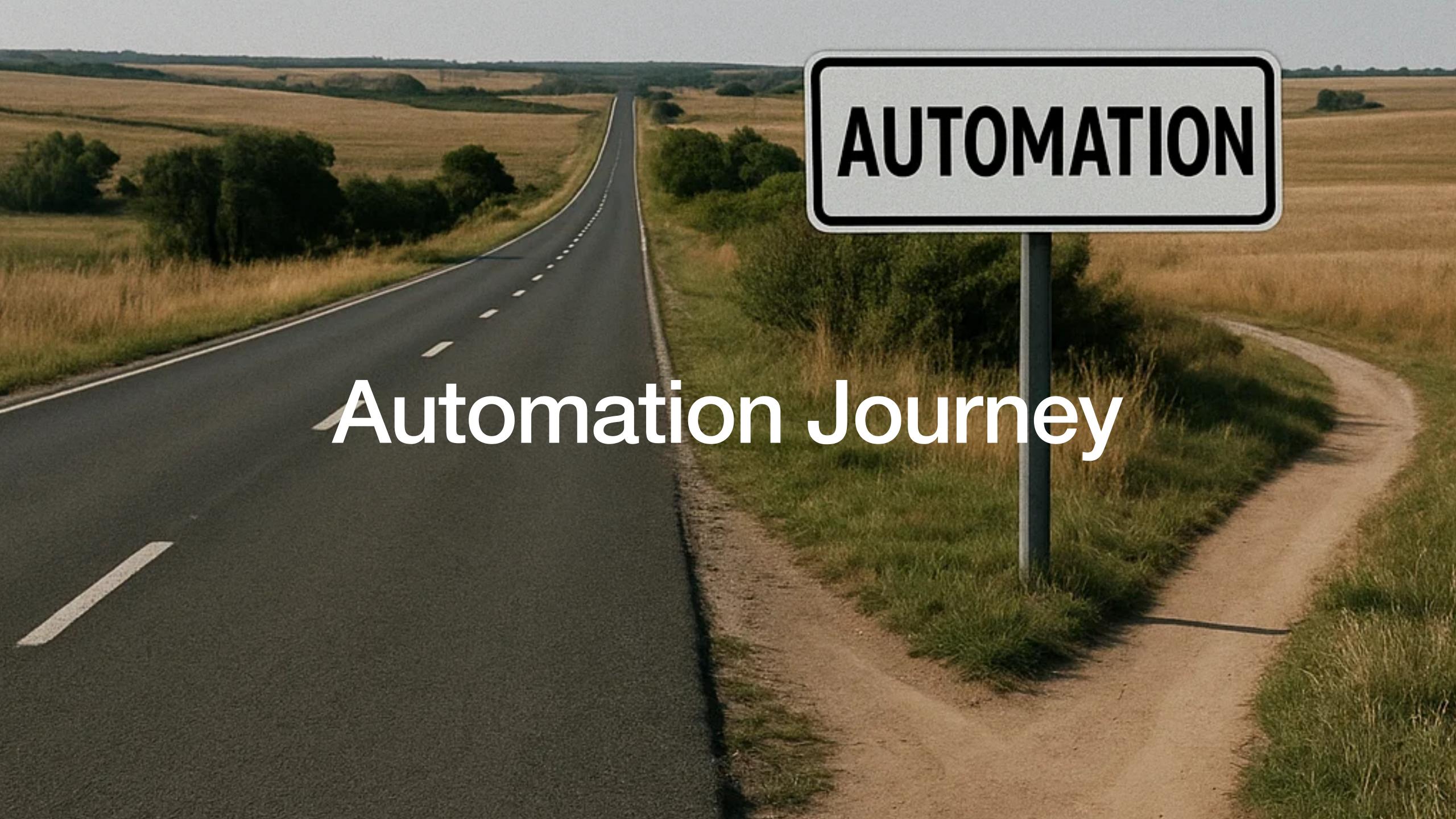






Displays Dialog over Login Window





Automation Journey

Our Road to Installomator

- The COVID-19 Catalyst (2020-2021)
 - Campus Closure: Sudden shift to remote learning and work
 - CARES Act Funding: Federal support for student technology access
 - Emergency Deployment: Rapid student checkout laptop program expansion
 - Decentralized Operations: Staff working from home, distributed device management







Automation Journey

Our Road to Installomator

- The COVID-19 Catalyst (2020-2021)
 - Campus Closure: Sudden shift to remote learning and work
 - CARES Act Funding: Federal support for student technology access
 - Emergency Deployment: Rapid student checkout laptop program expansion
 - Decentralized Operations: Staff working from home, distributed device management
- The Challenge Explosion
 - Volume Surge: From hundreds to 1,500+ laptops needing management
 - Software Requests: Exponential increase in diverse application needs
 - Remote Support: Troubleshooting and updates without physical access
 - Speed Requirements: Students needed devices configured and ready immediately









































Automation Journey

Our Road to Installomator

The COVID-19 Catalyst (2020-2021)

- Campus Closure: Sudden shift to remote learning and work
- CARES Act Funding: Federal support for student technology access
- Emergency Deployment: Rapid student checkout laptop program expansion
- Decentralized Operations: Staff working from home, distributed device management

The Challenge Explosion

- Volume Surge: From hundreds to 1,500+ laptops needing management
- Software Requests: Exponential increase in diverse application needs
- Remote Support: Troubleshooting and updates without physical access
- Speed Requirements: Students needed devices configured and ready immediately





































Traditional Bottlenecks

- Application Packaging Delays: 1-2 day to create/update packages for new software requests
- Infrastructure Strain: Manual processes overwhelmed
- Staff Burnout: Manual processes couldn't scale with demand



macOS Software Lifecycle

macOS Software Installation Lifecycle with Jamf Pro

Traditional Workflow



macOS Software Installation Lifecycle with Jamf Pro

Traditional Workflow

• Download & Expand Software - from vendor (DMG, ZIP, PKG, etc.).









macOS Software Installation Lifecycle with Jamf Pro

Traditional Workflow

- Download & Expand Software from vendor (DMG, ZIP, PKG, etc.).
- Repackage into PKG using Composer, Packages, munkipkg, etc.









macOS Software Installation Lifecycle with Jamf Pro

Traditional Workflow

- Download & Expand Software from vendor (DMG, ZIP, PKG, etc.).
- Repackage into PKG using Composer, Packages, munkipkg, etc.
- <u>Upload to Jamf Pro</u> configure policies, scope, smart groups.













macOS Software Installation Lifecycle with Jamf Pro

Traditional Workflow

- Download & Expand Software from vendor (DMG, ZIP, PKG, etc.).
- Repackage into PKG using Composer, Packages, munkipkg, etc.
- <u>Upload to Jamf Pro</u> configure policies, scope, smart groups.
- Deploy to clients Jamf pushes package to Macs.









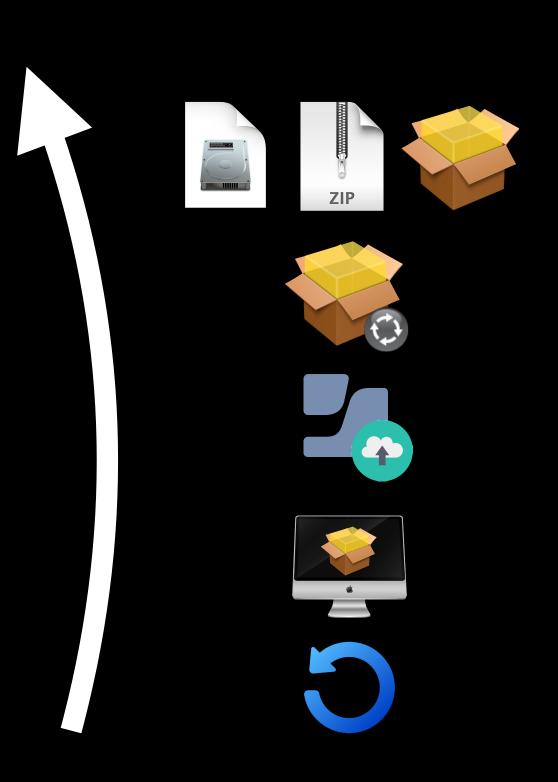




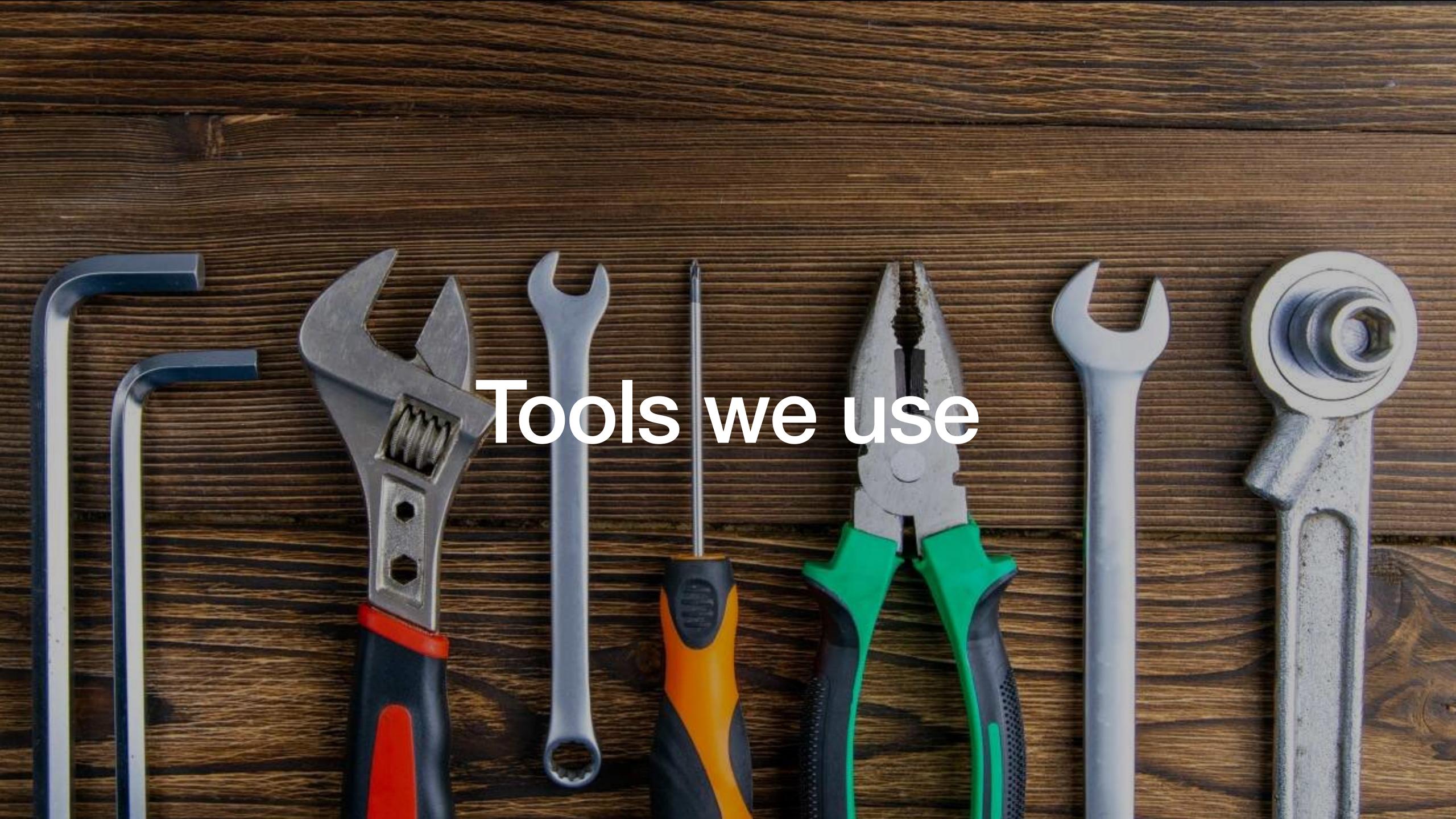
macOS Software Installation Lifecycle with Jamf Pro

Traditional Workflow

- Download & Expand Software from vendor (DMG, ZIP, PKG, etc.).
- Repackage into PKG using Composer, Packages, munkipkg, etc.
- <u>Upload to Jamf Pro</u> configure policies, scope, smart groups.
- Deploy to clients Jamf pushes package to Macs.
- Repeat for updates cycle restarts for each new version.







Building the Foundation Before Installomator

munkipkg Implementation

https://github.com/munki/munki-pkg

- Open Source
 Support by a community of MacAdmins & developers
- Collaboration-Friendly
 Projects can be easily checked into Git and shared among teams
- Package Format
 Standard Apple installer packages
- Automatable
 It's suited for CI/CD workflows and automated build processes
- Human-Readable Configuration
 Configurations are managed in easy-to-read YAML, JSON, or plist formats





Building the Foundation Before Installomator

python-jamf & jctl

JNUC 2021 - https://www.youtube.com/watch?v=2YLriNwyP3s

python-jamf

Library for easy access to Jamf Pro API data.

https://github.com/univ-of-utah-marriott-library-apple/python-jamf

• jctl

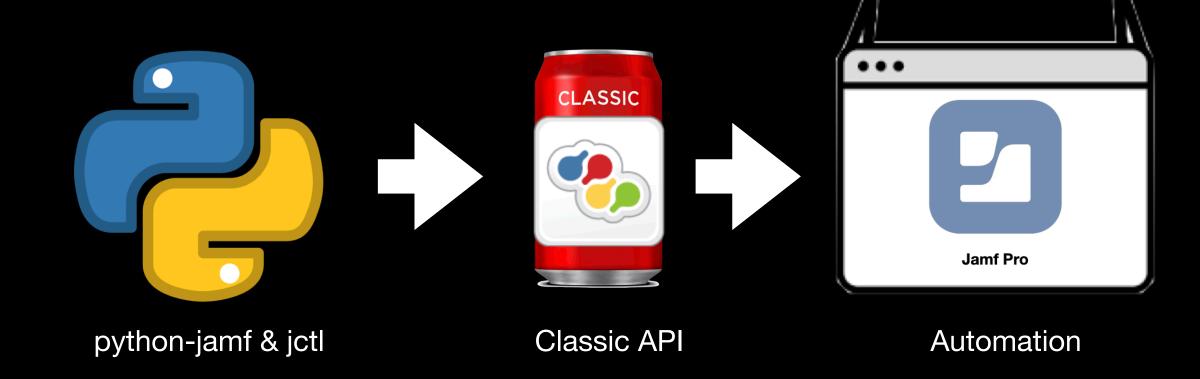
CLI tool using python-jamf for CRUD operations on Jamf Pro/Cloud, simplifying repetitive UI tasks into single commands.

Create, Read, Update, Delete—on Jamf Pro or Jamf Cloud servers

https://github.com/univ-of-utah-marriott-library-apple/jctl

Usage

- Policy Management Update Policies & Installers or Scripts
- Patch Management Update Patch Policies (dev, test, prod)
- Any manual task supported by Jamf API & tools





Patch Management

• Jamf Patch Management

For notable applications and updates can manage repackaged application patching

https://learn.jamf.com/en-US/bundle/jamf-pro-documentation-current/page/PatchManagement.html



Jamf maintains a centralized catalog of software definitions that includes version information, download locations, and installation criteria.

https://learn.jamf.com/en-US/bundle/jamf-app-catalog/page/Patch_Management_Software_Titles.html

Jamf Title Editor

A cloud-hosted solution that enhances Jamf Pro's patch management by enabling custom software title definitions. Powered by Kinobi technology, it serves as an external patch source in Jamf Pro.

https://learn.jamf.com/en-US/bundle/title-editor/page/About_Title_Editor.html

• Patch management leverages Jamf's Smart Groups

Current software versions, operating system compatibility, user assignments or departmental groupings





Apps That Don't Use Installomator

- Mac App Store Applications
 - Repackage Mac App Store Applications
 Use munkipkg to repackage Mac App Store applications.





- Mac App Store Applications
 - Repackage Mac App Store Applications
 Use munkipkg to repackage Mac App Store applications.
- Why?



















- Mac App Store Applications
 - Repackage Mac App Store Applications
 Use munkipkg to repackage Mac App Store applications.
- Why?
 - <u>Post-Installation Workflows / Extra Content</u>

 Mac App Store application require post install workflows or content (Xcode, Logic Pro, GarageBand, Final Cut Pro)







- Mac App Store Applications
 - Repackage Mac App Store Applications
 Use munkipkg to repackage Mac App Store applications.
- Why?
 - <u>Post-Installation Workflows / Extra Content</u>

 Mac App Store application require post install workflows or content (Xcode, Logic Pro, GarageBand, Final Cut Pro)
 - <u>Install Failures</u>
 Mac App Store installations frequently timeout or fail in enterprise environments



















- Mac App Store Applications
 - Repackage Mac App Store Applications
 Use munkipkg to repackage Mac App Store applications.
- Why?
 - <u>Post-Installation Workflows / Extra Content</u>
 Mac App Store application require post install workflows or content (Xcode, Logic Pro, GarageBand, Final Cut Pro)
 - <u>Install Failures</u>
 Mac App Store installations frequently timeout or fail in enterprise environments
 - <u>Licensing Complexity</u>
 Managing device vs user licenses across large organizations becomes administratively burdensome



















Building the Foundation Before Installomator

Mac App Store Applications

Repackage Mac App Store Applications
 Use munkipkg to repackage Mac App Store applications.

Why?

• <u>Post-Installation Workflows / Extra Content</u>

Mac App Store application require post install workflows or content (Xcode, Logic Pro, GarageBand, Final Cut Pro)

Install Failures

Mac App Store installations frequently timeout or fail in enterprise environments

• Licensing Complexity

Managing device vs user licenses across large organizations becomes administratively burdensome

Customization Options

Installers can include enterprise-specific configurations, licenses, or modifications



















Building the Foundation Before Installomator

Mac App Store Applications

Repackage Mac App Store Applications
 Use munkipkg to repackage Mac App Store applications.

Why?

• <u>Post-Installation Workflows / Extra Content</u>

Mac App Store application require post install workflows or content (Xcode, Logic Pro, GarageBand, Final Cut Pro)

• Install Failures

Mac App Store installations frequently timeout or fail in enterprise environments

Licensing Complexity

Managing device vs user licenses across large organizations becomes administratively burdensome

Customization Options

Installers can include enterprise-specific configurations, licenses, or modifications

Version Management

IT can maintain more easily manage specific app versions without forced updates



















Building the Foundation Before Installomator

Mac App Store Applications

Repackage Mac App Store Applications
 Use munkipkg to repackage Mac App Store applications.

Why?

• <u>Post-Installation Workflows / Extra Content</u>

Mac App Store application require post install workflows or content (Xcode, Logic Pro, GarageBand, Final Cut Pro)

• Install Failures

Mac App Store installations frequently timeout or fail in enterprise environments

Licensing Complexity

Managing device vs user licenses across large organizations becomes administratively burdensome

Customization Options

Installers can include enterprise-specific configurations, licenses, or modifications

Version Management

IT can maintain more easily manage specific app versions without forced updates

<u>Limited Offline Capability</u>

Unlike traditional installers, VPP apps cannot be deployed in air-gapped or restricted network environments



















Non-Standard Commercial Installers

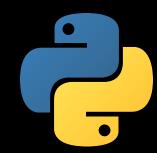


Non-Standard Commercial Installers





Non-Standard Commercial Installers



Repackage Commercial Applications

Use munkipkg to repackage Commercial applications with enterprise challenges

Non-Standard Installer

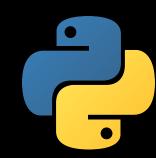
Some commercial applications use non-standard installed or install applications

- InstallAnywhere (Flexera) (MATLAB)
- Java-Based Installers (Autodesk)
- Virtual Disk / Self-Mounting Images (.iso, .toast) (Older Roxio or Corel products)
- Download/Installer Stub (like Fusion 360)





Non-Standard Commercial Installers



Repackage Commercial Applications

Use munkipkg to repackage Commercial applications with enterprise challenges

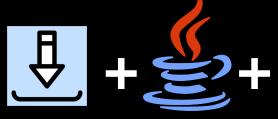
Non-Standard Installer

Some commercial applications use non-standard installed or install applications

- InstallAnywhere (Flexera) (MATLAB)
- Java-Based Installers (Autodesk)
- Virtual Disk / Self-Mounting Images (.iso, .toast) (Older Roxio or Corel products)
- Download/Installer Stub (like Fusion 360)

• Licensing Configuration

- Mathematica → Wolfram ID or license server
- Maple → FlexNet license server
- MATLAB → FlexNet or standalone license file
- Autodesk → Autodesk Licensing Service / FlexNet



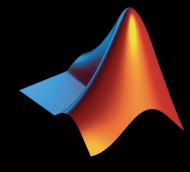




















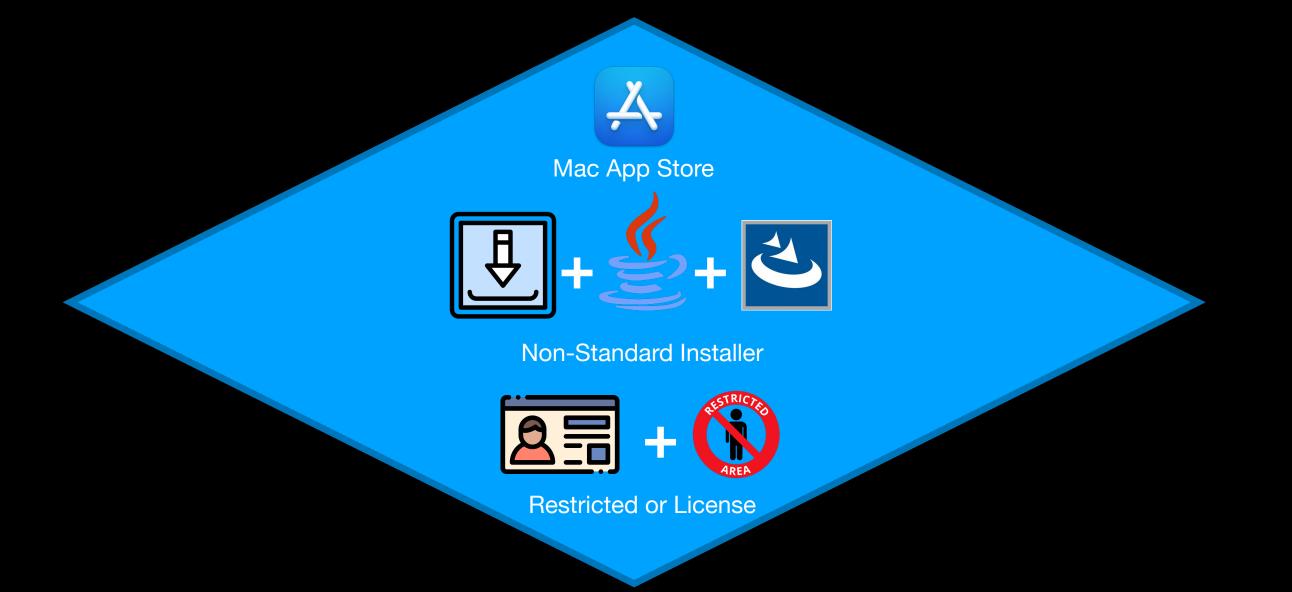
macOS Automation & Repackaging at Marriott Libra

- munkipkg
 Open-source packaging tool for automated, reproducible .pkg builds.
- <u>python-jamf & jctl</u>
 Simplify Jamf Pro/Cloud API operations (policy, patch, CRUD tasks).
- Repackaging Mac App Store Apps
 Handles post-install workflows, licensing, and failures (Xcode, Logic Pro, GarageBand, Final Cut Pro).
- Repackaging Commercial Apps
 For non-standard installers (MATLAB, Mathematica, Maple, Autodesk) with licensing/configuration needs.
- <u>Jamf Patch Management & Title Editor</u>

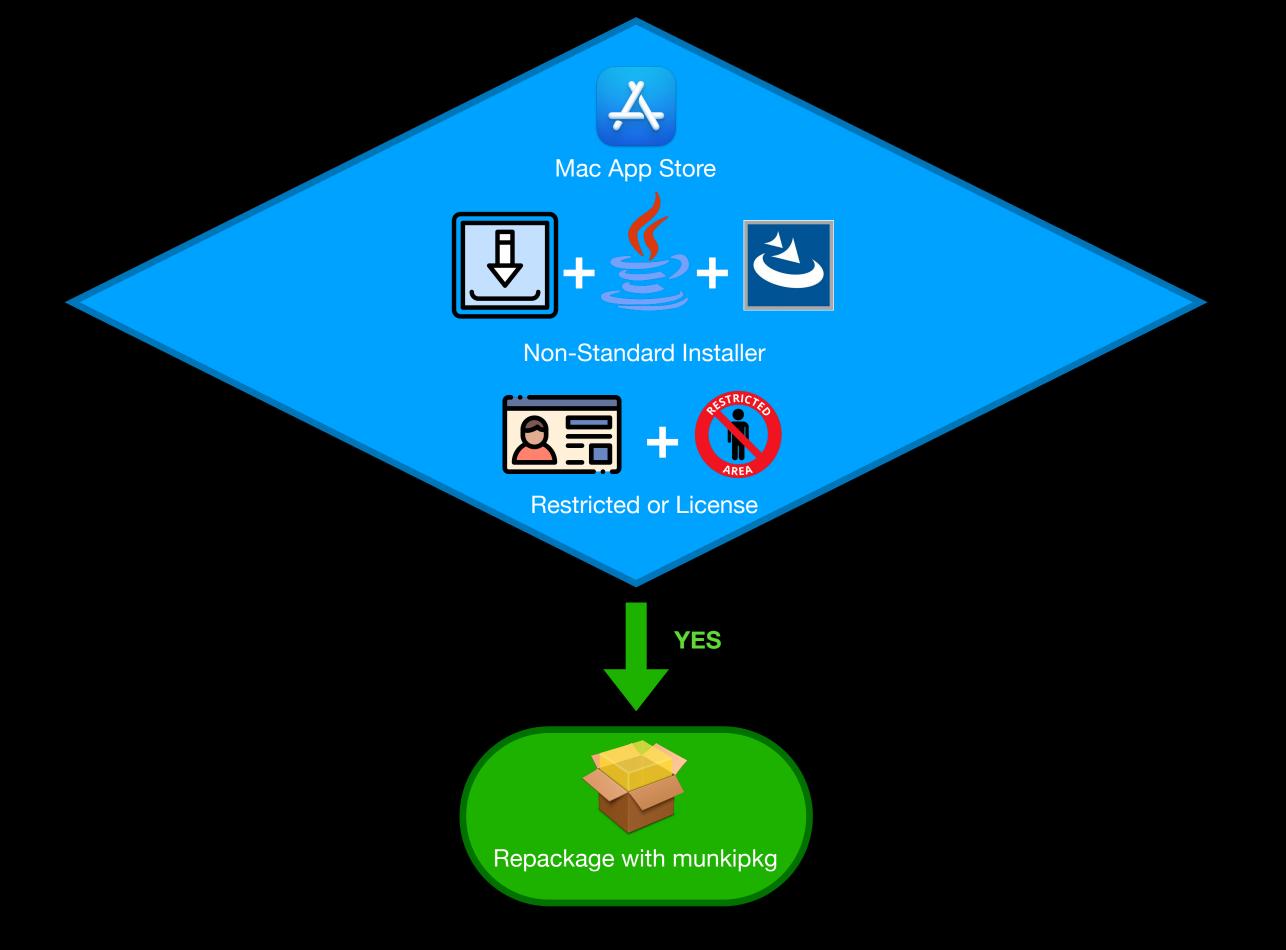
 Manage updates with Smart Groups, custom titles, and centralized software definitions.
- Custom Scripts & Keyboard Maestro
 Custom scripts & keyboard Maestro automate repetitive tasks, typing text, clicking, etc.



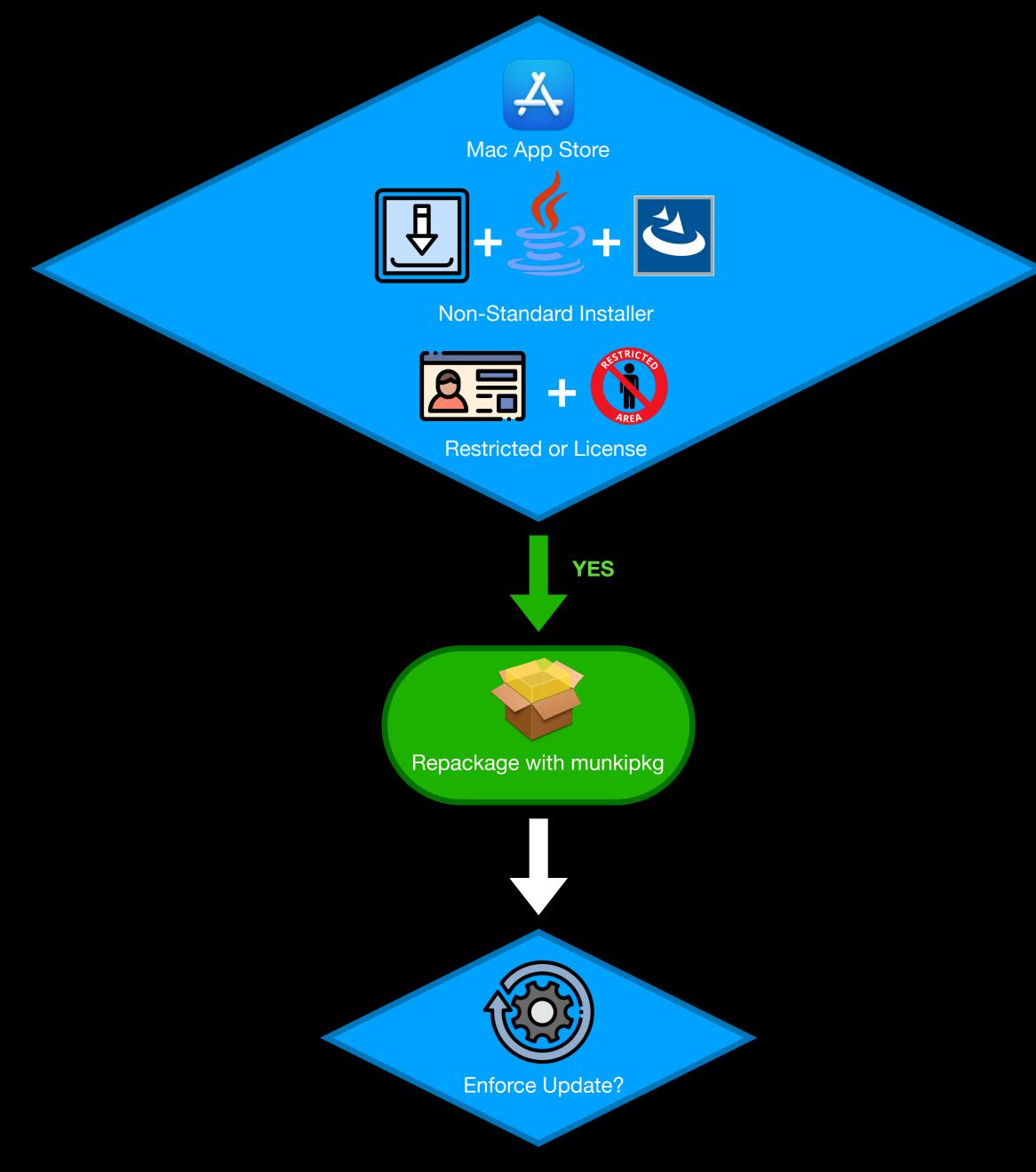




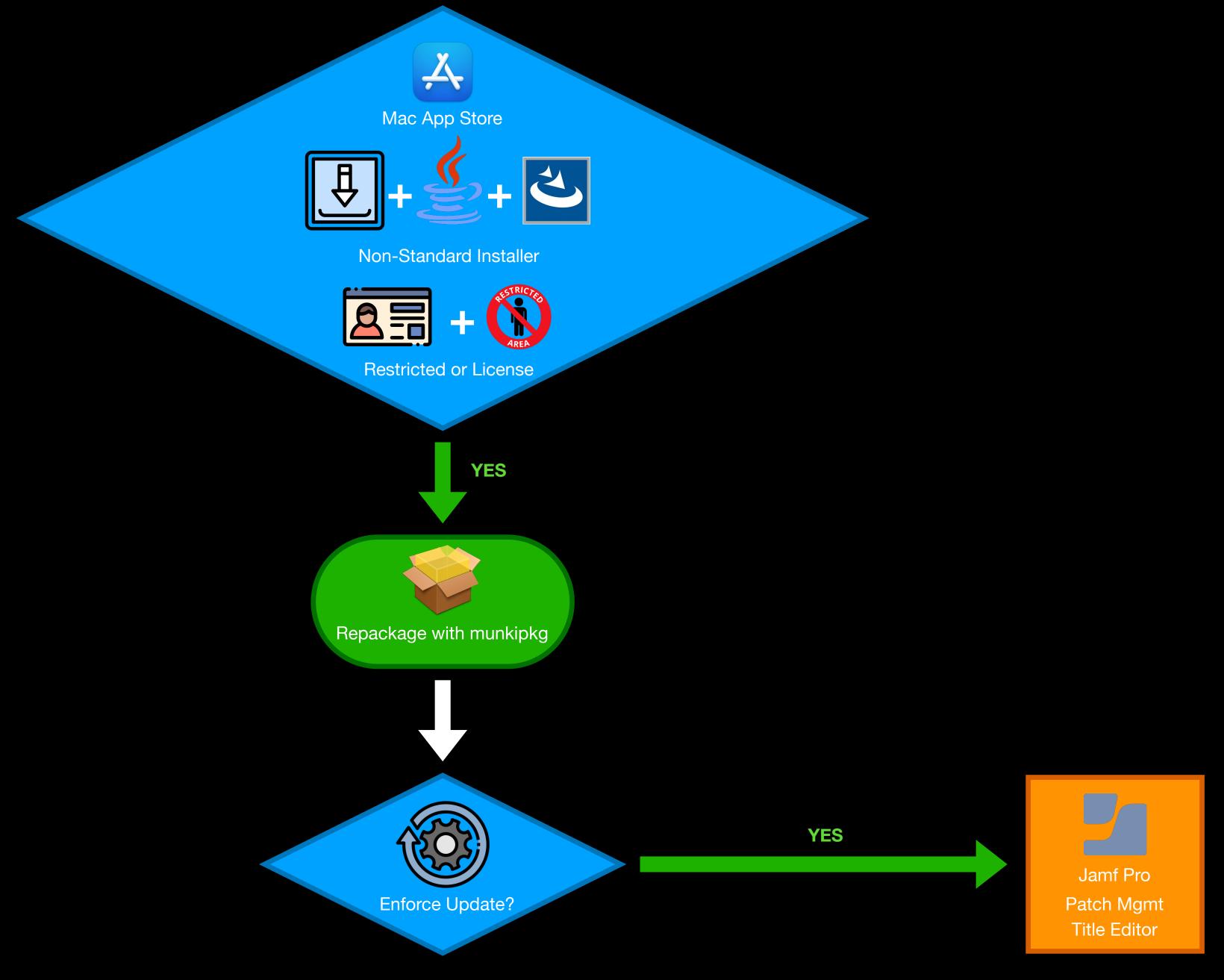




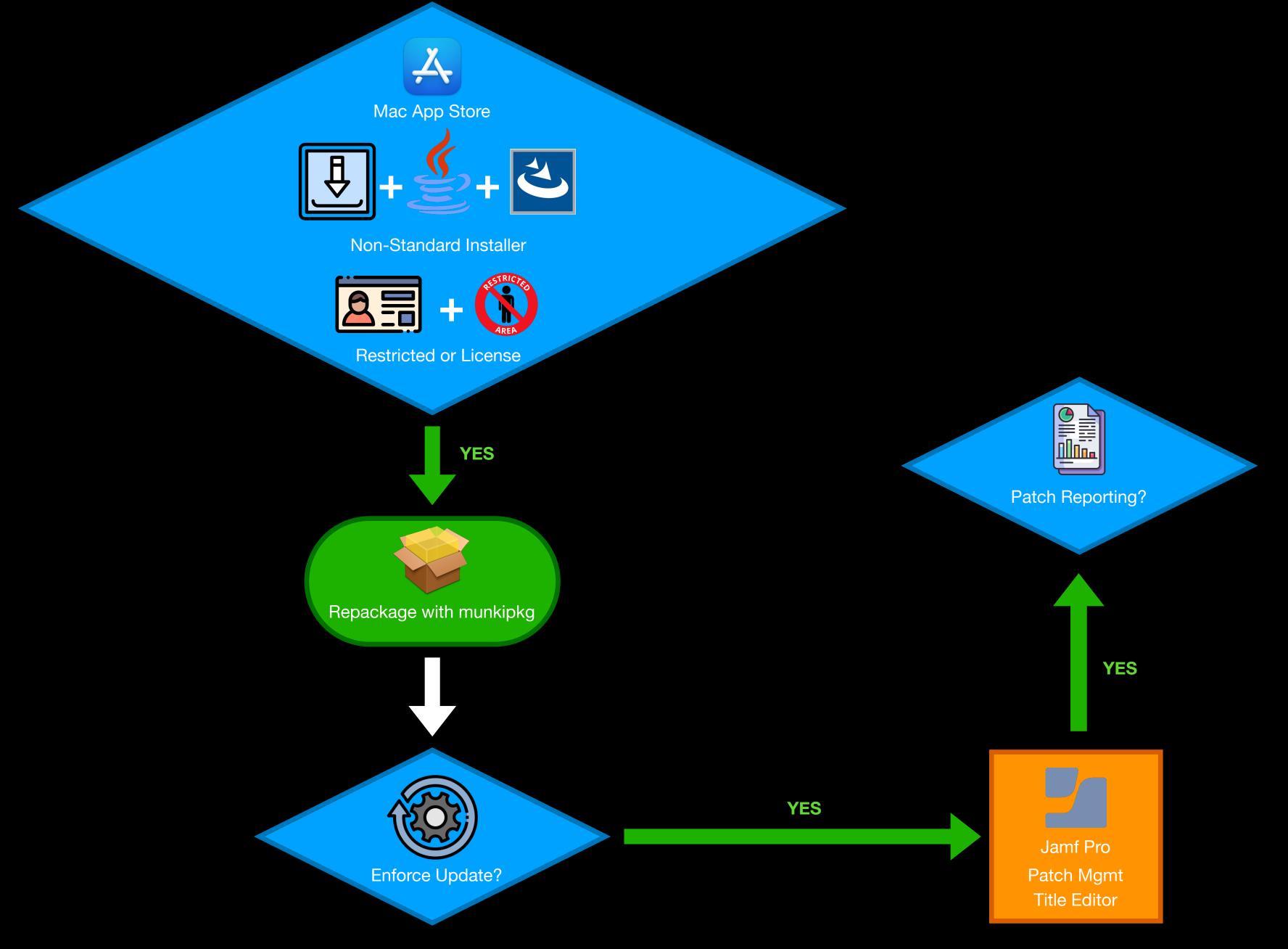






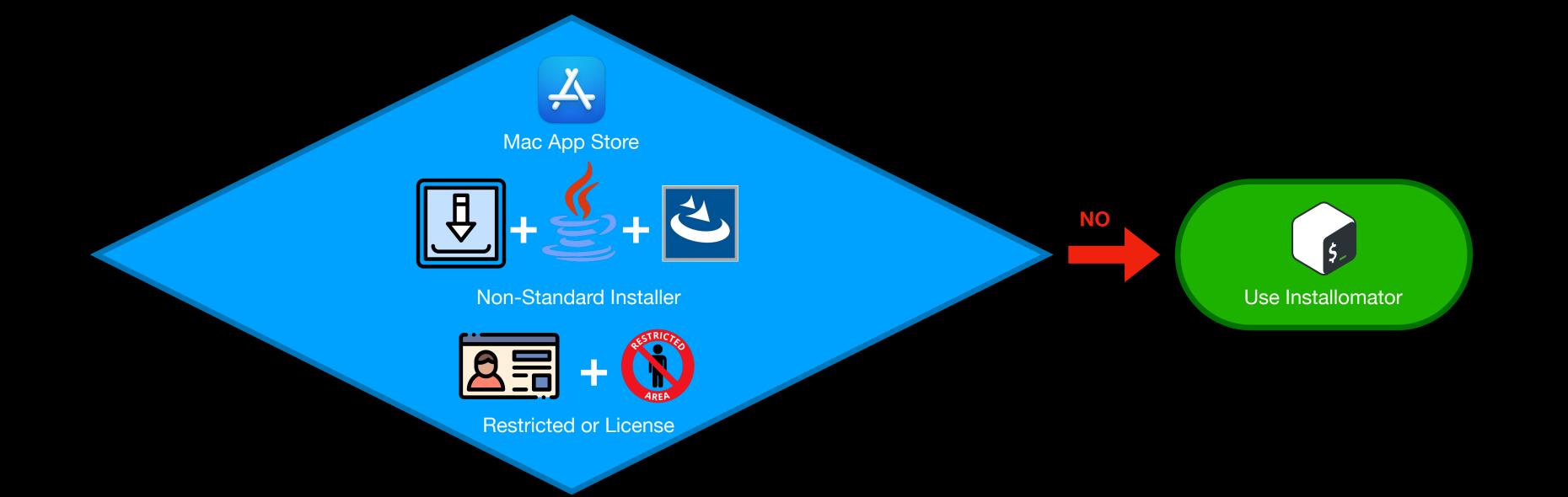




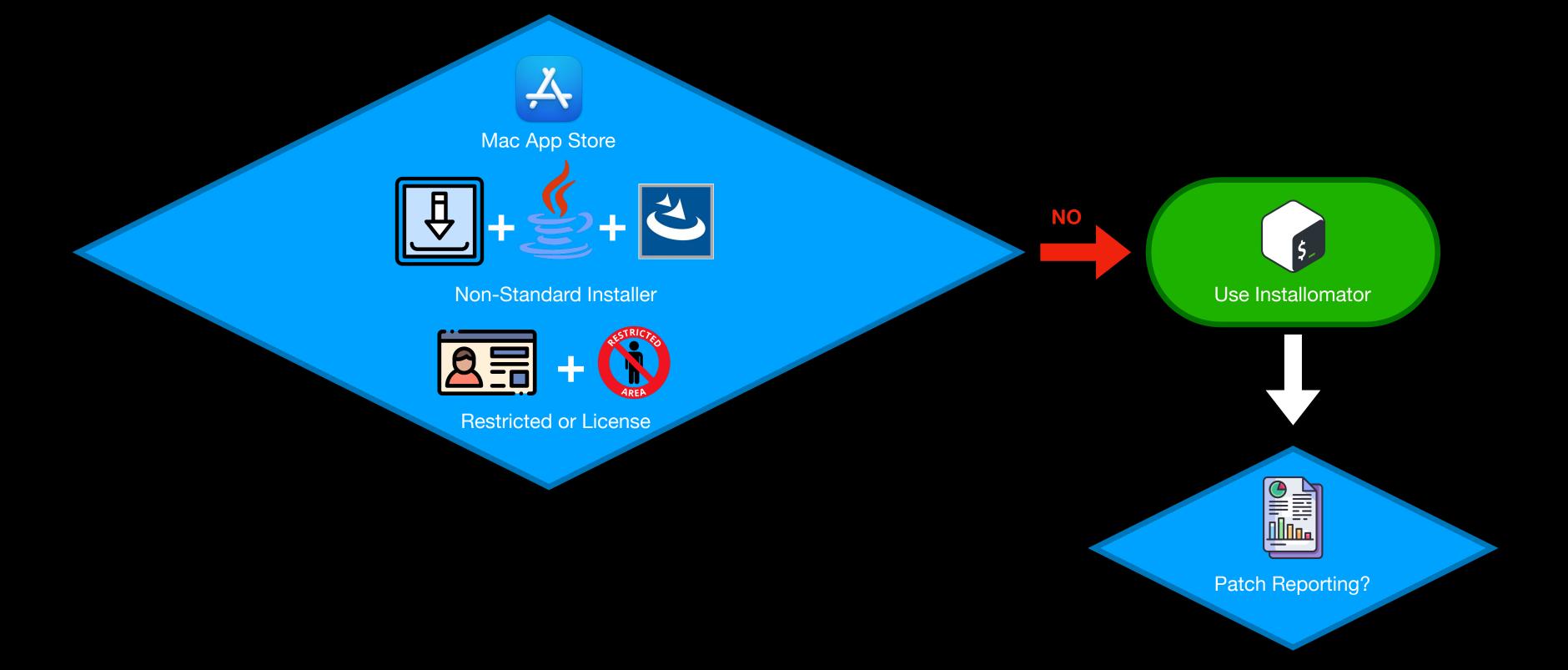




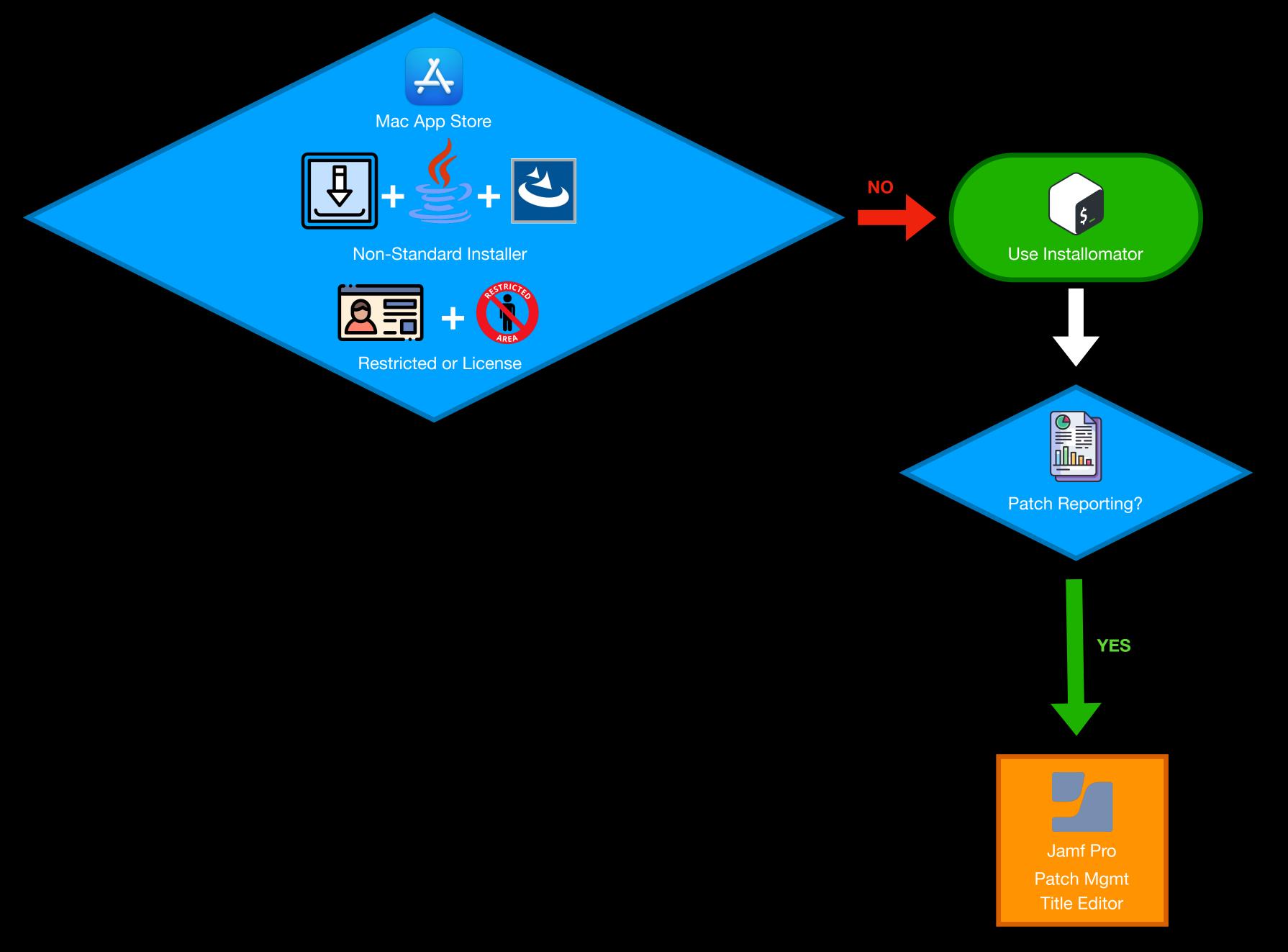
Installomator













macOS Automation & Repackaging at Marriott Library





macOS Automation & Repackaging at Marriott Library

Summary - Tools We Use to Streamline macOS Package Management

munkipkg
 Open-source packaging tool for automated, reproducible .pkg builds.





macOS Automation & Repackaging at Marriott Library

- munkipkg
 Open-source packaging tool for automated, reproducible .pkg builds.
- <u>python-jamf & jctl</u> Simplify Jamf Pro/Cloud API operations (policy, patch, CRUD tasks).





macOS Automation & Repackaging at Marriott Library

- munkipkg
 Open-source packaging tool for automated, reproducible .pkg builds.
- <u>python-jamf & jctl</u> Simplify Jamf Pro/Cloud API operations (policy, patch, CRUD tasks).
- Repackaging Mac App Store Apps
 Handles post-install workflows, licensing, and failures (Xcode, Logic Pro, GarageBand, Final Cut Pro).





macOS Automation & Repackaging at Marriott Library

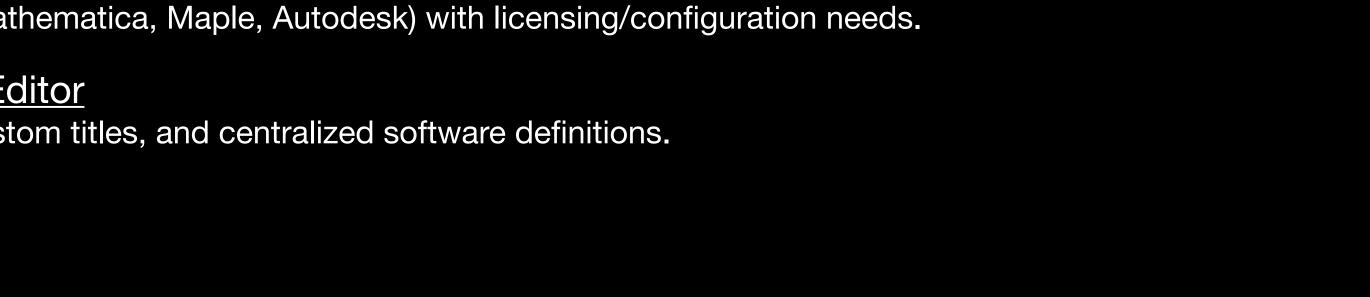
- munkipkg
 Open-source packaging tool for automated, reproducible .pkg builds.
- <u>python-jamf & jctl</u> Simplify Jamf Pro/Cloud API operations (policy, patch, CRUD tasks).
- Repackaging Mac App Store Apps
 Handles post-install workflows, licensing, and failures (Xcode, Logic Pro, GarageBand, Final Cut Pro).
- Repackaging Commercial Apps
 For non-standard installers (MATLAB, Mathematica, Maple, Autodesk) with licensing/configuration needs.





macOS Automation & Repackaging at Marriott Library

- munkipkg Open-source packaging tool for automated, reproducible .pkg builds.
- python-jamf & jctl Simplify Jamf Pro/Cloud API operations (policy, patch, CRUD tasks).
- Repackaging Mac App Store Apps Handles post-install workflows, licensing, and failures (Xcode, Logic Pro, GarageBand, Final Cut Pro).
- Repackaging Commercial Apps For non-standard installers (MATLAB, Mathematica, Maple, Autodesk) with licensing/configuration needs.
- Jamf Patch Management & Title Editor Manage updates with Smart Groups, custom titles, and centralized software definitions.





Automation JourneyBuilding the Foundation Before Installomator

macOS Automation & Repackaging at Marriott Library

Summary - Tools We Use to Streamline macOS Package Management

- munkipkg
 Open-source packaging tool for automated, reproducible .pkg builds.
- <u>python-jamf & jctl</u> Simplify Jamf Pro/Cloud API operations (policy, patch, CRUD tasks).
- Repackaging Mac App Store Apps
 Handles post-install workflows, licensing, and failures (Xcode, Logic Pro, GarageBand, Final Cut Pro).
- Repackaging Commercial Apps
 For non-standard installers (MATLAB, Mathematica, Maple, Autodesk) with licensing/configuration needs.
- <u>Jamf Patch Management & Title Editor</u>

 Manage updates with Smart Groups, custom titles, and centralized software definitions.
- <u>Custom Scripts & Keyboard Maestro</u>

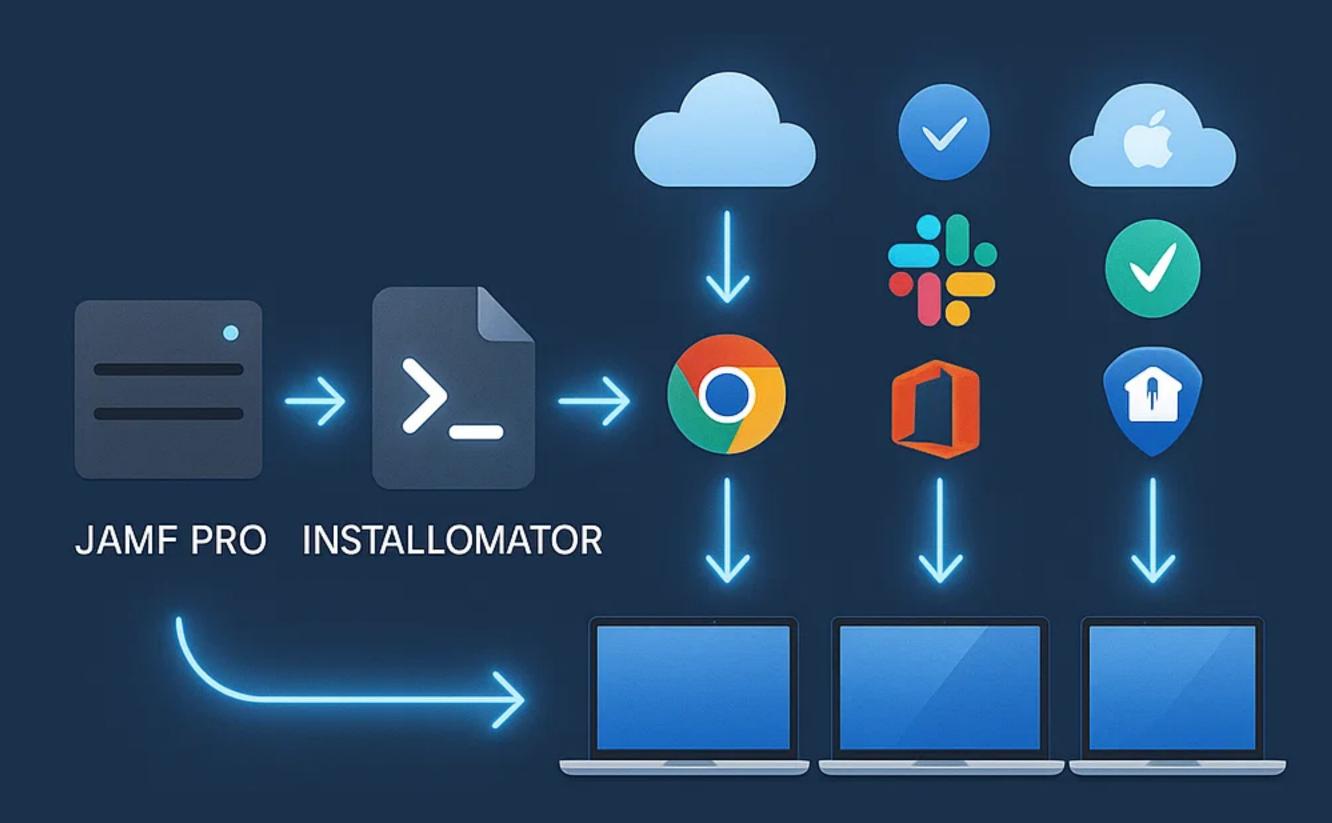
 Custom scripts & keyboard Maestro automate repetitive tasks, typing text, clicking, etc.





Installomator

When to use or not use?



Installomator - When to use or not use?

Installomator - When to Use?



Installomator - When to use or not use?

Installomator - When to Use?

https://github.com/Installomator/Installomator

Reduced infrastructure Complexity & Resources



Installomator - When to use or not use?

Installomator - When to Use?

- Reduced infrastructure Complexity & Resources
- Avoiding technical debt from complex toolchains



Installomator - When to use or not use?

Installomator - When to Use?

- **Reduced** infrastructure Complexity & Resources
- Avoiding technical debt from complex toolchains
- **zsh script** very easy to understand, troubleshoot, and contribute



Installomator - When to use or not use?

Installomator - When to Use?

- Reduced infrastructure Complexity & Resources
- Avoiding technical debt from complex toolchains
- **zsh script** very easy to understand, troubleshoot, and contribute
- **Light-touch** management approaches



Installomator - When to use or not use?

Installomator - When to Use?

- **Reduced** infrastructure Complexity & Resources
- Avoiding technical debt from complex toolchains
- **zsh script** very easy to understand, troubleshoot, and contribute
- <u>Vight-touch</u> management approaches
- **User-driven Installs** or **Updates** via Jamf Pro Self Service



Installomator - When to use or not use?

Installomator - When to Use?

- **Reduced** infrastructure Complexity & Resources
- Avoiding technical debt from complex toolchains
- **zsh script** very easy to understand, troubleshoot, and contribute
- **Light-touch** management approaches
- **V** User-driven Installs or Updates via Jamf Pro Self Service
- **Wighly distributed** mobile fleets without local infrastructure



Installomator - When to use or not use?

Installomator - When to Use?

- **Reduced** infrastructure Complexity & Resources
- Avoiding technical debt from complex toolchains
- **zsh script** very easy to understand, troubleshoot, and contribute
- Light-touch management approaches
- **V** User-driven Installs or Updates via Jamf Pro Self Service
- **Wighly distributed** mobile fleets without local infrastructure
- V Organizations wanting <u>automatic latest versions</u>



Installomator - When to use or not use?

Installomator - When to Use?

- Reduced infrastructure Complexity & Resources
- Avoiding technical debt from complex toolchains
- **zsh script** very easy to understand, troubleshoot, and contribute
- Light-touch management approaches
- **User-driven Installs** or **Updates** via Jamf Pro Self Service
- **Wighly distributed** mobile fleets without local infrastructure
- V Organizations wanting <u>automatic latest versions</u>
- V No preparation cycle benefits (eliminates repackaging)



Installomator - When to use or not use?

Installomator - When NOT to Use?



Installomator - When to use or not use?

Installomator - When NOT to Use?

https://github.com/Installomator/Installomator

• X Version control requirements with locked-down versions (workaround using overrides)



Installomator - When to use or not use?

Installomator - When NOT to Use?

- X Version control requirements with locked-down versions (workaround using overrides)
- Approval and testing processes for updates



Installomator - When to use or not use?

Installomator - When NOT to Use?

- X Version control requirements with locked-down versions (workaround using overrides)
- Approval and testing processes for updates
- Network constraints like bandwidth limitations in schools (Workaround is Squid Web Proxy Cache)

 https://github.com/squid-cache/squid



Installomator - When to use or not use?

Installomator - When NOT to Use?

- Version control requirements with locked-down versions (workaround using overrides)
- Approval and testing processes for updates
- Network constraints like bandwidth limitations in schools (Workaround is Squid Web Proxy Cache)

 https://github.com/squid-cache/squid
- K Firewall/proxy restrictions blocking vendor downloads



Installomator - When to use or not use?

Installomator - When NOT to Use?

- X Version control requirements with locked-down versions (workaround using overrides)
- Approval and testing processes for updates
- Network constraints like bandwidth limitations in schools (Workaround is Squid Web Proxy Cache)

 https://github.com/squid-cache/squid
- K Firewall/proxy restrictions blocking vendor downloads
- Applications requiring <u>authentication to download</u>



Installomator - When to use or not use?

Installomator - When NOT to Use?

- X Version control requirements with locked-down versions (workaround using overrides)
- Approval and testing processes for updates
- Network constraints like bandwidth limitations in schools (Workaround is Squid Web Proxy Cache)

 https://github.com/squid-cache/squid
- K Firewall/proxy restrictions blocking vendor downloads
- Applications requiring <u>authentication to download</u>
- Licensed software with restricted access



Installomator - Mix & Match Needs with Tools

Installomator - Mix & Match With Needs & Tools

https://github.com/Installomator/Installomator

Choose the best tool per application based on requirements



Installomator - Mix & Match Needs with Tools

Installomator - Mix & Match With Needs & Tools

- Choose the best tool per application based on requirements
 - Manual



Installomator - Mix & Match Needs with Tools

Installomator - Mix & Match With Needs & Tools

- Choose the best tool per application based on requirements
 - Manual
 - Jamf Pro Patch Management (Patch Reporting, Sources & Policy Workflow) https://learn.jamf.com/en-US/bundle/jamf-pro-documentation-current/page/PatchManagement.html



Installomator - Mix & Match Needs with Tools

Installomator - Mix & Match With Needs & Tools

- Choose the best tool per application based on requirements
 - Manual
 - Jamf Pro Patch Management (Patch Reporting, Sources & Policy Workflow) https://learn.jamf.com/en-US/bundle/jamf-pro-documentation-current/page/PatchManagement.html
 - Title Editor (External & Custom Patch Reporting, Sources & Policy Workflow) https://learn.jamf.com/en-US/bundle/title-editor/page/About_Title_Editor.html



Installomator - Mix & Match Needs with Tools

Installomator - Mix & Match With Needs & Tools

- Choose the best tool per application based on requirements
 - Manual
 - Jamf Pro Patch Management (Patch Reporting, Sources & Policy Workflow) https://learn.jamf.com/en-US/bundle/jamf-pro-documentation-current/page/PatchManagement.html
 - Title Editor (External & Custom Patch Reporting, Sources & Policy Workflow) https://learn.jamf.com/en-US/bundle/title-editor/page/About_Title_Editor.html
 - Jamf Pro App Installers (Requires Jamf Cloud)
 https://learn.jamf.com/en-US/bundle/jamf-app-catalog/page/App_Installers_Software_Titles.html



Installomator - Mix & Match Needs with Tools

Installomator - Mix & Match With Needs & Tools

- Choose the best tool per application based on requirements
 - Manual
 - Jamf Pro Patch Management (Patch Reporting, Sources & Policy Workflow) https://learn.jamf.com/en-US/bundle/jamf-pro-documentation-current/page/PatchManagement.html
 - Title Editor (External & Custom Patch Reporting, Sources & Policy Workflow) https://learn.jamf.com/en-US/bundle/title-editor/page/About_Title_Editor.html
 - Jamf Pro App Installers (Requires Jamf Cloud)
 https://learn.jamf.com/en-US/bundle/jamf-app-catalog/page/App_Installers_Software_Titles.html
 - AutoPKG (Mac or Runner, Storage for Repositories, Automated Workflows for Testing & Deployment) https://github.com/autopkg/autopkg



Installomator - Mix & Match Needs with Tools

Installomator - Mix & Match With Needs & Tools

- Choose the best tool per application based on requirements
 - Manual
 - Jamf Pro Patch Management (Patch Reporting, Sources & Policy Workflow) https://learn.jamf.com/en-US/bundle/jamf-pro-documentation-current/page/PatchManagement.html
 - Title Editor (External & Custom Patch Reporting, Sources & Policy Workflow) https://learn.jamf.com/en-US/bundle/title-editor/page/About_Title_Editor.html
 - Jamf Pro App Installers (Requires Jamf Cloud)
 https://learn.jamf.com/en-US/bundle/jamf-app-catalog/page/App_Installers_Software_Titles.html
 - AutoPKG (Mac or Runner, Storage for Repositories, Automated Workflows for Testing & Deployment)
 https://github.com/autopkg/autopkg
- Commercial Solutions
 - Electrona Patch https://www.alectrona.com/patch
 - App Catalog https://appcatalog.cloud/





Automating Software Lifecycle with Installomator

Installomator Main Features



Automating Software Lifecycle with Installomator

Installomator Main Features

https://github.com/Installomator/Installomator

• Thousand of prebuilt labels – 1000+ apps supported (Chrome, Slack, Zoom, Firefox, etc.)



Automating Software Lifecycle with Installomator

Installomator Main Features

- Thousand of prebuilt labels 1000+ apps supported (Chrome, Slack, Zoom, Firefox, etc.)
- Jamf Policy runs Installomator on Mac triggered at enrollment, recurring check-in, or Self Service



Automating Software Lifecycle with Installomator

Installomator Main Features

- Thousand of prebuilt labels 1000+ apps supported (Chrome, Slack, Zoom, Firefox, etc.)
- Jamf Policy runs Installomator on Mac triggered at enrollment, recurring check-in, or Self Service
- **Downloads apps directly from the vendor** supports DMG, ZIP, PKG, and other archive formats



Automating Software Lifecycle with Installomator

Installomator Main Features

- Thousand of prebuilt labels 1000+ apps supported (Chrome, Slack, Zoom, Firefox, etc.)
- Jamf Policy runs Installomator on Mac triggered at enrollment, recurring check-in, or Self Service
- **Downloads apps directly from the vendor** supports DMG, ZIP, PKG, and other archive formats
- Performs security validation Gatekeeper, Developer ID, and Team ID checks



Automating Software Lifecycle with Installomator

Installomator Main Features

- Thousand of prebuilt labels 1000+ apps supported (Chrome, Slack, Zoom, Firefox, etc.)
- Jamf Policy runs Installomator on Mac triggered at enrollment, recurring check-in, or Self Service
- **Downloads apps directly from the vendor** supports DMG, ZIP, PKG, and other archive formats
- Performs security validation Gatekeeper, Developer ID, and Team ID checks
- Silent installs and updates no user interaction required, but optional



Automating Software Lifecycle with Installomator

Installomator Main Features

- Thousand of prebuilt labels 1000+ apps supported (Chrome, Slack, Zoom, Firefox, etc.)
- Jamf Policy runs Installomator on Mac triggered at enrollment, recurring check-in, or Self Service
- **Downloads apps directly from the vendor** supports DMG, ZIP, PKG, and other archive formats
- Performs security validation Gatekeeper, Developer ID, and Team ID checks
- Silent installs and updates no user interaction required, but optional
- Self Service integration users can install optional apps on demand



Automating Software Lifecycle with Installomator

Installomator Main Features

- Thousand of prebuilt labels 1000+ apps supported (Chrome, Slack, Zoom, Firefox, etc.)
- Jamf Policy runs Installomator on Mac triggered at enrollment, recurring check-in, or Self Service
- **Downloads apps directly from the vendor** supports DMG, ZIP, PKG, and other archive formats
- Performs security validation Gatekeeper, Developer ID, and Team ID checks
- Silent installs and updates no user interaction required, but optional
- Self Service integration users can install optional apps on demand
- Customizable User Behavior adjust notifications, force reinstalls, relaunch apps, etc.



Automating Software Lifecycle with Installomator

Installomator Main Features - More



Automating Software Lifecycle with Installomator

Installomator Main Features - More

https://github.com/Installomator/Installomator

• Smart Group compatibility – deploy only when missing, outdated, or scoped to specific devices



Automating Software Lifecycle with Installomator

Installomator Main Features - More

- Smart Group compatibility deploy only when missing, outdated, or scoped to specific devices
- Patch Management support combine with Jamf Patch for update tracking



Automating Software Lifecycle with Installomator

Installomator Main Features - More

- Smart Group compatibility deploy only when missing, outdated, or scoped to specific devices
- Patch Management support combine with Jamf Patch for update tracking
- Ideal for distributed fleets works well for remote devices outside corporate networks



Automating Software Lifecycle with Installomator

Installomator Main Features - More

- Smart Group compatibility deploy only when missing, outdated, or scoped to specific devices
- Patch Management support combine with Jamf Patch for update tracking
- Ideal for distributed fleets works well for remote devices outside corporate networks
- **Debug & test modes** simulate downloads/installs with DEBUG=1 or DEBUG=2



Automating Software Lifecycle with Installomator

Installomator Main Features - More

- Smart Group compatibility deploy only when missing, outdated, or scoped to specific devices
- Patch Management support combine with Jamf Patch for update tracking
- Ideal for distributed fleets works well for remote devices outside corporate networks
- Debug & test modes simulate downloads/installs with DEBUG=1 or DEBUG=2
- Open source & community maintained large contributor base adding new apps and features



Automating Software Lifecycle with Installomator

Installomator Main Features

https://github.com/Installomator/Installomator

Works with various MDM solutions:



lamf Pro

https://github.com/Installomator/Installomator/wiki/MDM:-Jamf-Pro



Microsoft Intune

https://github.com/Installomator/Installomator/wiki/MDM:-Microsoft-Intune-(Endpoint-Manager)



Filewaye

https://kb.filewave.com/books/software-deployment-recipes-macos/page/installomator-the-one-installer-script-to-rule-them-all-macos-pkg



Workspace One

https://github.com/Installomator/Installomator/wiki/MDM:WorkspaceONE



Mosyle

https://github.com/Installomator/Installomator/wiki/MDM:-Mosyle-(Business,-Fuse,-and-Manager)



Addigy

https://github.com/Installomator/Installomator/wiki/MDM:-Addigy



Jamf School

https://github.com/Installomator/Installomator/wiki/MDM:-Jamf-School



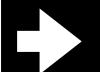
Automating Software Lifecycle with Installomator

Installomator Main Features

https://github.com/Installomator/Installomator

Works with various MDM solutions:





Notable MDM' used at University of Utah



Workspace One

https://github.com/Installomator/Installomator/wiki/MDM:WorkspaceONE



Mosyle

https://github.com/Installomator/Installomator/wiki/MDM:-Mosyle-(Business,-Fuse,-and-Manager)



Addigy

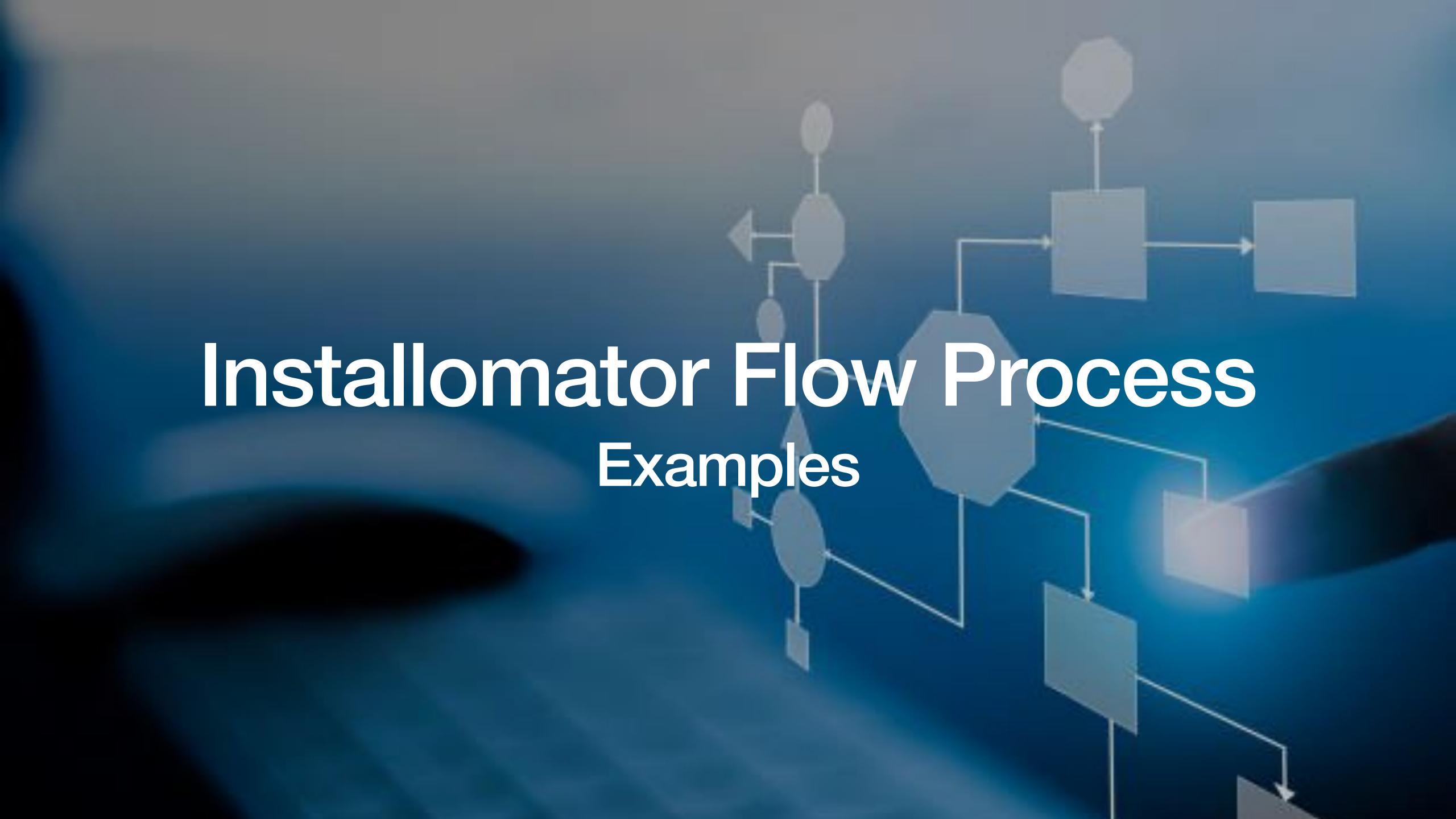
https://github.com/Installomator/Installomator/wiki/MDM:-Addigy



Jamf School

https://github.com/Installomator/Installomator/wiki/MDM:-Jamf-School

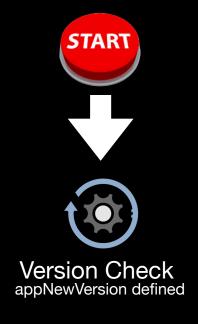




Installomator Flow Process No Update



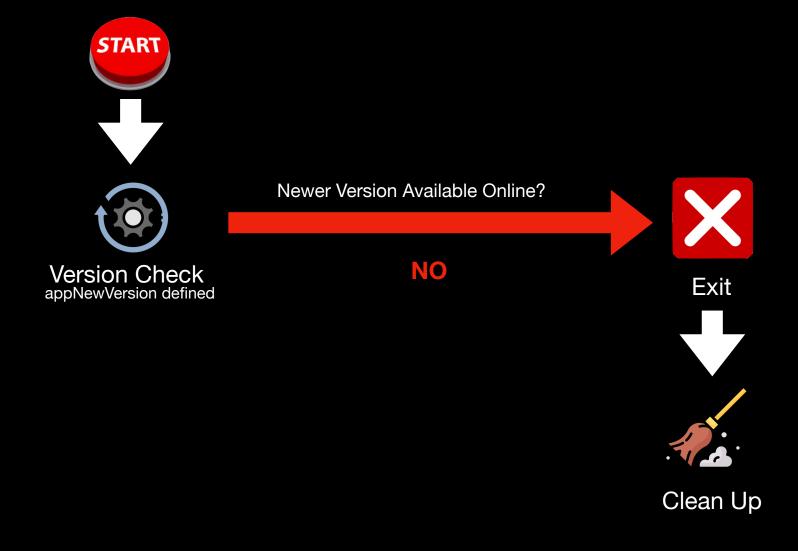




















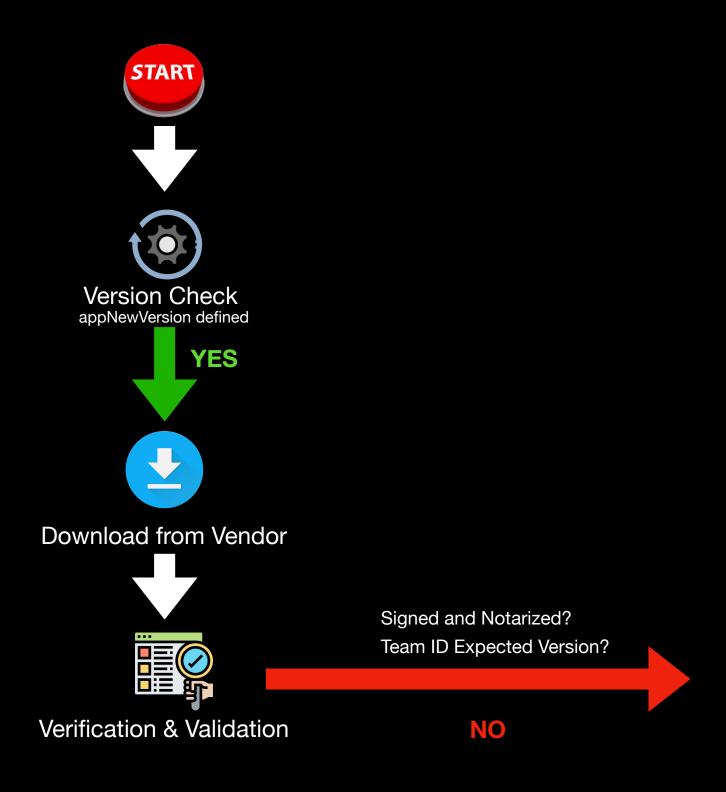




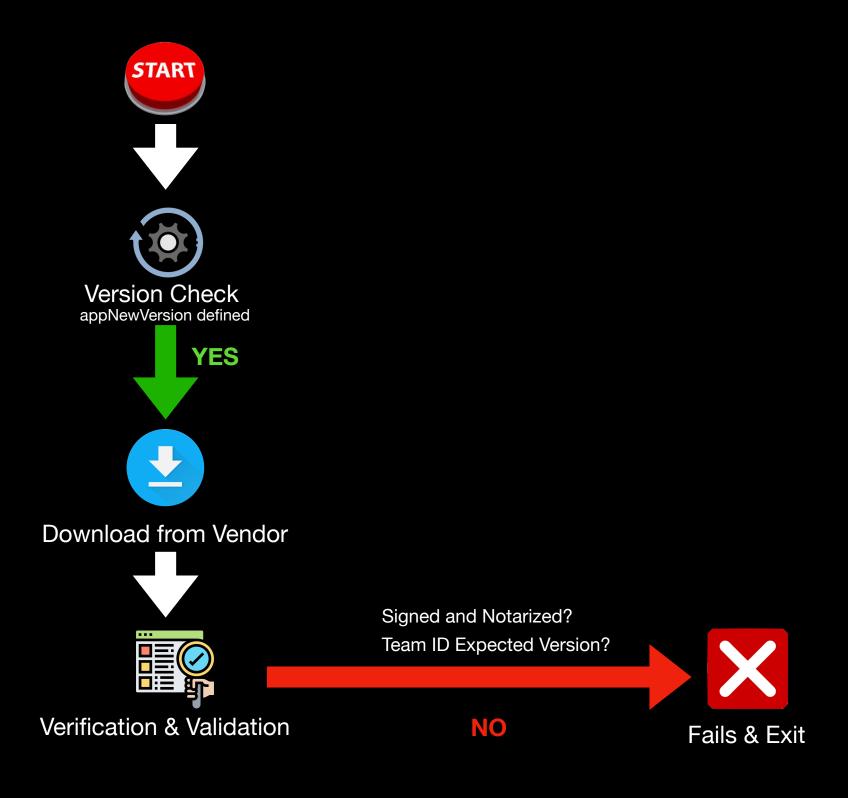




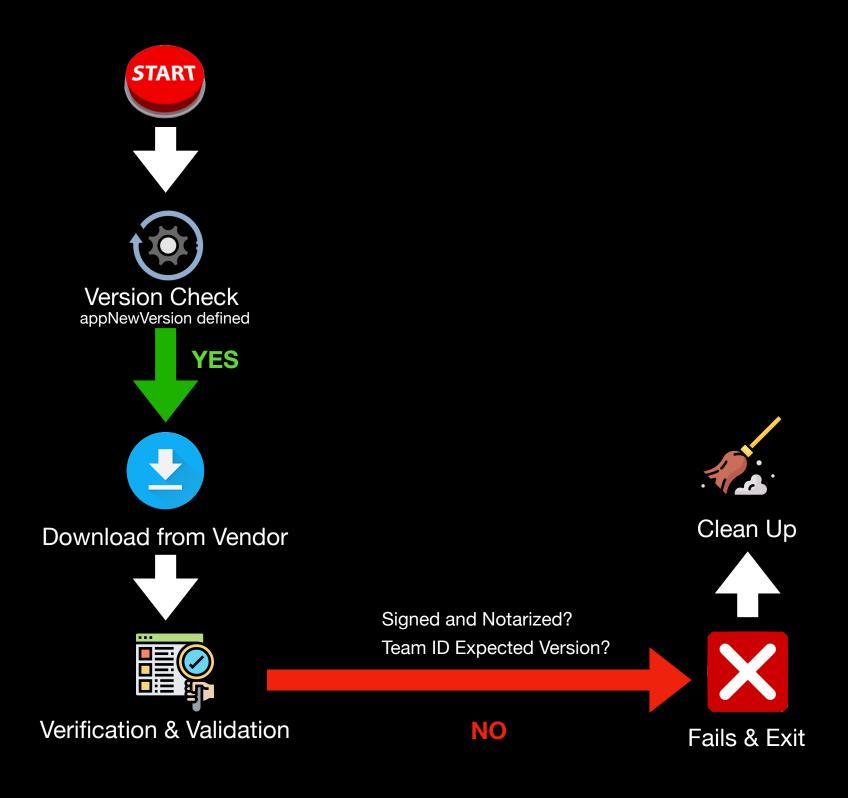
















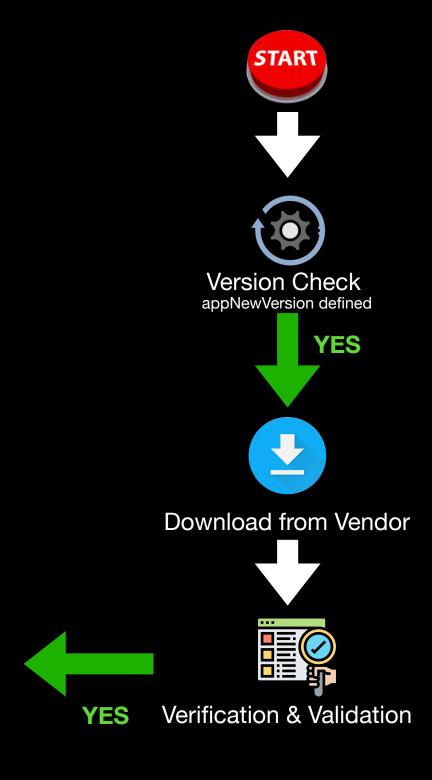




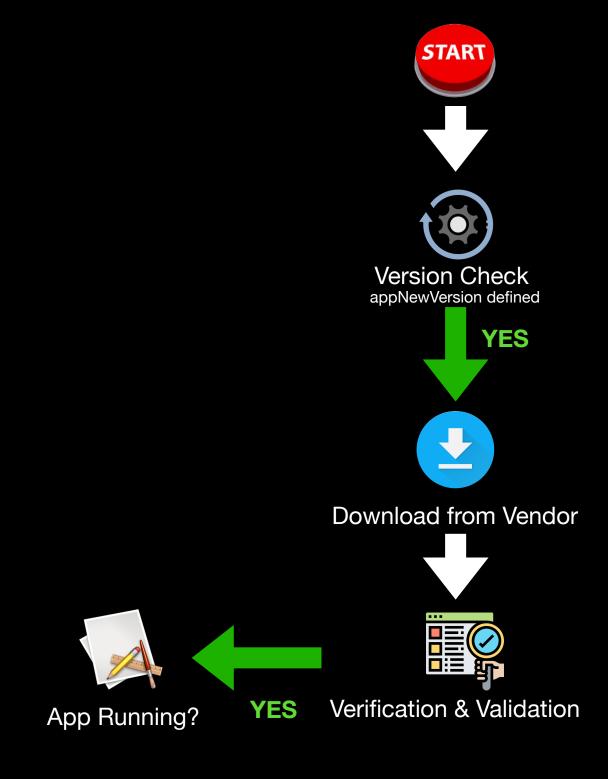




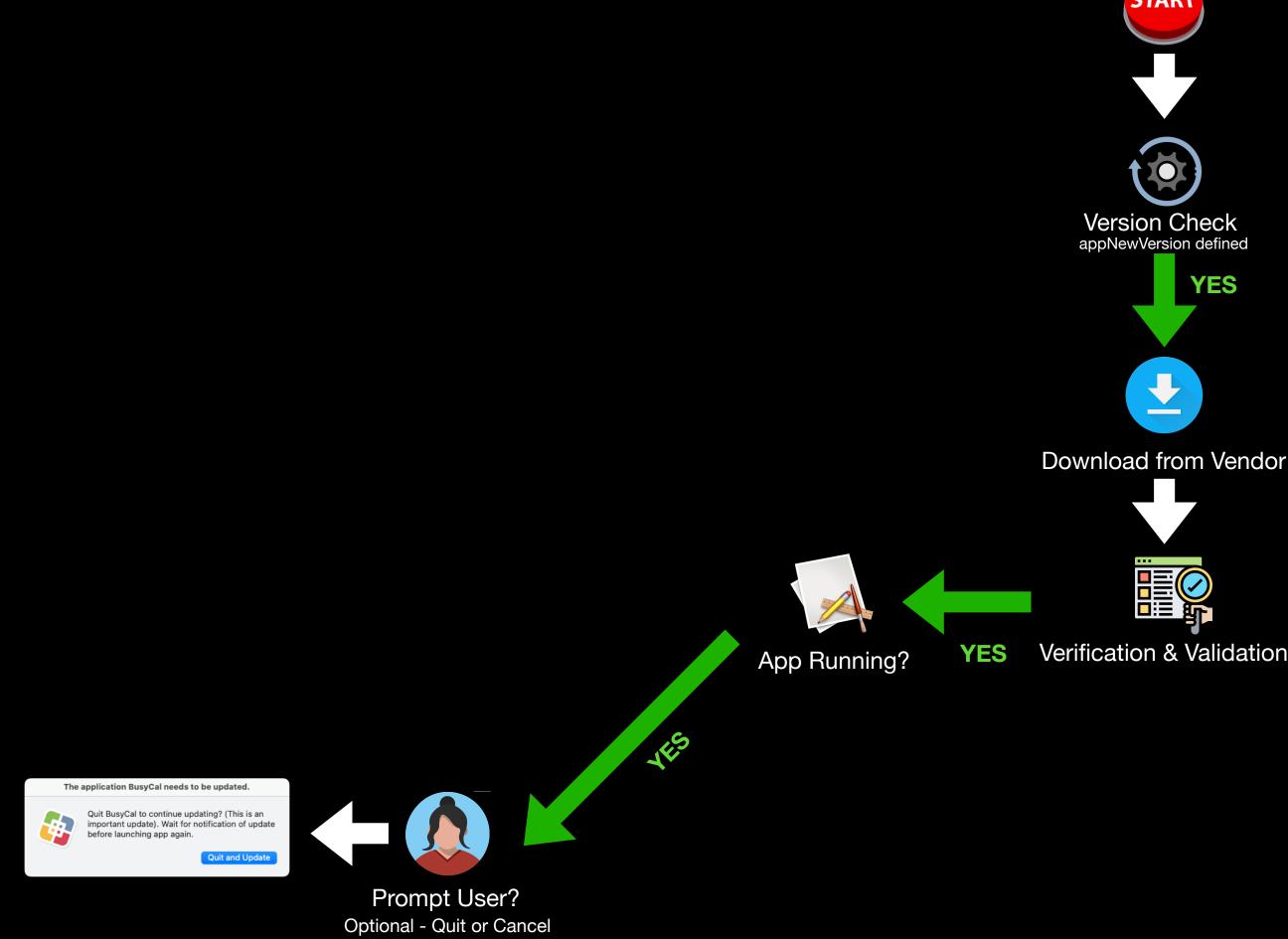




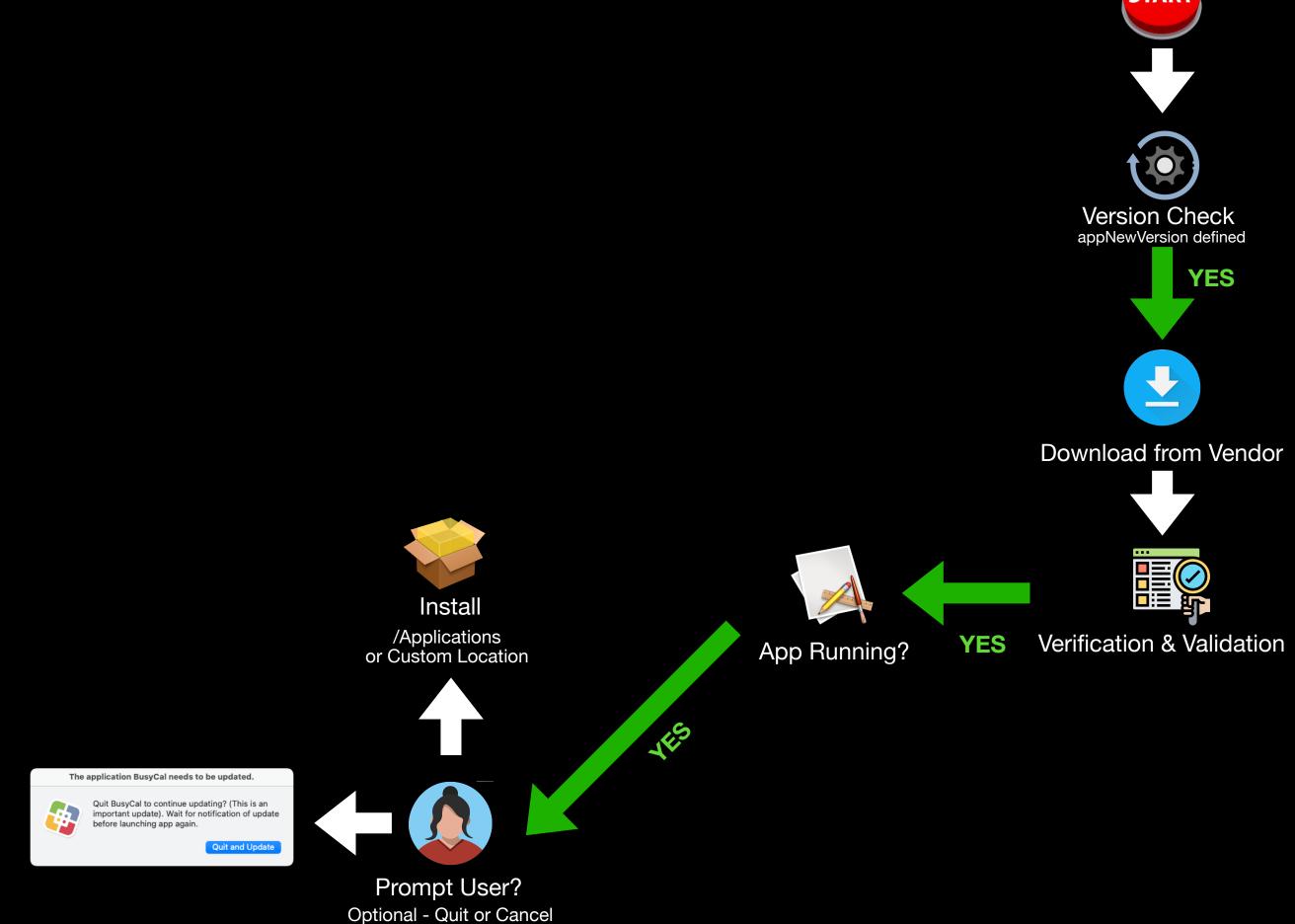




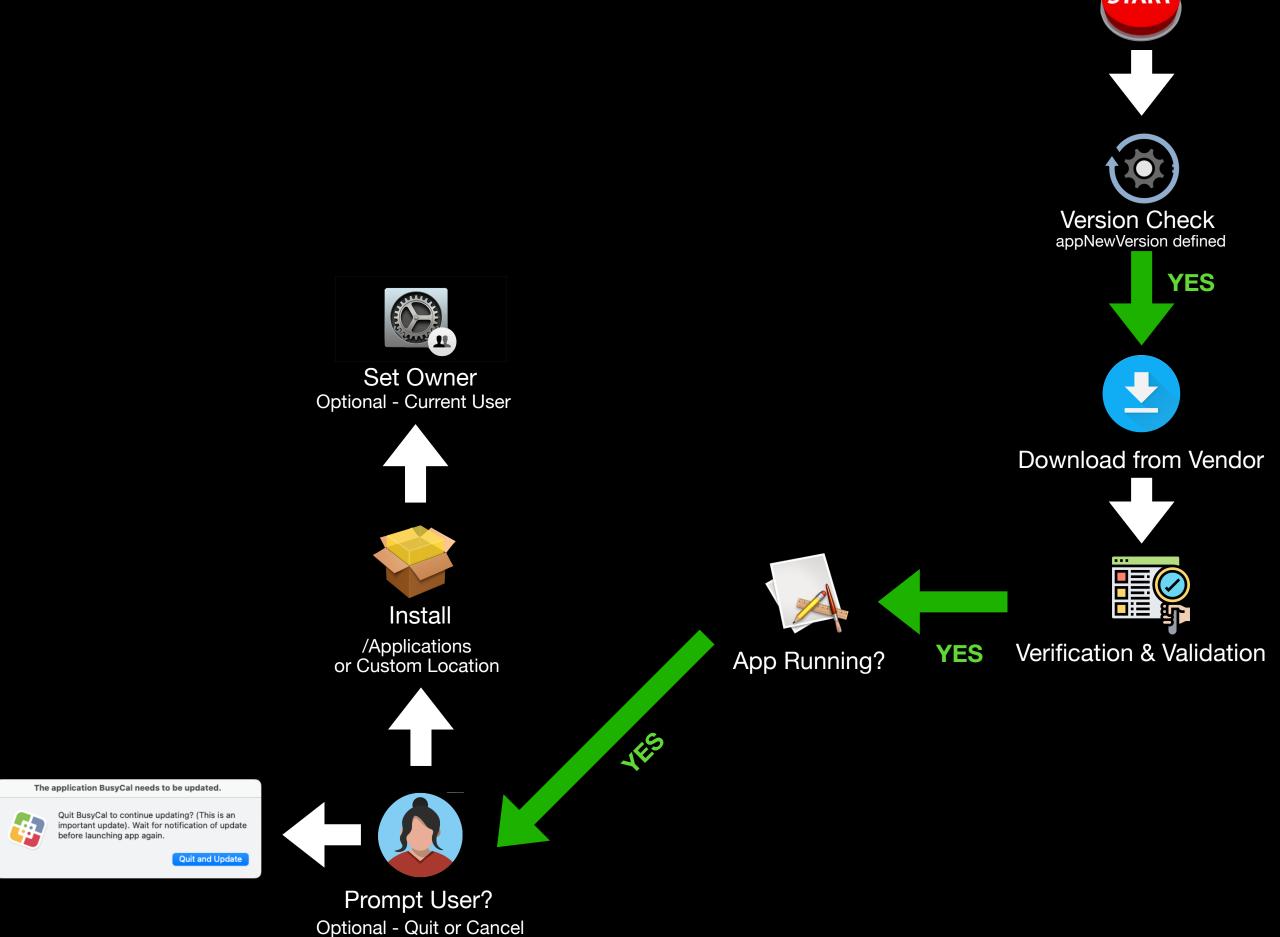




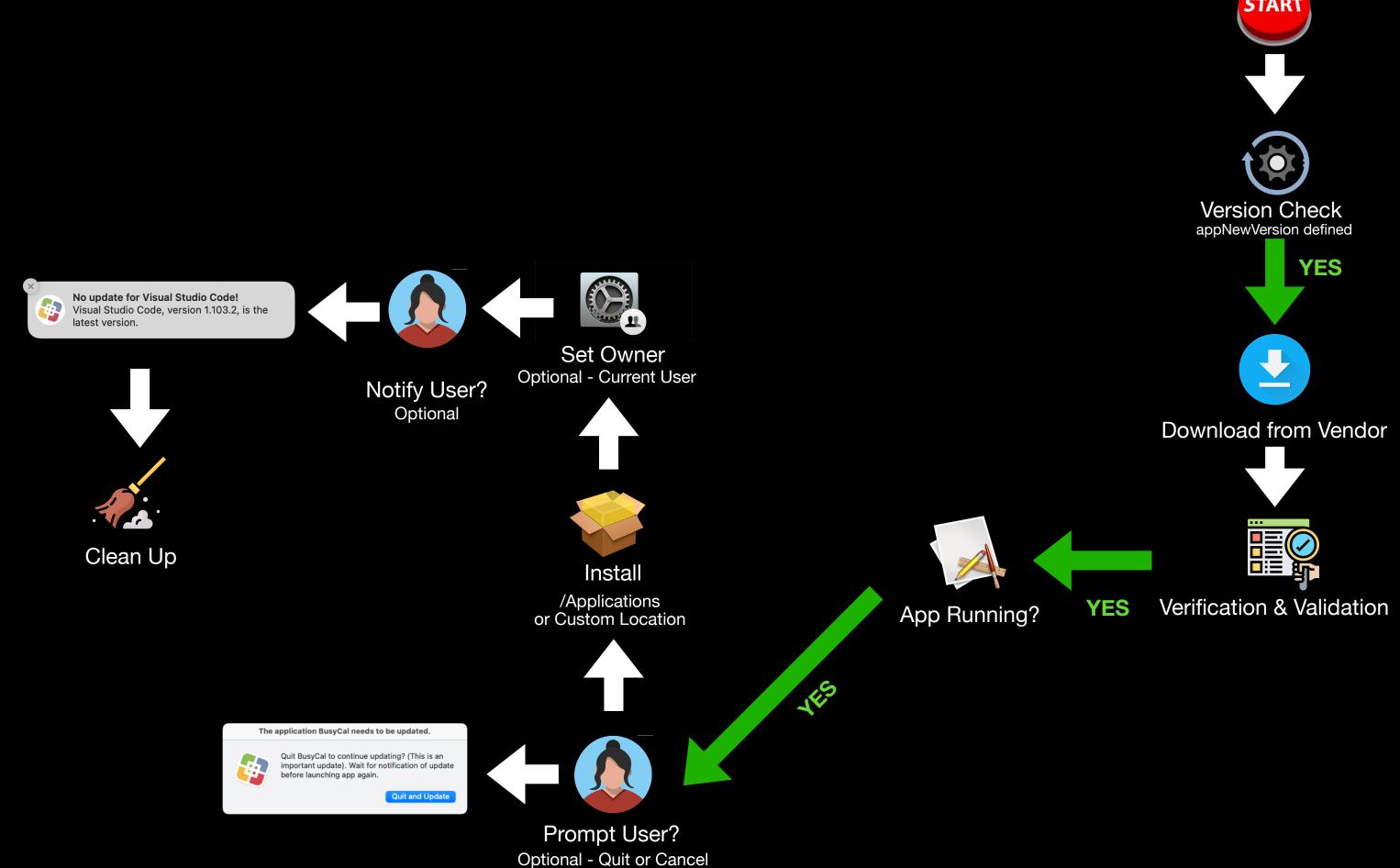




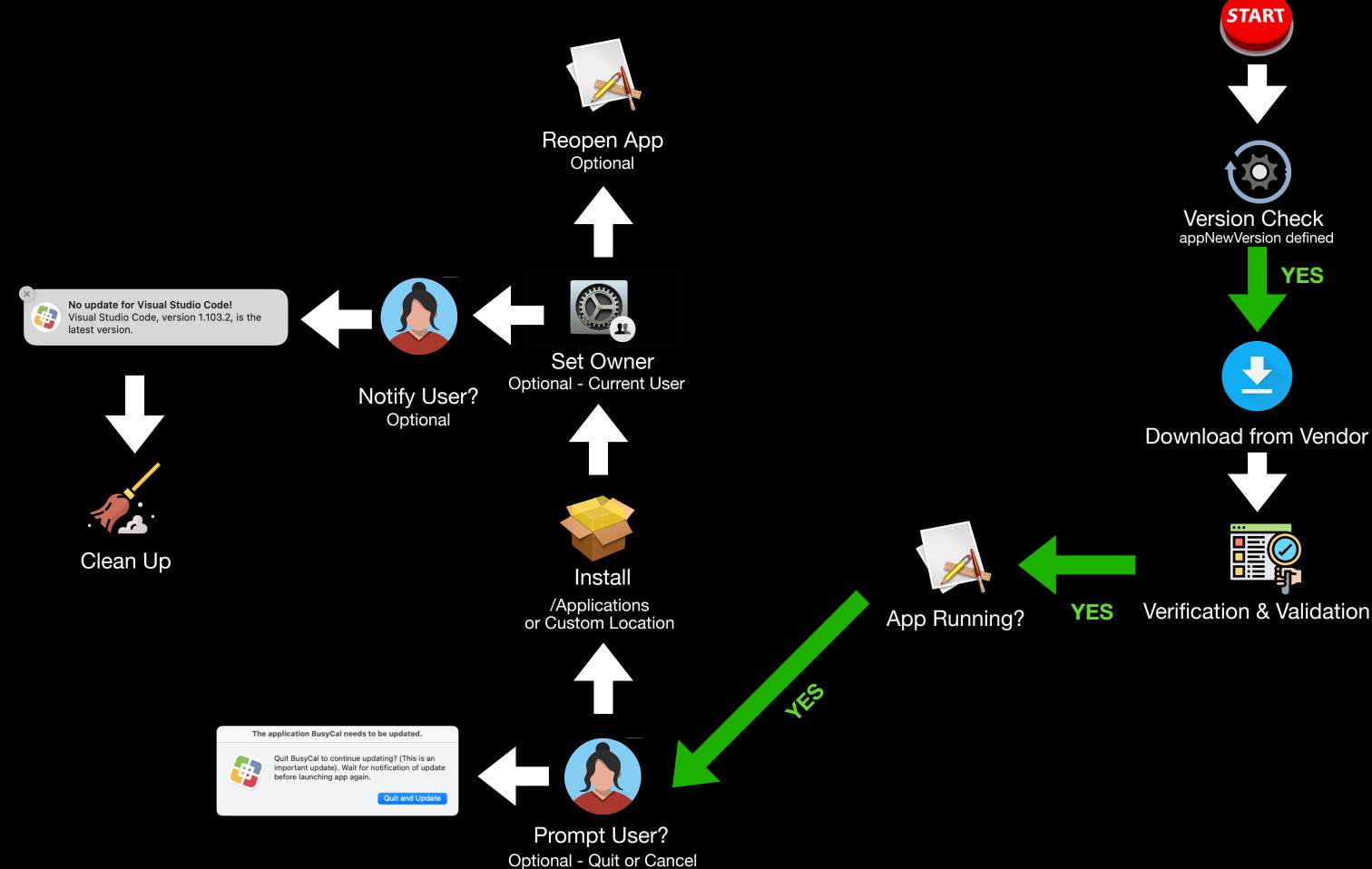
















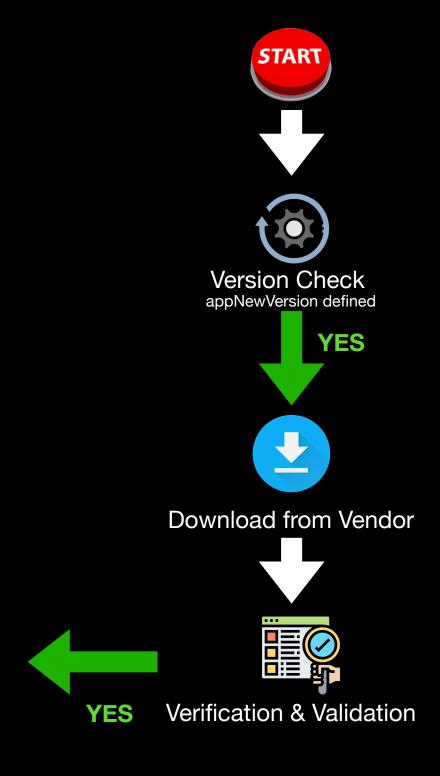




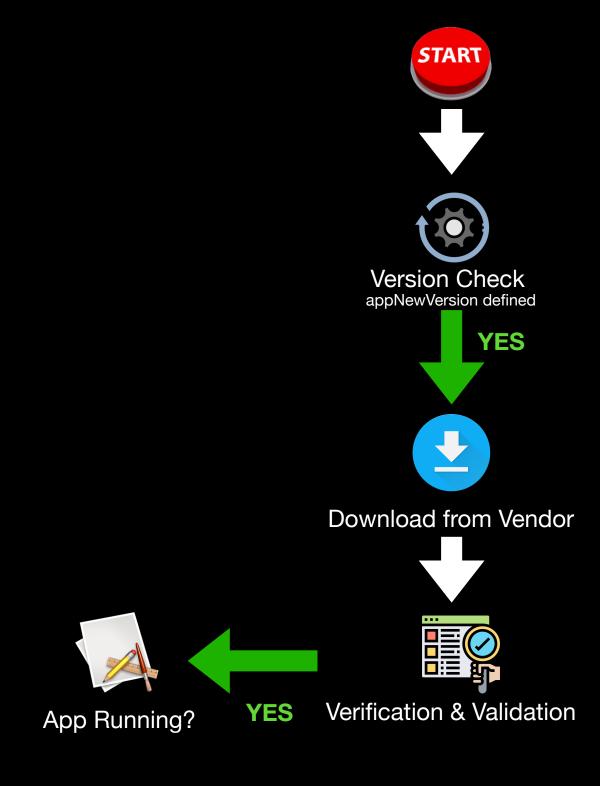




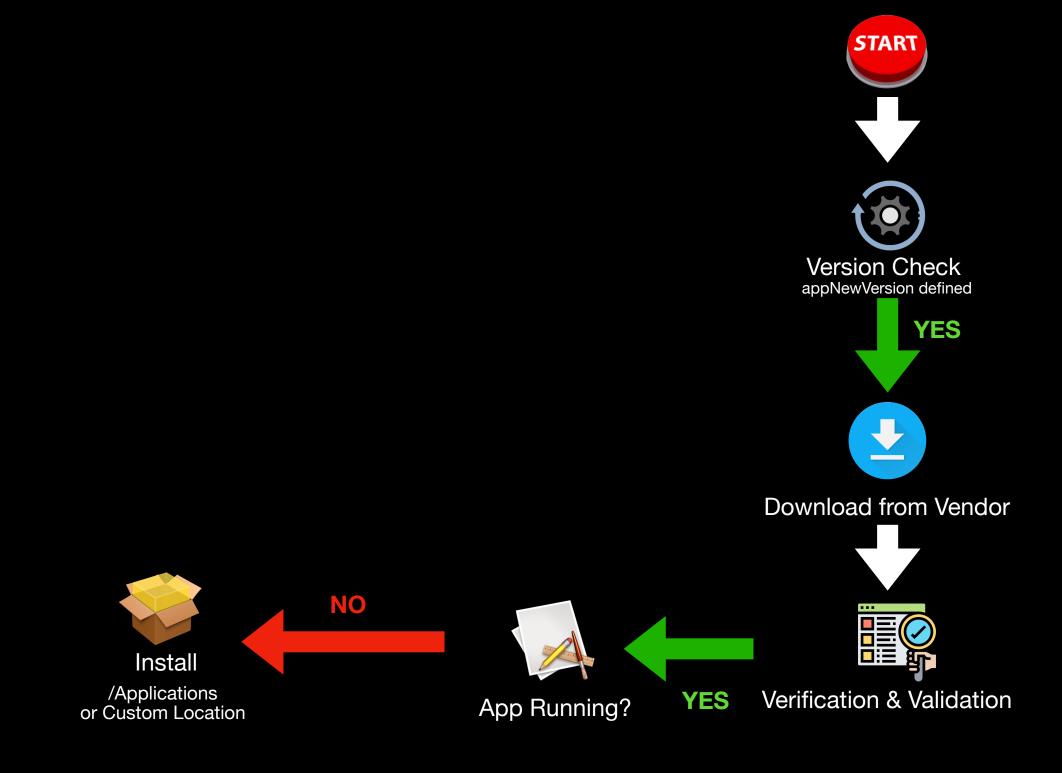




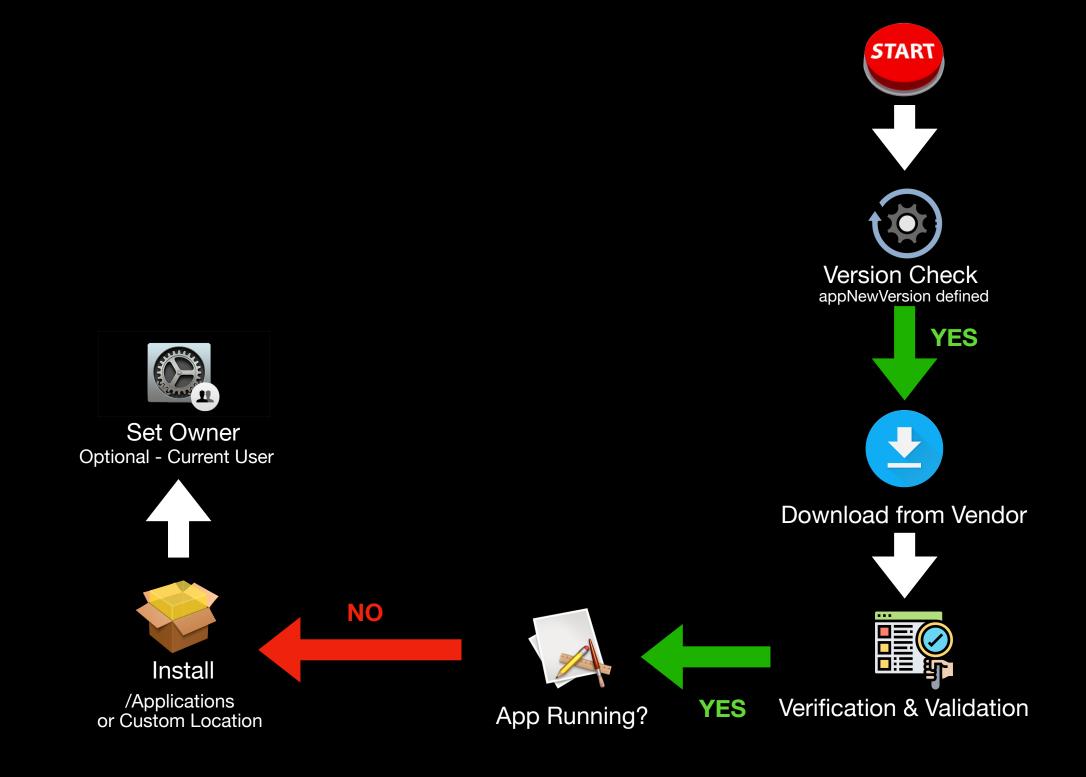




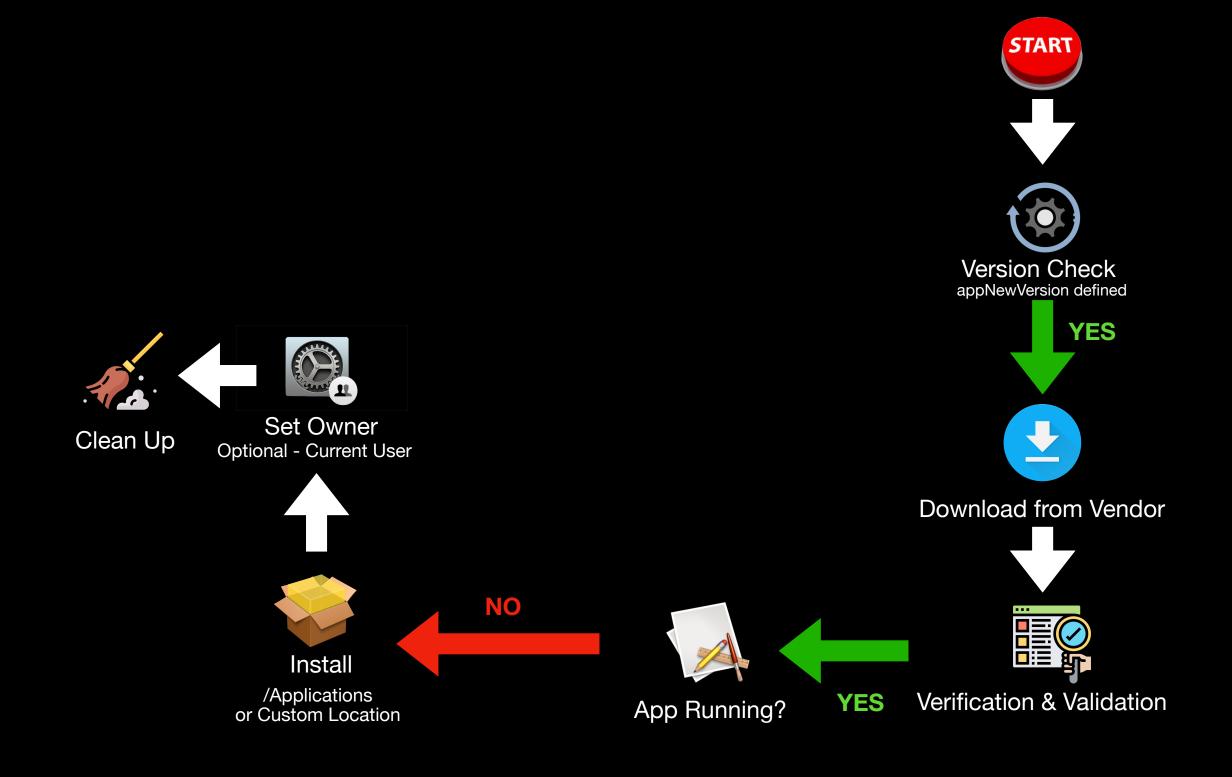
















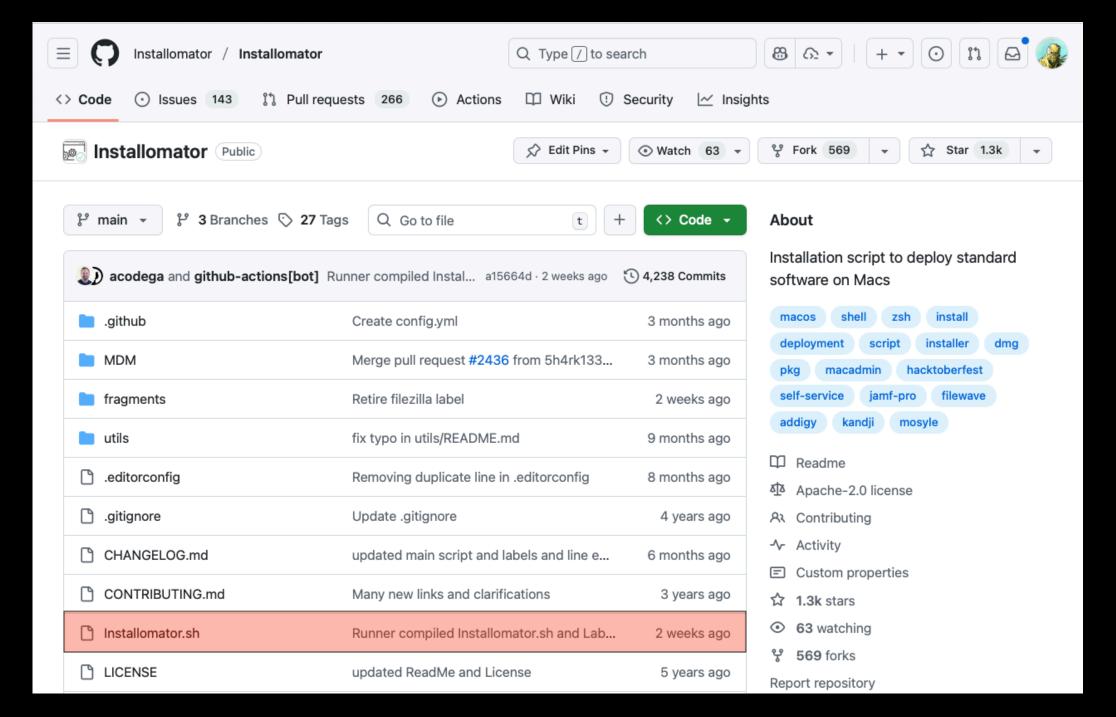
Jamf Pro Installomator Quick Start

Jamf Pro - Installomator Quick Start

Go to the Installomator on Repository on GitHub

https://github.com/Installomator/Installomator

Navigate to the <u>Installomator.sh</u> file inside the repository



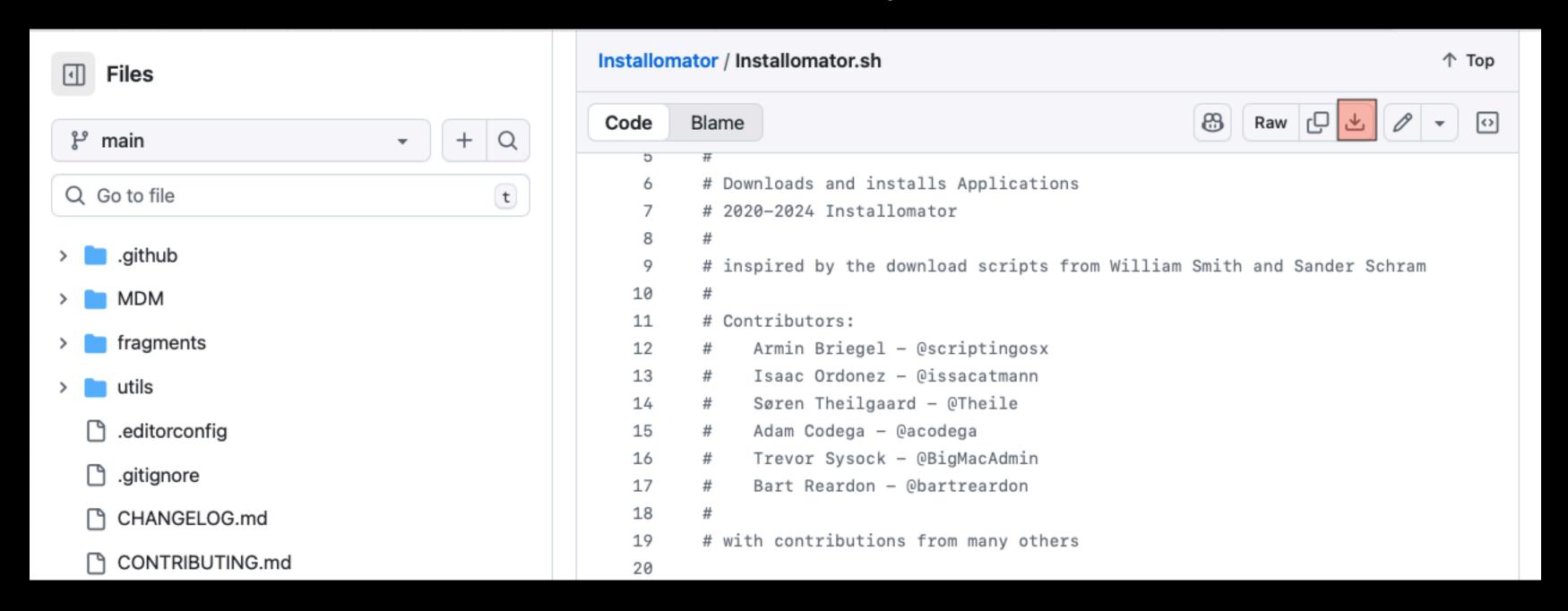


Jamf Pro - Installomator Quick Start

Go to the Installomator on Repository on GitHub

https://github.com/Installomator/Installomator

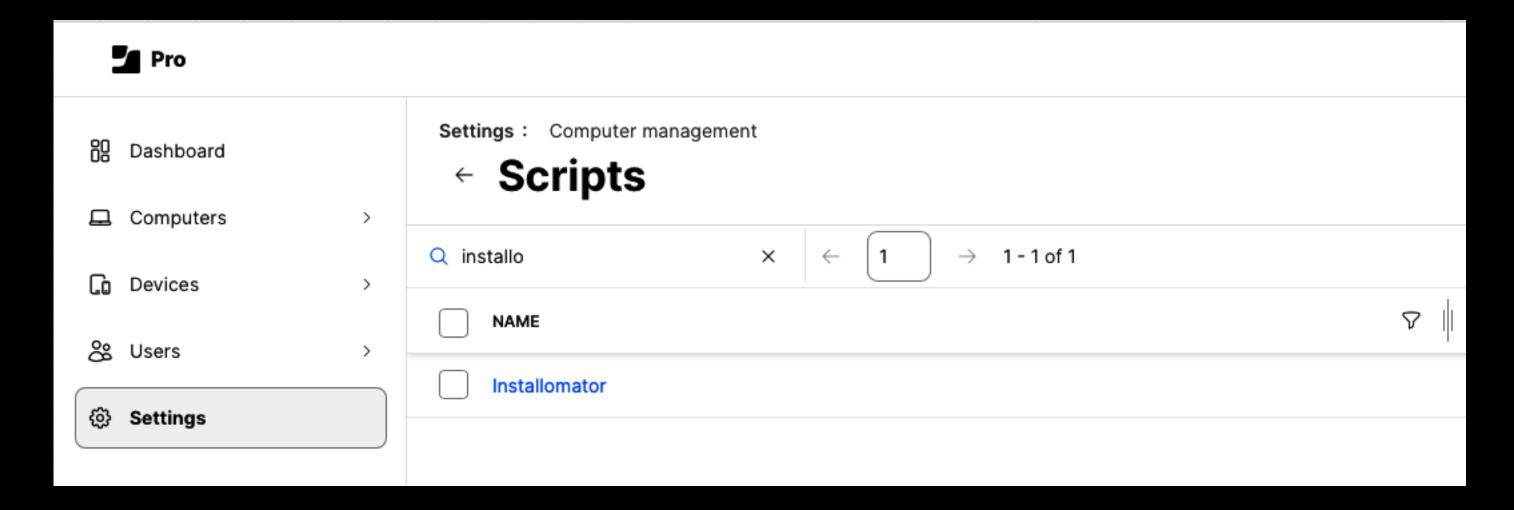
Click "Download raw file" icon to save it locally





Jamf Pro - Installomator Quick Start

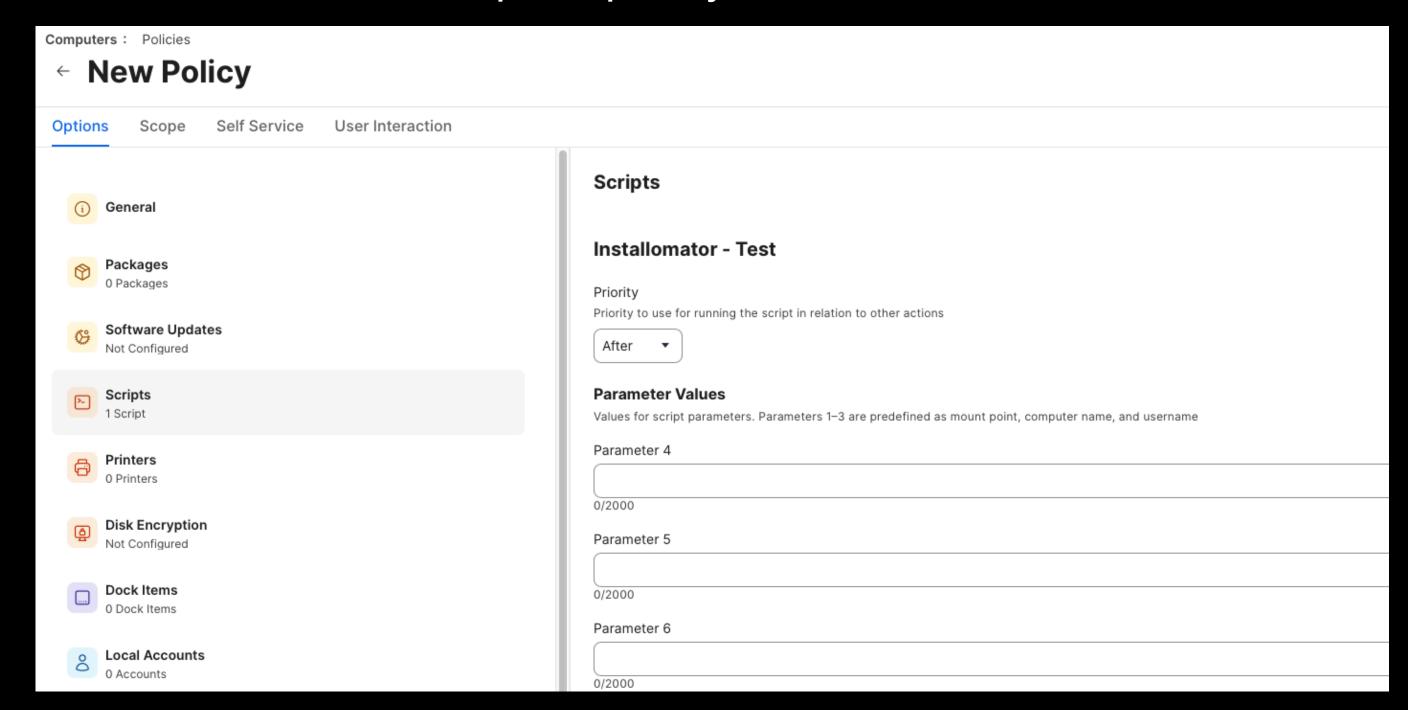
- Upload to Jamf Pro
 - Add the script to your <u>Jamf Pro Scripts library</u>





Jamf Pro - Installomator Quick Start

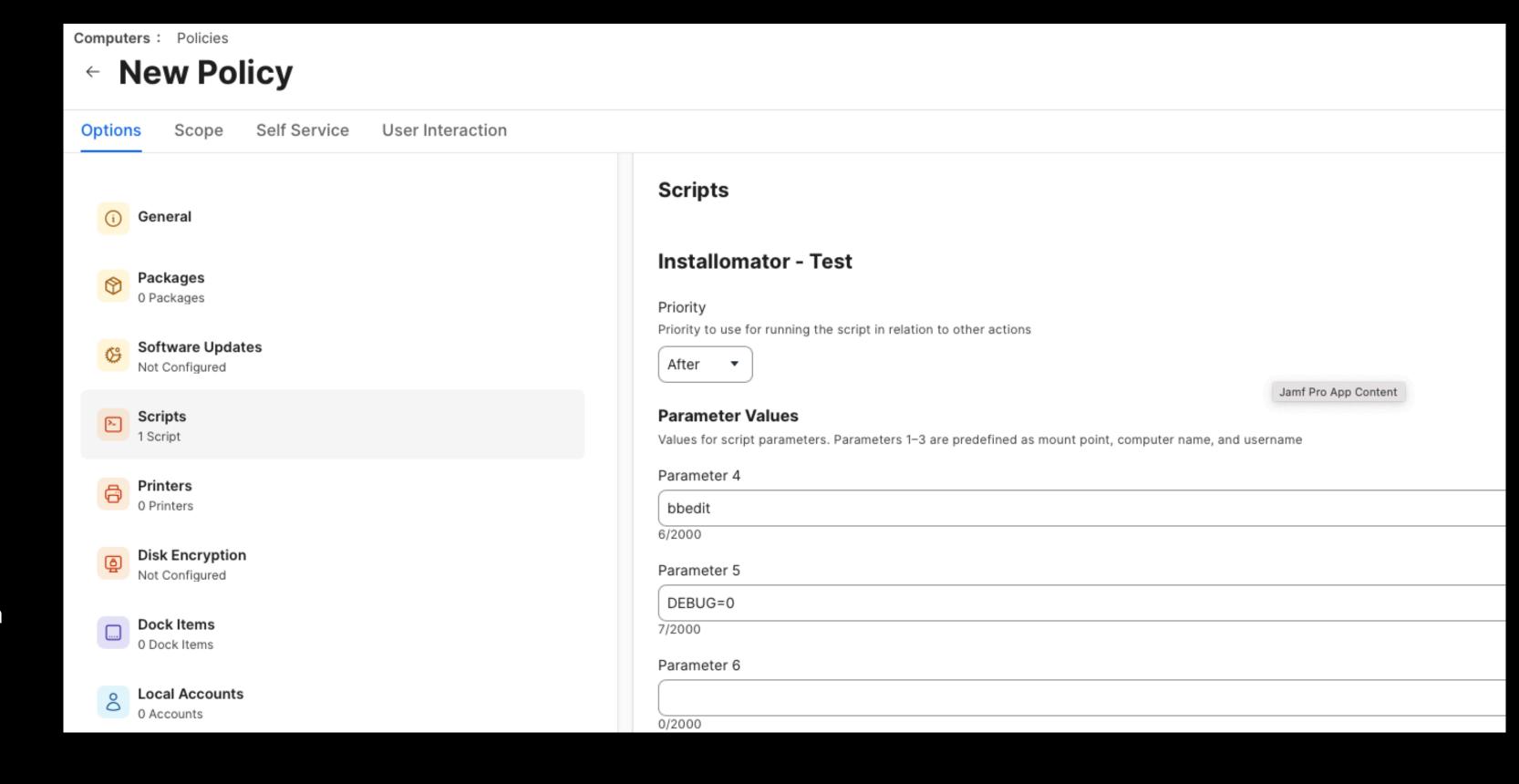
- Create a Policy
 - Add Installomator script to policy





Jamf Pro - Installomator Quick Start

- Create a Policy
 - Fill out script parameter fields
 - Parameter 4
 Add label name "bbedit"
 - Parameter 5
 Add Debug Mode "DEBUG=0"
 - <u>DEBUG=0</u> Default for deployment
 - <u>DEBUG=1</u>
 Useful for testing downloads & verification
 - <u>DEBUG=2</u>
 Useful for testing quit/notify workflow

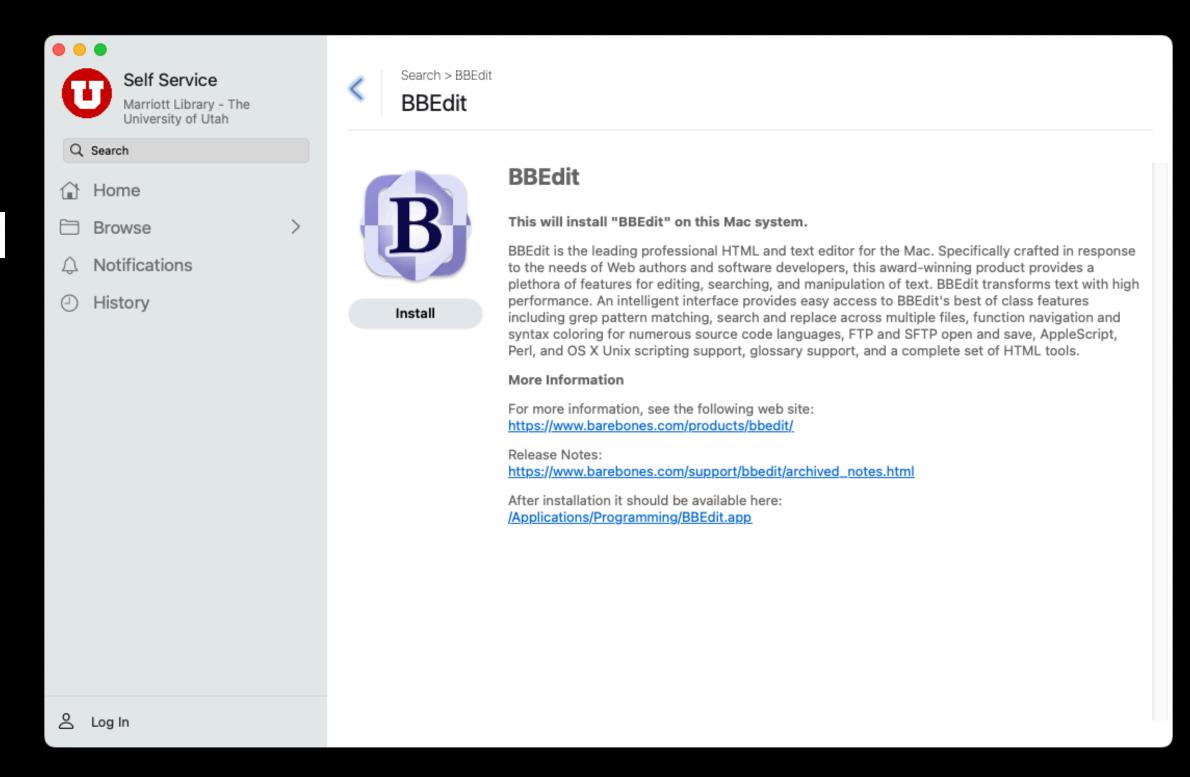


Note - Fill out parameters sequentially without empty parameters



Jamf Pro - Installomator Quick Start

- Scope Policy to Test Mac
- Execute Policy to Test BBEdit Install
 - Verify that "BBEdit" installed in /Applications
 - To troubleshooting...
 - Policy Logs
 Policy -> Logs -> Details
 - Computer Logs
 Computer -> History -> Policy Logs





Microsoft Intune Installomator Quick Start

Microsoft Intune - Installomator Quick Start

https://github.com/Installomator/Installomator/wiki/MDM:-Microsoft-Intune-(Endpoint-Manager)

- Background
 - Script size limits Microsoft Intune doesn't handle large scripts well
 - No variable support You can't dynamically pass arguments
- Install Installomator



- In the Intune admin center, go to:
 - Apps → macOS → Add → Line-of-business app.



• Once deployed, the Installomator script will live on the managed Macs (default: /usr/local/Installomator/Installomator.sh)



Create a Wrapper Script

Create a lightweight script for each application you want to deploy



Microsoft Intune - Installomator Quick Start

https://github.com/Installomator/Installomator/wiki/MDM:-Microsoft-Intune-(Endpoint-Manager)

Create a Wrapper Script Example

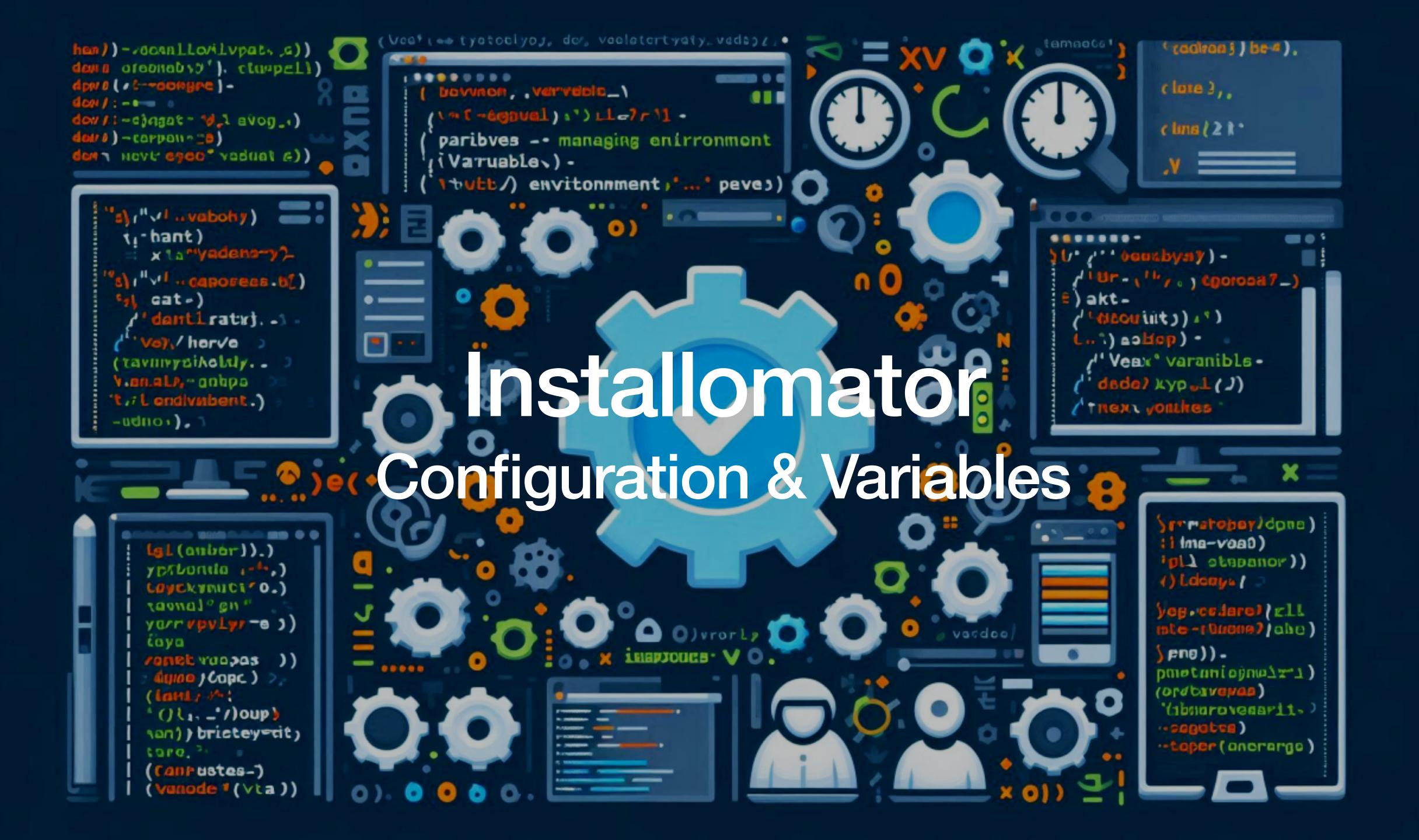
```
#!/bin/bash
if [ ! -f "/usr/local/Installomator/Installomator.sh" ]; then
    echo "Installomator not found"
    exit 1
fi
/usr/local/Installomator/Installomator.sh LABLE_NAME BLOCKING_PROCESS_ACTION=quit DEBUG=0
exit $?
```

Intuneomator

Intuneomator is a Swift-based open-source tool that automates macOS application lifecycle management via Microsoft Intune, leveraging the label database from Installomator for automated discovery, packaging, deployment, version updates, and integration with Intune's Graph API

https://github.com/gilburns/Intuneomator





Installomator - Configuration & Variables

https://github.com/Installomator/Installomator/wiki/Configuration-and-Variables

DEBUG Modes

Safely test Installomator without actually installing apps

- DEBUG=1
 - Default Mode

Downloads to script directory; skips version checks & installation

• DEBUG=2

Uses temporary folder; checks versions & blockers; skips install

DEBUG=0

Production - full install/update functionality



Installomator - Configuration & Variables

https://github.com/Installomator/Installomator/wiki/Configuration-and-Variables

- Logging Levels
 Control how much detail Installomator writes to logs
 - DEBUG: Full logging (detailed)
 - INFO: Standard logging (default)
 - WARN: Warnings only
 - **ERROR**: Errors only
 - **REQ**: Required-level messages
 - Use LOGGING=DEBUG when testing variable overrides



Installomator - Configuration & Variables

https://github.com/Installomator/Installomator/wiki/Configuration-and-Variables

Notifications

Controls what user notifications are displayed

Options

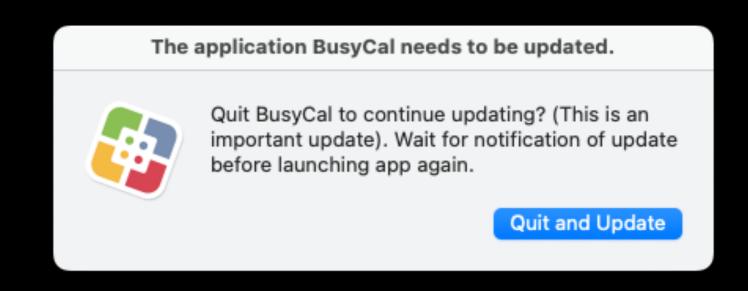
- success (default) notify only on successful install/update
- **silent** no notifications
- <u>all</u> show notifications for start, success, and failure
- **fail** notify only on failure

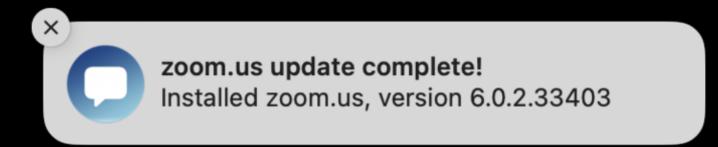
Behavior

By default → standard macOS system alerts (triggered via AppleScript).

If **swiftDialog** is installed and configured → Installomator can send updates to a dialog window

"Quit and Update" prompts are always done with AppleScript, and NOT swiftDialog







Installomator - Configuration & Variables

https://github.com/Installomator/Installomator/wiki/Configuration-and-Variables

Prompt Timeout

Number of seconds Installomator waits for a user response before continuing

Purpose

- Prevents scripts from hanging indefinitely if a user doesn't respond.
- Ensures MDM agents (Jamf, Intune, etc.) continue their workflow without getting stuck

Jamf Pro Timeout

It is recommended to use PROMPT_TIMEOUT=600 to avoid potential issues



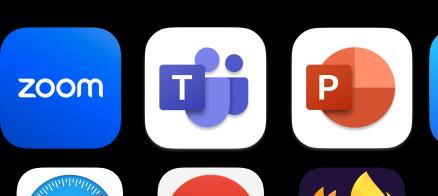
Installomator - Configuration & Variables

https://github.com/Installomator/Installomator/wiki/Configuration-and-Variables

Do Not Disturb (DND) Checks

Ensure it respects macOS "Display Sleep Assertions" and full-screen app states

Not related to macOS "Focus" modes



- INTERRUPT_DND
 - <u>yes (default)</u> Ignores DND and full-screen apps, proceeds with notifications & prompts
 - <u>no</u> Installomator pauses or exits if active DND/full-screen apps are detected
- IGNORE_DND_APPS
 Comma-separated list of apps that Installomator will ignore even if DND is active



Installomator - Configuration & Variables

https://github.com/Installomator/Installomator/wiki/Configuration-and-Variables

LOGO Variable

Controls the icon shown in dialog boxes (does not affect notifications)

Value	Icon Displayed	Example
appstore	Apple App Store (default) "/System/Applications/App Store.app/Contents/Resources/Applcon.icns"	**
jamf	Jamf Pro "/Library/Application Support/JAMF/Jamf.app/Contents/Resources/Applcon.icns"	
microsoft	Microsoft Endpoint Manager (Intune) "/Library/Intune/Microsoft Intune Agent.app/Contents/Resources/Applcon.icns"	
filewave	Filewave "/usr/local/sbin/FileWave.app/Contents/Resources/fwGUI.app/Contents/Resources/kiosk.icns"	
mosylem	Mosyle Manager (Education) "/Applications/Manager.app/Contents/Resources/Applcon.icns"	
ws1	Workspace ONE (AirWatch) "/Applications/Workspace ONE Intelligent Hub.app/Contents/Resources/Applcon.icns"	



Installomator - Configuration & Variables

https://github.com/Installomator/Installomator/wiki/Configuration-and-Variables

SYSTEMOWNER Variable

Determines ownership of apps installed from DMG or archives





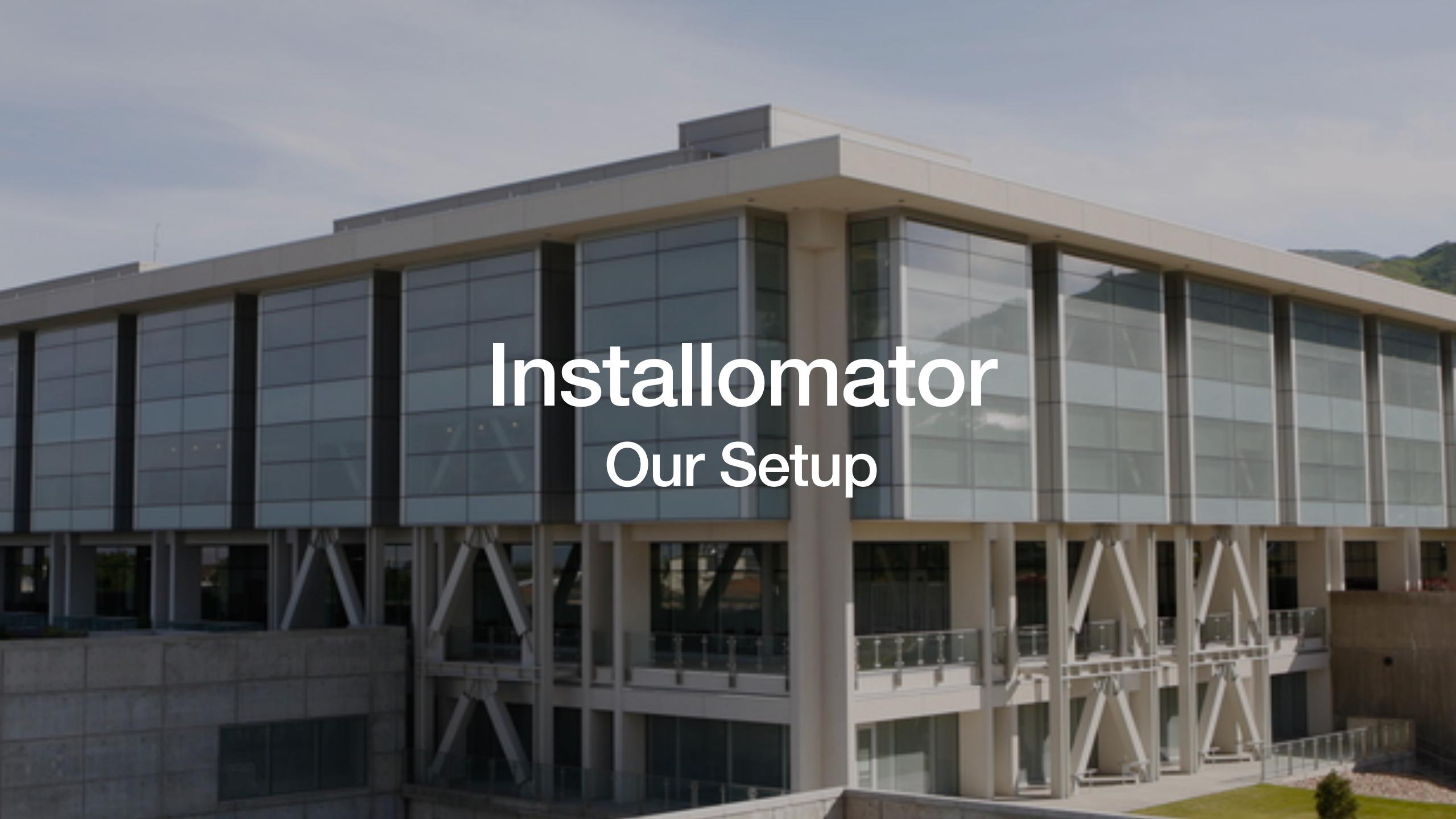


- <u>SYSTEMOWNER=0 (default)</u>
 Current user owns the app (mimics drag-and-drop install). If no user is logged in, owner defaults to <u>root:wheel</u>.
- <u>SYSTEMOWNER=1</u>
 Owner is set to <u>root:wheel</u>. Useful for shared machines or kiosk setups. Built-in app updates will be blocked.



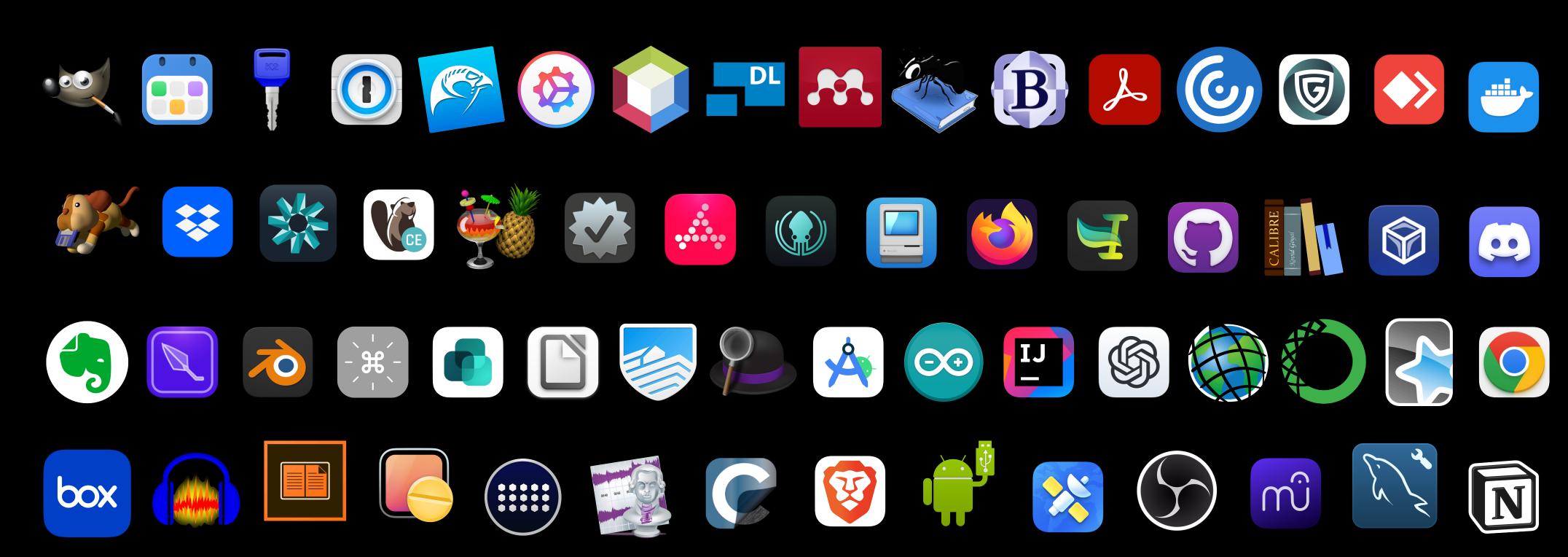
For security & policy, in our environment we always use SYSTMEOWNER=1





Installomator - Marriott Library's Setup

Currently, using Installomator with ≈ 200+ Applications



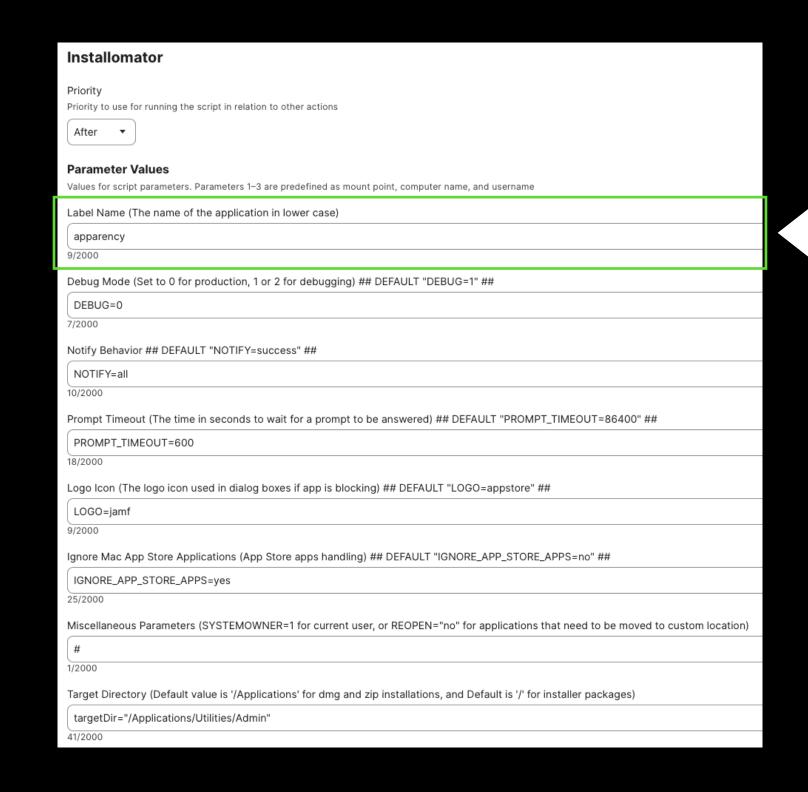


Installomator - Marriott Library's Setup

Install Policy

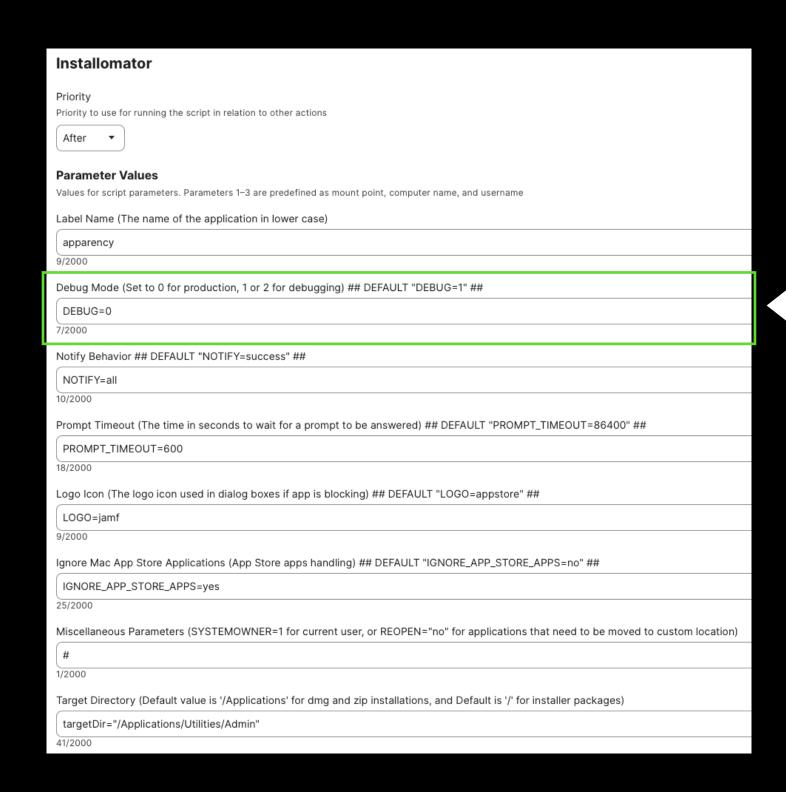


- Install Policy Install policy can be used to automatically install or via Self Service
 - Parameter #4: Label Name



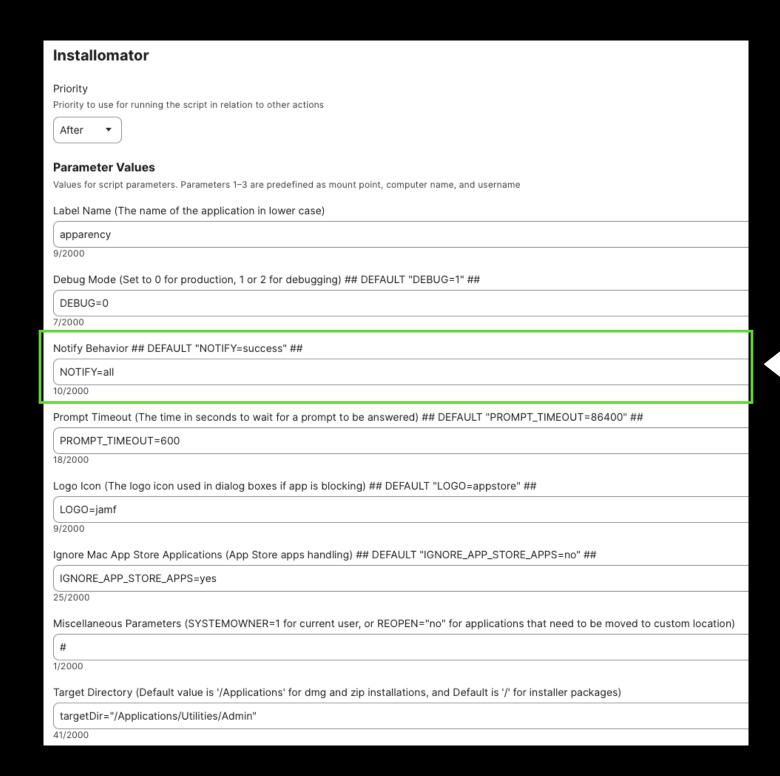


- Install Policy Install policy can be used to automatically install or via Self Service
 - Parameter #4: Label Name
 - Parameter #5: DEBUG=0



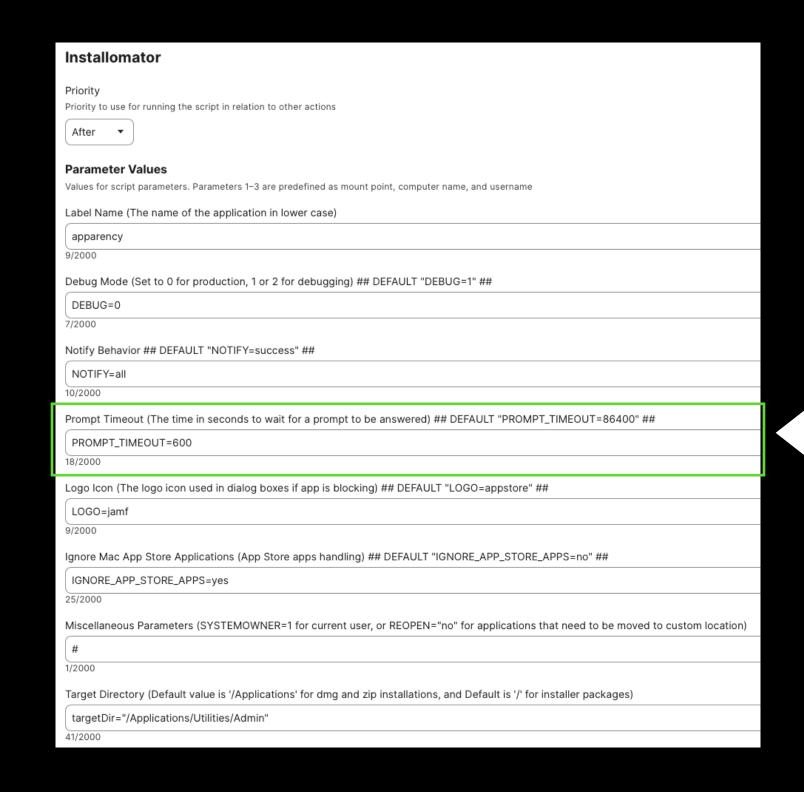


- Install Policy
 Install policy can be used to automatically install or via Self Service
 - Parameter #4: Label Name
 - Parameter #5: DEBUG=0
 - Parameter #6: NOTIFY=all



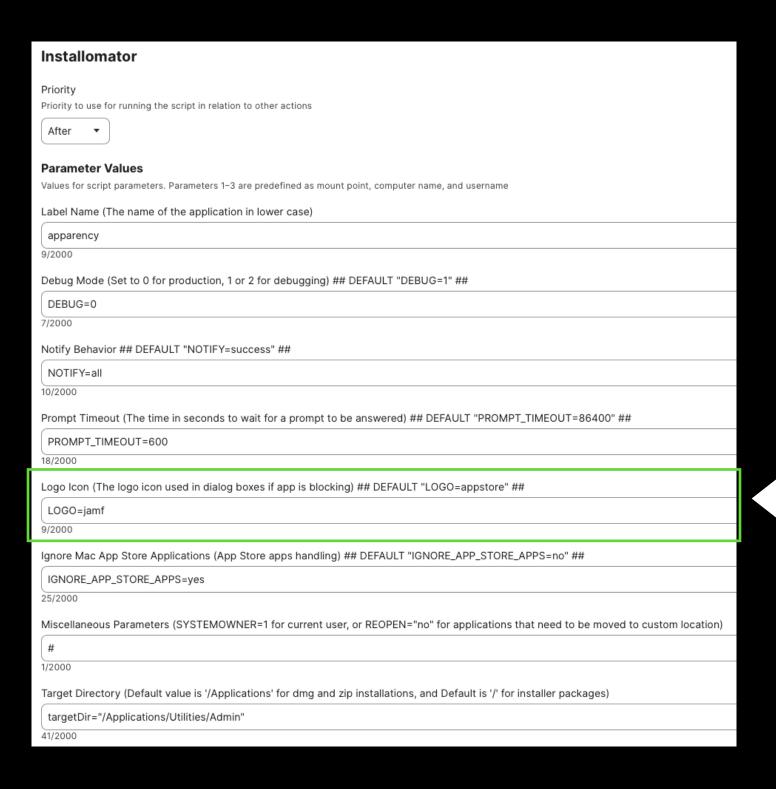


- Install Policy
 Install policy can be used to automatically install or via Self Service
 - Parameter #4: Label Name
 - Parameter #5: DEBUG=0
 - Parameter #6: NOTIFY=all
 - Parameter #7: PROMPT_TIMEOUT=600





- Install Policy
 Install policy can be used to automatically install or via Self Service
 - Parameter #4: Label Name
 - Parameter #5: DEBUG=0
 - Parameter #6: NOTIFY=all
 - Parameter #7: PROMPT_TIMEOUT=600
 - Parameter #8: LOGO=jamf

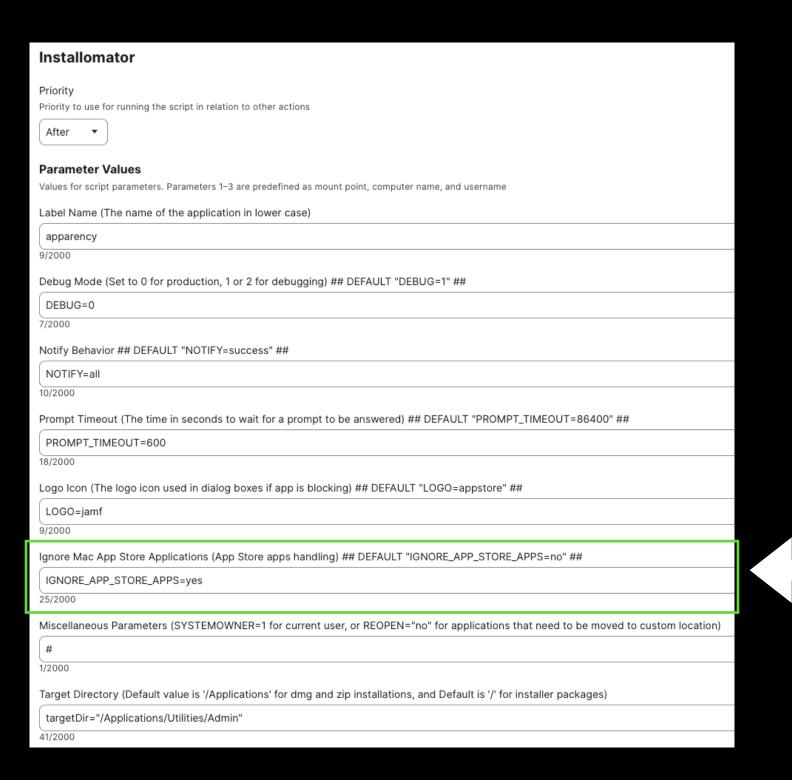




Installomator - Marriott Library's Setup

Install Policy

- Parameter #4: Label Name
- Parameter #5: DEBUG=0
- Parameter #6: NOTIFY=all
- Parameter #7: PROMPT_TIMEOUT=600
- Parameter #8: LOGO=jamf
- Parameter #9: IGNORE_APP_STORE_APPS=yes

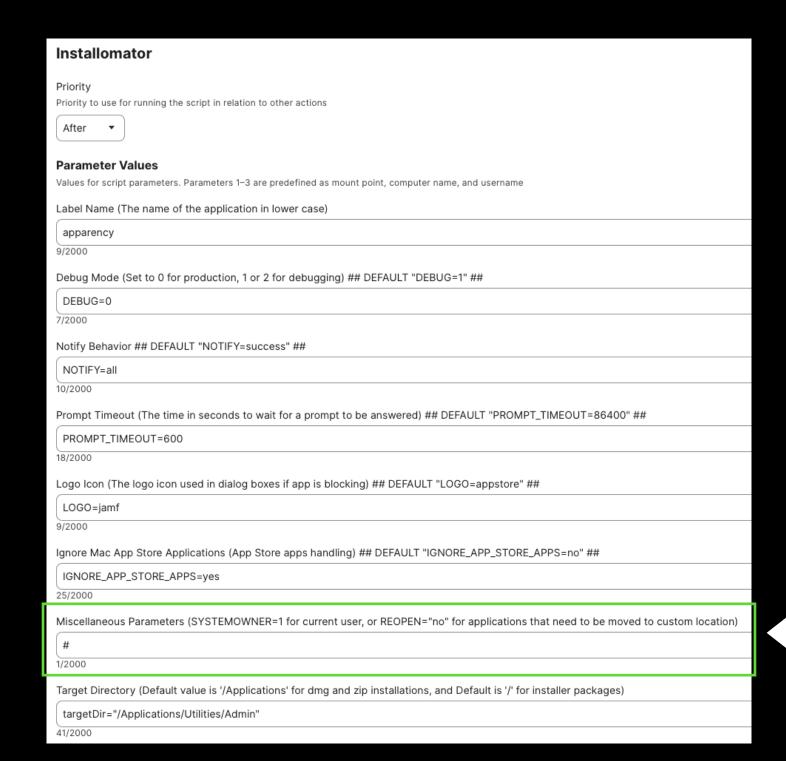




Installomator - Marriott Library's Setup

Install Policy

- Parameter #4: Label Name
- Parameter #5: DEBUG=0
- Parameter #6: NOTIFY=all
- Parameter #7: PROMPT_TIMEOUT=600
- Parameter #8: LOGO=jamf
- Parameter #9: IGNORE_APP_STORE_APPS=yes
- Parameter #10: Miscellaneous (System Owner, Re-open App, etc.)

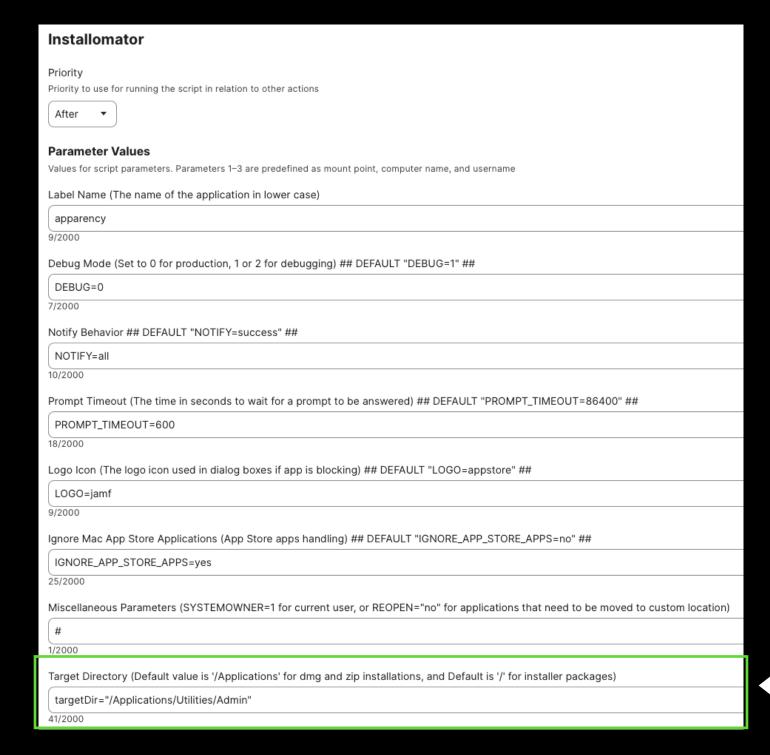




Installomator - Marriott Library's Setup

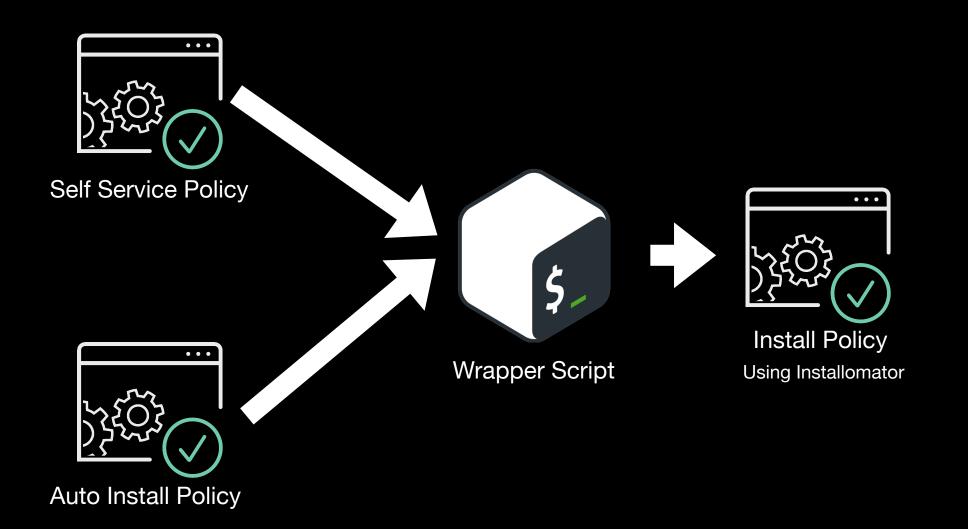
Install Policy

- Parameter #4: Label Name
- Parameter #5: DEBUG=0
- Parameter #6: NOTIFY=all
- Parameter #7: PROMPT_TIMEOUT=600
- Parameter #8: LOGO=jamf
- Parameter #9: IGNORE_APP_STORE_APPS=yes
- Parameter #10: Miscellaneous (System Owner, Re-open App, etc.)
- Parameter #11: Target Directory (DMG's & ZIP's, etc.)



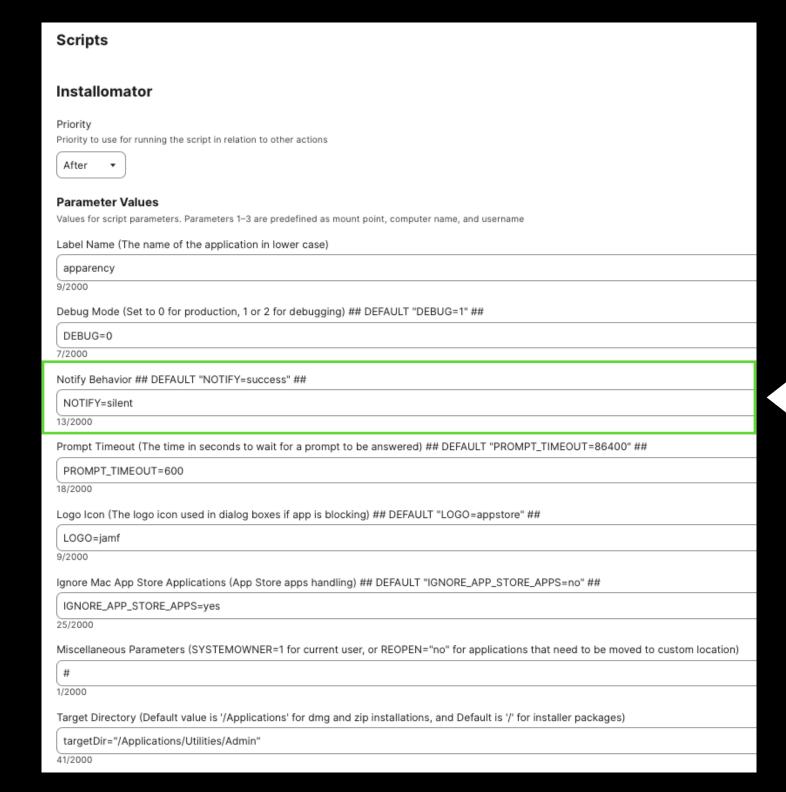


- Install Policy
 - Install policy can be used to automatically install or via Self Service
 - Trigger Name: [app_name]_install
 - Service Service Policy or Auto Install Policy Uses Trigger Name via wrapper script





- Update Policy
 Update policy can be used to automatically update
 - Parameter #4: Label Name
 - Parameter #5: DEBUG=0
 - Parameter #6: NOTIFY=silent
 - Parameter #7: PROMPT_TIMEOUT=600
 - Parameter #8: LOGO=jamf
 - Parameter #9: IGNORE_APP_STORE_APPS=yes
 - Parameter #10: Miscellaneous (System Owner, Re-open App, etc.)
 - Parameter #11: Target Directory (DMG's & ZIP's, etc.)





Installomator - Marriott Library's Setup

- Initial Update Policy Scoping Nested Groups
 Update policy can be used to automatically update
 - Trigger Name: [app_name]_update
 - Auto Update Policy Uses Trigger Name via wrapper script
 - First Smart Group (Reporting)

 Create a smart group that uses "patch reporting" as its criterion



Patch Reporting Software Title

Smart Computer Groups -> Criteria -> Patch Reporting Software Title -> [SOFTWARE] -> Latest Version



Installomator - Marriott Library's Setup

- Initial Update Policy Scoping Nested Groups
 Update policy can be used to automatically update
 - Trigger Name: [app_name]_update
 - Auto Update Policy Uses Trigger Name via wrapper script
 - First Smart Group (Reporting)
 Create a smart group that uses "patch reporting" as its criterion





Smart Computer Groups -> Criteria -> Patch Reporting Software Title -> [SOFTWARE] -> Latest Version

Second Smart Group (Scoping)
 This first smart group cannot be used directly to scope a policy.



Member of Group

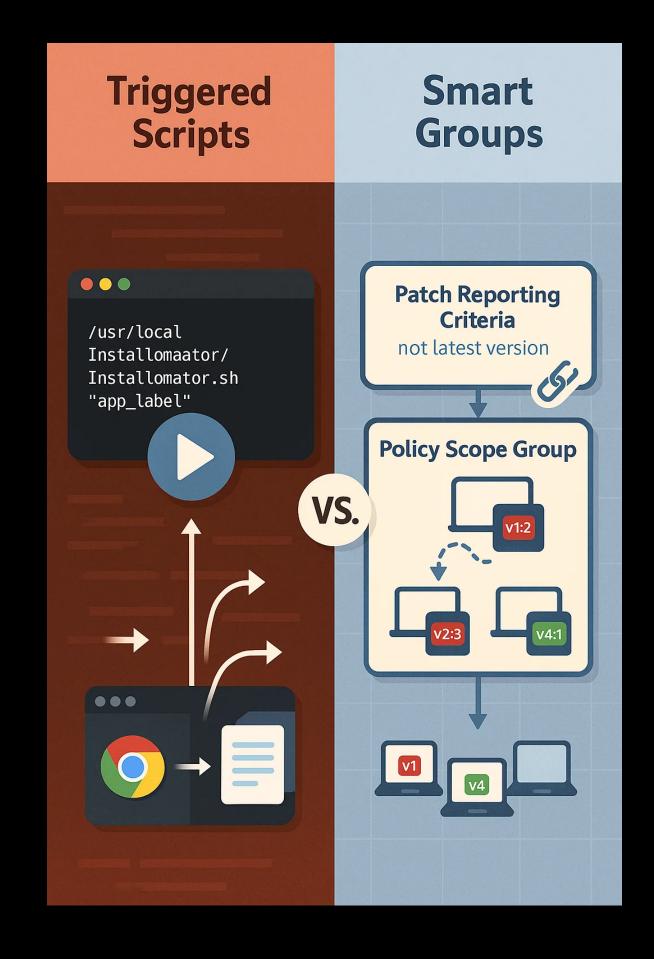
Smart Computer Groups -> Criteria -> Computer Group -> [SOFTWARE] - Patch Reporting Software Title



Installomator - Marriott Library's Setup

- Current Update Policy Moved to Script
 To lessen server resources move moved to script
 - Trigger Name: [app_name]_update
 - Auto Update Policy Uses Trigger Name via wrapper script
 - Validates Potential Installation Paths to Update
 - Policy Executes on Schedule (weekly)







Installomator - Marriott Library's Setup

Update Policy - Script Example

Wrapper script using case statement to validate paths and trigger update

```
CASE_ITEM_NAME)

ITEM_NAME="[APP_NAME]"

APP_CATEGORY="[APP_CATEGORY]"

UPDATE_TRIGGER_NAME="[APP_NAME]_udpate"

UPDATE_METHOD="installomator" or Jamf Installer Update or Adobe RUM Update

VERSION_KEY_VALUE="" CFBundleShortVersionString of CFBundleVersion or CFBundleGetInfoString

DEFAULT_PATH_CHECK=""

ALTERNATIVE_PATH_CHECK_1=""

ALTERNATIVE_PATH_CHECK_2=""

ALTERNATIVE_PATH_CHECK_3=""

;;
```



Installomator - Marriott Library's Setup

CVE Update Policy - Script Example

Wrapper script using case statement to validate paths and trigger update But, timeframe can be shortened to force quicker update (use jamf trigger & execution options)

```
CASE_ITEM_NAME)

ITEM_NAME="[APP_NAME]"

APP_CATEGORY="[APP_CATEGORY]"

UPDATE_TRIGGER_NAME="[APP_NAME]_udpate"

UPDATE_METHOD="installomator" or Jamf Installer Update or Adobe RUM Update

VERSION_KEY_VALUE="" CFBundleShortVersionString of CFBundleVersion or CFBundleGetInfoString

DEFAULT_PATH_CHECK=""

ALTERNATIVE_PATH_CHECK_1=""

ALTERNATIVE_PATH_CHECK_2=""

ALTERNATIVE_PATH_CHECK_3=""

;;
```



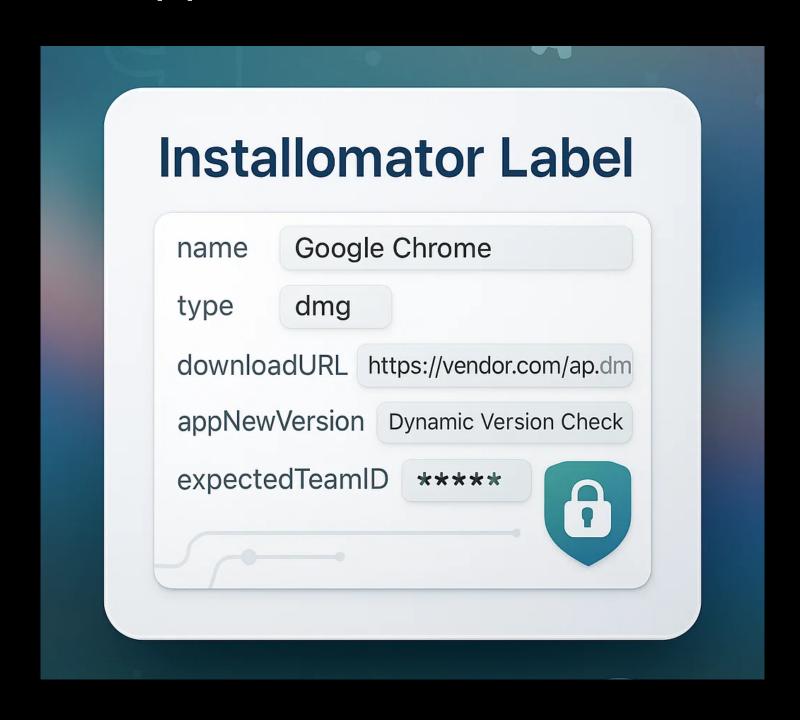
Or could use a nested smart group patch report/member & policy to address this need.





Installomator - Labels

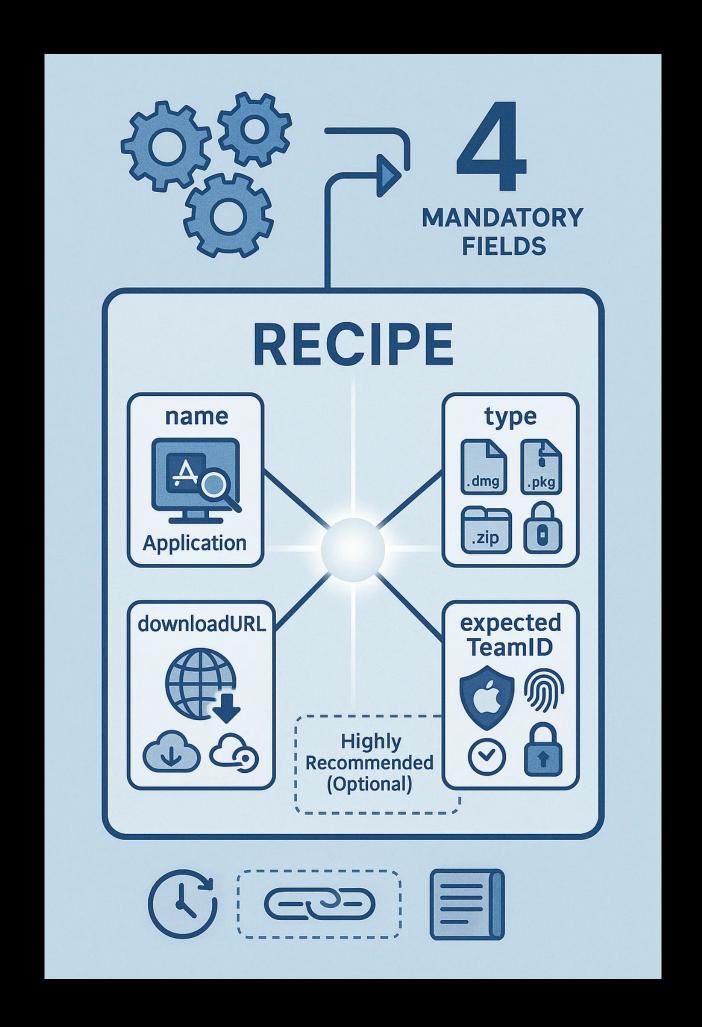
- Introduction
 - Is a script fragment that guides Installomator in handling an specfic application
 - Defines critical information such as
 - Application Name
 - Download URL
 - Expected Apple Developer Team ID
 - And other optional information





Installomator - Labels

- Mandatory Fields The Core "Recipe"
 - Installomator requires "4" mandatory fields
 - Used for a specific application install & update





Installomator - Labels





Installomator - Labels



Mandatory Fields

name

The application's display name, used for searching on the system.

```
suspiciouspackage)
    name="Suspicious Package"
    type="dmg"
    downloadURL="https://mothersruin.com/software/downloads/SuspiciousPackage.dmg"
    expectedTeamID="936EB786NH"
    ;;
```



Installomator - Labels



Mandatory Fields

name

The application's display name, used for searching on the system.

type

The installer format, such as dmg, pkg, zip, tbz, pkglnDmg, applnDmglnZip, or pkglnZip

```
suspiciouspackage)
    name="Suspicious Package"
    type="dmg"
    downloadURL="https://mothersruin.com/software/downloads/SuspiciousPackage.dmg"
    expectedTeamID="936EB786NH"
    ;;
```



Installomator - Labels



Mandatory Fields

name

The application's display name, used for searching on the system.

type

The installer format, such as dmg, pkg, zip, tbz, pkglnDmg, applnDmglnZip, or pkglnZip

downloadURL

The link from which to download the software.

```
suspiciouspackage)
   name="Suspicious Package"
   type="dmg"
   downloadURL="https://mothersruin.com/software/downloads/SuspiciousPackage.dmg"
   expectedTeamID="936EB786NH"
   ;;
```



Installomator - Labels



Mandatory Fields

name

The application's display name, used for searching on the system.

type

The installer format, such as dmg, pkg, zip, tbz, pkglnDmg, applnDmglnZip, or pkglnZip

downloadURL

The link from which to download the software.

expectedTeamID

The Apple Developer Team ID of the developer that signed the software

codesign -dv "/Applications/AppName.app" 2>&1 | grep "TeamIdentifier"

```
suspiciouspackage)
    name="Suspicious Package"
    type="dmg"
    downloadURL="https://mothersruin.com/software/downloads/SuspiciousPackage.dmg"
    expectedTeamID="936EB786NH"
;;
```



Installomator - Labels



(X) Highly Recommended Fields



Installomator - Labels



Highly Recommended Fields

appNewVersion

For determining the latest available version of the software directly from the vendor appNewVersion=\$(curl -fsIL "\$downloadURL" | grep -i location | cut -d'/' -f5 | tr -d '\r\n')

```
suspiciouspackage)
  name="Suspicious Package"
  type="dmg"
  downloadURL="https://mothersruin.com/software/downloads/SuspiciousPackage.dmg"
  appNewVersion=$(curl -fs https://mothersruin.com/software/SuspiciousPackage/get.html | grep 'class="version"' | sed -E 's/.*>([0-9\.]*) \(.*/\l/g')
  expectedTeamID="936EB786NH"
  ;;
```



Installomator - Labels



Highly Recommended Fields

appNewVersion

For determining the latest available version of the software directly from the vendor appNewVersion=\$(curl -fsIL "\$downloadURL" | grep -i location | cut -d'/' -f5 | tr -d '\r\n')

blockingProcesses

Lists application processes to quit or manage before an update

```
istatmenus)
   name="iStat Menus"
   type="zip"
   downloadURL="https://download.bjango.com/istatmenus/"
   expectedTeamID="Y93TK974AT"
   appNewVersion=$(curl -fs https://bjango.com/mac/istatmenus/versionhistory/| grep "<h3>" | head -1 | sed -E 's/<h3>([0-9.]*)<\/h3>/\1/')
   blockingProcesses=( "iStat Menus" "iStatMenusAgent" "iStat Menus Status" )
;;
```



Installomator - Labels



Highly Recommended Fields

appNewVersion

For determining the latest available version of the software directly from the vendor appNewVersion=\$(curl -fsIL "\$downloadURL" | grep -i location | cut -d'/' -f5 | tr -d '\r\n')

blockingProcesses

Lists application processes to quit or manage before an update

installerTool / CLIInstaller

Allows for custom installation methods, such as utilizing a command-line interface (CLI) installer provided by the vendor

```
grammarly)
    name="Grammarly Desktop"
    type="dmg"
    packageID="com.grammarly.ProjectLlama"
    downloadURL="https://download-mac.grammarly.com/Grammarly.dmg"
    expectedTeamID="W8F64X92K3"
    installerTool="Grammarly Installer.app"
    CLIInstaller="Grammarly Installer.apr/Contents/MacOS/Grammarly Desktop"
;;
```



Installomator

App Discovery & Version Detection Methods

App Discovery & Version Detection Methods



- This method is used when vendors provide fixed, predictable download links for their software
- Sometimes the version is part of the URL path, allowing for easy extraction



App Discovery & Version Detection Methods



- This method is used when vendors provide fixed, predictable download links for their software
- Sometimes the version is part of the URL path, allowing for easy extraction

```
downloadURL="https://vendor.com/downloads/app-latest.dmg"
```

```
appVersion=$(curl -fsL "URL" | tr -d '\n\r')
downloadURL="https://vendor.com/app-v${appVersion}.dmg"
```

Real World Example - Slack

```
downloadURL="https://slack.com/ssb/download-osx-universal"
```



App Discovery & Version Detection Methods

API-Based Discovery



JSON APIs

Many apps offer JSON APIs that return structured data containing versions and download URLs appNewVersion=\$(getJSONValue "\$firefoxVersions" "LATEST_FIREFOX_VERSION")



XML/RSS Feeds

Some update mechanisms (like Sparkle update feeds) often provide information in XML or RSS formats downloadURL=\$(curl -s https://versioncheck.barebones.com/BBEdit.xml | grep dmg | sort | tail -n1 | cut -d">" -f2 | cut -d"<" -f1)



GitHub API

Installomator can interact with the GitHub API to find release information and asset downloads downloadURL="\$(downloadURLFromGit swiftDialog swiftDialog)"



App Discovery & Version Detection Methods

API-Based Discovery

```
appVersion=$(curl -fsL "API_URL" | grep -o '"tag_name": *"[^"]*"' | cut -d'"' -f4 | sed 's/^v//')"
```

```
downloadURL=$(curl -fsL "URL/appcast.xml" | xpath '//item[1]/enclosure/@url' 2>/dev/null | cut -d'"' -f2)
```

Real World Example - Firefox - JSON API's

appNewVersion=\$(getJSONValue "\$firefoxVersions" "LATEST_FIREFOX_VERSION")

Real World Example - BBEdit - XML/RSS Feeds

```
downloadURL=$(curl -s https://versioncheck.barebones.com/BBEdit.xml | grep dmg | sort | tail -n1 | cut -d">" -f2 | cut -d"<" -f1)
```



App Discovery & Version Detection Methods

Web Scraping & HTML Parsing

This involves using command-line tools like curl to download the web page content, followed by grep, sed, awk, and regular expressions to extract the required data.



HTTP Headers Analysis

Inspect HTTP headers (e.g., Content-Disposition, Location from redirects) returned during a download request To find version information or the final download URL after redirects

appNewVersion="\$(curl -fsIL \${downloadURL} | grep -i ^location | cut -d "/" -f5)"



User-Agent Spoofing Some websit

Some websites require specific User-Agent headers to deliver the correct content

downloadURL=\$(curl -fs -H "User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15)" "https://evernote.com/download" | grep -i ".dmg" | grep -ioe "href.*" | cut -d '"' -f2)



App Discovery & Version Detection Methods

Web Scraping & HTML Parsing

```
downloadURL=$(curl -fsL -w "%{url_effective}" -o /dev/null "URL")
```

```
downloadURL=$(curl -fsI "URL" | grep -i "location:" | cut -d' ' -f2 | tr -d '\r')
```

Real World Example - Zoom - HTML Parsing

```
appNewVersion="$(curl -fsIL ${downloadURL} | grep -i ^location | cut -d "/" -f5)"
```

Real World Example - Evernote - User-Agent Spoofing

```
downloadURL=$(curl -fs -H "User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15)" "https://evernote.com/download" | grep -i ".dmg" | grep -ioe "href.*" | cut -d '"' -f2)
```



Installomator Testing & Debugging

Testing & Debugging

Primary Diagnostic Tool: DEBUG Modes

Installomator includes a built-in debug mode to facilitate testing without making changes to client systems

DEBUG=1

Downloads, verifies, and works through all installation steps but does NOT install anything or change apps

Purpose: Ideal for initial testing and debugging new or modified labels

• DEBUG=2

Performs a dry run as a regular user (not root)

Purpose: Useful for testing label logic from a user's perspective during development & testing



Testing & Debugging

- Log Analysis
 - Installomator Logs
 Installomator provides comprehensive logging to /var/log/Installomator.log
 - Jamf Pro Logs
 Real-time Monitoring: Use tail -f /var/log/jamf.log

```
Policy Logs
Policy -> Logs -> Details
```

<u>Computer Logs</u> Computer -> History -> Policy Logs



Testing & Debugging



Exit Codes

Various exit codes to indicate the success or specific failure conditions of an installation or update process https://github.com/Installomator/Installomator/wiki/Installomator-Exit-Codes

• 0: Success

This code means everything went well, or at least as expected

• 1: Unknown label

Could not find a "recipe" or label for the application name provided

• 2 : Download Error

Indicates a problem during the download process, which could be due to a broken downloadURL

• 4: Error verifying download

Include certain types of Gatekeeper issues, specifically those related to the downloaded file's code signing and notarization

• 5: Team IDs do not match

The **Team ID** of the downloaded software did not match the expectedTeamID specified in the Installomator label



Testing & Debugging

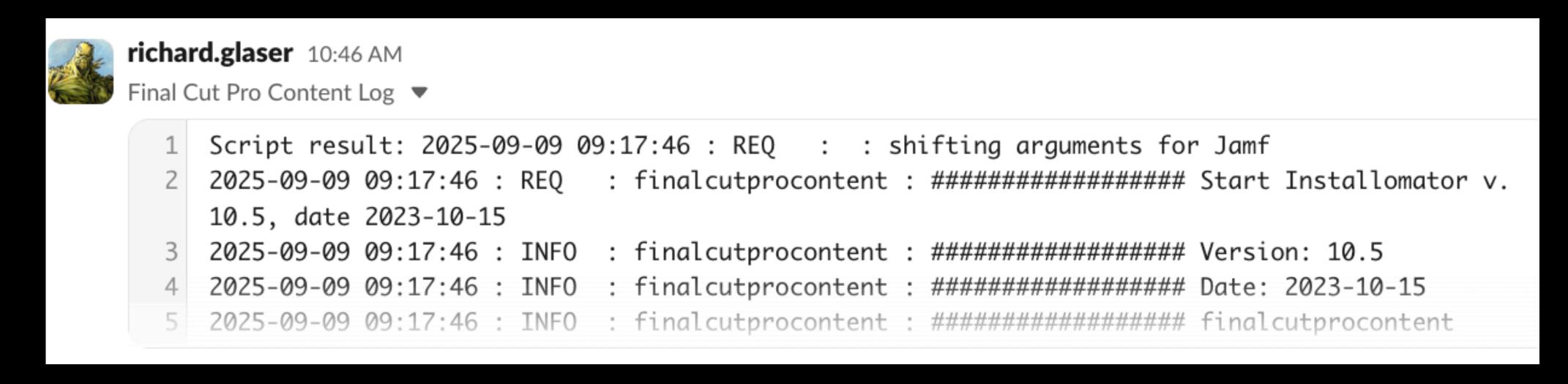


MacAdmins Slack Channel - For Support Questions

The #installomator Slack channel is the go-to place for discussion, support, and advice. If you're in doubt, ask there

Text Snippets

When sharing code snippets, logs, or detailed command outputs in Slack, it's recommended to use Slack's builtin "text snippet" feature to make them easier to read and analyze.





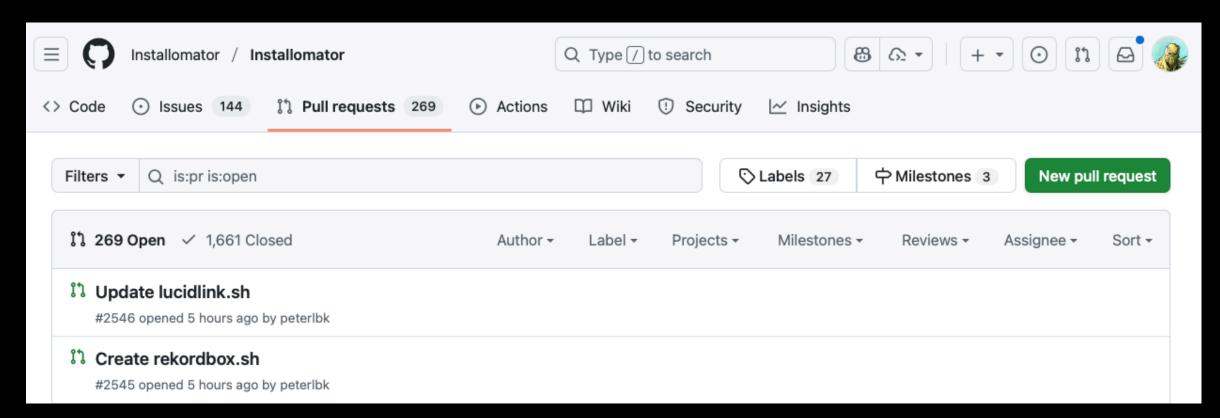
Testing & Debugging



GitHub - Contributing Solutions

https://github.com/Installomator/Installomator/pulls

- <u>Submit Pull Requests (PRs)</u>
 Create PRs for broken labels, improvements, or new features. Providing code via a PR is the preferred method https://github.com/Installomator/Installomator/wiki/GitHub-howto-create-PRs
- <u>Search Existing Resources First</u>
 Before asking a question or filing an issue, always search the GitHub repository for existing Pull Requests or Issues, and check the MacAdmins Slack channel for similar discussions.









Links & Resources

Presentation & Speaker Information

Marriott Library Student Checkout Devices Technical Details, Processes & Pain Points
 The presentation reviewed the Marriott Library's student device checkout programs, covering device details, onboarding processes for various OS, and key challenges like support issues, device management, and security concerns.

https://stream.lib.utah.edu/index.php?c=details&id=13631

JNUC 2021 - Python-jamf & jctl

This YouTube video presents a session from JNUC 2021 discussing the use of python-jamf and jctl for Jamf Pro API automation..

https://www.youtube.com/watch?v=2YLriNwyP3s



Links & Resources

Platform & Management Solutions

Microsoft Windows Preinstallation Environment (WinPE) Introductions

This Microsoft Learn article introduces WinPE, a lightweight operating system used for deploying Windows, which is relevant for automated Windows imaging...

https://learn.microsoft.com/en-us/windows-hardware/manufacture/desktop/winpe-intro?view=windows-11

Microsoft Endpoint Configuration Manager Introduction

This resource provides an introduction to Microsoft Endpoint Configuration Manager, utilized for staging and managing Windows device collections in an enterprise environment.

https://learn.microsoft.com/en-us/intune/configmgr/core/understand/introduction

Microsoft Endpoint Configuration Manager Introduction

This resource provides an introduction to Microsoft Endpoint Configuration Manager, utilized for staging and managing Windows device collections in an enterprise environment.

https://learn.microsoft.com/en-us/intune/configmgr/core/understand/introduction



Links & Resources

Platform & Management Solutions

Jamf Patch Management

This Jamf documentation details how Jamf Pro's patch management features can be used to manage repackaged application patching for notable applications and updates.

https://learn.jamf.com/en-US/bundle/jamf-pro-documentation-current/page/PatchManagement.html

Jamf Patch Management Software Titles

This Jamf documentation details how Jamf Pro's patch management features can be used to manage repackaged application patching for notable applications and updates.

https://learn.jamf.com/en-US/bundle/jamf-app-catalog/page/Patch_Management_Software_Titles.html

Jamf Title Editor

The Jamf Title Editor is a cloud-hosted solution powered by Kinobi technology that allows for custom software title definitions to enhance Jamf Pro's patch management capabilities.

https://learn.jamf.com/en-US/bundle/title-editor/page/About_Title_Editor.html

Jamf Pro App Installers (Requires Jamf Cloud)

This Jamf App Catalog resource pertains to App Installers, which are available for organizations using Jamf Cloud to manage software titles.

https://learn.jamf.com/en-US/bundle/jamf-app-catalog/page/App_Installers_Software_Titles.html



Links & Resources

Imaging & Maintenance Tools (Hardware Specific)

<u>iMazing Configurator (iPad Imaging)</u>

Mazing Configurator is a tool that automates iPad imaging, connecting devices to campus wireless and re-enrolling them in Jamf Pro.

https://imazing.com/configurator

Cambrionix ThunderSync3-C10 USB Hub

his link points to a high-speed, multi-device USB hub with advanced control features, suitable for iPad imaging workflows.

https://www.cambrionix.com/products/thundersync3-c10

DFU Blaster Pro (Apple Laptop Imaging)

DFU Blaster Pro is a tool that automates Apple laptop imaging, including DFU mode and re-enrollment in Jamf Pro.

https://twocanoes.com/products/mac/dfu-blaster/

Acroname USBHub3c Pro Edition Hub

This is a high-speed USB hub offering programmable control for imaging multiple Macs efficiently.

https://acroname.com/store/programmable-industrial-power-delivery-hub-s99-usbhub-3c-pro

Satechi 200W USB-C 6-Port GaN Charger

Students often return systems with low batteries, or they die between imaging sessions. This will keep them charged, allowing immediate imaging without waiting for a charge.

https://satechi.net/products/6-port-gan-charger?variant=40197715492952



Links & Resources

macOS Automation & Scripting Tools

S.U.P.E.R.M.A.N. (Open Source Script
 S.U.P.E.R.M.A.N. is an open-source script that automates software updates, upgrades, and restart scheduling for enterprise macOS environments.

https://github.com/Macjutsu/super

swiftDialog (Open Source Utility)

swiftDialog is an open-source utility for creating customizable dialog boxes and user interfaces, often used with administrative scripts and workflows.

https://github.com/swiftDialog/swiftDialog

outset (Open Source Tool

Outset is an open-source tool that runs scripts and applications at startup, login, or on-demand, based on folder organization.

https://github.com/macadmins/outset

Keyboard Maestro

Keyboard Maestro is a tool used for custom scripts and to automate repetitive tasks like typing text or clicking.

https://www.keyboardmaestro.com/main/



Links & Resources

macOS Automation & Scripting Tools

• <u>munkipkg (Open-source packaging tool)</u> munkipkg is an open-source packaging tool for automated, reproducible .pkg builds, often used for repackaging Mac App Store and commercial applications.

https://github.com/munki/munki-pkg

• <u>python-jamf (Library for Jamf Pro API data)</u> python-jamf is a library designed for easy access to Jamf Pro API data, simplifying policy and patch management tasks.

https://github.com/univ-of-utah-marriott-library-apple/python-jamf

• <u>jctl (CLI tool for Jamf Pro/Cloud CRUD operations)</u>

jctl is a command-line interface (CLI) tool that uses python-jamf to perform Create, Read, Update, and Delete (CRUD) operations on Jamf Pro or Jamf Cloud servers, simplifying repetitive UI tasks into single commands.

https://github.com/univ-of-utah-marriott-library-apple/jctl

AutoPKG

AutoPKG is a tool for automated workflows for testing and deployment, often used with repositories for Mac runner environments.

https://github.com/autopkg/autopkg



Links & Resources

Installomator - Core Resources

Installomator GitHub Repository

This is the main GitHub repository for Installomator, an open-source tool that automates the installation and updating of macOS applications.

https://github.com/Installomator/Installomator

Installomator Releases (for signed PKG)

This page provides access to signed PKG releases of Installomator, which can be downloaded and uploaded to MDM solutions like Intune.

https://github.com/Installomator/Installomator/releases/

Installomator Exit Codes (Wiki)

This wiki page details the various exit codes Installomator uses to indicate the success or specific failure conditions of an installation or update process.

https://github.com/Installomator/Installomator/wiki/Installomator-Exit-Codes

• Installomator GitHub: How-to create Pull Requests (Wiki)

This wiki provides guidance on how to create pull requests for Installomator, which is the preferred method for contributing code, improvements, or new features.

https://github.com/Installomator/Installomator/wiki/GitHub-howto-create-PRs



Links & Resources

Installomator - MDM Integrations (Wiki Pages)

Installomator with Jamf Pro

This wiki page provides specific instructions and best practices for integrating Installomator with Jamf Pro.

https://github.com/Installomator/Installomator/wiki/MDM:-Jamf-Pro

Installomator with Microsoft Intune (Endpoint Manager)

This wiki page offers guidance on integrating Installomator with Microsoft Intune (Endpoint Manager) for macOS app deployment.

https://github.com/Installomator/Installomator/wiki/MDM:-Microsoft-Intune-(Endpoint-Manager)

Installomator with Filewave

This Filewave knowledge base article provides a recipe for using Installomator as a universal installer script with Filewave.

https://kb.filewave.com/books/software-deployment-recipes-macos/page/installomator-the-one-installer-script-to-rule-them-all-macos-pkg



Links & Resources

Related & Alternative Solutions

Squid Web Proxy Cache

Squid Web Proxy Cache is a workaround solution for network constraints and bandwidth limitations, especially in school environments, to manage vendor downloads efficiently.

https://github.com/squid-cache/squid

Electrona Patch

Electrona Patch is a commercial solution for patch management, presented as an alternative to or alongside Installomator.

https://www.alectrona.com/patch

App Catalog

App Catalog is a commercial solution for managing applications, also mentioned as a tool in the broader software lifecycle.

https://appcatalog.cloud

Intuneomator (Intune + Installomator project)

Intuneomator is a project that combines Microsoft Intune with Installomator to simplify the deployment and updating of Mac applications via Intune, especially to overcome Intune's script limitations.

https://github.com/gilburns/Intuneomator



Links & Resources

Related & Alternative Solutions

App Auto Patch

App Auto-Patch is a Mac application patch management tool that automates app discovery, updates, and reporting through Jamf Pro and other MDMs using swiftDialog and Installomator integration.

https://github.com/App-Auto-Patch/App-Auto-Patch

Patchomator

Patchomator is a management script that extends Installomator, automating the discovery, configuration, and updating of Mac applications.

https://github.com/Mac-Nerd/patchomator

InstallomatorMate

nstallomator Mate is an app designed for macOS 14.2 and above that helps manage software installations using customized scripts and integrates with GitHub repositories..

https://hennig.nu/appar/installomatormate/installomator-mate-faq

