



App Auto-Patch: Hands-Off Application Updates



App Auto-Patch: Hands-Off Application Updates



Rob Schroeder

Senior Apple Engineer II



Andrew Spokes

Client Engineer





App Auto-Patch

Agenda

- Overview
 - Variables and Configuration
 - Deployment
 - Future Development
 - Resources / Q&A
- 

What is App Auto-Patch?

- App Auto-Patch is designed to make life easier for Mac Admins by automating and simplifying the application patching process.
- Our goal is to provide a tool that integrates smoothly with existing systems, supports various applications, and helps create a more productive and secure Mac environment for everyone.
- App Auto-Patch was inspired by Patchomator, a script that utilizes Installomator with specific app labels to help update applications



How App Auto-Patch Works?

- **Discovery Phase**

- Installed applications are checked against Installomator labels
- Version comparison between app installed and what's available from the vendor
- If newer versions are available, apps are queued for updates

- **Deferral Check**

- If enabled, deferral prompts are displayed using SwiftDialog
- Check for Display Assertions (Screen Sharing, etc.)

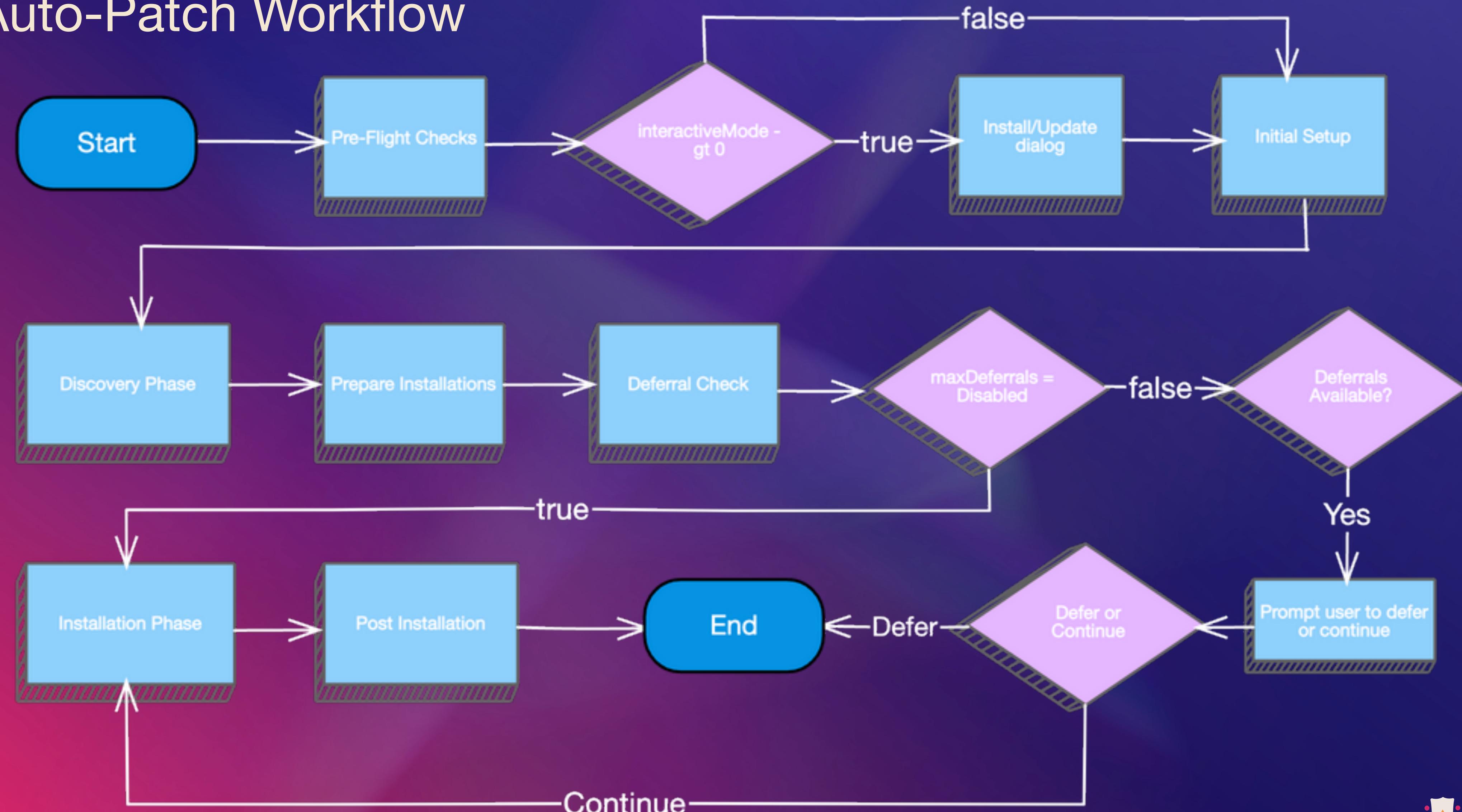
- **Installation Phase**

- Execution of updates for queued applications



Technical Overview

App Auto-Patch Workflow



End-User Experience

There are three options for how App Auto-Patch behaves:

Mode	Description
(0) Completely Silent	Discovers and Patches apps silently in the background with no user interaction
(1) Silent Discovery, Interactive Patching	Discovers apps silently, displays deferrals (if enabled) and installation status if updates are required
(2) Fully Interactive	Displays all activity including Discovery, Deferrals and installation phase



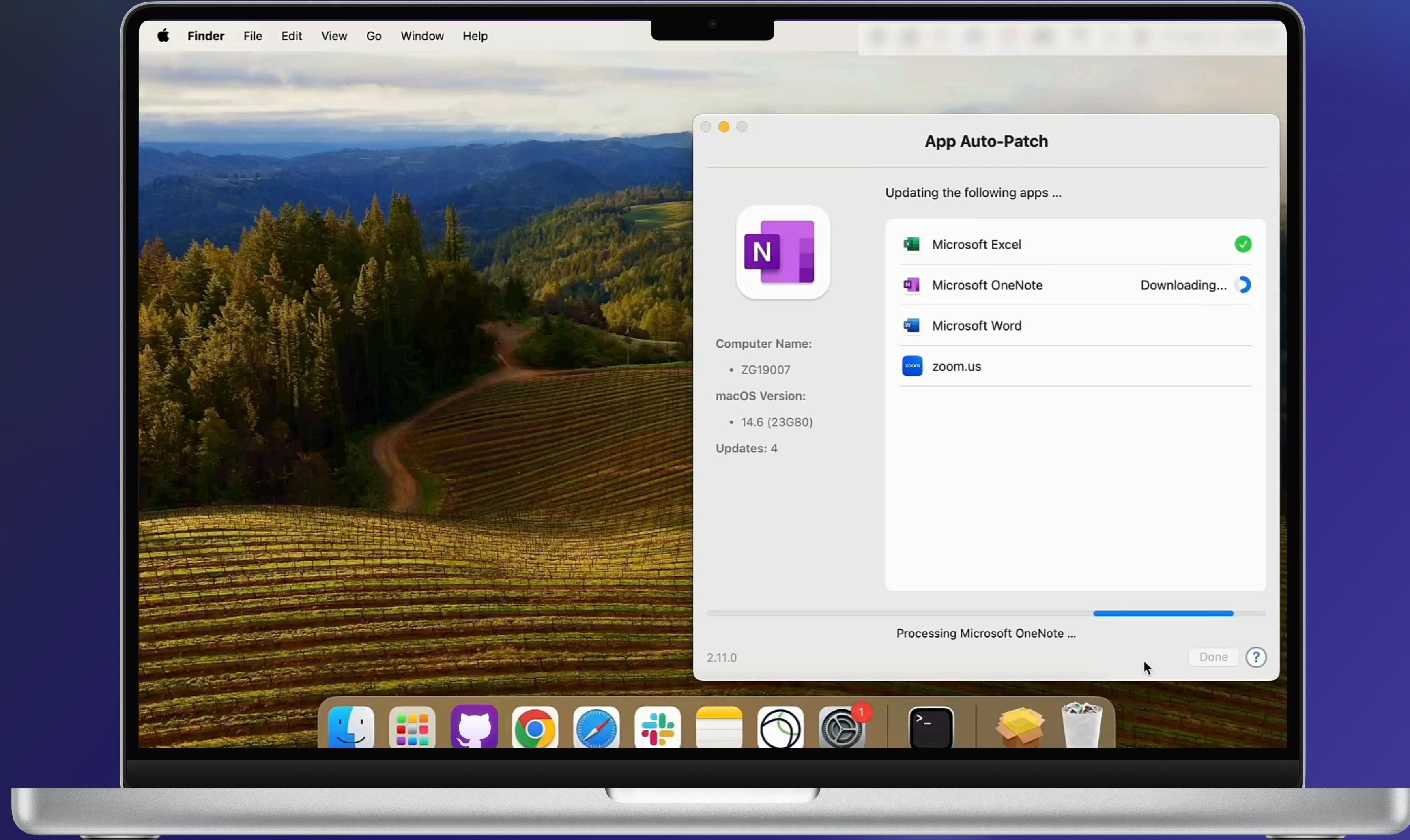
End-User Experience



End-User Experience



End-User Experience



Installlomator Label Categories

App Auto-Patch scans for apps matching all possible labels, but you can have a tailored approach using the following label categories:

- Ignored Labels - Any matches in this list will be ignored
- Required Labels - Any labels added to this list will be forced installed/Updated
- Optional Labels - Install/update will be forced only if app is installed



Variables and Configuration

Jamf Pro Parameter Variables

```
# Parameter 4: Interactive Mode
interactiveMode="${4:="2"}"

# Parameter 5: A space-separated list of Installomator labels to ignore
ignoredLabels="${5:=""}"

# Parameter 6: A space-separated list of required Installomator labels
requiredLabels="${6:=""}"

# Parameter 7: A space-separated list of optional Installomator
optionalLabels="${7:=""}"

# Parameter 8: A space-separated list of options to override default Installomator options
# (i.e., BLOCKING_PROCESS_ACTION=prompt_user NOTIFY=silent LOGO=appstore)
installomatorOptions="${8:-""}"

# Parameter 9: Number of Deferrals Allowed
maxDeferrals="${9:-"Disabled"}"
```



Variables and Configuration

Feature Variables

```
### Deferral Options ###

# Time given to the user to respond to deferral prompt if enabled
deferralTimer=3600

# What happens when the deferral timer expires [ Defer | Continue ]
deferralTimerAction="Defer"

# Number of days after the Patch Start Date to reset patch status to False
daysUntilReset=7

# The day of week to set the start date for weekly patching: (1=Mon 2=Tue...7=Sun)
patchWeekStartDay="3"

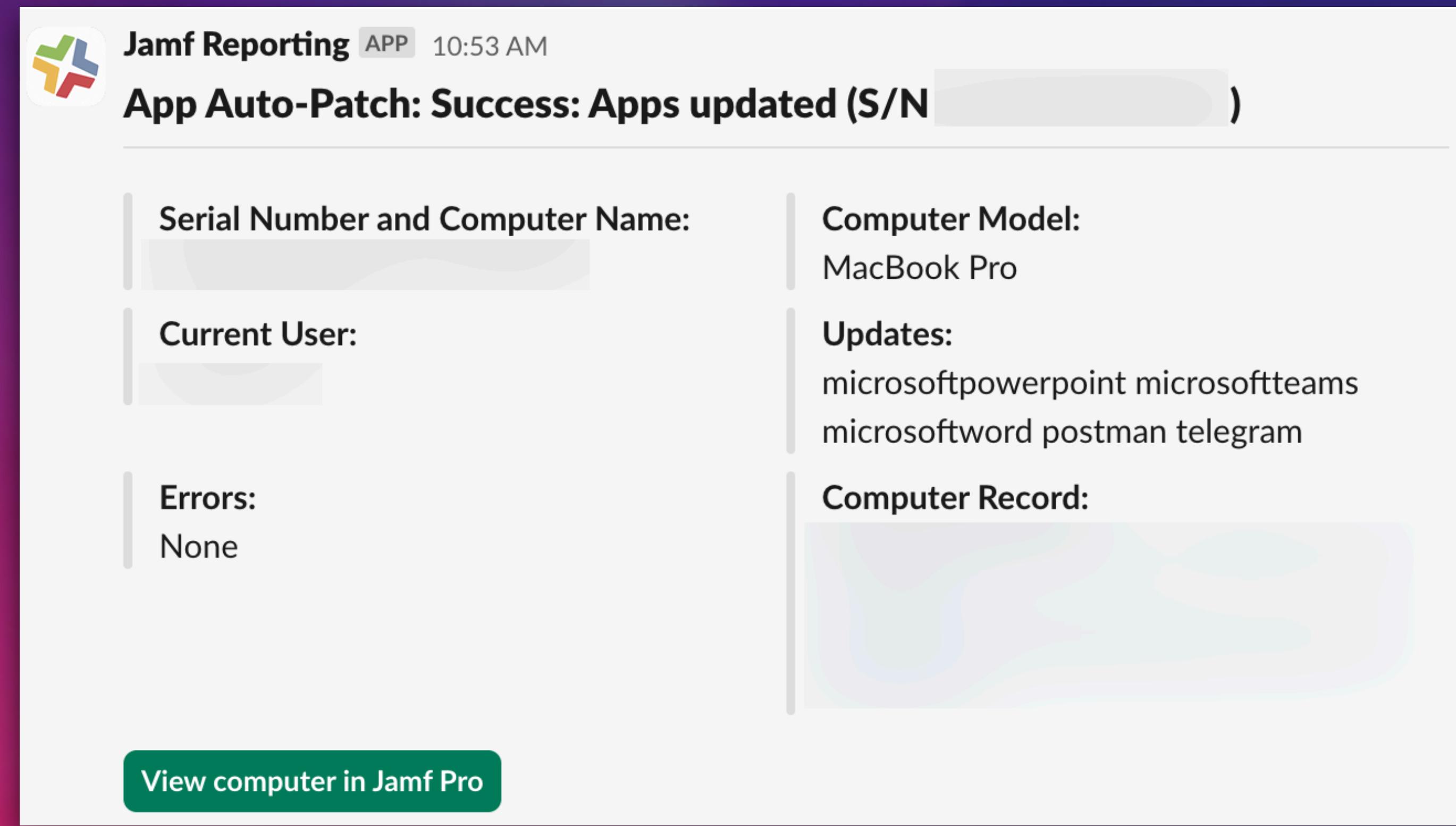
# The maximum number of deferred attempts from Display Assertions
maxDisplayAssertionCount="1"
```



Variables and Configuration

Webhooks

- A Webhook URL can be provided for Slack or Teams to either notify each time App Auto-Patch or when failures are detected



Variables and Configuration

Additional Variables

- Here are some additional variables within the script:
 - runDiscovery: Re-run discovery of installed applications
 - removeInstallomatorPath: Removes Installomator script after App Auto-Patch has completed
 - convertAppsInHomeFolder: Remove apps in /Users/* and install them to the default path
 - ignoreAppsInHomeFolder: Ignore apps found in /Users/*
 - useLatestAvailableInstallomatorScriptVersion: Compares the VERSIONDATE of the local installomator script against the latest available on Github and updates if needed



Deployment and Configuration

Getting Started with Jamf Pro - Policy Configuration

- Upload the script to Jamf and set Parameter Labels
- Create a new Policy and configure the General payload options:
 - Display Name
 - Trigger
 - Execution Frequency
- Configure the Script Payload to use the App Auto-Patch script previously uploaded
- Set the parameters for the script according to your organization's needs
- Adjust scope and test



Deployment and Configuration

The screenshot shows the Jamf Pro web interface under the 'Computers : Policies' section. The main title is 'Tool - App Auto-Patch - Deployment'. On the left sidebar, under 'Policies', the 'Scripts' option is selected. The main content area shows a 'General' section with a 'Priority' dropdown set to 'After'. Below it is a 'Scripts' section containing a single entry named 'AAP-App-Auto-Patch'. This script has several parameter fields:

- Priority:** After
- Parameter Values:** Values for script parameters. Parameters 1-3 are predefined as mount point, computer name, and username.
- Interactive Mode:** [0 (Completely Silent) | 1 (Silent Discovery, Interactive Patching) | 2 (Full Interactive) (default)]
Value: 1
- ignoredLabels:** A space-separated list of Installomator labels to ignore (i.e., "microsoft* googlechrome* jamfconnect zoom* 1password* firefox* swiftdialog")
Value: 1password* androidstudio azuredatastudio bettertouchtool beyondcomparepro clickshare deonotify displaylinkmanager docker dropbox firefox firefox_da firefox_intl fi
- requiredLabels:** A space-separated list of required Installomator labels (i.e., "firefoxpkg_intl")
Value:
- optionalLabels:** A space-separated list of optional Installomator labels (i.e., "renew") ** Does not support wildcards **
Value:
- installomatorOptions:** A space-separated list of options to override default Installomator options (i.e., BLOCKING_PROCESS_ACTION=prompt_user NOTIFY=silent LOGO=appstore)
Value: BLOCKING_PROCESS_ACTION=quit_kill NOTIFY=silent LOGO=appstore IGNORE_APP_STORE_APPS=yes
- maxDeferrals:** Number of times a user is allowed to defer before being forced to install updates. A value of "Disabled" will not display the deferral prompt. ['integer' | Disabled (default)]
Value: 2



Future Development



Future Development

App Auto-Patch Version 3.0

- Managed using Configuration Profiles (MDM or Local)
- Command Line Triggers
- Automation and Scheduling via LaunchDaemon
- Enhanced Deferral Options
- Process application updates at login window
- MDM agnostic

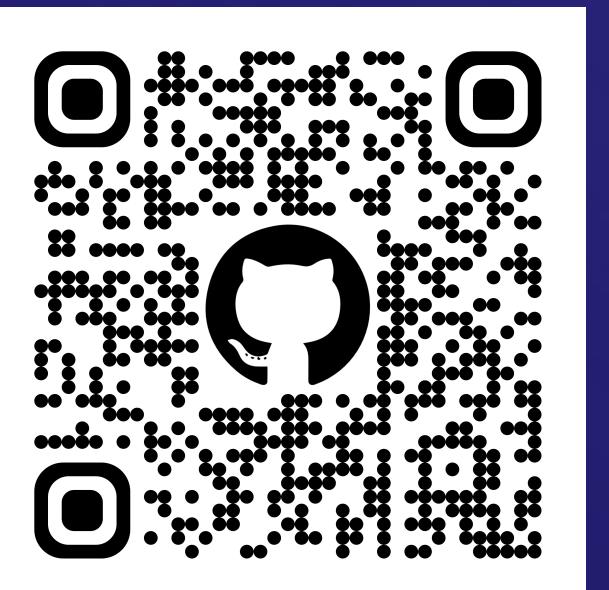
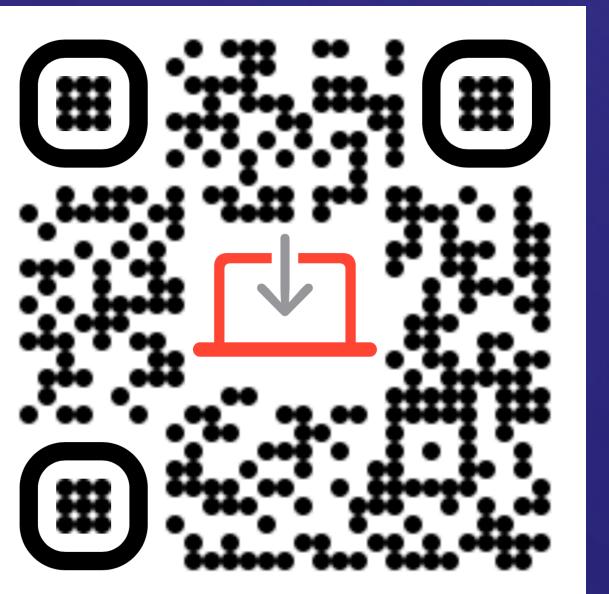


Resources / Q & A



Resources / Q & A

- App Auto-Patch Blog
 - <https://techitout.xyz/app-auto-patch>
- App Auto-Patch wiki
 - <https://github.com/App-Auto-Patch/App-Auto-Patch/wiki>
- MacAdmins Slack
 - #app-auto-patch



App Auto-Patch: Hands-Off Application Updates



Thank You!

