Training Day 2022

FileWave Technical Sessions

filewave

Using Installomator to support 3rd party apps (macOS)

Presenter: Josh Levitsky

filewave

2 main approaches to Mac app management

Always be on the latest release (Installomator)
- Fileset revisions are irrelevant
- Fileset will automatically update app to latest version on regular basis
- Everyone always gets the latest revision
- Set it and forget approach = less work

Tightly controlled testing, versioning, and release management by IT (AutoPkg)
- New versions are imported as revisions of a fileset
- IT admins test and choose when to switch default revision to latest release
- Lets you deploy different versions/revisions of an app to different groups of Macs
- More control = more work on your part
- Less likely to deploy a bad release of an app

Introducing Installomator

Shell script that downloads and installs > 450 different Mac apps
- Takes a parameter, a label, for name of the app to install
- Normally only installs app if latest version is newer than installed version
  - Be careful - there are exceptions to this so test, test, test
- Single FileWave fileset to install and update app
- Handy configuration options to
  - Notify the user when done
  - Close app before installing (important)
  - Prompt user before closing app (really important)
  - Re-launch app after update if it was open
  - Ignore App Store apps
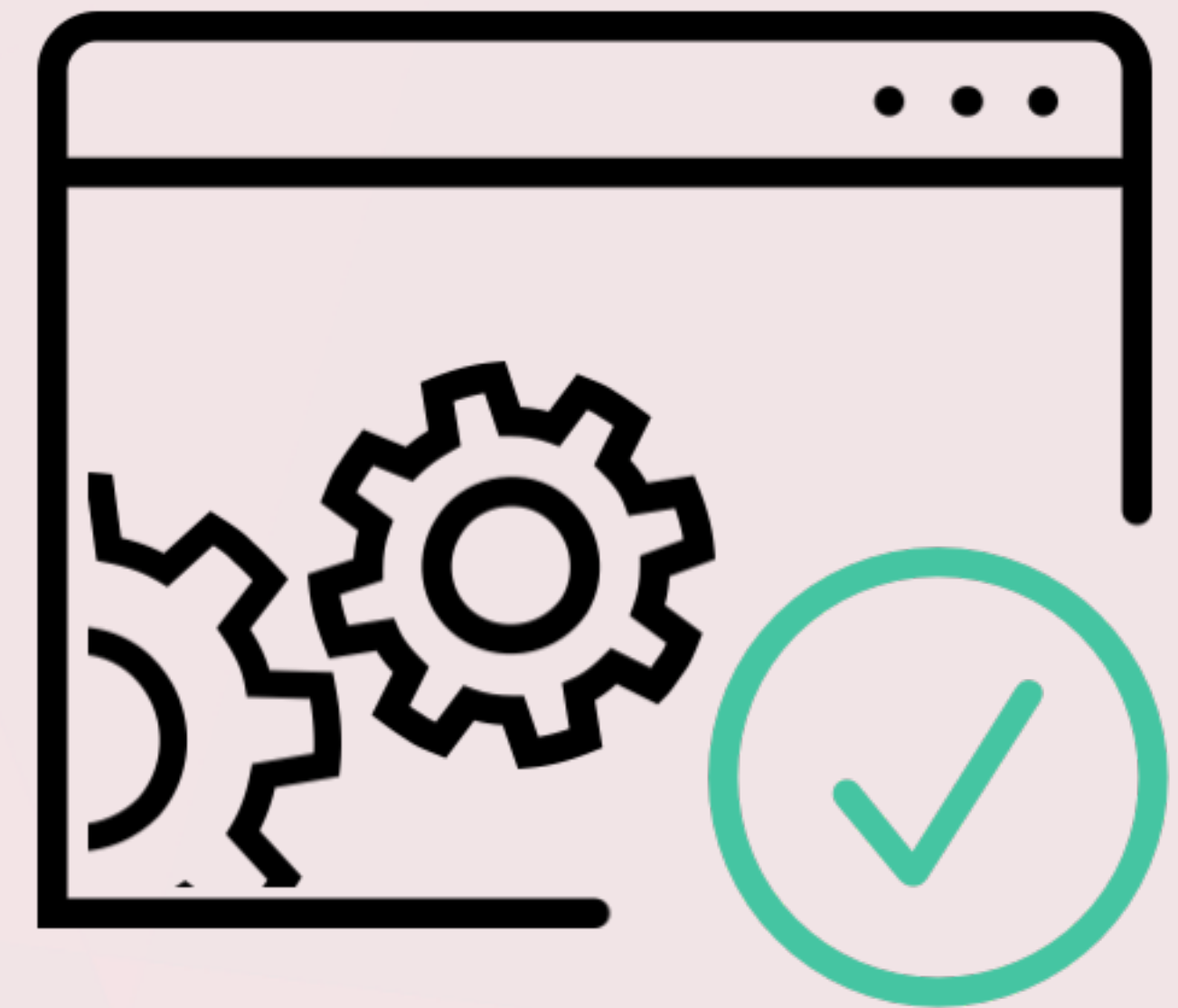
Deploy Installomator to all your Macs

1. Download PKG from https://github.com/Installomator/Installomator and install on Mac Admin
   - Script file installed to /usr/local/Installomator/Installomator.sh
2. Edit Installomator.sh to customize configuration options
3. Create empty fileset and add Installomator.sh to /usr/local/bin folder
4. Assign fileset to all your Macs
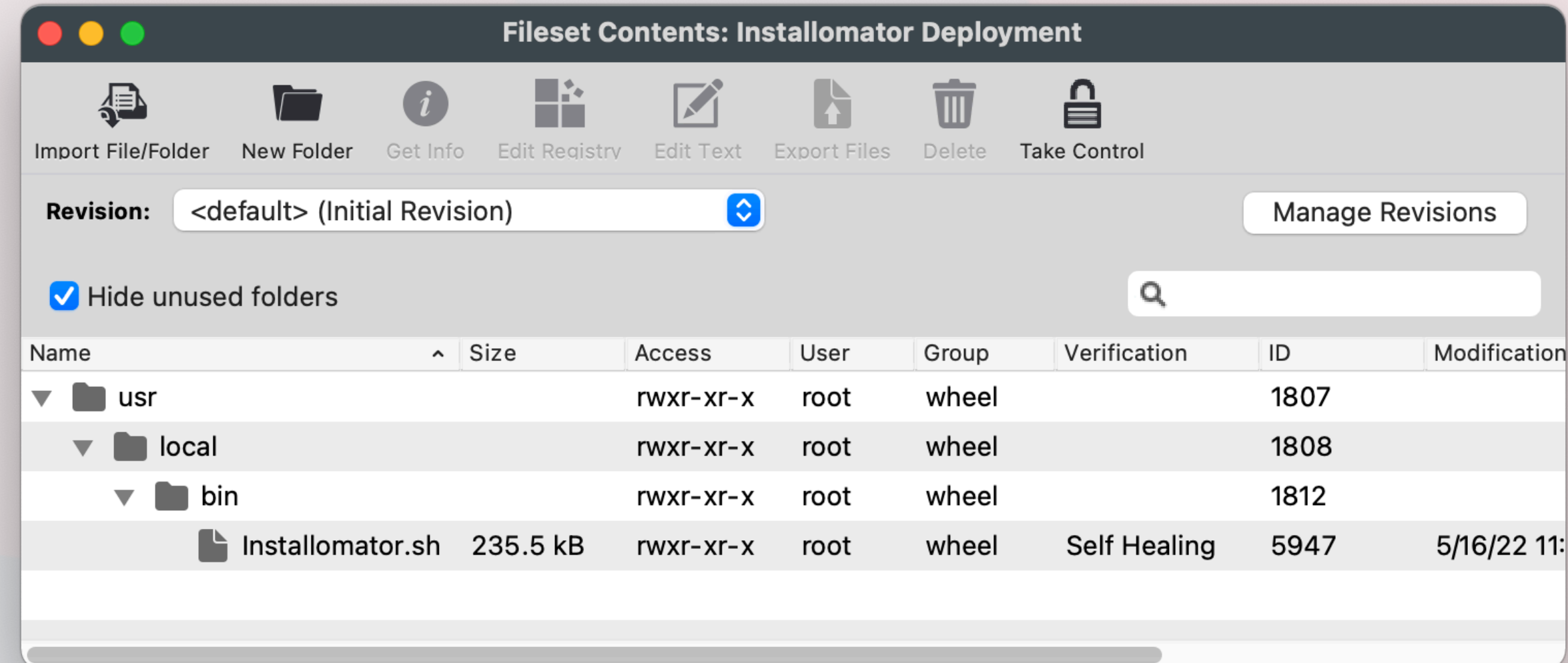
Build fileset to deploy & update app via Installomator

Editing check_installomator.sh

check_installomator.sh

```
#!/bin/bash

if [ -f "/usr/local/bin/Installomator.sh" ]; then
        exit 0
else
        echo "Could not find Installomator"
fi
```

Line Endings: ● Unix   ○ Windows   Cancel   OK

> Requirements script: Activate fileset only if Installomator is present

> Activation script for initial deployment

Editing install.sh

install.sh

```
#!/bin/bash
/usr/local/bin/Installomator.sh $1 LOGO=/usr/local/sbin/FileWave.app/Contents/Resources/fwGUI.app/Contents/Resources/kiosk.icns
```

Line Endings: ● Unix   ○ Windows   Cancel   OK

Editing verify.sh

verify.sh

```
#!/bin/bash
/usr/local/bin/Installomator.sh $1 LOGO=/usr/local/sbin/FileWave.app/Contents/Resources/fwGUI.app/Contents/Resources/kiosk.icns
```

Line Endings: ● Unix   ○ Windows   Cancel   OK

> Verification script for automatic updates

Build fileset to deploy & update app via Installomator

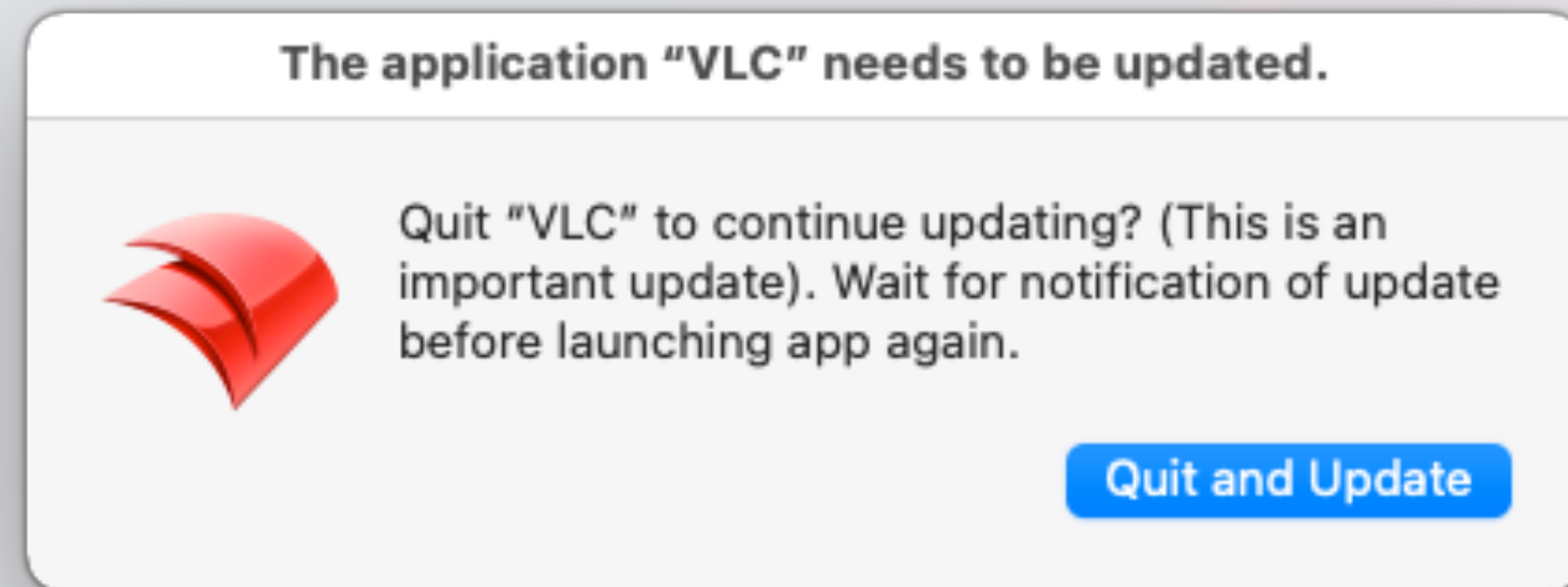Run Installomator.sh by itself to view all labels

Find label for app to be installed

Specify label for app to install/ update as launch argument in your activation and verification scripts

Duplicate your fileset, rename it, and change launch argument to label for next app

What it looks like from the client side

The application "VLC" needs to be updated.

Quit "VLC" to continue updating? (This is an important update). Wait for notification of update before launching app again.

Quit and Update

```
37
38    # behavior when blocking processes are found
39    BLOCKING_PROCESS_ACTION=tell_user
40    # options:
```

```
29
30    # notify behavior
31    NOTIFY=success
```

Fri Jul 29 12:02 AM

VLC update complete!
Installed VLC, version 3.0.17.3
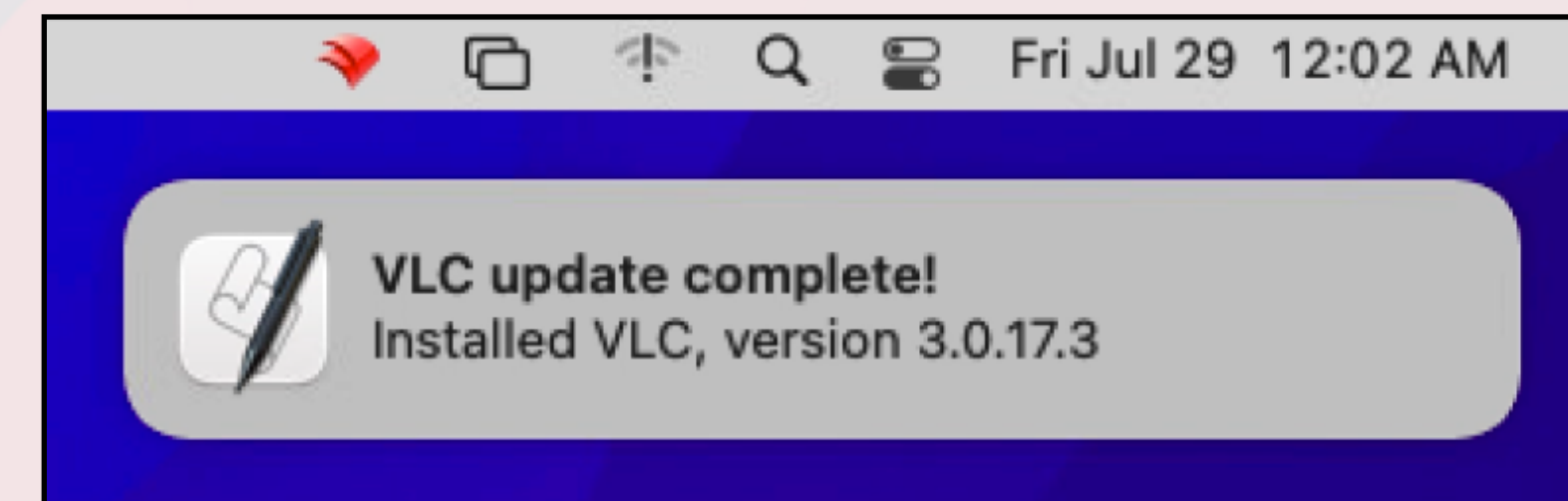
Be careful as some apps/labels may install even if the app is already on the latest version

Q&A

Next is Using winget to support 3rd party apps (Windows) with Tony

filewave