

Installomator - The one installer script to rule them all (macOS PKG)

Description

This article delves into the application of [Installomator](#), a shell script, in combination with FileWave for the maintenance of macOS applications. Installomator supports the downloading and installation of over 450 different applications, operating directly from the vendor's public URLs. This approach is compared with the use of AutoPkg, a tool that provides more stringent control over versioning, testing, and release management.

Installomator can be a valuable tool when deploying the latest publicly available version of an application is the goal. It verifies the authenticity and validity of each download, though it's not a fully automated solution. For certain applications like Chrome, Zoom, and Microsoft Office, their update processes may be better managed via a config profile. Additionally, attention may be needed for built-in auto update prompts of other apps.

A key characteristic of Installomator is its handling of versioning. Instead of installing only when the downloaded version is newer, Installomator initiates installation if the version numbers differ. This is due to the complexities in determining which version number is higher. For instance, if Installomator downloads version 4 after a vendor has released version 5 due to a URL change, it will attempt to install version 4 because the version numbers differ.

Ingredients

- FW Admin
- swiftDialog - [Deploying swiftDialog via FileWave \(macOS PKG\)](#)
- Installomator - [Releases · Installomator/Installomator \(github.com\)](#)

Directions

Deploying Installomator on all your Macs

Installomator operates by taking a 'label' parameter representing the application to be installed. It generally only installs an application if the latest version is newer than the installed version, though exceptions exist, emphasizing the importance of thorough testing. It's deployed via a single FileWave Fileset, with numerous configuration options to improve user experience and system stability. These include user notifications upon completion, the closing of the app prior to installation, prompts before closing the app, and the ability to re-launch the app post-update if it was initially open. Installomator does not interact with App Store applications except to replace them with non-App Store versions if desired. See [Configuration Options](#) further down for how to customize the operation of the script.

Installing Installomator

1. Download the PKG for [SwiftDialog](#) and install the PKG as a FileSet.
2. Download the PKG for Installomator from [Releases · Installomator/Installomator \(github.com\)](#) install the PKG as a FileSet. This will place the script file to `/usr/local/Installomator/Installomator.sh`



PKG Filesets are easy to make. You can start by dragging and dropping the PKG on the Filesets window in FileWave Central or use our example here: [PKG - Installomator.fileset.zip](#)

Identifying an App to install

Run Installomator by itself in Terminal and get the full list of apps it can install:

```
/usr/local/Installomator/Installomator.sh
```

This is the output of that command:

```
jlevitsk@JoshuaseWaveMBP ~ % /usr/local/Installomator/Installomator.sh
2023-07-22 12:45:33 : REQ : : no label provided, printing labels
1password7
1password8
1passwordcli
4kvideodownloader
8x8
abetterfinderrename11
abstract
acroniscyberprotectconnect
acroniscyberprotectconnectagent
adobeacrobatprodc
adobebrackets
adobeconnect
adobecreativeclouddesktop
adobereaderdc
adobereaderdc-install
adobereaderdc-update
affinitydesigner2
affinityphoto2
affinitypublisher2
aftermath
```

Or run Installomator in Terminal with grep to filter the results:

```
/usr/local/Installomator/Installomator.sh | grep -i bbedit
```

This is the output of that command:

```
jlevitsk@JoshuaseWaveMBP ~ % /usr/local/Installomator/Installomator.sh | grep -i bbedit
bbedit
bbeditpkg
jlevitsk@JoshuaseWaveMBP ~ %
```

Installing an App

At this point you have installed both [swiftDialog](#) and Installomator. For the next steps here is an example [BBDit - Installomator.fileset.zip](#) to look at while you follow along.

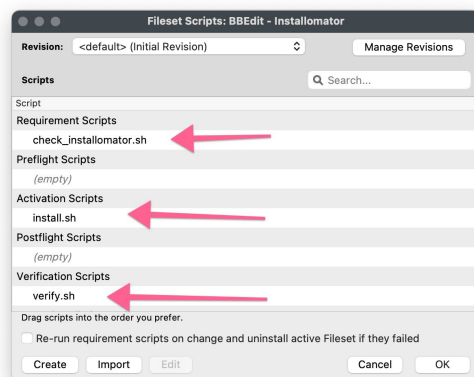
The BBDit example is a very simple Fileset containing 3 scripts, and that's all.

The scripts are fairly simple and listed below:

check_installomator.sh

```
#!/bin/zsh
# Checks every 2 minutes for Installomator to be
# present.
# Add the contents of your script below:

if [ -f "/usr/local/Installomator/Installomator.sh" ];
then
    exit 0
else
    echo "Could not find Installomator"
    exit 1
fi
```



install.sh

```
#!/bin/zsh
# Remember to change the properties for both install.sh
and verify.sh
# to have the right app label since it is passed to the
script from properties.

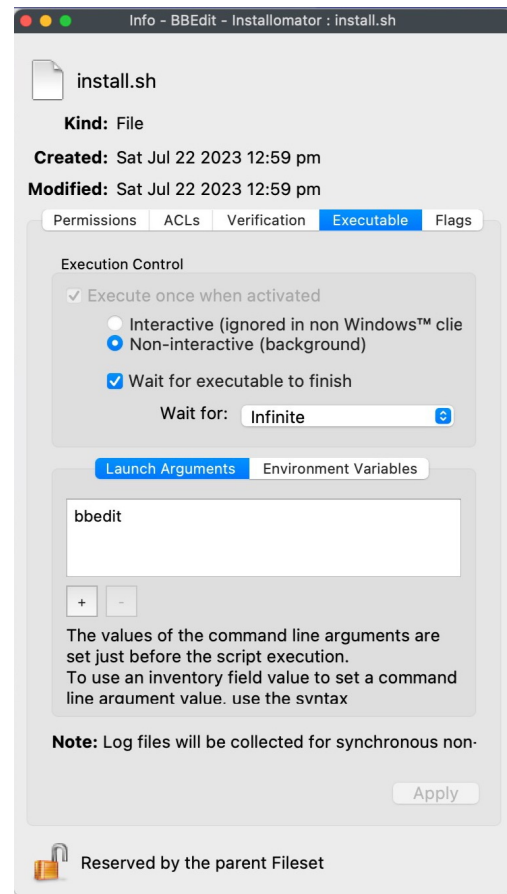
/usr/local/Installomator/Installomator.sh $1
NOTIFY_DIALOG=1 NOTIFY=success
LOGO=/usr/local/sbin/FileWave.app/Contents/Resources/fwG
UI.app/Contents/Resources/kiosk.icns
```

verify.sh

```
#!/bin/zsh
# Remember to change the properties for both install.sh
and verify.sh
# to have the right app label since it is passed to the
script from properties.

/usr/local/Installomator/Installomator.sh $1
NOTIFY_DIALOG=1 NOTIFY=success
LOGO=/usr/local/sbin/FileWave.app/Contents/Resources/fwG
UI.app/Contents/Resources/kiosk.icns
```

In our example both `install.sh` and `verify.sh` are identical but they don't have to be depending on your needs.



So now you may be wondering about where "bbedit" is since that label is the label we want to use? Highlight the BBEdit Fileset in your admin console and click the Scripts button in the toolbar. Now right click on `install.sh` and pick Properties. The dialog will look like the one here. You'll want to change `bbedit` to whatever app you want this to install. Do the same exact thing for `verify.sh`.

Configuration Options

In our example, we pass the LOGO parameter to the script. This action assigns a FileWave logo to the script, derived from the FileWave Kiosk icon file located on a device with FileWave installed. You can also pass additional parameters to further refine Installomator's behavior. This is covered in greater detail at the following link: [Configuration and Variables](#).

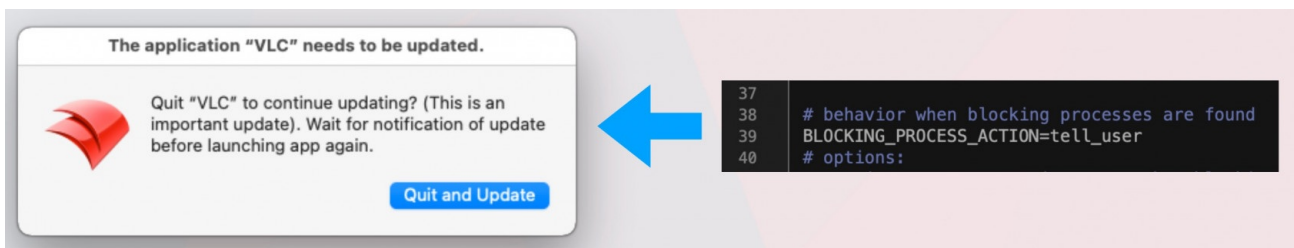
[Installomator/Installomator Wiki \(github.com\)](#) One setting you may want to change is NOTIFY=success to be NOTIFY=all if you are using Kiosk to allow someone to install apps.

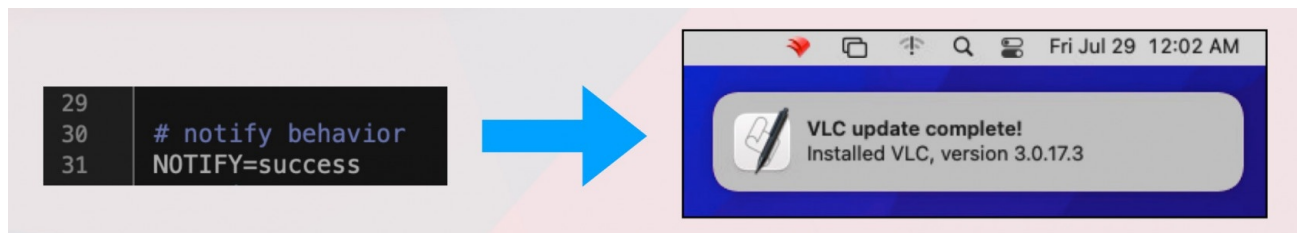
What it looks like

To install an app this way you simply [Associate](#) the [swiftDialog](#) Fileset, [Installomator](#) Fileset, and [BBEdit - Installomator.fileset.zip](#) Filesets and you should see it install.

⚠ Be careful as some apps/labels may install even if the app is already on the latest version

Depending on what dialogs you have enabled you may show status messages like the below. We recommend not editing the Installomator script itself but rather pass the options as parameters in your Filesets.





You should now be able to deploy anything that Installomator supports. Just remember that this tool is designed to always keep things up to date which can be a good or bad thing for you depending on your use case.

Troubleshooting

You will be able to see what the Fileset is doing by leveraging the [Fileset logs](#). All 3 of the scripts will log to `/var/log/fwcd/<fileset_id>` where `<fileset_id>` is the same as the ID you see in FileWave for the Fileset.

Name	Size	Version	Files	Modified	ID	Default Revision
Installomator Group					10243	
VSC - Installomator	842 B	1	3		10247	Initial Revision
BBEdit - Installomator	842 B	3	3		10246	Initial Revision
Installomator	340.5 kB	3	1		10245	10.4
PKG - swiftDialog-2.2.1-4591	6 MB	2	1		10244	Initial Revision

You may also be able to use FileWave Central to right click on the log and view it, but the log on the machine when testing is most simple and will exist 100% of the time.

The screenshot shows the FileWave Central interface for a client named 'Joshua's Apple Studio (2) - Client Info'. The client information includes: Last Connected: 7/23/23 12:47 PM, From: 173.44.70.23, Free Space: 200.3 GB, Platform: macOS (Intel), Model: 705, Version: Not connected, Enrollment Type: User approved enrollment, and Missing Updates: 0 missing | 0 critical. Below this, there are tabs for 'Filesets Status', 'Device Details', 'Command History', 'Installed Apps', 'Installed Profiles', 'Users', 'Policies', and 'Software Updates'. The 'Filesets Status' tab is selected, showing a table of filesets. The table has columns for Fileset, Fileset ID, Revision, Script, Status, Activation State, Date/Time, and Directory. The filesets listed are BBEEdit - Installomator, FileWave_macOS_Client_15.0.1_a8a2059a67, Installomator, Microsoft Defender macOS, PKG - swiftDialog-2.2.1-4591, and Profile - Microsoft Defender - Background Service. The 'BBEEdit - Installomator' fileset is highlighted, and a red arrow points to its 'check_install...' script. Another red arrow points to the 'View script output' button next to the 'verify.sh' script. A third red arrow points to the 'Reinstall Selected Filesets' button at the bottom of the table.

Related Links

- [Installomator GitHub](#) - Installomator official GitHub page
- [Deploying swiftDialog via FileWave \(macOS PKG\)](#)
- [AutoPkgr with FileWave](#) - Information on the AutoPkgr tool
- Training Day 2022 Presentation - [Using Installomator to support 3rd party apps - Mac.pdf](#)

Digging Deeper

The choice between Installomator and AutoPkgr ultimately hinges on your administrative preferences and needs. Installomator, with its ability to streamline the download and installation of different versions of applications, serves as a viable option when the goal is to deploy the latest publicly available versions. However, it requires careful management, especially considering its approach to version comparison and updates. On the other hand, AutoPkgr presents a viable alternative for those seeking a more tightly controlled release process, offering thorough testing, versioning, and release management.

Revision #34

★ Created 22 July 2023 12:37:12 by Josh Levitsky

✎ Updated 15 October 2024 17:02:18 by Josh Levitsky