

Integrations

The Integrations section provides valuable information and resources on integrating FileWave with other essential systems and tools. Discover a range of integration possibilities to enhance the functionality and streamline the management of your FileWave environment. Explore documentation, guides, and best practices for seamless integration with popular platforms and services. By integrating FileWave with your existing systems, you can achieve enhanced automation, data synchronization, and streamlined workflows, optimizing your device management processes and simplifying administration tasks. Unlock the power of integrations to maximize the efficiency and effectiveness of your FileWave deployment.

- [Agnosys](#)
 - [Integrating EasyLAPS with FileWave](#)
 - [Integrating MacOnboardingMate \(MOM\) with FileWave](#)
 - [Integrating Telepod with FileWave](#)
- [AutoPkg with FileWave](#)
 - [AutoPkg - FileWave Integrated \(v15.5+\)](#)
 - [Using AutoPkg with FileWave for Advanced macOS Software Deployment](#)
 - [Autopkg\(r\) FAIL_RECIPES_WITHOUT_TRUST_INFO](#)
- [Cloudflare WARP integration with FileWave \(macOS/iOS/Windows/Android\)](#)
- [DeepFreeze \(macOS/Windows\)](#)
- [Hello-IT integration with FileWave \(macOS\)](#)
- [Invgate integration with FileWave](#)
- [ServiceNow integration with FileWave](#)
- [Slack integration with FileWave](#)
- [Truce Family integration with FileWave](#)
- [OTRS integration with FileWave](#)

Agnosys

Agnosys was founded on April 7, 1999. Agnosys is a Qualiopi Certified Authorized Training Center, a member of the Apple Consultants Network (ACN). Agnosys' core business is training on Apple technologies. They offer a range of trainings in support, integration, deployment and maintenance of Apple products, some of which are preparatory to obtaining official Apple certifications.

Integrating EasyLAPS with FileWave

What

This article focuses on [EasyLAPS](#), a tool developed to routinely rotate the local administrator account password of a Mac and store it in a Mobile Device Management (MDM) solution, including FileWave. EasyLAPS main function is to maintain unique passwords across a Mac fleet, centralized in the MDM console.

When/Why

EasyLAPS is beneficial when the need arises to manage and rotate local administrator passwords across a number of Mac devices, ensuring unique passwords are utilized and safely stored within the MDM. This tool is particularly useful for enhancing the security of your network by preventing unauthorized access and reducing the risk of password-related security breaches.

How

EasyLAPS operates in two different functioning logics, both supported by FileWave:

Logic #1: In this mode, the password is stored in encrypted form both in the MDM and in the EasyLAPS Keychain. EasyLAPS manages the password rotation using the locally stored password, with the new generated password then stored in the MDM. The public key used for encryption is part of the EasyLAPS configuration file, while the private key is not present on the device and must be kept securely. This mode is most suitable when a large number of technicians have access to the MDM console, and only those possessing a copy of the EasyLAPS-Toolkit with the private key can access the rotated password.

Logic #2: Here, the password is stored in clear text in the MDM and not stored locally unless a password reversion fails. EasyLAPS uses the MDM-stored password to manage the rotation to the new generated one, which is then stored in the MDM. This logic is best when a limited number of technicians have access to the MDM console and can access a rotated password.

After the first successful rotation, the new password is visible in the device inventory record.

EasyLAPS operates a true rotation of the local administrator password, so the account keeps its cryptographic status. That means that once the password is changed, the account is still a Crypto user and Volume owner, able to unlock the device, install macOS updates, make changes to the startup security policy, initiate an Erase All Content and Settings, and more.

Complete documentation on how to use EasyLAPS with FileWave is provided upon purchase, offering detailed instructions and support. Please note that EasyLAPS supports a variety of MDM solutions with FileWave included.

The screenshot displays the FileWave console interface for a device named "MacBook Pro de ladmin". The top section shows key information: Last Connected (06/07/2023 23:37), From (176.187.169.40), Free Space (213 GB), Platform (macOS (Intel)), Model (61), Version (Not connected), Enrollment Type (User approved enrollment), and Missing Updates (1 missing, 0 critical). Below this is a navigation bar with tabs: Filessets Status, Device Details (selected), Command History, Installed Apps, Installed Profiles, Users, Policies, and Software Updates. A search bar labeled "Filter Device Details" is also present. The main content area shows a table of device properties and values.

Property	Value	Last Update Time
Device Name	MacBook Pro de ladmin	
Device Product Name	MacBookPro15,2	
EACS Preflight	success	
EasyLAPS	!@ground-6observing1-last rotation: 2023-07-06 23:27:08	07/07/2023 01:27
Enroll Date	06/07/2023 21:12	
Enrolled via DEP	false	
Enrollment Approved By User	true	
Enrollment State	Enrolled	
Enrollment Type	User approved enrollment	
Enrollment Username		

Related Links

- [EasyLAPS Official Website](#) - Comprehensive information about EasyLAPS, its features, and support.
- [EasyLAPS - Management solutions support - EN - Agnosys](#)
- [Integrating Telepod with FileWave](#)
- [Integrating MacOnboardingMate \(MOM\) with FileWave](#)

Integrating MacOnboardingMate (MOM) with FileWave

What

This article is about MacOnboardingMate (MOM), a tool that streamlines the onboarding and migration of Mac devices across different Mobile Device Management (MDM) solutions, including FileWave.

When/Why

MOM is utilized when a Mac needs to be onboarded to an MDM solution or migrated from one MDM to another. This is relevant when a new device is added to your network, a device is being transferred to a different MDM platform, or when you are initiating an MDM switch project. MOM retains the Automated Device Enrollment configuration during migration, a key feature for most organizations.

How

MOM operates in two different execution modes: Launcher and AutoLauncher.

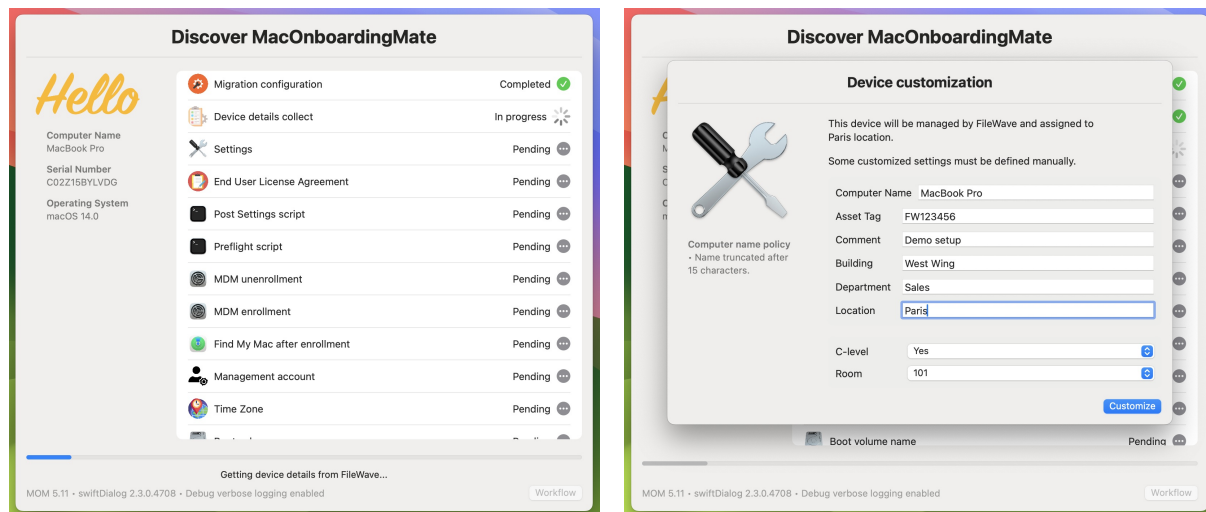
Launcher is used when MOM is manually run outside of an MDM, while AutoLauncher is used when MOM is operated from within an MDM, either automatically or manually through a Self Service.

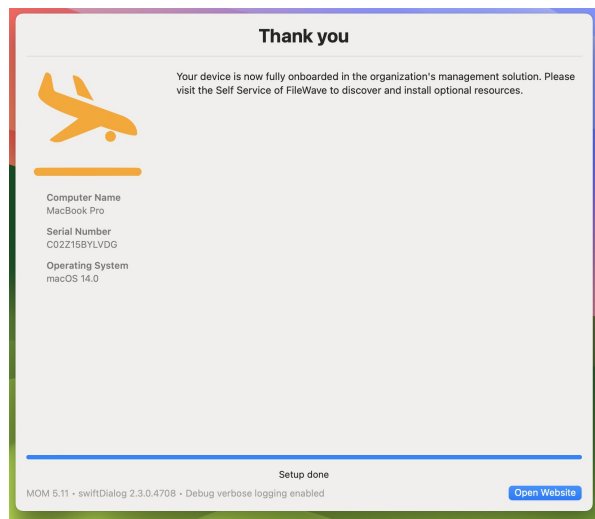
MOM will facilitate the onboarding or migration process, managing both the unenrollment from the previous MDM and the enrollment into FileWave. For onboarding, MOM is run from within the FileWave MDM and started during the Setup Assistant. For migration to FileWave, MOM is executed from the MDM that the device leaves.

MOM's latest version includes a new user interface based on swiftDialog. The previous interface, based on DEPNotify, is still available to ensure a smooth transition for existing users. MOM White glove provisioning combined with macOS Automated Device Enrollment offers a similar experience as Windows Autopilot for pre-provisioned deployment.

The tool supports multiple languages, currently available in English and French, and can be localized to other languages as required. An important advantage of using MOM is its "turnkey" nature, requiring no scripting knowledge for implementation or upgrade. If necessary, it can be augmented with scripts at key steps of the workflows, offering flexibility and customizability.

Complete documentation on how to use MOM with FileWave is provided upon purchase, offering detailed instructions and support. Please note that MOM supports a variety of MDM solutions with FileWave included.





Demos

Demo : Migration of a Mac between two MDM (DEPNotify)



Demo : Onboarding of a Mac enrolled during the Setup Assistant (DEPNotify)



Demo : Onboarding of a Mac enrolled from an opened user's session (DEPNotify)



Related Links

- [MacOnboardingMate Official Website](#) - Comprehensive information about MOM, its features, and support.
- [MacOnboardingMate - Management solutions support - EN - Agnosys](#)
- [Integrating EasyLAPS with FileWave](#)
- [Integrating Telepod with FileWave](#)

Integrating Telepod with FileWave

What

This article discusses [Telepod](#), an automaton created to streamline the lifecycle of an iOS device. Telepod enables the backup and restoration of new iOS devices without iCloud, remotely monitored by IT support. It can work as part of a Mobile Device Management (MDM) solution, like FileWave, to retain the Automated Device Enrollment configuration during MDM migrations.

When/Why

Telepod is essential when setting up new iOS devices, replacing existing devices, or migrating devices between MDM solutions. It offers a streamlined and efficient method to manage iOS device lifecycles, which is particularly useful in large-scale environments where multiple devices need to be managed simultaneously.

How

Telepod operates through highly customizable workflows, launched from an assistant available in the Self Service of a Mac enrolled in an MDM solution. Currently, there are four types of workflows:

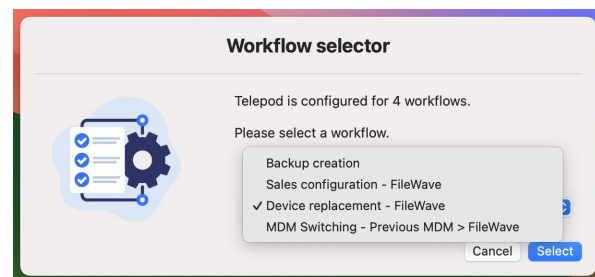
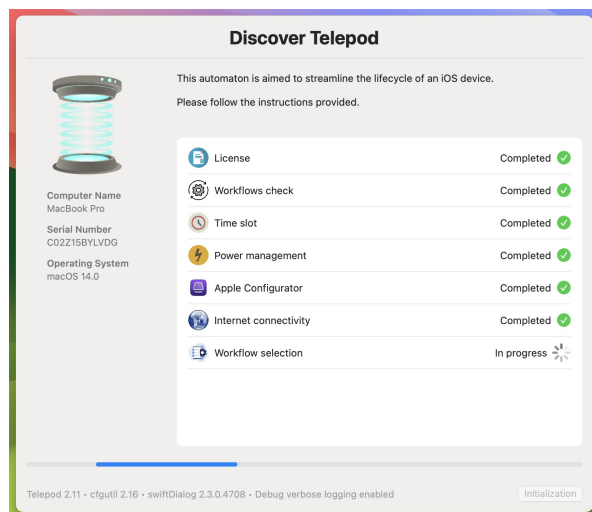
1. **Migration (MDM switching):** This migrates a device from one MDM to another. No data transfer occurs, and devices are enrolled in the new MDM using Device Enrollment.
2. **Replacement:** This replaces a current device with a new one. It supports two main use cases: device switching under the current MDM and MDM switching to a new MDM.
3. **Setup:** This sets up a new device from the backup of another device acting as a model.
4. **Backup:** This creates a backup of a device acting as a model, allowing other new devices to be set up.

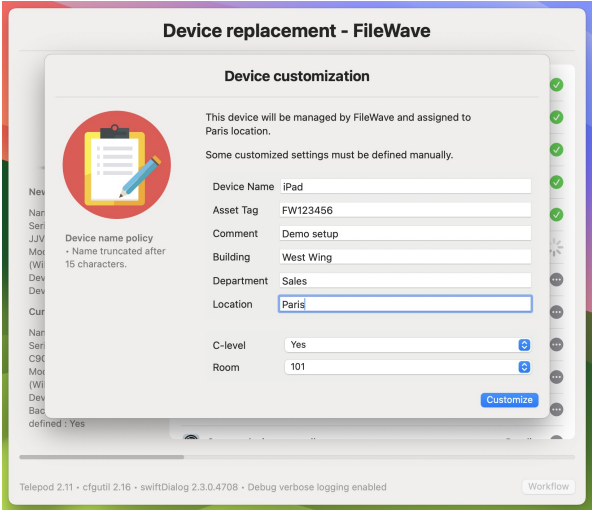
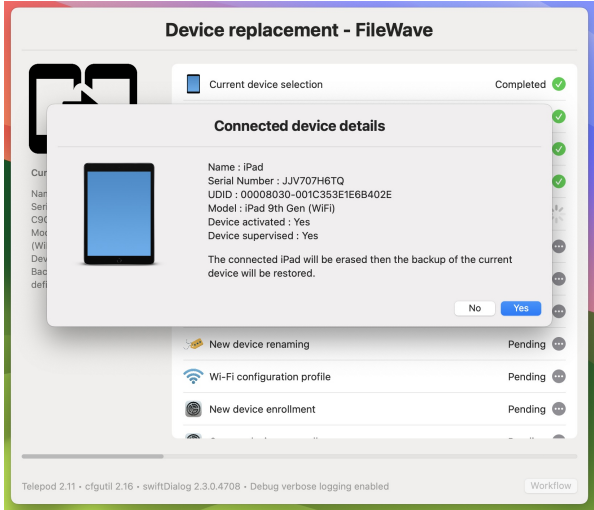
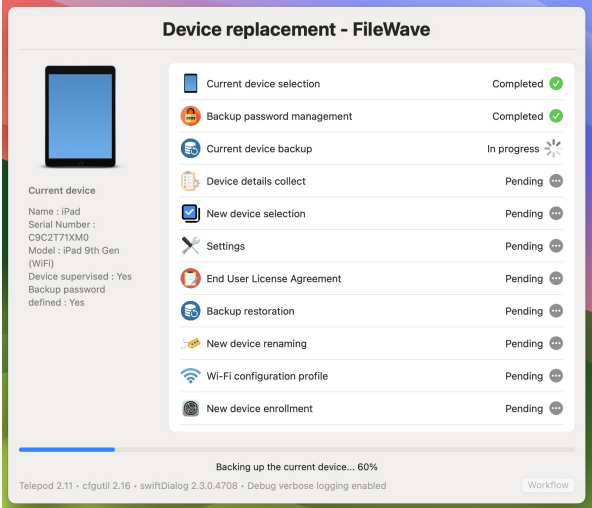
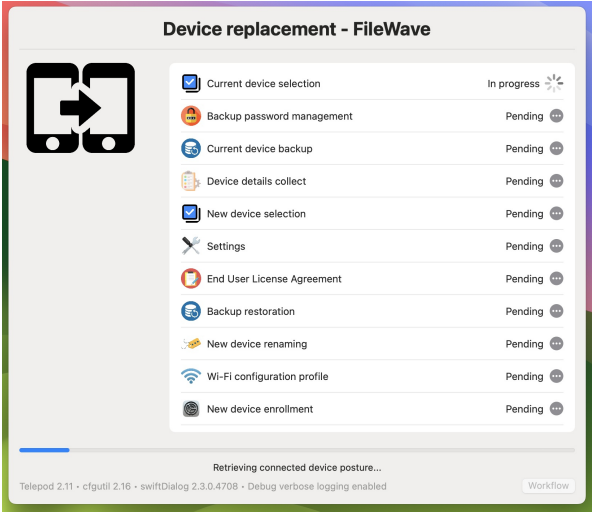
Data is transferred locally over a wired connection for Backup, Replacement, and Setup workflows, bypassing the need for iCloud. Backups can be stored centrally in a distribution point, making them available worldwide.

Telepod's latest version includes a new user interface based on swiftDialog. The previous interface, based on DEPNotify, is still available to ensure a smooth transition for existing users.

The tool supports multiple languages, currently available in English and French, and can be localized to other languages as required. Telepod is a "turnkey" software, meaning no scripting knowledge is required for implementation or upgrade.

Complete documentation on how to use Telepod with FileWave is provided upon purchase, offering detailed instructions and support. Please note that Telepod supports a variety of MDM solutions with FileWave included.





Device replacement - FileWave

New device

Name : iPad

Serial Number : JJV707H6TQ

Model : iPad 9th Gen (WiFi)

Device activated : Yes

Device supervised : Yes

Current device

Name : iPad

Serial Number : C9C2T71XM0

Model : iPad 9th Gen (WiFi)

Device supervised : Yes

Backup password defined : Yes

Current device selection

Completed

Backup password management

Completed

Current device backup

Completed

Device details collect

Completed

New device selection

Completed

Settings

Completed

End User License Agreement

Completed

Backup restoration

In progress

New device renaming

Pending

Wi-Fi configuration profile

Pending

New device enrollment

Pending

Restoring the backup on the new device... 81%

Telepod 2.11 • cfgutil 2.16 • swiftDialog 2.3.0.4708 • Debug verbose logging enabled

Workflow

Device replacement - FileWave

New device

Name : iPad

Serial Number : JJV707H6TQ

Model : iPad 9th Gen (WiFi)

Device activated : Yes

Device supervised : Yes

Current device

Name : iPad

Serial Number : C9C2T71XM0

Model : iPad 9th Gen (WiFi)

Device supervised : Yes

Backup password defined : Yes

Backup restoration

Completed

New device renaming

Completed

Wi-Fi configuration profile

Completed

New device enrollment

Completed

Current device unenrollment

Completed

Current device backup deletion

Completed

Configuration profiles

Completed

Documents

Completed

Wallpaper

Completed

Device details update

Completed

Device inventory

Completed

Telepod 2.11 • cfgutil 2.16 • swiftDialog 2.3.0.4708 • Debug verbose logging enabled

Workflow

Thank you

Your new device is now fully onboarded in FileWave and can be safely disconnected.

Workflow done

Telepod 2.11 • cfgutil 2.16 • swiftDialog 2.3.0.4708 • Debug verbose logging enabled

Workflow

2023-07-26 15:10:54 CEST C02Z15BYLVDG MacBook Pro - Status message : Workflow Device replacement - FileWave started

2023-07-26 15:12:26 CEST C02Z15BYLVDG MacBook Pro - Status message : iPad C9C2T71XM0 confirmed as the current device

2023-07-26 15:17:25 CEST C02Z15BYLVDG MacBook Pro - Status message : iPad JJV707H6TQ confirmed as the new device

2023-07-26 15:19:04 CEST C02Z15BYLVDG MacBook Pro - Status message : EULA agreed by ladmin (ladmin)

2023-07-26 15:26:07 CEST C02Z15BYLVDG MacBook Pro - Status message : Remote Unenroll command requested to FileWave for iPad C9C2T71XM0

2023-07-26 15:28:33 CEST C02Z15BYLVDG MacBook Pro - Status message : Replacement of iPad C9C2T71XM0 by iPad JJV707H6TQ completed

2023-07-26 15:28:36 CEST C02Z15BYLVDG MacBook Pro - Status message : Workflow Device replacement - FileWave done

Demo

Demo : Replacement / Device switching (DEPNotify)



Related Links

- [Telepod Official Website](#) - Comprehensive information about Telepod, its features, and support.
- [Management solutions support \(agnosys.com\)](#)
- [Telepod - Capacities - EN - Agnosys](#)
- [Integrating MacOnboardingMate \(MOM\) with FileWave](#)
- [Integrating EasyLAPS with FileWave](#)

AutoPkg with FileWave

The AutoPkg chapter of our Knowledge Base provides comprehensive guidance on using AutoPkg and AutoPkgr with FileWave to streamline software deployment. It includes articles on the new integrated AutoPkg feature in FileWave 15.5 for easy package creation, as well as detailed instructions on leveraging the full AutoPkg and AutoPkgr tools for advanced package management and customization.

AutoPkg - FileWave Integrated (v15.5+)

What

In FileWave version 15.5.0, we have introduced direct integration with AutoPkg, significantly streamlining the process of creating and deploying software packages. Administrators can now create new Filesets by selecting AutoPkg, allowing them to search for a software package and generate the Fileset with a single click. This integration simplifies the deployment workflow by eliminating the need for external tools or complex configurations.

Within the Fileset properties, administrators have the option to select whether the Fileset should be uninstalled when the association between the Fileset and any clients is removed. Version management is also made more accessible. By managing revisions in the Fileset properties, you can choose to deploy the latest version of the software or select an older version as needed. For example, if you need to deploy a specific version of Google Chrome, such as 108.105.10.2, for a certain period, you can select that version and later update to a newer release when appropriate.

Unlike the traditional use of AutoPkg, which typically involves installing it on a macOS system and customizing recipes with various repositories, FileWave's implementation focuses on simplicity and security. We have limited the integration to a curated set of known repositories to prevent the addition of potentially rogue repos. While this means that some software may not be available, we are open to considering requests for additional repositories to be added.

This streamlined approach is designed to make it easier for FileWave administrators who want a hassle-free method to find and deploy installers without dealing with complex configurations. For power users who require more advanced functionality, [traditional use of AutoPkg remains available](#). They can continue to build custom recipes and add their PKGs to FileWave as PKG Filesets, just as before.

When/Why

When to Use

This feature is ideal when you need a quick and straightforward way to deploy common software packages to your devices. If you're looking to reduce the time and complexity involved in creating Filesets, the AutoPkg integration provides a user-friendly solution. It is particularly useful for administrators who prefer not to delve into the intricacies of AutoPkg configurations but still want the benefits of automated package management.

Why This Feature Matters

The integration of AutoPkg directly into FileWave 15.5 enhances efficiency and simplifies the software deployment process. By offering a curated set of repositories and an intuitive interface, administrators can save time by quickly creating and deploying software packages without the need for external tools or manual configurations. This approach maintains control over software versions, allowing you to ensure that specific versions are deployed when necessary and updated on your schedule.

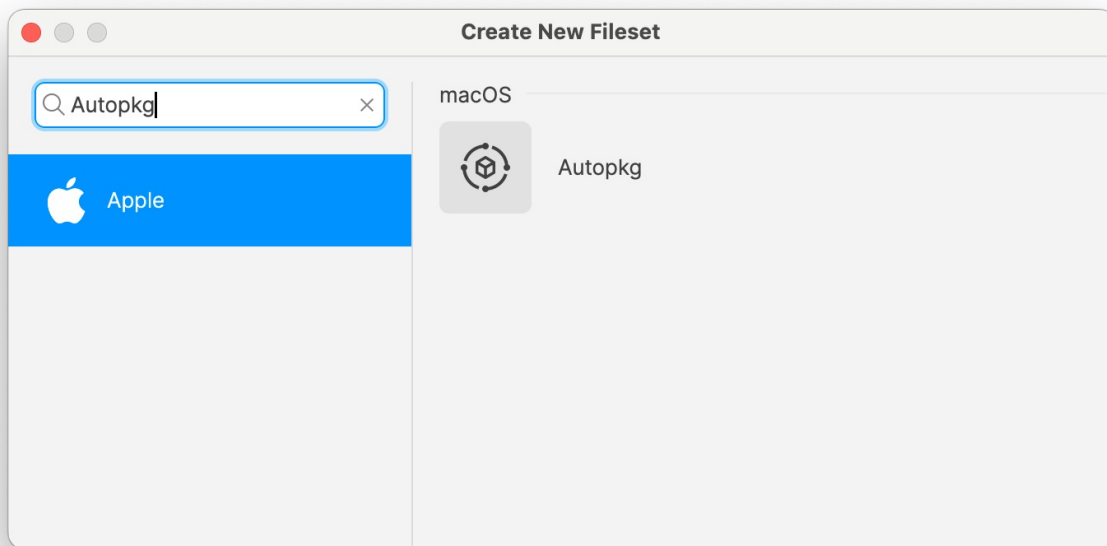
Security is also enhanced by using known repositories, preventing the introduction of unverified or malicious software into your environment. The reduced complexity lowers the learning curve associated with AutoPkg, making package deployment accessible to administrators with varying levels of experience. By focusing on ease of use, this feature empowers administrators to manage software deployments more effectively, without sacrificing control or security.

How

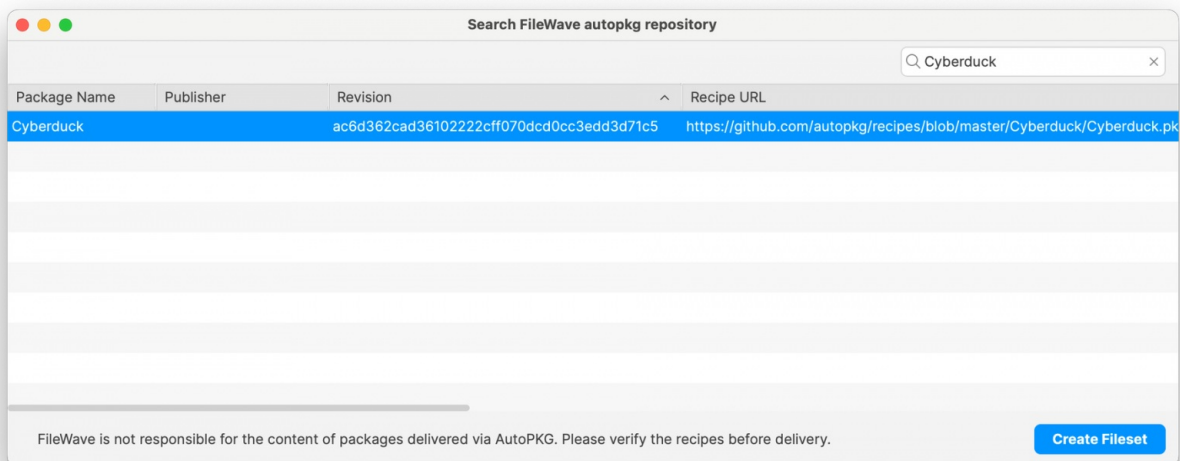
Creating an AutoPkg Fileset:

With the AutoPkg integration, creating a new Fileset is as simple as selecting the AutoPkg option when adding a new Fileset. You can then search for the desired software package and create it with a single click. The Fileset properties allow you to configure uninstallation behavior and manage software versions according to your deployment needs.

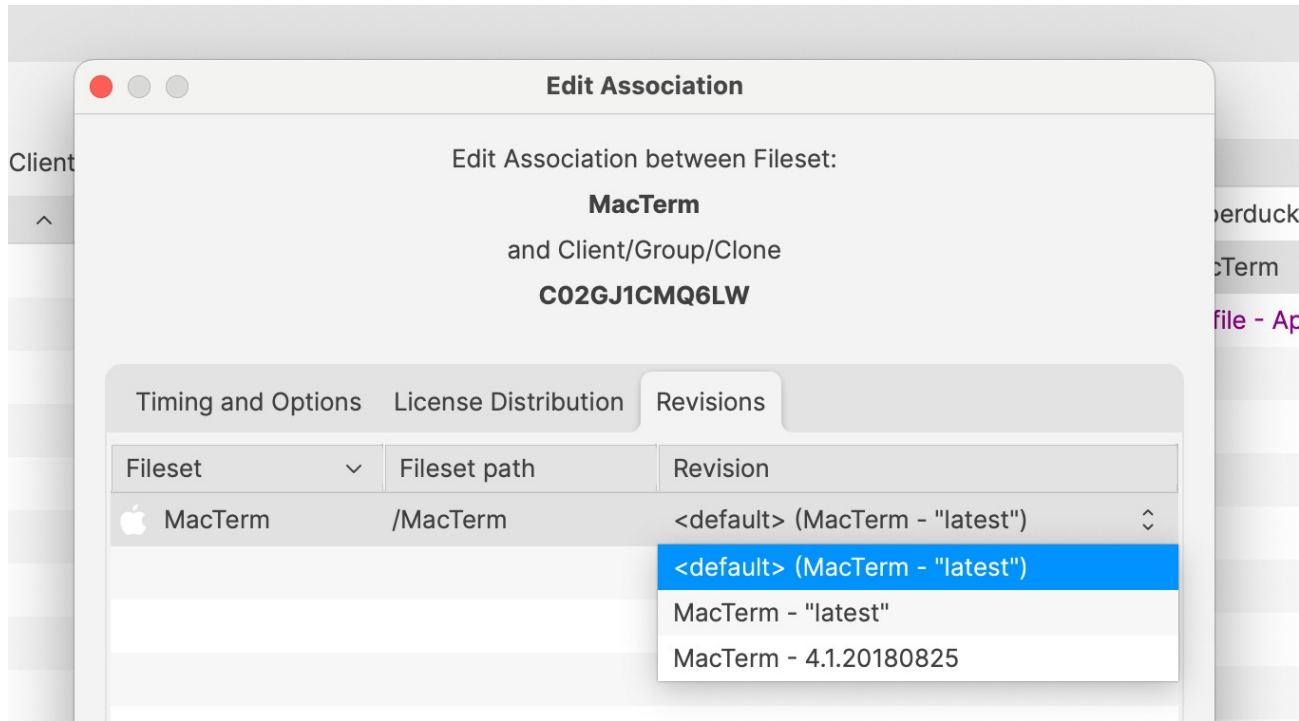
- Create a New Fileset, and select Apple so that you can pick Autopkg as the type:



- Search for software. Note that the feed of repos does not contain descriptions. We will be working on labeling things, but note that this process is manual and there are many items. The Recipe URL can be very helpful to understand where it comes from, in order to assess what it is and how to locate more detail.

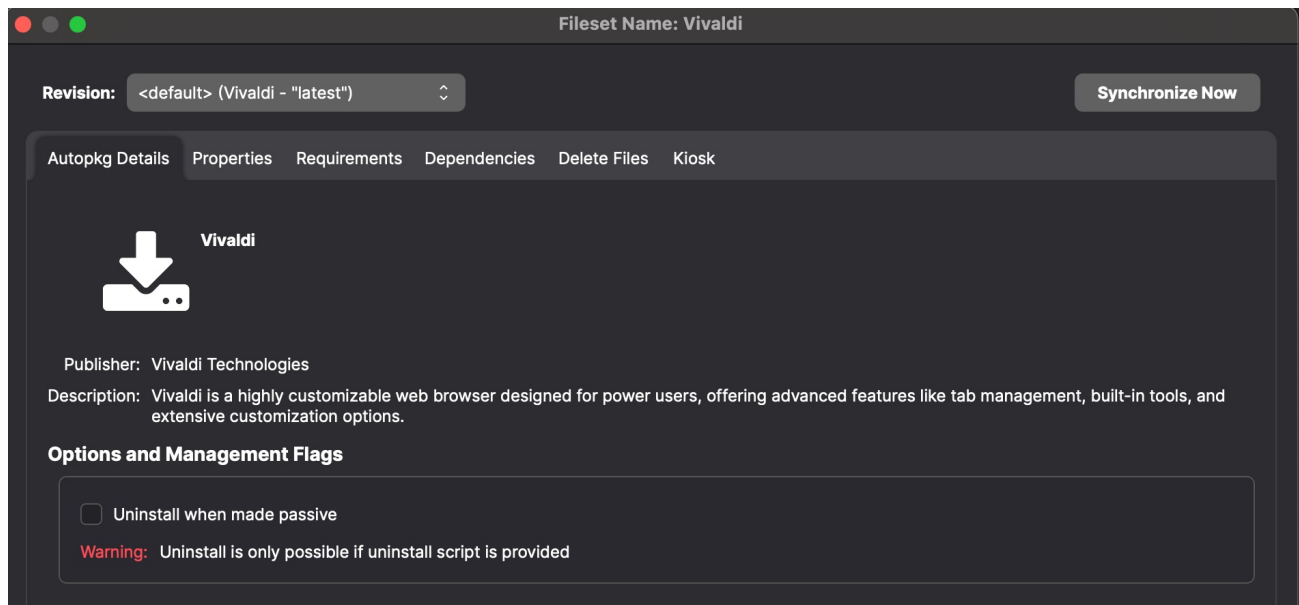


- Create an Association or Deployment. Selecting "latest" ensures upgrades can occur automatically. Otherwise, specify a specific version to stay at that version until later changed.



- Selecting Properties for the Fileset will allow you to select if "Uninstall when made passive" is enabled which will cause an uninstallation to occur when the Association or Deployment is removed.

Please note that only software that includes an uninstall script will act on this option. This is up to the person who created the recipe so it is recommended to test that removal can occur for any software where this is important to you.



Related Content

- [AutoPkg with FileWave](#)

Digging Deeper

Your FileWave Server will make requests to `le7bqzikha.execute-api.us-east-1.amazonaws.com` in order to create Filesets with this integration. It is important that if you run your own FileWave Server that it be able to reach that host if you want to take advantage of the integrated AutoPkg functionality.

Using AutoPkg with FileWave for Advanced macOS Software Deployment

Description

AutoPkg is an automation framework for macOS software packaging and distribution, oriented toward the tasks one would normally perform manually to prepare third-party software for mass deployment to managed clients. An important use in conjunction with FileWave is to provide a way to turn 3rd party software updates into Filesets on an automated basis.

AutoPkg is an automation framework for macOS software packaging and distribution, designed to automate the tasks one would normally perform manually to prepare third-party software for mass deployment to managed clients. While FileWave version 15.5 and later introduces an integrated AutoPkg feature for simplified package creation (as detailed in our new article "[Integrated AutoPkg \(v15.5+\)](#)"), power users seeking advanced functionality may prefer using the full AutoPkg and AutoPkg tools. This article focuses on leveraging AutoPkg with FileWave to automate the process of turning third-party software updates into Filesets on an automated basis, providing greater control and customization options for sophisticated deployment scenarios.

Ingredients

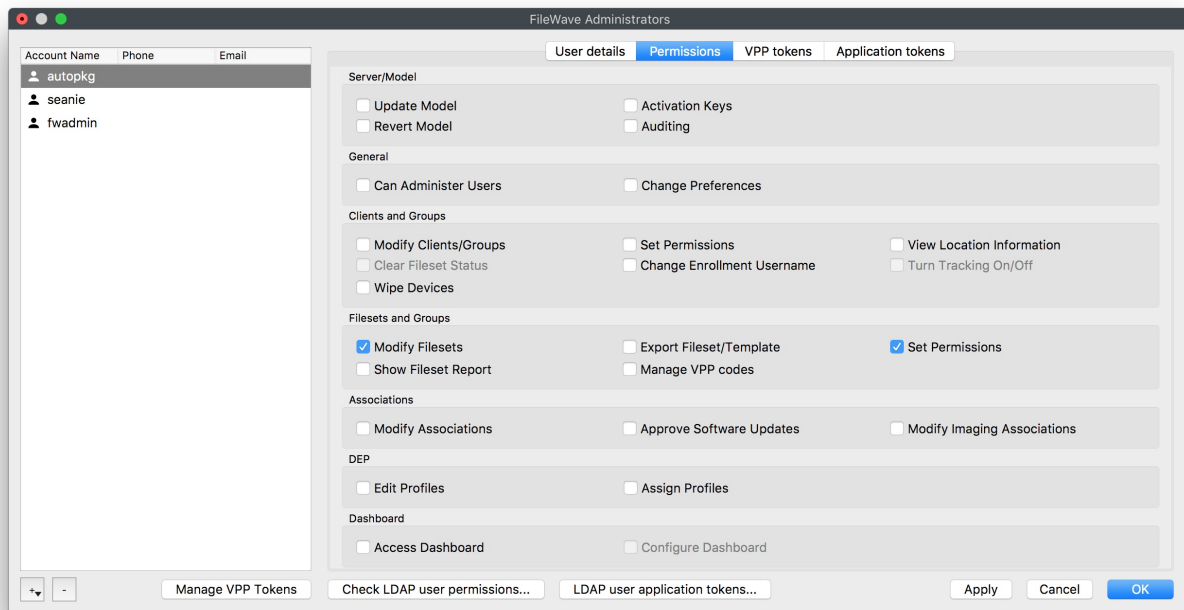
- FW Admin
- AutoPkg Installer
- An 'always on' computer

Directions

Complementing the below setup, there is also a Foundry presentation about this and, additional configuration and typical stumbling blocks: [FileWave and AutoPkg](#)

Setup FileWave

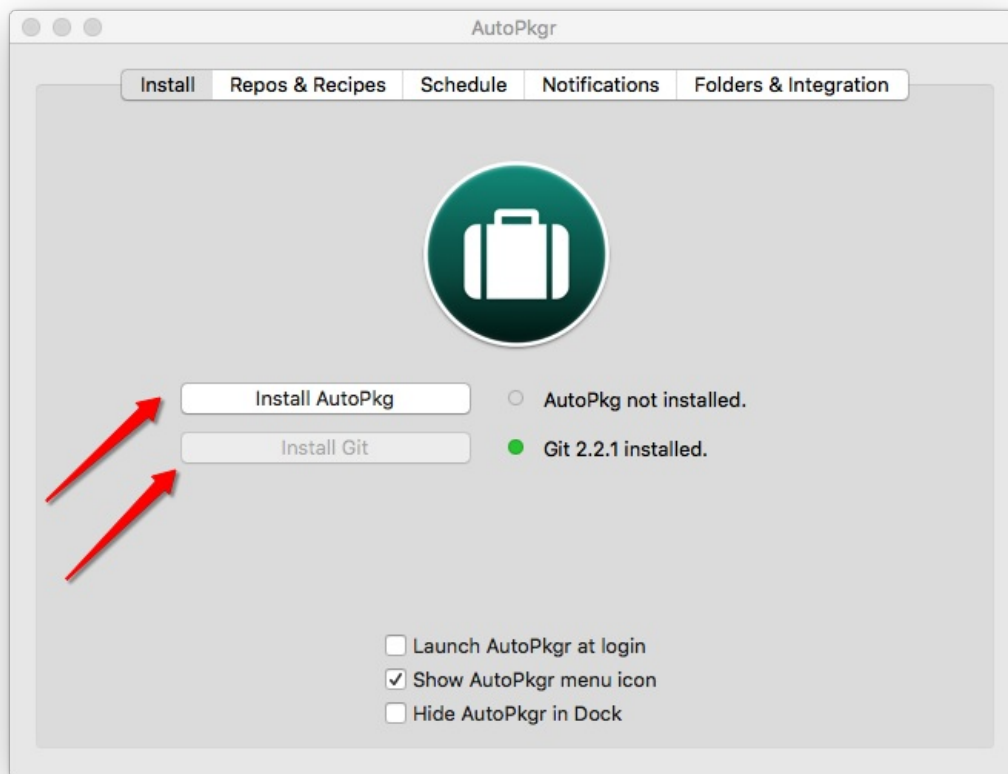
- Go to FileWave Admin -> Assistants -> Manage Administrators



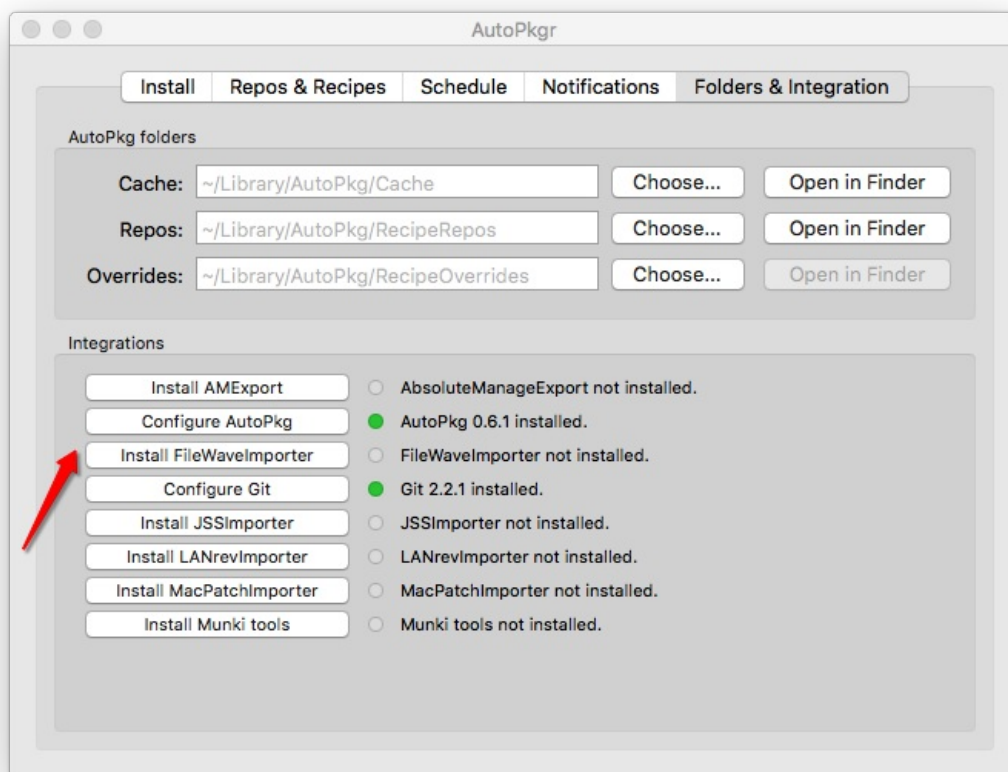
- Click on the + button to add a new Administrator
- Select Local Account, (for example autopkg and the password autopkg as well)
- Go to Permissions tab and click on Select None
- Allow the autopkg user to 'Modify Filesets' and 'Set Permissions' as per the above screenshot.
- Click Apply
- Confirm with OK

Setup AutoPkgr

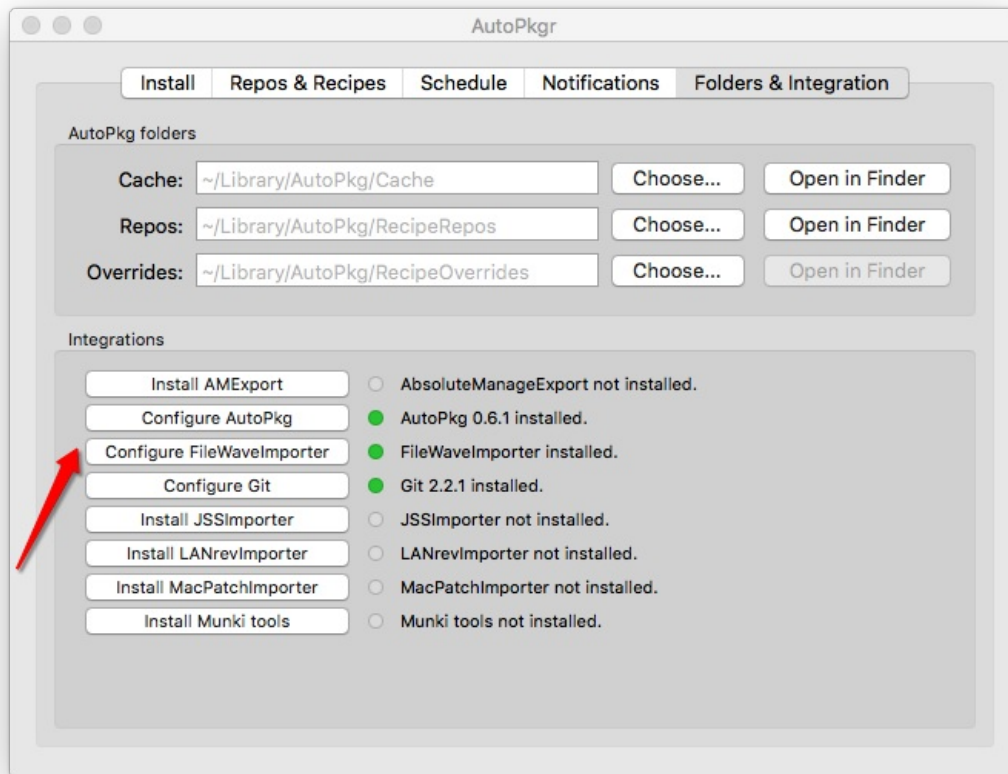
- Go to <https://github.com/lindegroup/autopkgr/releases/latest>
- Download, install and launch AutoPkgr
- Launch AutoPkgr, Click on "Install AutoPkgr", and "Install Git"



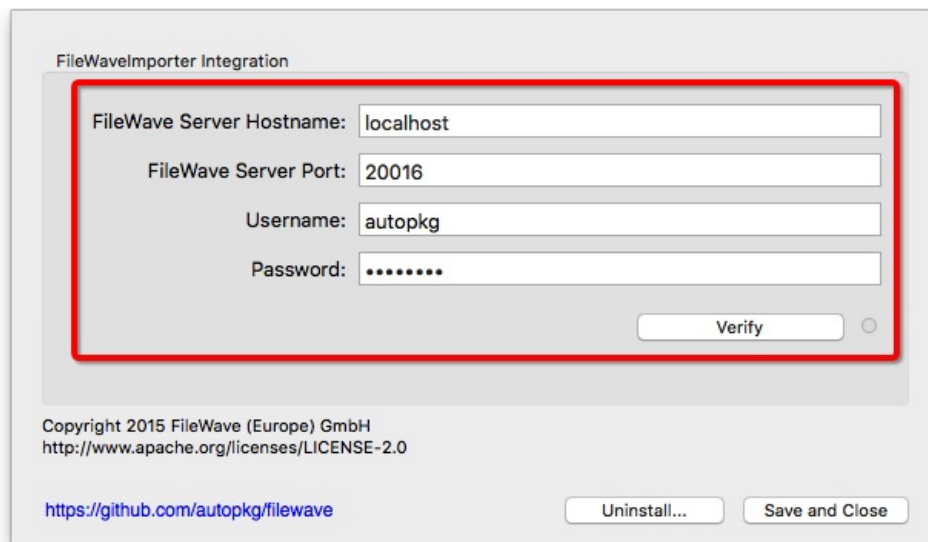
- Go to Folders & Integration and click on Install FileWaveImporter:



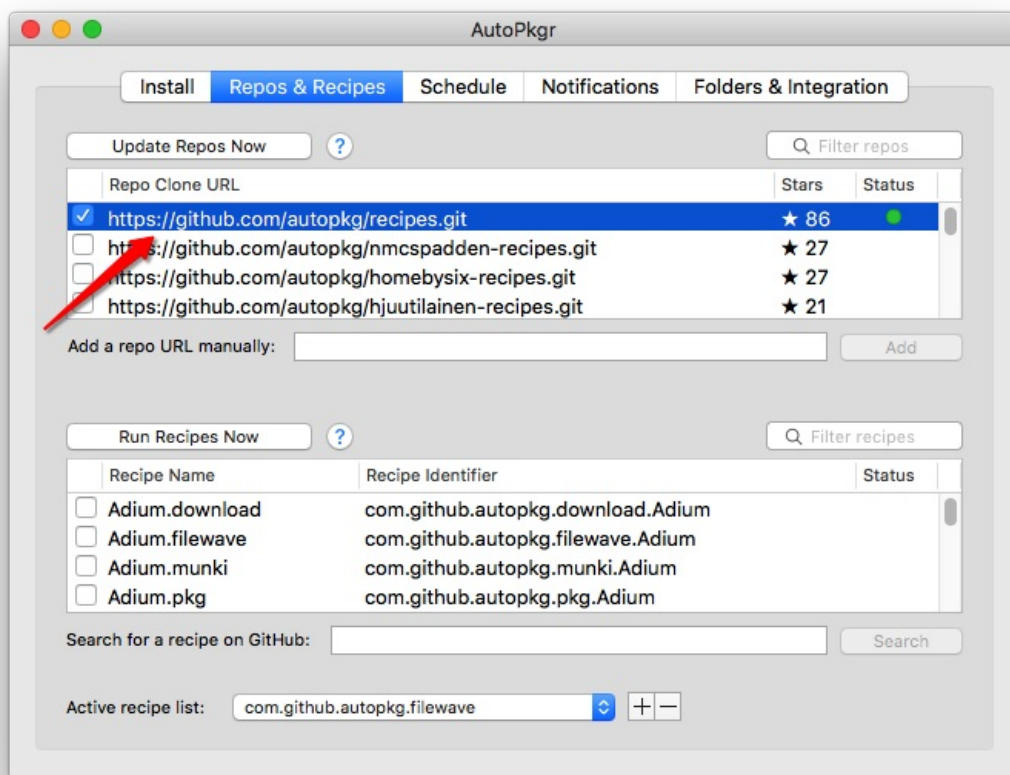
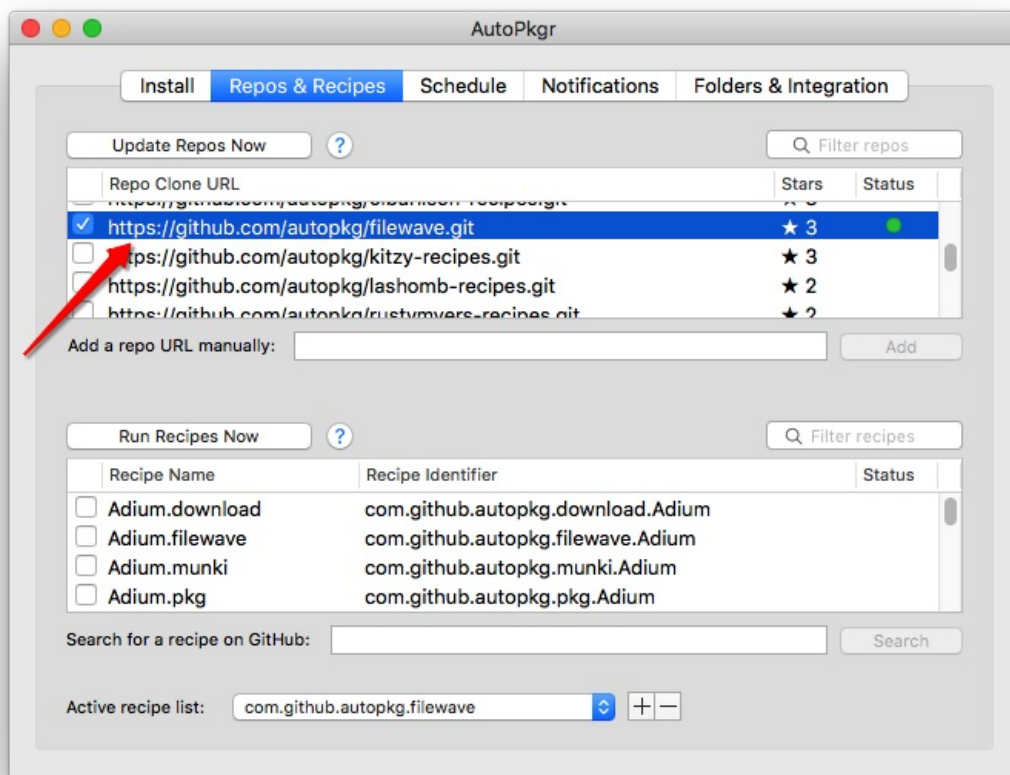
- Click on Configure FileWaveImporter:



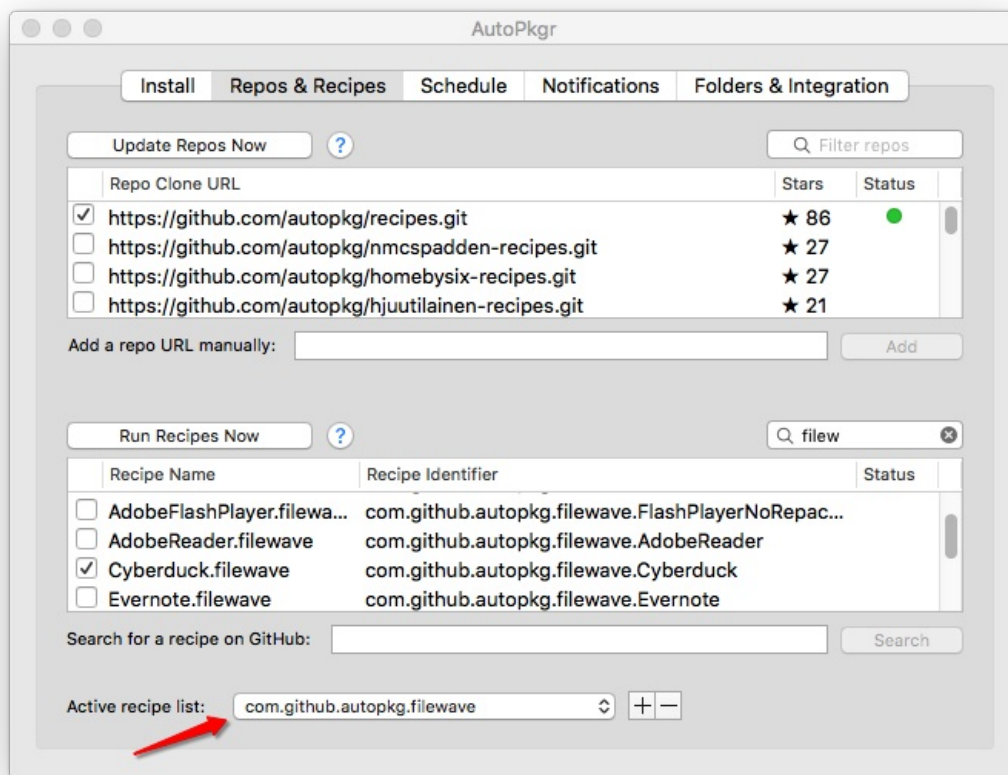
- Enter your FileWave Server Hostname



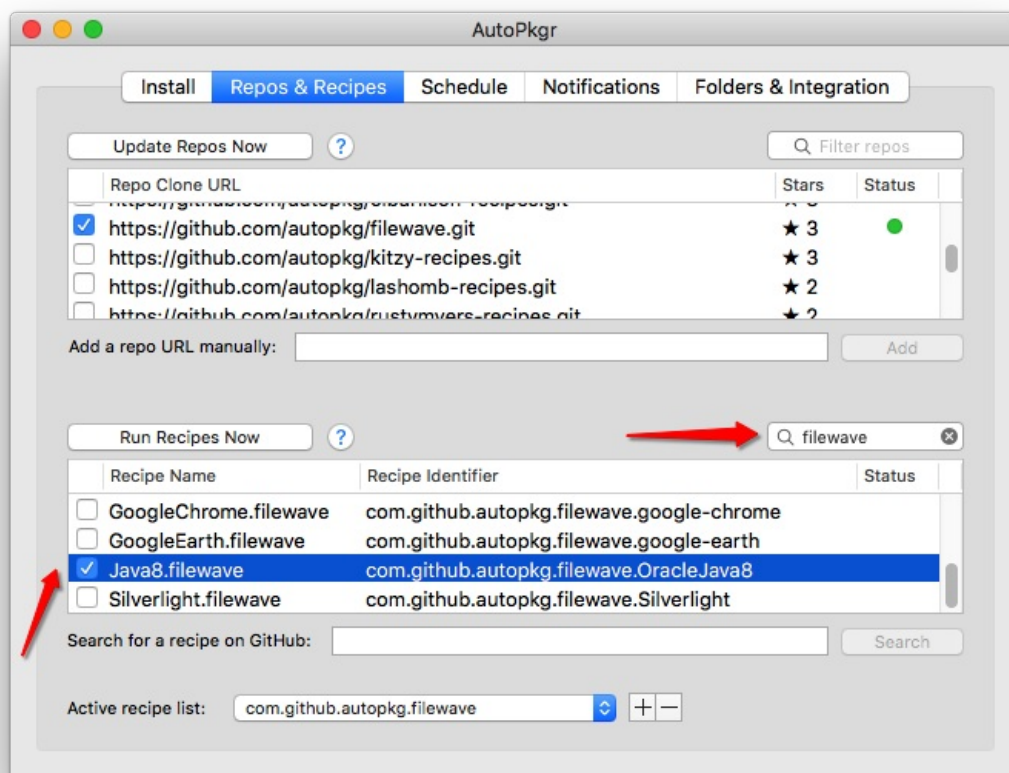
- FileWave Server Port is already set to 20016
- Username is e.g. autopkg
- Password is e.g. autopkg
- Click on Verify to validate the setup
- Click on Save and Close
- Go to Repos & Recipes and verify that <https://github.com/autopkg/recipes.git> and <https://github.com/autopkg/filewave/git> are checked



- Make sure that 'Active recipe list' has added com.github.autopkg.filewave:



- Now You can run a Recipe for example Java8. To find it quickly enter filewave on search bar and check the Java8 recipe:

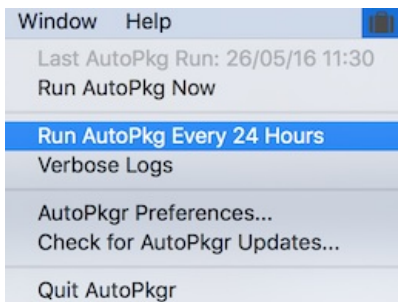


Security and Trust Relationship

- Running recipes directly from a cloned repo will bypass AutoPkg's security mechanism. As such an Override recipe should always be created and run. This builds a local recipe with a trust relationship between this and any linked 'parent' recipes, see

below. For additional information on Override recipes and more, please view the Foundry video: [FileWave and AutoPkg](#)

- Congratulations ! Your AutoPkg setup is now complete. Choose the recipes you would like to run on a regular basis , and then schedule AutoPkg to run every 24 hours.



Recipe Updates

On occasion recipes that were working will fail to run. Typically this is because something has changed regarding the 3rd party's website or download. This will require the author of the relevant recipe to update their recipe to implement this change. In this instance of failed recipes, check for recipe updates 'Update Recipes Now'. For any updated recipe, changes should be observed and then the trust relationship will need to be updated too; see below. Override recipes can additionally customise the Fileset, Fileset groups, etc.

Override Recipes & Trust Relationship

For security, a trust relationship was added between recipes. The idea is the child recipe is made to trust it's parent recipes. If an updated version of a parent recipe is pulled from a repository, then this parent will no longer be trusted by that child, until the trust relationship is manually updated. AutoPkg does not offer the ability to change trust and so this must be done via the command line.

With no trust, when a recipe is run that relies on parent recipes you will see an error similar when running the recipe from Terminal:

```
$ autopkg run -v OracleJava8.filewave.local
Processing OracleJava8.filewave.local...
Failed local trust verification.
Receipt written to /tmp/receipts/OracleJava8.filewave-receipt-20180409-141621.plist

The following recipes failed:
  OracleJava8.filewave.local
    No trust information present.

Nothing downloaded, packaged or imported.
```

In this example, Creating a Recipe Override will create a recipe that has trust added for us. Using the above Java8 example, first make an Override recipe. The Override recipe and initial trust can be created in either AutoPkg or using the command line. The Java 8 override recipe will be called "Java8.filewave.override". The last entry is reference to the parent recipe to be overridden (this can be either be recipe name or it's identifier, recipe name used in this example):

```
$ autopkg make-override -n Java8.filewave.override Java8.filewave
```

By making the override file in this way, the trust relationship has been added automatically to the Override recipe. Now there is a trust relationship, the override file can be used to run the recipe (either through Terminal or AutoPkg):

```
$ autopkg run -v OracleJava8.filewave.override
Processing OracleJava8.filewave.override...

[lines removed]

The following fileset was imported:
  Fw Fileset Id  Fw Fileset Group  Fw Fileset Name
-----
  194266        Root              Java - 1.8.161.12

The following packages were copied:
  Pkg Path
-----
```

```
/Users/Shared/Autopkg/Cache/local.override.filewave.OracleJava8/Java-1.8.161.12.pkg
```

The following new items were downloaded:

Download Path

```
/Users/Shared/Autopkg/Cache/local.override.filewave.OracleJava8/downloads/Java.dmg
```

If after updating repos, the trust relationship error is flagged against any recipes, this indicates that a parent has been updated and trust is no longer in place. At this point, the parent should be reviewed to observe the changes made. Changes to a recipe can easily be viewed by navigating to the relevant recipe on GitHub and viewing the 'History'.

Once confirmation has been made that the changes are acceptable, a new trust relationship should be created. As an override file already exists, the trust will need to be updated for the Java 8 override recipe; as such re-trusting all parents:

```
$ autopkg update-trust-info Java8.filewave.override
```

Although it is possible to disable trust relationship, this should not be recommended for security reasons. Current status can be seen by running the following and checking the value of 'FAIL_RECIPES_WITHOUT_TRUST_INFO':

```
$ autopkg info
```

It is possible to temporarily override the trust relationship, such that it is ignored:

```
$ autopkg run --ignore-parent-trust-verification-errors [name of recipe]
```

Important



FOR SECURITY REASONS, IT IS ALWAYS RECOMMENDED THAT RECIPES ARE CHECKED BEFORE INGESTING INTO YOUR FILEWAVE SERVER AND CREATED FILESETS ARE SUBSEQUENTLY CHECKED ON TEST MACHINES BEFORE DEPLOYING TO LARGER GROUPS OF MACHINES

Related Content

- [Integrated AutoPkg \(v15.5+\)](#)
- [Autopkg\(r\) FAIL_RECIPES_WITHOUT_TRUST_INFO](#)
- FileWave & AutoPkg Instructions - <https://github.com/autopkg/filewave>
- The primary site for AutoPkg - <http://autopkg.github.io/autopkg>
- Github Site for AutoPkg - <https://github.com/autopkg/autopkg>
- AutoPkgr website - <http://www.lindegroup.com/autopkgr>
- AutoPkg Trust - <https://github.com/autopkg/autopkg/wiki/AutoPkg-and-recipe-parent-trust-info>

Autopkg(r)

FAIL_RECIPES_WITHOUT_TRUST_INFO

AutoPkg(r) FAIL_RECIPES_WITHOUT_TRUST_INFO

Description

Autopkg provides security through trust relationship. Each recipe is set to trust any parents. If those parents change, the trust will be broken until the recipe is informed to trust these updated parent recipes. Message may read as follows with no exit status error:

```
WARNING: com.github.autopkg.filewave.OracleJava8 is missing trust info and FAIL_RECIPES_WITHOUT_TRUST_INFO is not set.
Proceeding...
```

This is a generic Autopkg(r) message and details on Trust Info configuration to address this may be found at [AutoPkg and recipe parent trust info](#)

Typical parent updates are due to URL changes in a download recipe.

FileWave 13

After to upgrading to FileWave 13, the following errors may be experienced:

Exit Status 108:

```
WARNING: com.github.autopkg.filewave.OracleJava8 is missing trust info and FAIL_RECIPES_WITHOUT_TRUST_INFO is not set. Proceeding...
```

```
Command '['/Applications/FileWave/FileWave Admin.app/Contents/MacOS/FileWave Admin', '-u', u'autopkg', '-p', u'autopkg', '-H', u'filewave.server.com', '-P', '20016', '--listFilesets']' returned non-zero exit status 108
```

or

Exit Status 109:

```
WARNING: com.github.autopkg.filewave.Evernote is missing trust info and FAIL_RECIPES_WITHOUT_TRUST_INFO is not set. Proceeding...
```

```
Error in com.github.autopkg.filewave.Evernote: Processor: com.github.autopkg.filewave.FWTool/FileWaveImporter:
Error: Error importing the folder
'/Users/username/Library/AutoPkg/Cache/com.github.autopkg.filewave.Evernote/Evernote/Evernote.app' into FileWave
as a fileset called 'Evernote - 7.7'. Reason: Command '['/Applications/FileWave/FileWave
Admin.app/Contents/MacOS/FileWave Admin', '-u', u'autopkg', '-p', u'autopkg', '-H', u'filewave.server.com', '-P',
'20016', '--importFolder',
u'/Users/username/Library/AutoPkg/Cache/com.github.autopkg.filewave.Evernote/Evernote/Evernote.app', '--name',
u'Evernote - 7.7', '--root', u'/Applications/Evernote.app']' returned non-zero exit status 109
```

FileWave 13 has increased security and the server certificate is part of this security. There are also changes and additional options for FileWave Administrator Preferences. As such, some configuration changes will be necessary.

FileWave Admin

- 1 Additionally, if using a self-signed certificate, please observe the necessary steps for FileWave Admin in the following article to ensure you have a local copy of the certificate: [Self-Signed SSL Certificates Going Forward](#)

Directions

Exit Status 108

This is likely to be one of the following:

- Server Certificate
- Autopkg(r) setting - Server Name

- Autopkg(r) setting - User/Password
- An old expired certificate is in the Keychain

Server Certificate

Confirm that your server meets necessary requirements. For example:

- Server Common Name matches Server Name
- Certificate has not expired

Further details on certificates can be seen at: [Root Trusted SSL Certificate \(Using and Renewing\)](#)

Server Name

The following preference for server name, configured for Autopkg(r), needs to match the server address/common name and may not be, for example, IP or "localhost".

The following command may be use to confirm the current server settings of Autopkg(r). This should be run as the user and not root:

```
defaults read com.github.autopkg FW_SERVER_HOST
```

If the response of the server does not match the server's address/common name, then the value will need to be amended to match. Using the example above, server address/common name "filewave.server.com", the command should be:

```
defaults write com.github.autopkg FW_SERVER_HOST filewave.server.com
```

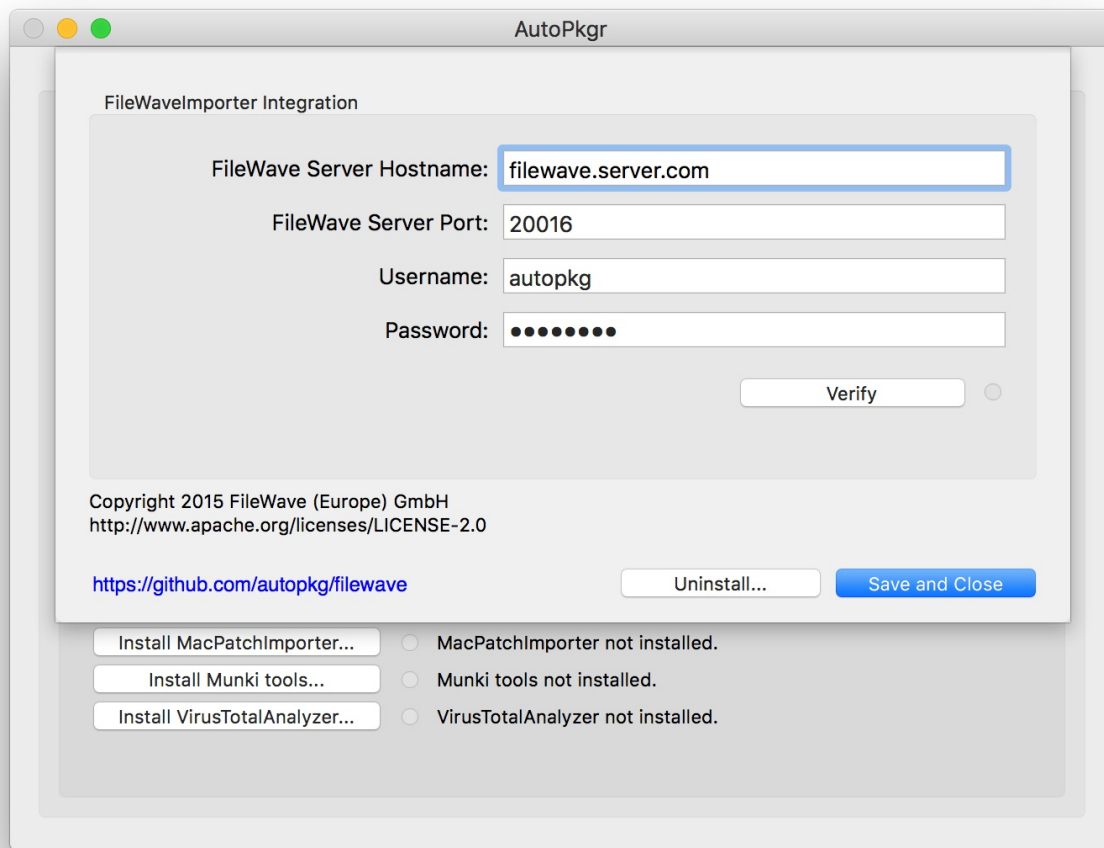
User settings

Prior to FileWave 13, the settings for the user, e.g password, could be left blank and the default password would be used. They must now be filled in.

The following command may be used to set the user and password (example username and password of autopkg): This should be run as the user and not root:

```
defaults write com.github.autopkg FW_ADMIN_USER autopkg
defaults write com.github.autopkg FW_ADMIN_PASSWORD autopkg
```

Note, both the above may be observed and set through Autopkg: 'Folders & Integration' > 'Configure FileWaveImporter'



Expired Certificate

Remove old expired certificates from the keychain. Check to ensure they are removed from both:

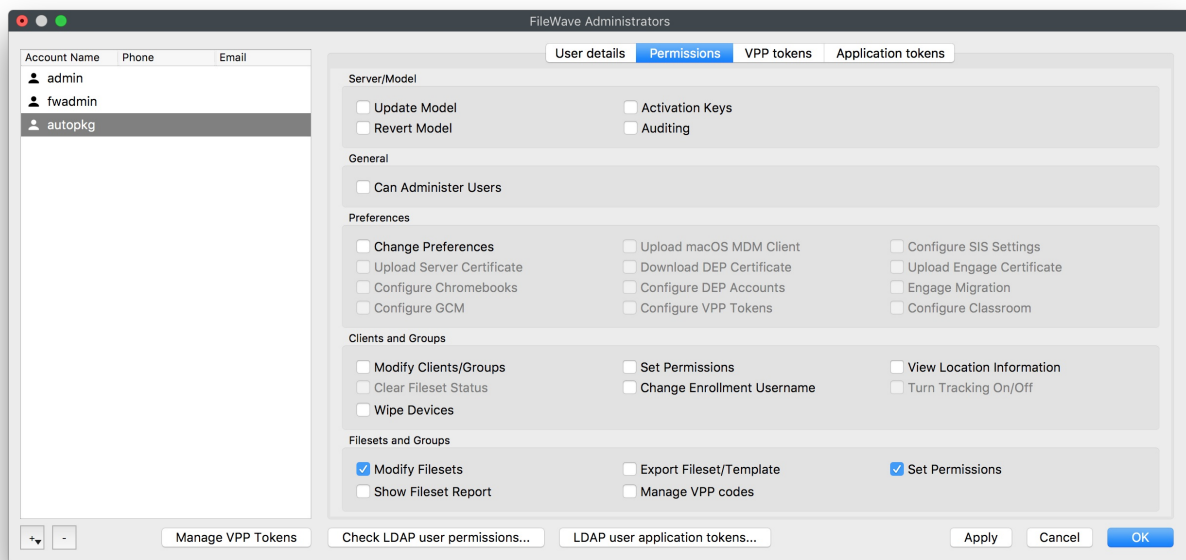
- login
- System

Exit Status 109

Manage Administrators

FileWave 13 has additional options and amended default settings for Administrator Preferences. If exit status 109 is seen, this may indicate that the settings for the 'autopkg' Administrator account need addressing.

Ensure the 'autopkg' user has permissions to modify Filesets:



Test

Once any of the above have been amended, re-run the recipes.

Cloudflare WARP integration with FileWave (macOS/iOS/Windows/Android)

Description

Cloudflare WARP is a popular choice of software to deploy to devices. This process can be simplified with FileWave.



Third Party Software

Slack is a third party application. The details provided are for example only and are unsupported by FileWave.

Ingredients

- Software installer for macOS and/or Windows, PKG/MSI
- VPP for iOS
- EMM for Android
- Configuration files

Information and resources are provided via the [Cloudflare Managed Deployment](#) documentation and the [Cloudflare WARP Download](#) page.

Directions

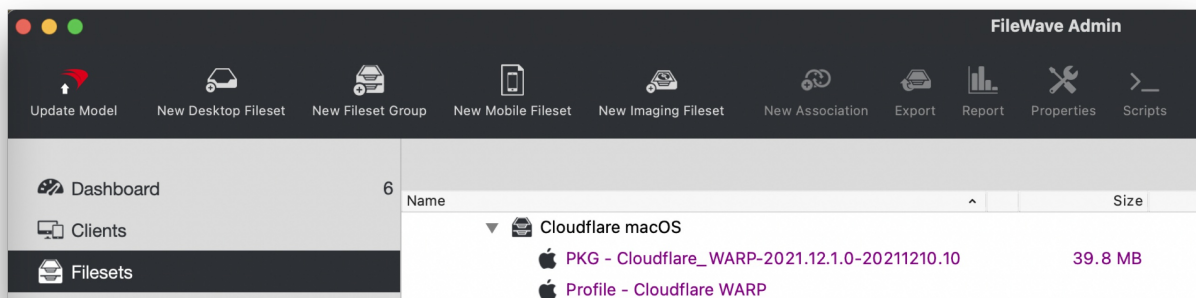
Cloudflare WARP has options for macOS, Windows, iOS and Android.

macOS

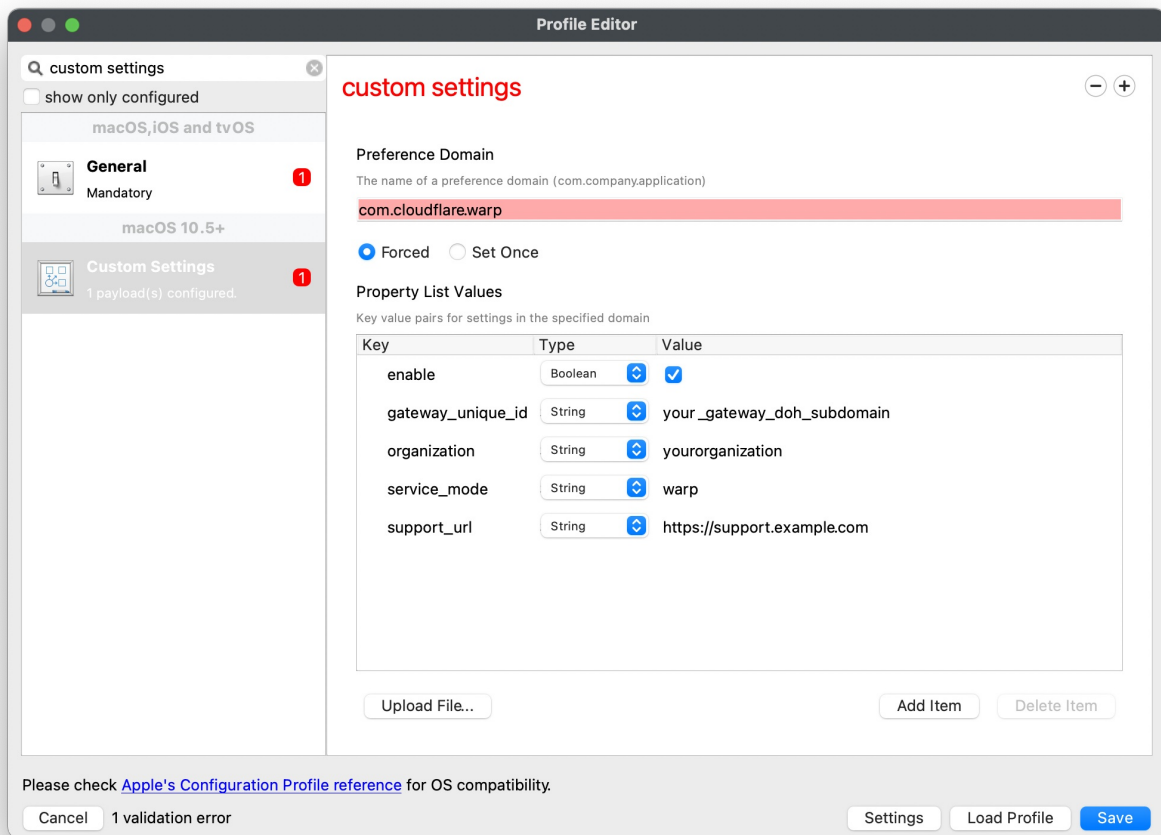
Cloudflare WARP for macOS requires the PKG installer, plus a configuration profile.

1. Create a Fileset Group and name as desired, e.g Cloudflare macOS
2. Download the Cloudflare PKG
3. Drag the PKG into the FileWave Admin Fileset Group created above
4. Highlight the Fileset Group and choose to create a New Desktop Fileset
5. Select Profile
6. In the Profile Editor window choose the Custom Settings Payload and Configure
7. The profile may be built from scratch or the Cloudflare example template could be downloaded, benefitting from the 'Upload file' option within the Profile Editor
8. If building manually, the preference domain should be: com.cloudflare.warp
9. Edit or add/remove keys and values as required. Details of [Cloudflare Parameters](#) below
10. Associate the Cloudflare Fileset Group to one or more test devices and deploy once satisfied all is well

Example Fileset Group



Example Custom Settings Payload, edit details as required

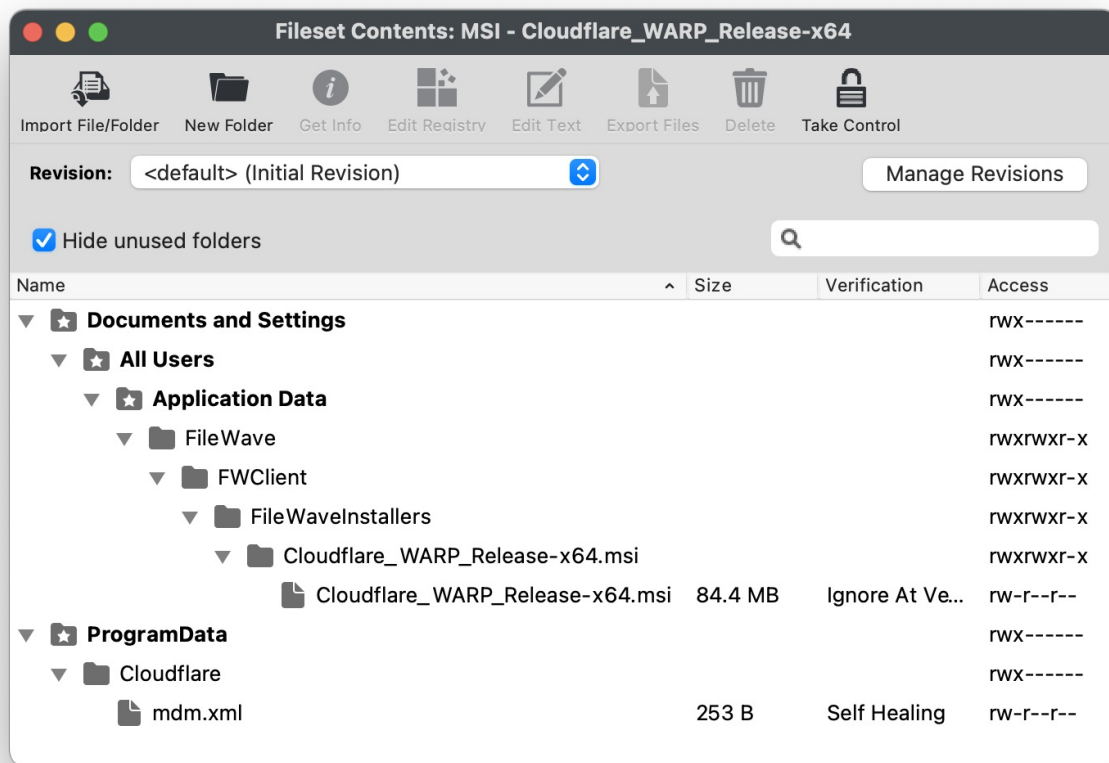


Windows

Cloudflare WARP for Windows requires the MSI installer, plus an XML configuration file

1. Download the Cloudflare MSI
2. Drag the MSI into the FileWave Admin Fileset view, creating a new Fileset
3. Download the mdm.xml template provided by Cloudflare or create one
4. Create a folder called Cloudflare within the ProgramData folder of the Fileset
5. Upload the XML file into this Cloudflare directory
6. Edit or add/remove keys and values within the XML as required. Details of [Cloudflare Parameters](#) below
7. Associate the Cloudflare Fileset to one or more test devices and deploy once satisfied all is well

Example Cloudflare Fileset



Example mdm.xml file, edit details as required

```
<dict>
  <key>organization</key>
  <string>exampleorg</string>
  <key>service_mode</key>
  <string>warp</string>
  <key>gateway_unique_id</key>
  <string>fmxk762nrj</string>
  <key>support_url</key>
  <string>http://support.example.com</string>
</dict>
```

iOS

Cloudflare for iOS requires the 1.1.1.1: Faster Internet App, with additional configuration

1. Purchase the free App Store App: [1.1.1.1: Faster Internet](#) through either Apple Business or School Manager
2. Associate the App to the relevant VPP licence and sync VPP in FileWave Preferences; accepting the creation of the new Fileset.
3. Double click the newly created Fileset and choose the Configuration tab
4. Manually create keys and values or upload the Cloudflare example template file
5. Edit or add/remove keys and values within the xml as required. Details of [Cloudflare Parameters](#) below
6. Associate the 1.1.1.1: Faster Internet Fileset to one or more test devices and deploy once satisfied all is well

Example Custom Settings Payload, edit details as required

Fileset Name: iOS App - ABM - 1.1.1.1: Faster Internet

Details Kiosk **Configuration** Associated Domains (iOS 13+)

Application configuration

Key	Type	Value
auto_connect	Number	1
organization	String	yourorganization
service_mode	String	warp
support_url	String	https://support.example.com
switch_locked	Boolean	<input type="checkbox"/>

Upload File... Export to File... Add Item Delete Item

Per-App VPN: None

*This App should be managed in order to use Per-App VPN
*Make sure that selected Per-App VPN payload is deployed to your device

Cancel OK

Android

Cloudflare for Android requires the 1.1.1.1: Faster & Safer Internet App, with additional configuration

1. Choose New Mobile Fileset
2. Select Play Store
3. Purchase the free Play Store App: [1.1.1.1: Faster & Safer Internet](#)
4. Double click the newly created Fileset and choose the Managed Properties tab
5. Edit or add/remove values as required. Details of [Cloudflare Parameters](#) below
6. Associate the 1.1.1.1: Faster & Safer Internet Fileset to one or more test devices and deploy once satisfied all is well

Example Managed Properties, edit values as required

Fileset Name: Play Store App - 1.1.1.1: Faster & Safer Internet

Configuration Permissions Managed Properties

Organization:
The user will be asked to sign into this organization. The name of the organization is case-insensitive.

Service mode:
Choose the mode in which the tunnel should run.

Gateway Unique ID:
Cloudflare Gateway DoH Subdomain. This option is not compatible with 1.1.1.1 for Families.

Enable the service: ☒
Force-enable the service in the selected (WARP/1.1.1.1) mode.

Support URL:
The URL provided here will be opened using the browser when the user uses the Send feedback option in the app. Supported on: iOS, Android.

Custom ID:
Custom/internal user identifier stored on the device.

Show onboarding: ☐
Show onboarding

Switch Locked: ☒
Switch Locked

Auto Connect:
Auto Connect

Cancel OK


Cloudflare Parameters


All the above example configurations will require adapting to include appropriate keys and values for each environment. Explanations of each key and its possible requirements and values can be found in the [Cloudflare Parameters](#) list.

DeepFreeze (macOS/Windows)

Description

DeepFreeze is a common pieces of software used by Administrators, many people have asked us how the two interact. Simply putting the two on one computer would cause the software to conflict, as FileWave would deploy software and DeepFreeze would remove it upon reboot. However, using a slightly modified deployment strategy, you can use FileWave to activate Deep Freeze's commands. In this article, I will describe the necessary steps for deploying a fileset to machines Frozen by DeepFreeze.

 **Third Party Software**
DeepFreeze is a third party application. The details provided are for example only and are unsupported by FileWave.

 You will invariably want to do this while users are not using the computers, since you do not want them to make modifications during its thawed boot. Therefore, this script will reboot the computer immediately. To make this script, you'll need to know the Deep Freeze administrator name and password. In your script, substitute your name/pass for 'your_admin' and 'your_password'

Ingredients

- FW Admin

Directions

- Create a new empty fileset (From the Fileset View : New Desktop Fileset > Empty : Type a Name > OK)
- With it selected: Click the Scripts item in the button bar
- Create a new Activation Script and name it.
- Paste the script below and edit

This is to Thaw macOS

```
#!/bin/sh
/Applications/Faronics/DFXControl.app/Contents/MacOS/DFXControl command your_admin your_password bootThawed
/sbin/reboot
```

This is to Thaw Windows

```
@echo off "%ProgramFiles%\FaronicsDeep Freeze EnterpriseDFC.exe" yourPassword /BOOTTHAWED shutdown -r
```

- Create a new Pre-Uninstallation Script and name it
- Paste the script below and edit

This is to Freeze macOS

```
#!/bin/sh
/Applications/Faronics/DFXControl.app/Contents/MacOS/DFXControl command your_admin your_password bootFrozen
/sbin/reboot
```

This is to Freeze Windows

```
@echo off "%ProgramFiles%\FaronicsDeep Freeze EnterpriseDFC.exe" yourPassword /BOOTFROZEN shutdown -r
```

It will reboot the machine in frozen mode after the fileset has been set to delete

 **Don't put passwords in scripts. See: [Script Best Practices](#)**

- Associate and schedule the script
 - Schedule the activation time for before you want to make changes
 - Schedule the delete time for after you are done making changes

Edit Association

Edit Association between Fileset:
Script - DeepFreeze - Thaw-n-freeze
and Client/Group/Clone
Site A

Timing

License Distribution

☐ Start downloading at:

☒ Activate files at:

8/3/20 9:00 PM

☐ Make files inactive at:

☒ Delete files at:

8/3/20 9:00 PM

☐ Kiosk Association

Cancel

OK

- Schedule the filesets you want to make changes to the computers

Fileset	Client/Group/Clone	Start Download	Become Active	Start Passive	Delete Files
Script - DeepFreeze - Thaw-n-freeze	Site A	-	8/3/20 9:00 PM	-	8/3/20 9:00 PM
Filesets to Apply changes	Site A	-	8/3/20 10:00 PM	-	-

Hello-IT integration with FileWave (macOS)

Description

Hello-IT is a 3rd party tool for macOS designed as a Menu Item to launch scripts as a Self Service tool. The below is an example of how FileWave can deliver and integrate with the tool. Hello-IT has multiple options for the Menu Item list, including:

- Reporting on server status
- Links to Websites
- Reporting information in the Menu Item
- Additional Scripting for Self Service (Including Integration with Slack)

Instructions on Hello-IT may be found via the following link:

<https://github.com/ygini/Hello-IT>



Third Party Software

Slack is a third party application. The details provided are for example only and are unsupported by FileWave.

Requirements

- Hello-IT installer
- Hello-IT configuration file



Self-Signed Certificates

Although many features of Hello-IT will work fine, the option to report on server status relies upon the ability to pull data from a server. If the server is not trusted, then this will fail. As such, servers with self-signed certificates will always report a failure when using public.test.http (see below)

Some examples below implement Slack, based upon our KB:

[Slack integration with FileWave](#)

Directions

Configuration of Hello-IT can be via a Custom Settings payload profile, which may be delivered through FileWave to devices, the basic example of which can be uploaded into a Custom Settings payload and is located within the GitHub repository: <https://github.com/ygini/Hello-IT/blob/master/example/Basic%20Example/com.github.ygini.Hello-IT.plist>

The Menu Items are linked to functions, below are some examples of Public functions.

public.open.resource

This function provides URLs that will launch in the default browser. Editing the provided Fileset, can allow for important websites, e.g. Intranet pages. In this example, FileWave Foundry and Website are offered:

public.open.resource

```
Dict {
  settings = Dict {
    title = FileWave Website
    URL = https://www.filewave.com
  }
  functionIdentifier = public.open.resource
}
Dict {
  settings = Dict {
    title = FileWave Foundry
    URL = https://foundry.filewave.com
  }
  functionIdentifier = public.open.resource
}
```

public.test.http

This function tests for access to a Web server by running a download command and checking the output of the download. If successful all will be well, but if failed, the Menu Item text will become red and a red dot will be highlighted next to the text in the Drop Down.

An md5 checksum is required for the resource, and may be obtained using a curl command. For example, if using the URL: <https://custom.filewave.com>, the following would be run

Web Resource md5

```
curl https://custom.filewave.com | md5
```

% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current
			Dload	Upload	Total	Spent	Left
100	1406	100	1406	0	0	11051	0
						--:--:--	--:--:--
						--:--:--	--:--:--
							11070

```
63e398fd52c3dc883d13401531339f51
```

The payload would then be amended thus:

public.test.http

```
Dict {
  settings = Dict {
    repeat = 60
    ignoreSystemState = true
    mode = md5
    originalString = 63e398fd52c3dc883d13401531339f51
    title = FileWave Server
    URL = https://custom.filewave.com
  }
  functionIdentifier = public.test.http
}
```

Configuration could be set to point to your FileWave server for example, so users can see that a connection is established.

public.script.item

This function provides the ability to run scripts. In this example, consider the [KB article that uses Slack](#) as an IT reporting tool and further utilising this for the following requests:

- Printer ink replacement
- Call the user back

The format of this section of the file could look like:

public.script.item

```
Dict {
  settings = Dict {
    content = Array {
      Dict {
        settings = Dict {
          script = slack_printer_ink.sh
          title = Dict {
            fr = Demander l'encre d'imprimante
            en = Request Printer Ink
          }
        }
      }
      functionIdentifier = public.script.item
    }
  }
  Dict {
    settings = Dict {
      script = slack_request_callback.sh
      title = Dict {
        fr = Demander un rappel
        en = Request IT Callback
      }
    }
    functionIdentifier = public.script.item
  }
}
```

```

    title = Dict {
      fr = Libre-service
      en = Self-Service
    }
  }
  functionIdentifier = public.submenu
}

```

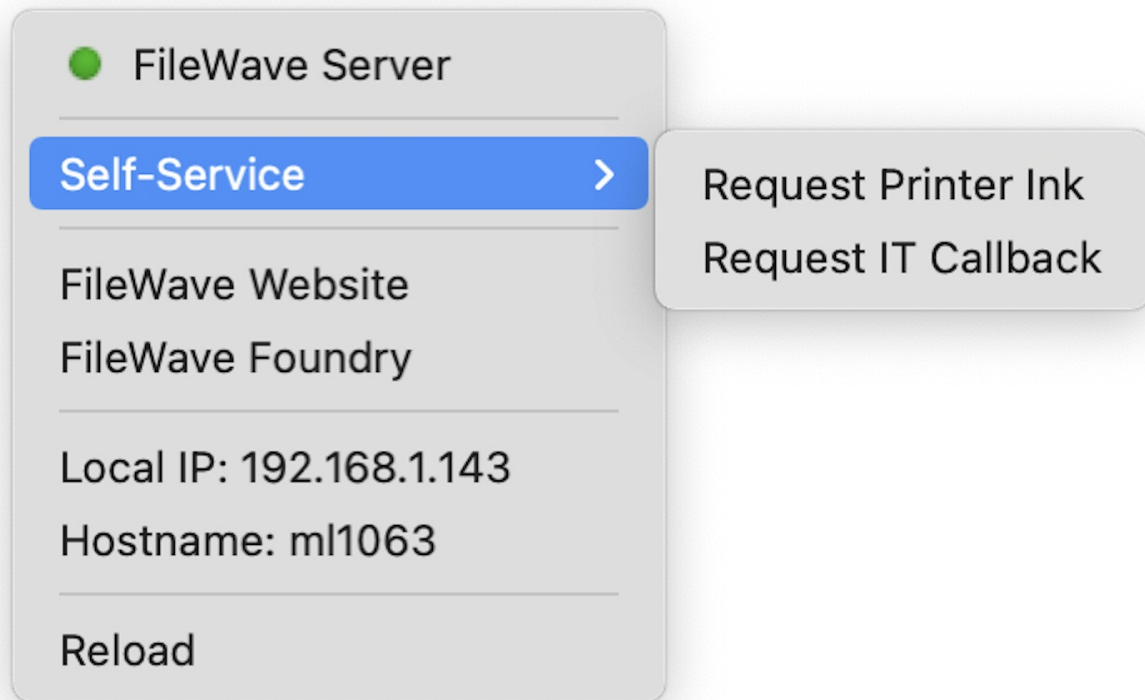
First thing to note is that these have been wrapped in their own 'Self-Service' sub Menu Item 'public.submenu'. In the example two scripts have been referenced:

- slack_printer_ink.sh
- slack_request_callback.sh

Scripts should be located in the following directory:

```
/Library/Application Support/com.github.ygini.hello-it/CustomScripts/
```


The user may then use the drop down menu to trigger these items, where IT would then receive a Slack message.



Example Filesets

Based upon the above, the following example Filesets include not only the configuration file, but the example scripts that implement the Slack integration. Consider placing these along with the Hello-IT installer in a Fileset Group and associate the group with devices:

▼ Hello-IT

 Hello-IT Slack Printer Ink

 Hello-IT Slack Request Callback

 **PKG - Hello-IT-1.6.0-Release-Distribution**

 **Profile - Hello-IT**

Please see the KB on [FileWave and Slack](#) with regards to creating the required Slack Webhook. Configuration of the scripts to set the Webhook are the same here, editing the Filesets, selecting the pre-installation script and then editing the 'Environment Variables' such that the 'slack_webhook' is set to the generated Webhook from Slack; replacing 'PLACE WEBHOOK HERE'.

Similarly, if using Legacy Webhooks, the slack_channel variable needs to be edited to match the name of the created channel; for App Webhooks it will be ignored. In the example, this channel is called 'fw_messages'

Profile - Hello-IT

[Profile - Hello-IT.fileset.zip](#)

Hello-IT Slack Request Callback



[Hello-IT Slack Request Callback.fileset.zip](#)

This script attempts to read the telephone number of the user based upon a directory service entry. If device is not bound or user is not a directory user, no number will be supplied.



Hello-IT Slack Printer Ink

[Hello-IT Slack Printer Ink.fileset.zip](#)

Example messages in Slack:

 **Helpdesk** APP 11:39 AM
 **Request Callback: 1015VMDEP**
Message from annie

Please call back: 0118 999 881 999 119 725 3
macOS: 10.15.1, client: 13.2.3 | Today at 11:39 AM

 **Helpdesk** APP 11:34 AM
 **Printer Ink Request: 1015VMDEP**
!!Printer Ink is Low!!

Request: Please Instal New Cartridge
macOS: 10.15.1, client: 13.2.3 | Today at 11:34 AM

Conclusion

This is just an example of how FileWave can be used to deliver an additional tool to devices, empowering users to easily request resources and where communication may be hindered, allow the user to easily request IT assistance; particularly useful for users who may be remote.

Invgate integration with FileWave

A common need of an IT admin is to see computer and mobile assets from within a Help Desk system. FileWave can send asset information, keyed by the FileWave "Last Logged in User" to the Invgate ITSM solution.



Third Party Software

Slack is a third party application. The details provided are for example only and are unsupported by FileWave.

Step-by-step guide

Setup the integration on the Invgate side:

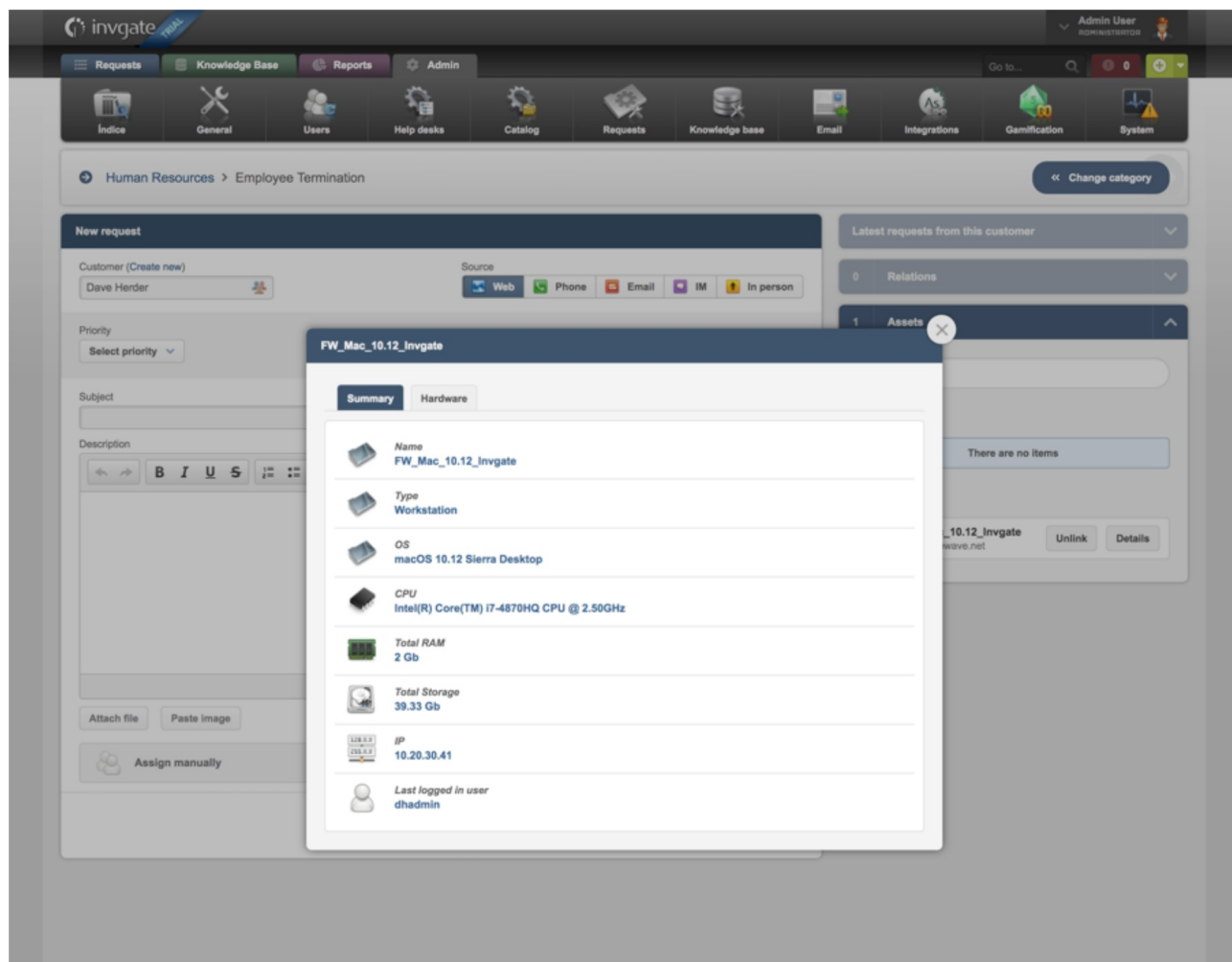
1. Ensure you are capturing User data in Invgate as well as FileWave (LDAP integration in both systems is wise to ensure you are dealing with the same user data across systems).
2. In the Invgate> "Admin"> "Integrations" tab, define the following fields:

The screenshot shows the Invgate Admin interface. The top navigation bar includes 'Requests', 'Knowledge Base', 'Reports', and 'Admin'. The 'Admin' tab is active, and the 'Integrations' sub-tab is selected. The main content area is titled 'Asset management' and contains a 'Configuration' section for the FileWave integration. The configuration fields are as follows:

- Application:** FileWave (Beta)
- Name:** [Redacted]filewave.net (Define a name to identify this integration.)
- Categories:** All (Categories in which it will be allowed to relate assets from this integration.)
- Link assets:** Enabled (Allow end users to link assets to requests.)
- Search for assets:** Enabled (Allow end users to search for assets.)
- Connection:** (Test button)
- Host:** [Redacted]filewave.net (Server name or IP.)
- Port:** 20443 (The default port is 20443.)
- SSL:** Enabled (When enabled a secure connection will be used.)
- Shared key:** [Redacted]MjExN30= (Share Base64 encoded key for FileWave inventory.)

On the right side of the configuration panel, there is a 'Save' button and a summary section titled 'Asset management' with the text 'Integration with ITAM solutions.' and a code icon for API settings.

1. Click "Save"
2. Wait for the data to be synchronized from FileWave to Invgate
3. Create a new request based on the new Asset information now populating Invgate. Note the new assets populating in the left hand column of the Invgate "New Request" dialog.



Related articles

- [How to write to a custom field using the FileWave API](#)

ServiceNow integration with FileWave

ServiceNow can be provided with device inventory information from FileWave to make the ServiceNow experience much more accurate and rewarding. Previously, a ServiceNow portal app was required to supply this FileWave data, but as of version Orlando of ServiceNow that is no longer required, and the FileWave API can be used directly. This guide, provided as a courtesy for a non-FileWave related system, will show you how to bring the information gathered by FileWave inventory into ServiceNow using the API.

- Third Party Software**
ServiceNow is a third party application. The details provided are for example only and are unsupported by FileWave.

Step-by-Step Guide

Step 1: Create Data Source

System Import Sets > Administration > Data Sources > New

1. NAME: FileWave REST API
2. IMPORT SET LABEL: (empty)
3. IMPORT SET TABLE NAME: u_fw_api
4. TYPE: REST
5. FORMAT: JSON
6. PATH FOR EACH ROW: /values/values
7. DISCARD ARRAYS: ✓
8. EXPAND: ✓
9. DATA IN SINGLE COLUMN: (unchecked)
10. APP: Global
11. REQUEST ACTION: Hit the "click here" and create one (see Step 2)

The screenshot shows the 'New Data Source' form in ServiceNow. The 'Name' field is 'FileWave REST API'. The 'Type' is 'REST (Integration only)'. The 'Format' is 'JSON'. The 'Path for each row' is '/values/values'. The 'Discard arrays' and 'Expand' checkboxes are checked. The 'Data in single column' checkbox is unchecked. The 'Request action' is 'GET FW REST'. The 'Application' is 'Global'. There is a 'Click here to understand how to configure a Data Source Request Action' link.

Step 2: Create Request Action

1. NAME: GET FW REST
2. ACCESSIBLE FROM: All application scopes
3. CATEGORY: (empty)
4. PROTECTION: —None—
5. DESCRIPTION: (empty)
6. APPLICATION: Global
7. IN-FLOW ANNOTATION: (empty)
8. Then hit "Submit"

(You should be redirected to the Flow Designer for GET FW REST...see Step 3)

The screenshot shows the 'New Request Action' form in ServiceNow. The 'Name' is 'GET FW REST'. The 'Accessible From' is 'All application scopes'. The 'Category' is empty. The 'Protection' is 'None'. The 'Application' is 'Global'. The 'In-Flow Annotation' field has a placeholder text: 'Type what you would like to have appear right under the Action title in the flow Designer view'. The 'Description' field is empty.

Step 3, Part 1: Create in Flow Designer

Create in FLOW DESIGNER, Select #1 "REST step"

- Connection Details -

- CONNECTION: Define connection inline
- CREDENTIALIAL ALIAS: (empty)
- USE MID: (unchecked)
- BASE URL: https://YOUR.FILEWAVE.FQDN:20445/inv/api/v1/query_result_extended/

-Request Details-

- BUILD REQUEST: Manually
- RESOURCES PATH: (empty)
- HTTP METHOD: POST

The reason this is a POST is so we don't rely on an inventory query that might be changed.

Customize the fields portion as needed.

This is an example of custom fields. If the field named "asset_tag" does not exist in your system, this will cause an error.

```
{
  "column": "asset_tag",
  "component": "CustomFields"
}
```

- QUERY PARAMETERS: (empty)
- HEADERS:
 1. Name: Authorization
Value: Your Users Application Token (FW Admin >

The screenshot shows the 'Flow Designer' for the 'GET FW REST' step. The 'Action Outline' shows the 'REST step' selected. The 'Connection Details' section shows the 'Connection' as 'Default Connection', 'Credential alias' as empty, 'Use MID' as unchecked, and 'Base URL' as 'https://YOUR.FILEWAVE.FQDN:20445/inv/api/v1/query_result_extended/'. The 'Request Details' section shows 'Build Request' as 'Manually', 'Resources Path' as empty, 'HTTP Method' as 'POST', and 'Query Parameters' as empty. The 'Request Content' section shows a JSON body:

```
{
  "column": "asset_tag",
  "component": "CustomFields"
}
```

- Assistants > Manage Administrators > (select admin) >
Application Tokens > (Copy the base64: e.g.
aalkjdIAKJDlakjdALkdsjaldksja=)
2. Name: Content-Type
Value: application/json

-Request Content-

- REQUEST TYPE: Text
- REQUEST BODY: - See "Request JSON" example below:

Request JSON

```
{
  "criteria": {
    "expressions": [
      {
        "column": "filewave_client_name",
        "component": "Client",
        "operator": "is_not",
        "qualifier": null
      },
      {
        "column": "last_check_in",
        "component": "Client",
        "operator": "!=",
        "qualifier": null
      }
    ],
    "logic": "all"
  },
  "fields": [
    {
      "column": "filewave_id",
      "component": "Client"
    },
    {
      "column": "filewave_client_name",
      "component": "Client"
    },
    {
      "column": "current_ip_address",
      "component": "Client"
    },
    {
      "column": "last_check_in",
      "component": "Client"
    },
    {
      "column": "latitude",
      "component": "GeoLocation"
    },
    {
      "column": "longitude",
      "component": "GeoLocation"
    },
    {
      "column": "type",
      "component": "OperatingSystem"
    },
    {
      "column": "version",
      "component": "OperatingSystem"
    },
    {
      "column": "filewave_client_version",
      "component": "AndroidClient"
    },
    {
      "column": "filewave_client_version",
      "component": "DesktopClient"
    },
    {
      "column": "cpu_count",
```

```

      "component": "Client"
    },
    {
      "column": "cpu_speed",
      "component": "Client"
    },
    {
      "column": "ram_size",
      "component": "Client"
    },
    {
      "column": "total_disk_space",
      "component": "Client"
    },
    {
      "column": "free_disk_space",
      "component": "Client"
    },
    {
      "column": "serial_number",
      "component": "Client"
    },
    {
      "column": "asset_tag",
      "component": "CustomFields"
    }
  ],
  "main_component": "Client"
}

```

-Retry Policy-

- ENABLE RETRY POLICY: (unchecked)
- Save
- Test Connection
 - Test > Use “test” for “Attachment Name” > Run Test
 - Select the “Action has been executed. To view action, click here.”
 - Verify: status code of 200 (means success)
 - Select the “(gear) GET FW REST” flow designer tab
 - Close the “Test Action” window with the X
- Select Publish

Close this browser window/tab to return to the “New Record | Data Source | ServiceNow” window/tab

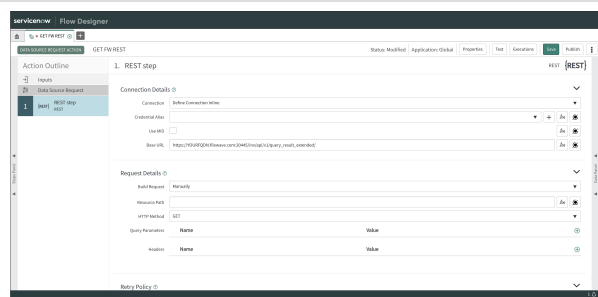
Step3, Part 2: Set Request Action

Under REQUEST ACTION, start typing the name of the Action we just created “GET FW REST” and select it from the drop-down.

Submit

Hitting ‘Submit’ will redirect you back to the Data Source list. Select the one we just made, FileWave REST API and select “Load All Records” under Related Links

Once the test has completed, go back to FileWave REST API's Data Source page and continue to next step.



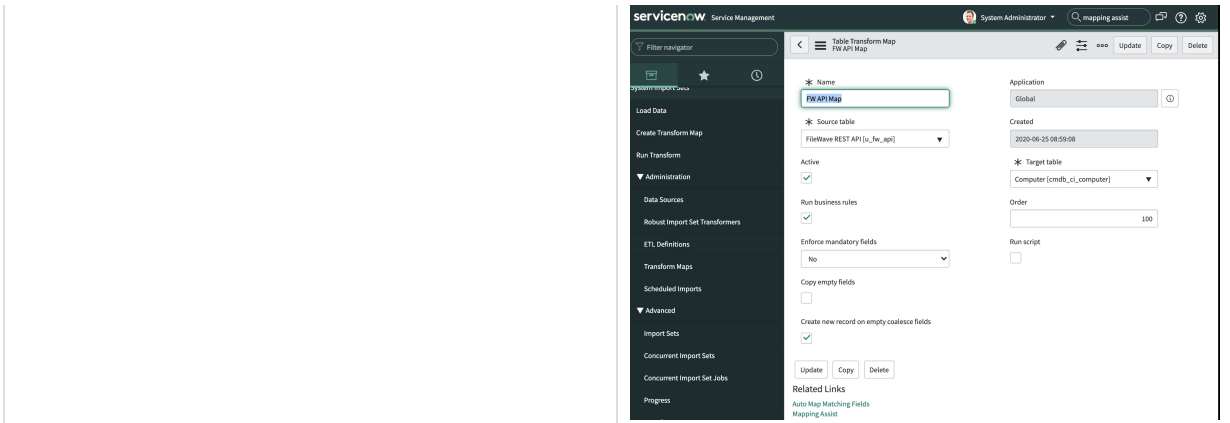
Step 3, Part 3: Generate Data Transform

Generate Transform(s)

— MAKING TRANSFORMS —

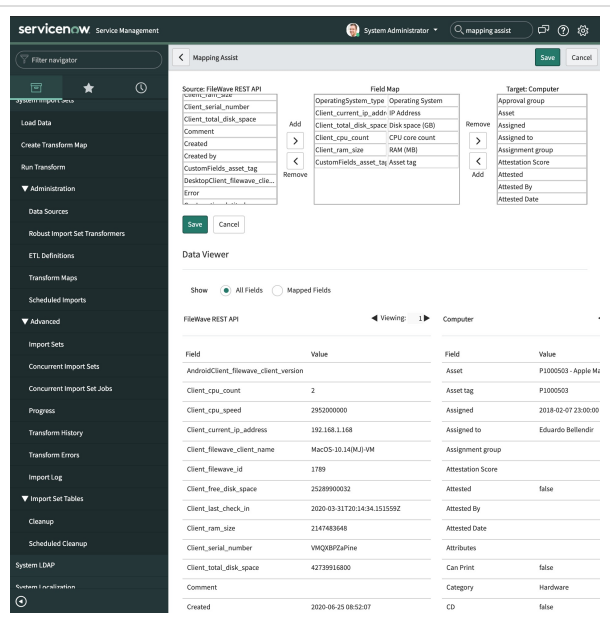
On the FileWave REST API page, scroll down to the “Transforms” tab and select “New”

- NAME: FW API Map
- SOURCE TABLE: (should auto fill the u_fw_api table)
- ACTIVE: ✓
- RUN BUSINESS RULES: ✓
- ENFORCE MANDATORY FIELDS: No
- COPY EMPTY FIELDS: (unchecked)
- CREATE NEW RECORD ON EMPTY COALESCE FIELDS: ✓
- APPLICATION: Global (input disabled)
- CREATED: Will auto-generate with Date/Time stamp when saved (input disabled)
- TARGET TABLE: Computer [cmdb_ci_computer]
- ORDER: 100
- RUN SCRIPT: (unchecked)



Select “Mapping Assist” under Related Links
You should be able to map a lot of the client's information, for example...

FileWave	ServiceNow
operatingsystem_type	Operating System
client_cpu_count	cpu_count
client_current_ip_address	ip_address
client_ram_size	ram
customfields_asset_tag	asset_tag
client_total_disk_space	disk_space
operatingsystem_version	os_version
client_cpu_speed	cpu_speed
client_filewave_client_name	name
client_serial_number	serial_number
...	



You can make more maps, like this example for location data:

Scroll down to the “Transforms” tab and select “New”

- NAME: FW Location Map
- SOURCE TABLE: (should auto fill the u_fw_api table)
- ACTIVE: ✓
- RUN BUSINESS RULES: ✓
- ENFORCE MANDATORY FIELDS: No
- COPY EMPTY FIELDS: (unchecked)
- CREATE NEW RECORD ON EMPTY COALESCE FIELDS: ✓
- APPLICATION: Global (disabled)
- CREATED: (disabled)
- TARGET TABLE: Location [cmn_location]
- ORDER: 100
- RUN SCRIPT: (unchecked)

Select “Mapping Assist”

FileWave	ServiceNow
geolocation_latitude	latitude
geolocation_longitude	longitude
client_filewave_client_name	Name

Once you're finished, save your data, update the Table Transform Map form and continue to next steps.

Part 4: Schedule the Job

Schedule the job

Schedule the import task you just created...recommend at least daily.

System Import Sets > Administration > Scheduled Imports.

https://docs.servicenow.com/bundle/orlando-platform-administration/page/administer/import-sets/task/t_ScheduleADataImport.html

Slack integration with FileWave

Description

In some instances it is desirable to receive automated notifications from devices, perhaps based upon time or hardware conditions changing. The following is a practical example using Slack and is provided as an idea of how IT teams can retrieve information. This particular example is based upon devices reporting if the system drive is filling up, with two thresholds set.



Third Party Software

Slack is a third party application. The details provided are for example only and are unsupported by FileWave.

Requirements

- Slack Account and Webhook
- Provided Fileset - [SlackDiskUsage.fileset.zip](#)

The provided Fileset is for macOS, but similar could be applied to Windows.

Information

Slack updated their Webhooks. For full compatibility the Fileset includes configuration for Legacy Webhooks, which are ignored by the newer App Webhooks.

```
\\"username\\": \\"Helpdesk\\",  
\\"channel\\": \\"#fw_messages\\",  
\\"icon_emoji\\": \":computer:",
```

The App Webhooks are dedicated to a Channel and the name and icon are now set via [Slacks's Website](#)

Details on Legacy Webhooks:

<https://api.slack.com/legacy/custom-integrations/incoming-webhooks>

Directions

Slack Webhook

Slack is a useful tool for communication and the basic option is free. There is also provision for automated posting of messages to a Slack channel, which requires a generated Slack Webhook, as per their guide:

<https://api.slack.com/incoming-webhooks>

Once generated, the Fileset may be configured to implement this Webhook.

Fileset Configuration

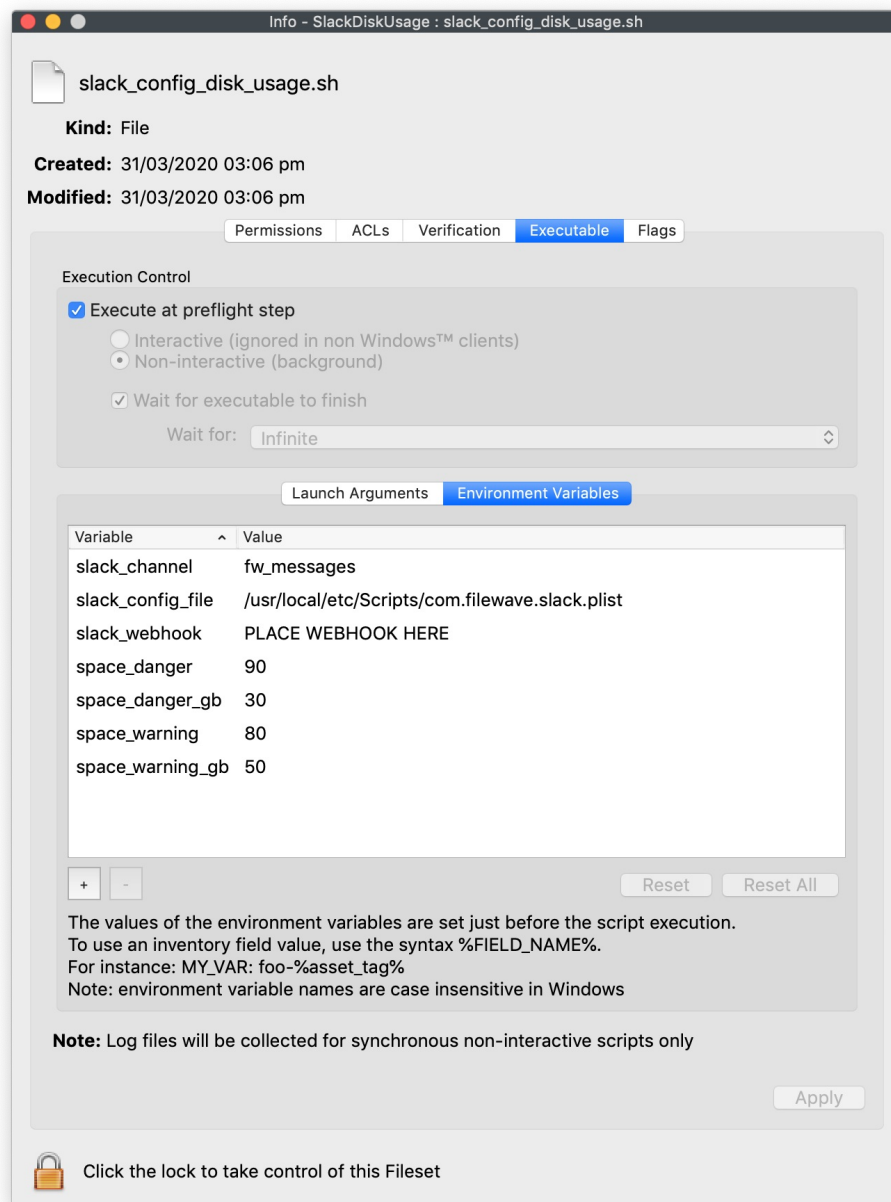
The Fileset has the following components:

- The script to report back to Slack
- A pre-instal script to create a plist configuration file to provide details for the Webhook and desired thresholds for the script
- A LaunchDaemon to trigger the script
- An Activation and Pre-Uninstallation Script, to load/unload the daemon

Plist Configuration File

Once the Webhook has been generated, the Fileset should be edited to match the provided URL by replacing 'PLACE WEBHOOK HERE' with the created Webhook. This needs to be done by:

- Editing the Fileset
- Selecting 'slack_config_disk_usage.sh'
- Choose 'Get Info'
- Select 'Executable'
- Select 'Environment Variables' tab



Similarly, if using Legacy Webhooks, the slack_channel variable needs to be edited to match the name of the created channel; for App Webhooks it will be ignored. In the example, this channel is called 'fw_messages'

There are also two pairs of thresholds. One pair for percentage of disk used:

- space_danger
- space_warning

and a pair for the amount of disk space available (value in GB):

- space_danger_gb
- space_warning_gb

From the example Fileset, a warning message will be delivered if either:

- The percentage used exceeds 80%, the value for space_warning
- The amount of disk space is lower than 50GB, the value of space_warning_gb

However, a more stringent message will be delivered if either:

- The percentage used exceeds 90%, the value of space_danger
- The amount of disk space is lower then 30GB, the value of space_danger_gb

These values should be edited to match desired requirement. Since the script is loaded as a launchd process, once loaded, changes to the script will only be adhered to if the launchd process is stopped and restarted. Providing an additional file, allows for values to be updated in the Fileset which the script will act upon when next triggered; without the need to restart the launchd process.

LaunchDaemon


The LaunchDaemon 'com.filewave.slack_drive_space.plist' is set to run once a week, based upon a random day (Mon-Fri) and a random time (09:00- 18:00). The random nature of the daemon is configured by the Activation Script. Having a random day and time per device, prevents all devices attempting checking in at the same day and time. Default values have been configured, but will be overwritten during Fileset Activation.


Details around the configuration of launchd may be seen at the following resource, should different timings be desired. Note that this would involve editing the Activation Script as well.

<https://www.launchd.info>

Example messages

Warning - Percentage drive space outside of threshold range and available disk space between Thresholds of 20GB and 40GB:

 **Helpdesk** APP 11:05 AM

 **Warning: 1015VMDEP**

System drive is low on space


Serial Number: CV02Z90SDLMNN


Drive Space Used: 30%

Available: 23GB

Thresholds 80% 20GB | Today at 11:05 AM

Danger - Percentage drive space outside of threshold range and available disk space lower than Thresholds of 30GB and 50GB:

 **Helpdesk** APP 11:05 AM

 **Warning: 1015VMDEP**

!!System drive space low!!

Serial Number: CV02Z90SDLMNN

Drive Space Used: 30%

Available: 23GB

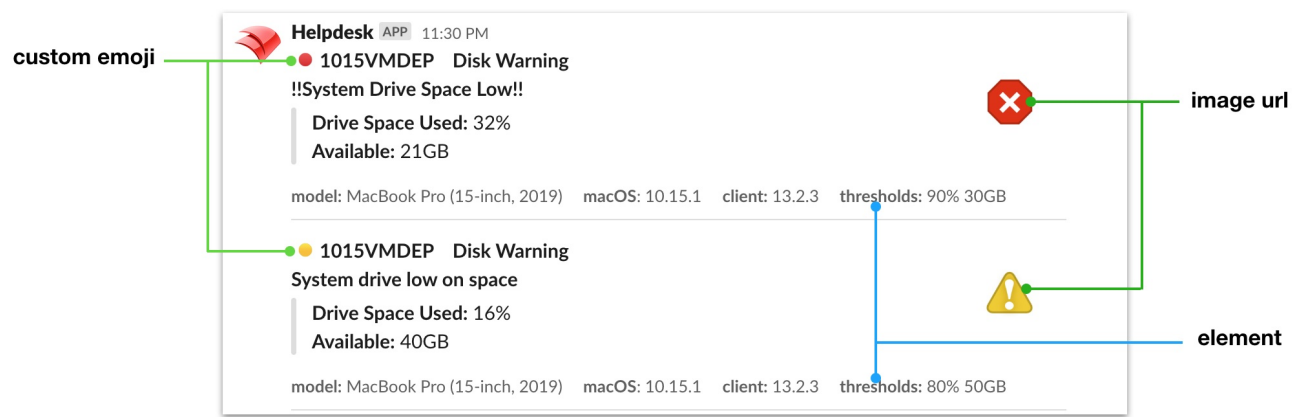
Thresholds 80% 30GB | Today at 11:05 AM

For self designing messages, the above main scripts were based upon [Slack's Secondary Attachments](#). Although Slack have only deprecated some items, indication is that all 'attachments' should not be relied upon and to consider their new Block feature.

<https://api.slack.com/reference/messaging/attachments>

Block Kit

Block Kit is really no different to 'attachments' and where some features are already incompatible with App Webhooks, others have been included through Block Kit. A similar look may be achieved:



For example,

- The deprecated attachment 'icon_emoji' may no longer be updated by a message and the icon is set through the settings for the App Webhooks. However, it is possible to add an image via a URL. Images may be added to the thread and then referenced by their URL; as the above example. Alternatively, numerous App Webhooks could be created with differing icons and the relevant Webhook could be referenced in each script.
- The attachment 'footer' can be replaced with an 'elements' block
- The 'mrkdwn' for Block Quotes does not allow for colouring the vertical line
- A 'divider' line may now be included to more clearly separate messages

Further details on designing text within Blocks:

<https://api.slack.com/reference/surfaces/formatting>

To reference images uploaded to Slack, they must be made publicly available. A public link should be of the format:

```
https://slack-files.com/{team_id}-{file_id}-{pub_secret}
```

The direct link to be referenced in a script has the format:

```
https://files.slack.com/files-pri/{team_id}-{file_id}/{filename}?pub_secret={pub_secret}
```

An example link may look like:

```
https://files.slack.com/files-pri/ABU1BH39Z-F0117AF5RRB/exclamation.jpg?pub_secret=5c56df27a5
```

Example json from the above. Variables have been replaced with values to allow for immediate testing:

```
json="{
  \"blocks\": [
    {
      \"type\": \"section\",
      \"text\": {
        \"type\": \"mrkdwn\",
        \"text\": \":red_circle:*1015VMDEP\\tDisk Warning*\n*!!System Drive Space
Low!*\\n>*Drive Space Used:* 32%\\n>*Available:* 21GB\\n>\"
      },
      \"accessory\": {
        \"type\": \"image\",
        \"image_url\":
        \"https://api.slack.com/img/blocks/bkb_template_images/beagle.png\",
        \"alt_text\": \"alt text for image\"
      }
    },
    {
      \"type\": \"context\",
      \"elements\": [
        {
          \"type\": \"mrkdwn\",
          \"text\": \"*model:* MacBook Pro (15-inch, 2019)\\n>\"
        },
        {
          \"type\": \"mrkdwn\",

```

```

        \text\: \*macOS*: 10.15.1\
    },
    {
        \type\: \mrkdn\,
        \text\: \*client:* 13.2.3\
    },
    {
        \type\: \mrkdn\,
        \text\: \*thresholds:* 90% 30GB\
    }
]
},
{
    \type\: \divider\
}
]
}"

```

Conclusion

The above shows how a Fileset may be delivered to devices in tandem with 3rd party tools, such that they will automatically report disk space usage at timed intervals. However, this is just an example framework and the scope for use is endless.

Truce Family integration with FileWave



What

TRUCE Family on FileWave is a dynamic device management solution designed to balance smartphone use in schools and at home. It allows schools and parents to customize and control access to apps and services based on location, activity, and time, ensuring that smartphones can be used for learning while minimizing distractions. By integrating TRUCE with FileWave, administrators can implement policies that support student focus in the classroom and promote responsible phone use outside of school hours.

When/Why

Smartphone distraction has become a major challenge in educational environments, with 75% of parents worried about excessive phone usage and 72% of high school teachers citing smartphones as a significant distraction. Traditional methods like locking phones away don't provide the flexibility needed in modern learning environments. TRUCE Family on FileWave provides the contextual control schools and families need to manage phone use effectively, without completely restricting access to critical communication or educational apps. This solution promotes focus and safety, making it ideal for schools that want to foster productive learning environments and for parents who want to encourage healthy digital habits.

How

To implement TRUCE Family on FileWave, follow these steps:

1. Install TRUCE on student devices: Using FileWave, deploy TRUCE Family onto student devices through an MDM (Mobile Device Management) profile, ensuring compliance with the school's cell phone policy.
2. Configure contextual policies: Use the TRUCE management console to define rules based on location (e.g., classrooms or homes), activity (e.g., school hours), or time (e.g., during classes) to enable or restrict certain apps and device functionality.
3. Monitor and adjust settings: TRUCE policies can be dynamically adjusted as needed. For example, allow only educational apps during school hours while enabling full phone access after school. All changes can be managed remotely via FileWave.

<https://www.youtube.com/embed/0csmV-eF20k>

Related Content

- [FileWave and Truce](#)

Digging Deeper

TRUCE Family offers extensive customization options, allowing schools to maintain a balance between accessibility and focus. By giving parents and educators the tools to define appropriate phone usage, students can develop responsible digital habits that carry over outside the classroom. For those interested in more advanced configurations, TRUCE Family also integrates with other FileWave features like reporting and remote support, ensuring smooth implementation and ongoing management.

OTRS integration with FileWave



What

The integration between OTRS and FileWave unifies IT ticketing and multi-platform device management into a single, efficient platform. This collaboration enhances automation, reduces manual processes, and ensures real-time data accuracy, all while strengthening security and compliance across your organization's devices.

When/Why

Use this integration when you need to streamline your IT processes, manage devices more efficiently, and enforce consistent security policies. It's especially beneficial for organizations looking to comply with evolving regulations like NIS-2 and DORA, as it simplifies policy enforcement and provides robust security features.

How

To leverage the OTRS and FileWave integration, synchronize your device and asset data between the FileWave Management Suite and OTRS's Configuration Management Database (CMDB). Utilize single sign-on (SSO) authentication to seamlessly navigate between systems, allowing your IT team to manage devices directly from OTRS and automate repetitive tasks for increased efficiency.

Related Content

- [OTRS and FileWave Integration Solution Page](#)
- [Press Release: FileWave and OTRS Join Forces](#)

Digging Deeper

By combining FileWave's expertise in multi-platform device management with OTRS's advanced service management solutions, this integration offers a holistic approach to IT operations. Real-time monitoring, automated patch deployments, and features like remote wipe and encryption bolster your organization's security posture. This unified platform not only saves time and resources but also enhances the quality of IT services, allowing your team to focus on strategic initiatives rather than routine tasks.