

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 \(v15.5+\)](#)
- [Using AutoPkgr with FileWave for Advanced macOS Software Deployment](#)
- [Autopkg\(r\) FAIL_RECIPES_WITHOUT_TRUST_INFO](#)

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.

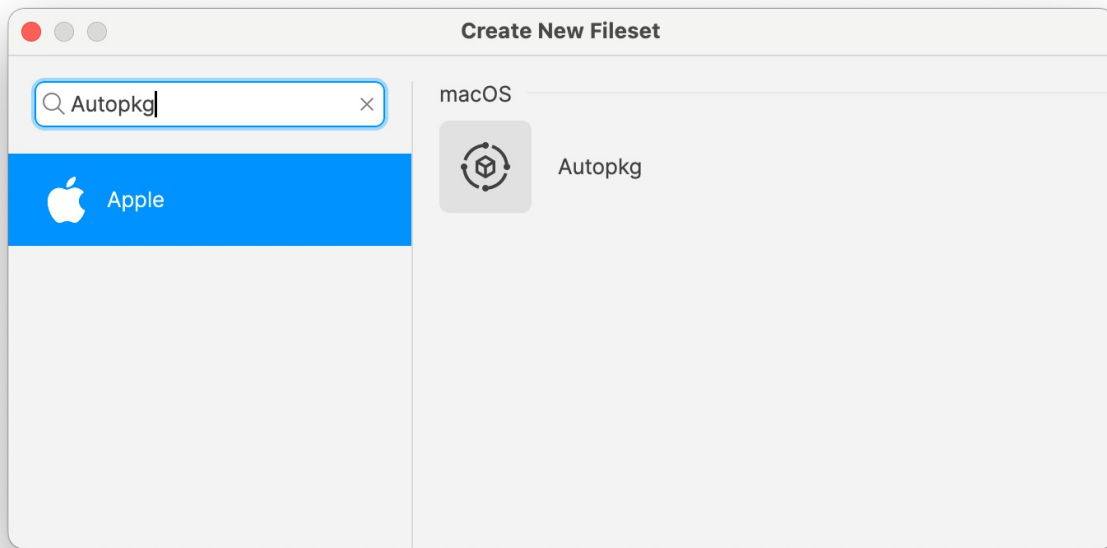
Security is also enhanced by using known repositories, preventing the introduction of unverified or malicious software into your environment. The reduced complexity lowers the learning curve associated with AutoPkg, making package deployment accessible to administrators with varying levels of experience.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

How

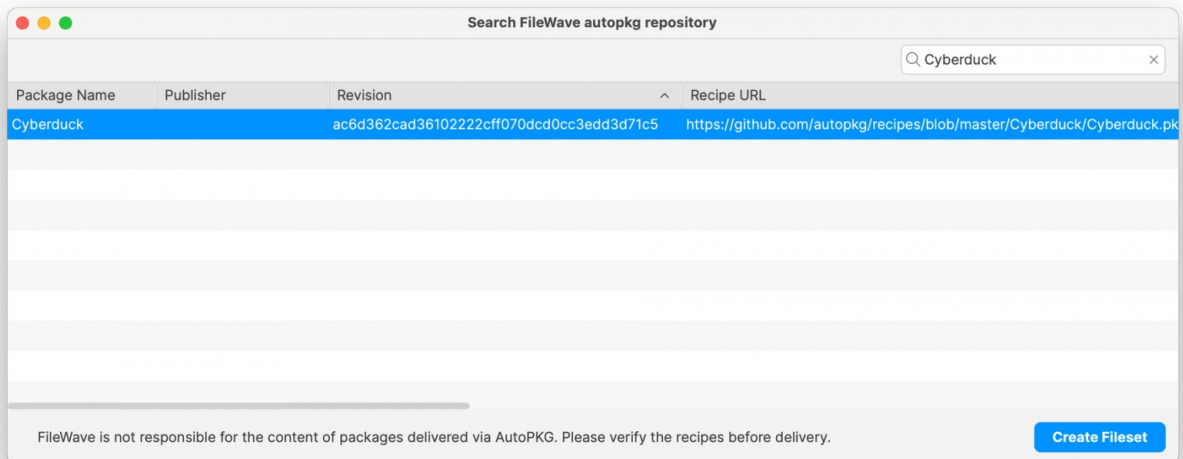
Creating an AutoPkg Fileset:

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.

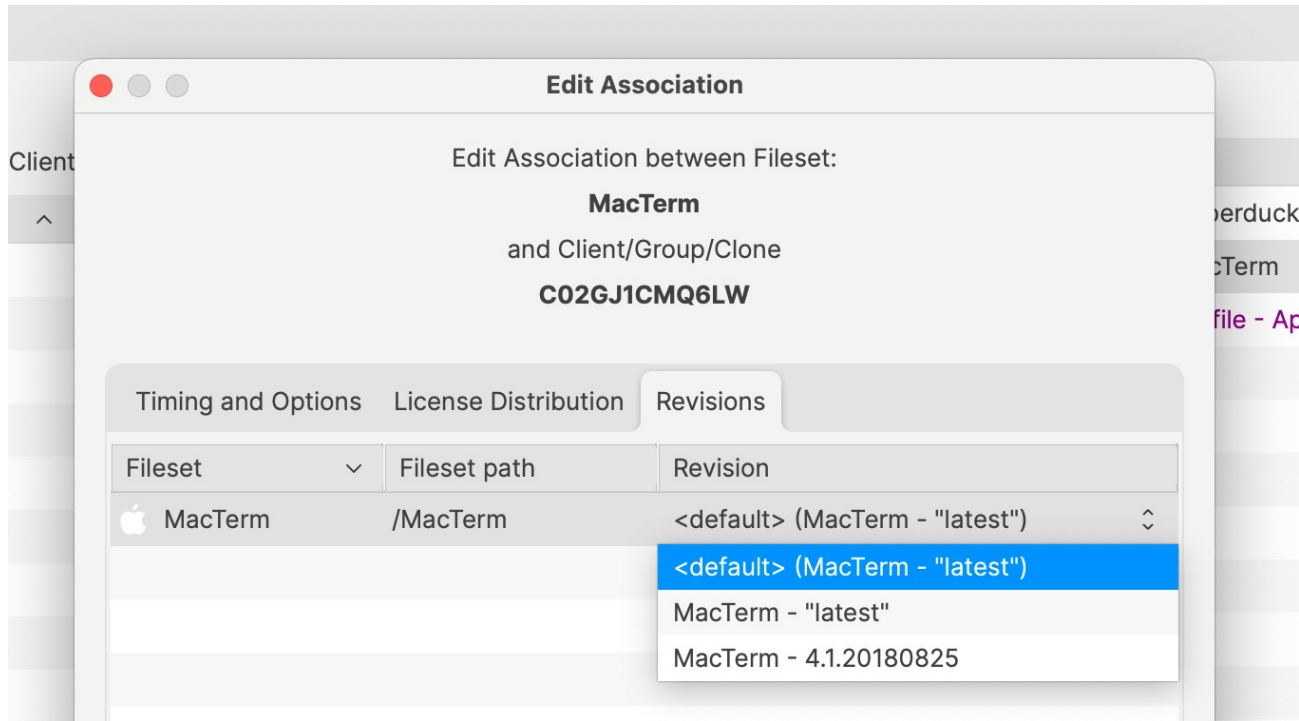
- Create a New Fileset, and select Apple so that you can pick Autopkg as the type:



- 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.

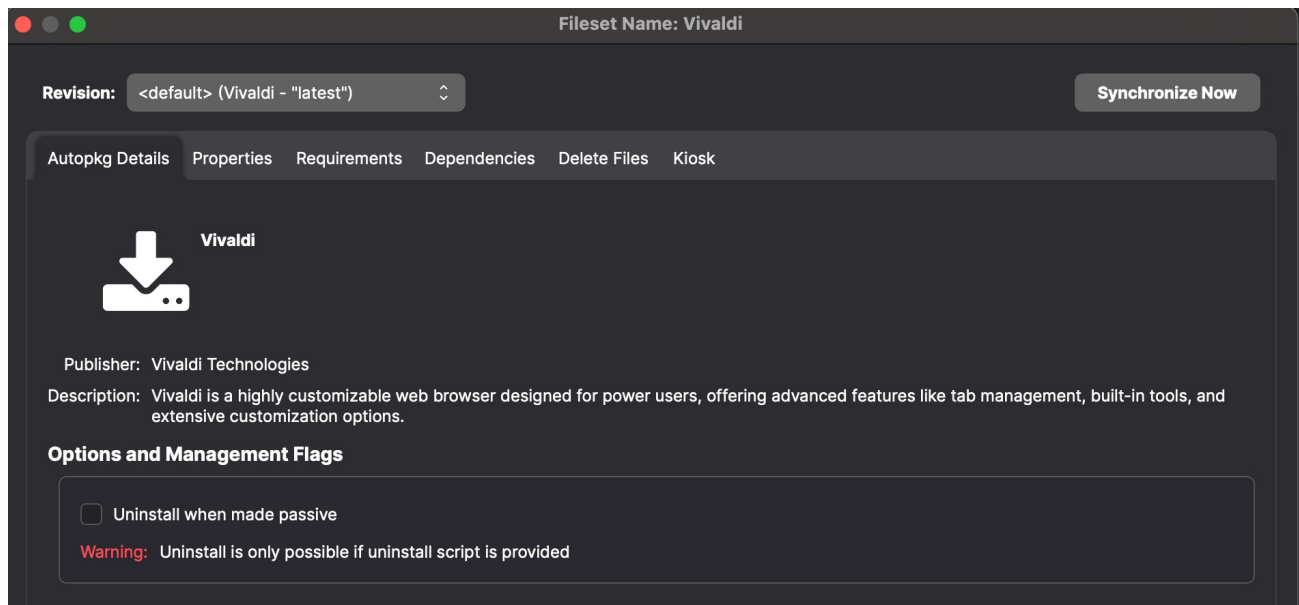


- Create an Association or Deployment. Selecting "latest" ensures upgrades can occur automatically. Otherwise, specify a specific version to stay at that version until later changed.



- 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

- [AutoPkgr with FileWave](#)

Using AutoPkgr with FileWave for Advanced macOS Software Deployment

Description

AutoPkgr 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.

AutoPkgr 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 AutoPkgr feature for simplified package creation (as detailed in our new article “[Integrated AutoPkgr \(v15.5+\)](#)”), power users seeking advanced functionality may prefer using the full AutoPkgr and AutoPkgr tools. This article focuses on leveraging AutoPkgr 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

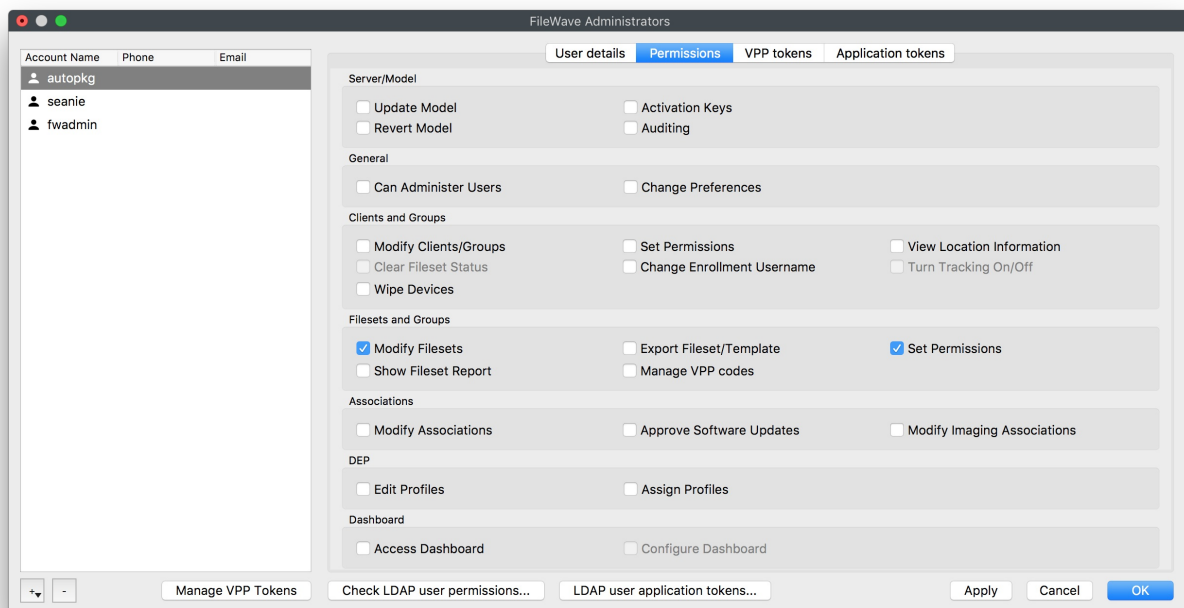
- FW Admin
- AutoPkgr Installer
- An 'always on' computer

Directions

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

Setup FileWave

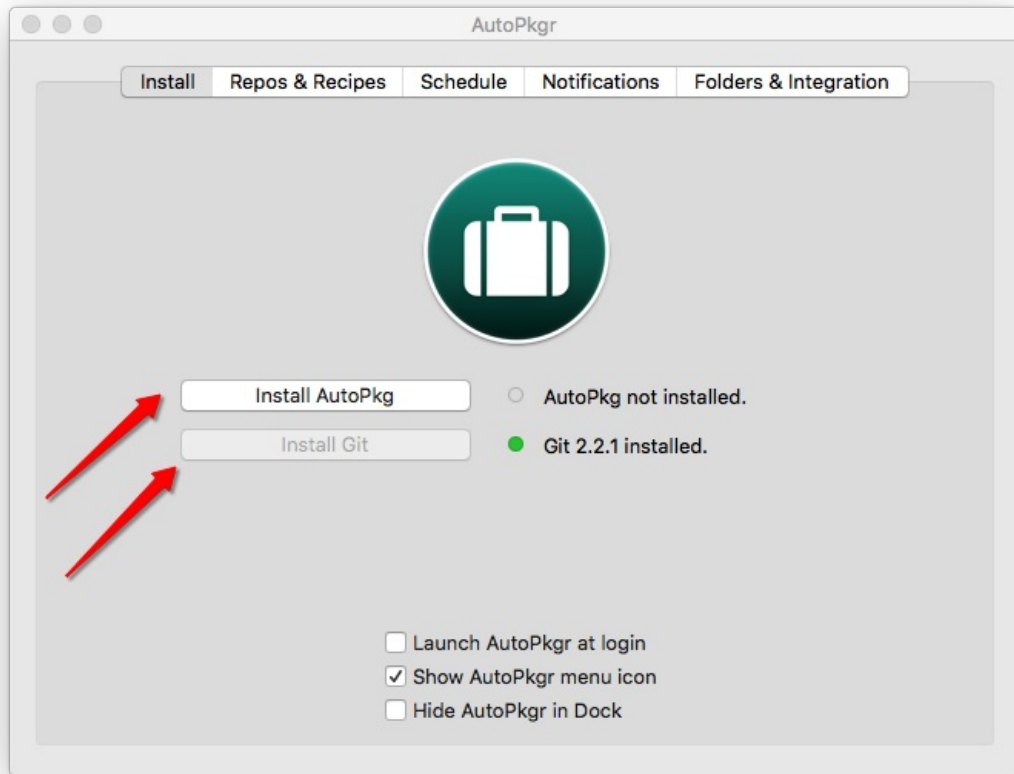
- Go to FileWave Admin -> Assistants -> Manage Administrators



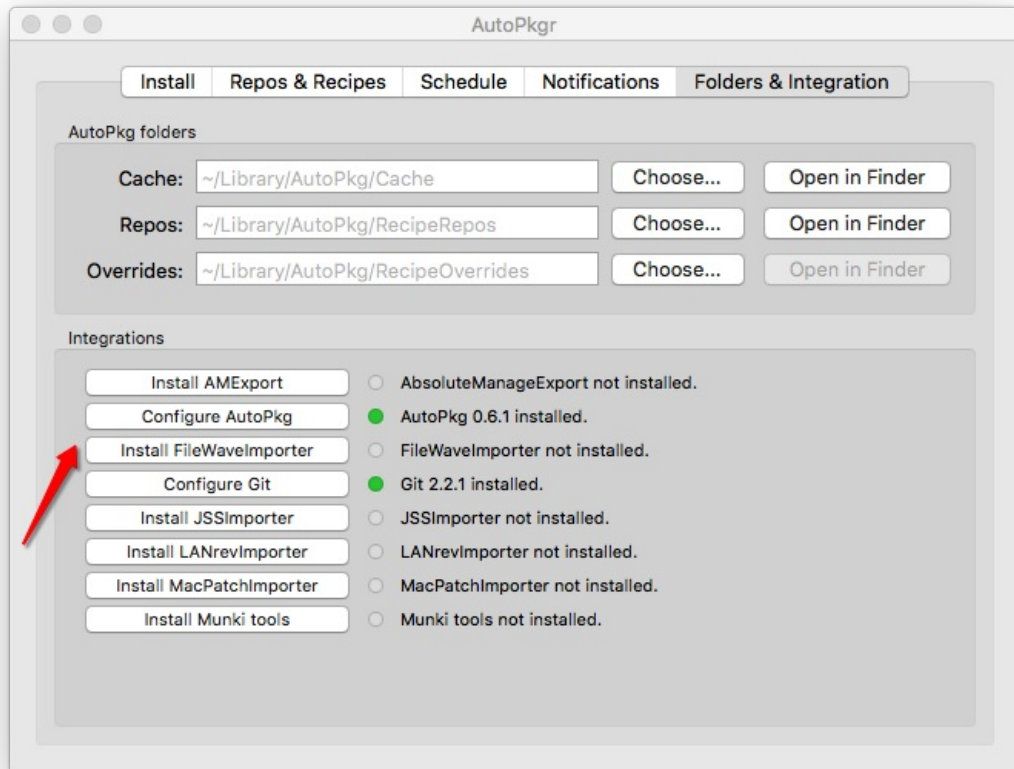
- Click on the + button to add a new Administrator
Select Local Account, (for example autopkg and the password autopkg as well)
- Go to Permissions tab and click on Select None
- Allow the autopkg user to 'Modify Filesets' and 'Set Permissions' as per the above screenshot.
- Click Apply
- Confirm with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

Setup AutoPkgr

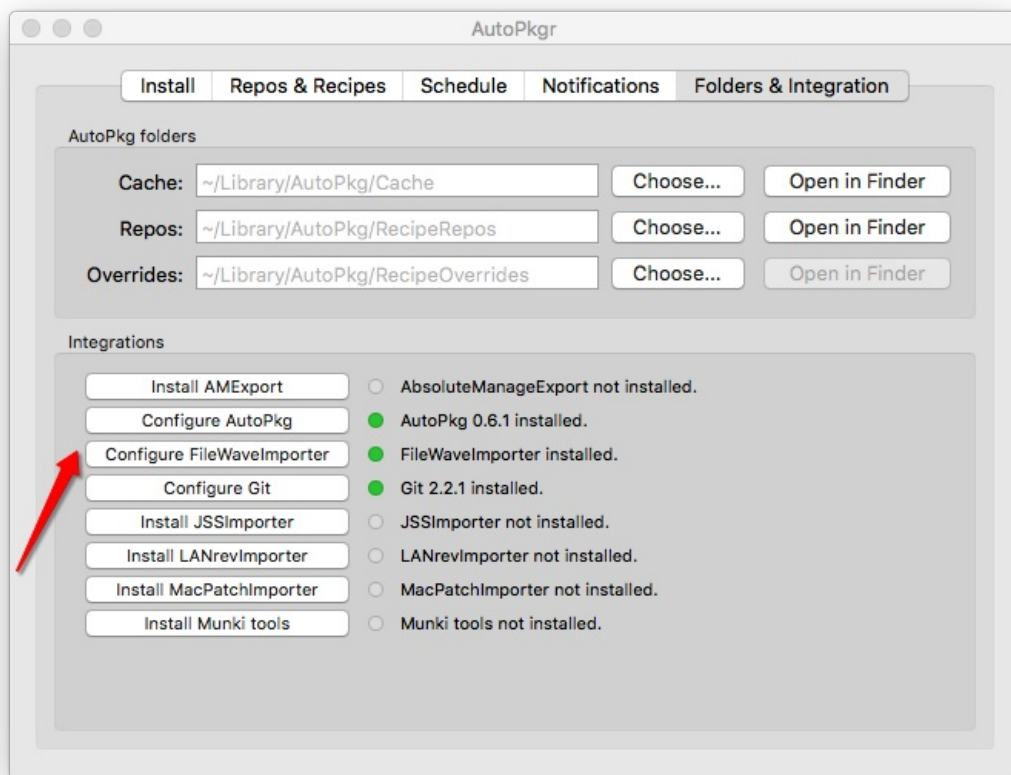
- Go to <https://github.com/lindegrouper/autopkg/releases/latest>
- Download, install and launch AutoPkg
- Launch AutoPkg, Click on "Install AutoPkg" , and "Install Git"



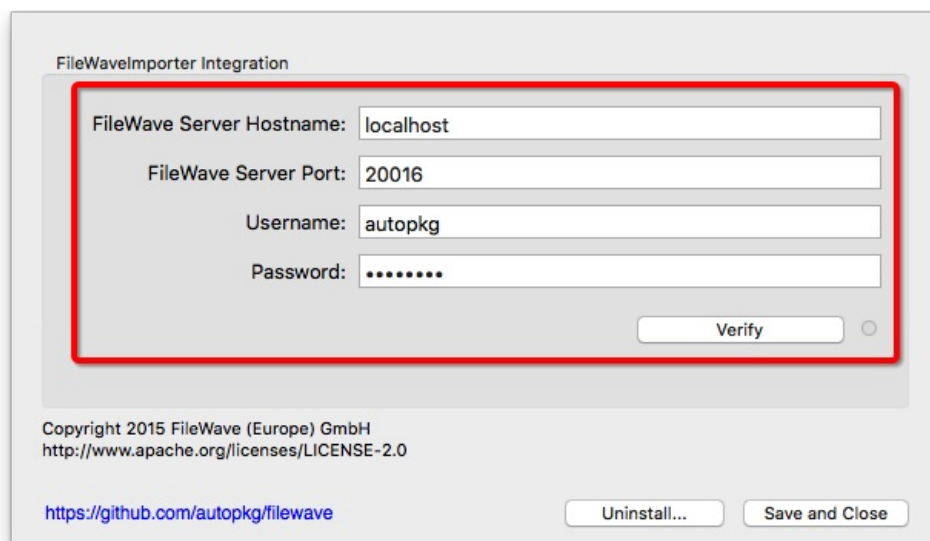
- Go to Folders & Integration and click on Install FileWaveImporter:



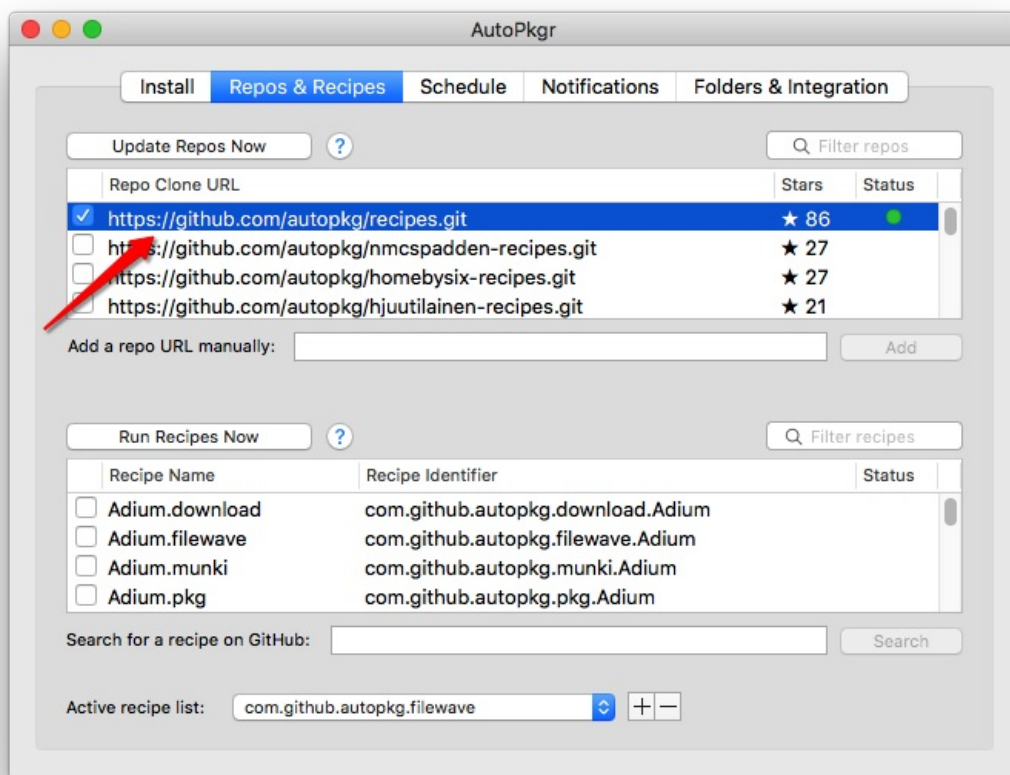
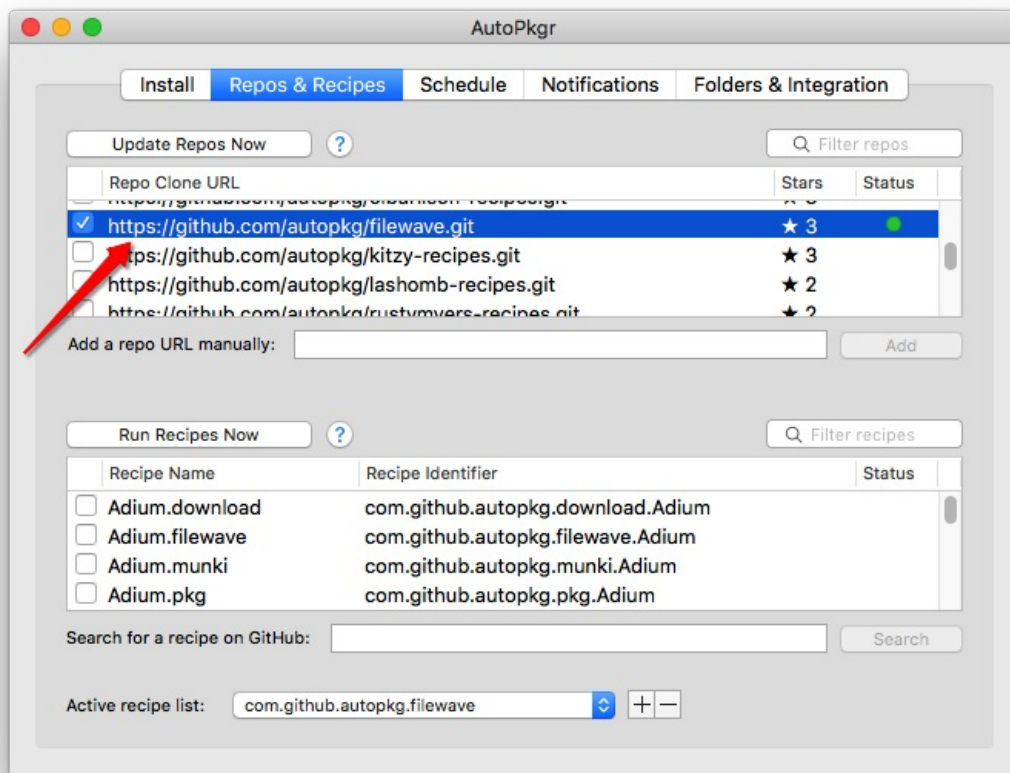
- Click on Configure FileWaveImporter:



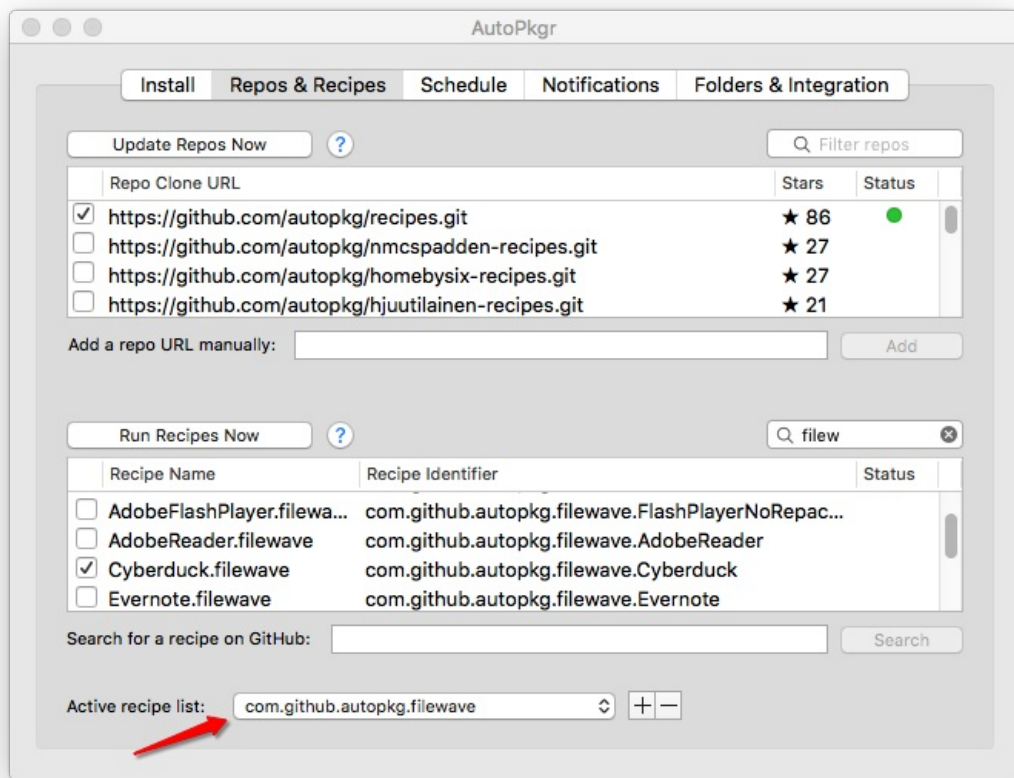
- Enter your FileWave Server Hostname



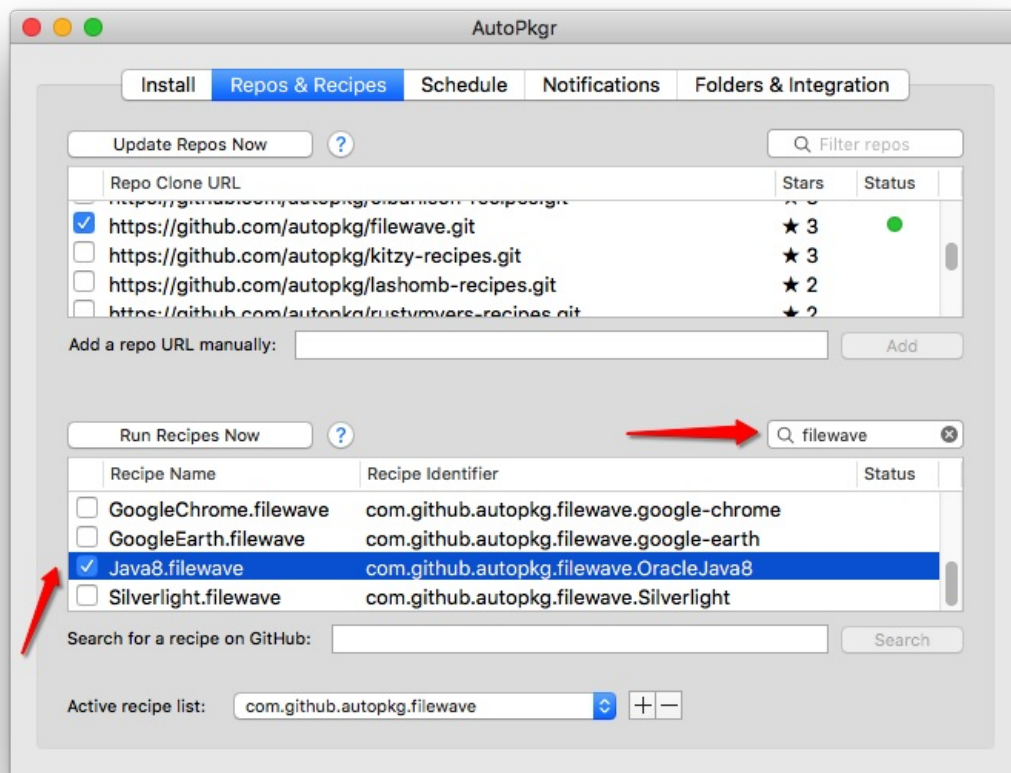
- FileWave Server Port is already set to 20016
- Username is e.g. autopkg
- Password is e.g. autopkg
- Click on Verify to validate the setup
- Click on Save and Close
- Go to Repos & Recipes and verify that <https://github.com/autopkg/recipes.git> and <https://github.com/autopkg/filewave/git> are checked



- 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:

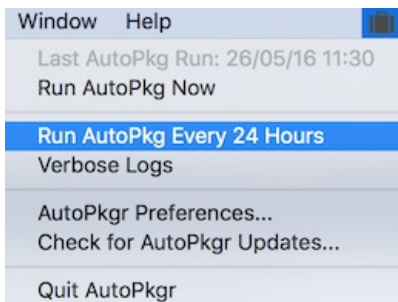


Security and Trust Relationship

- Running recipes directly from a cloned repo will bypass AutoPkgr'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

 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
```

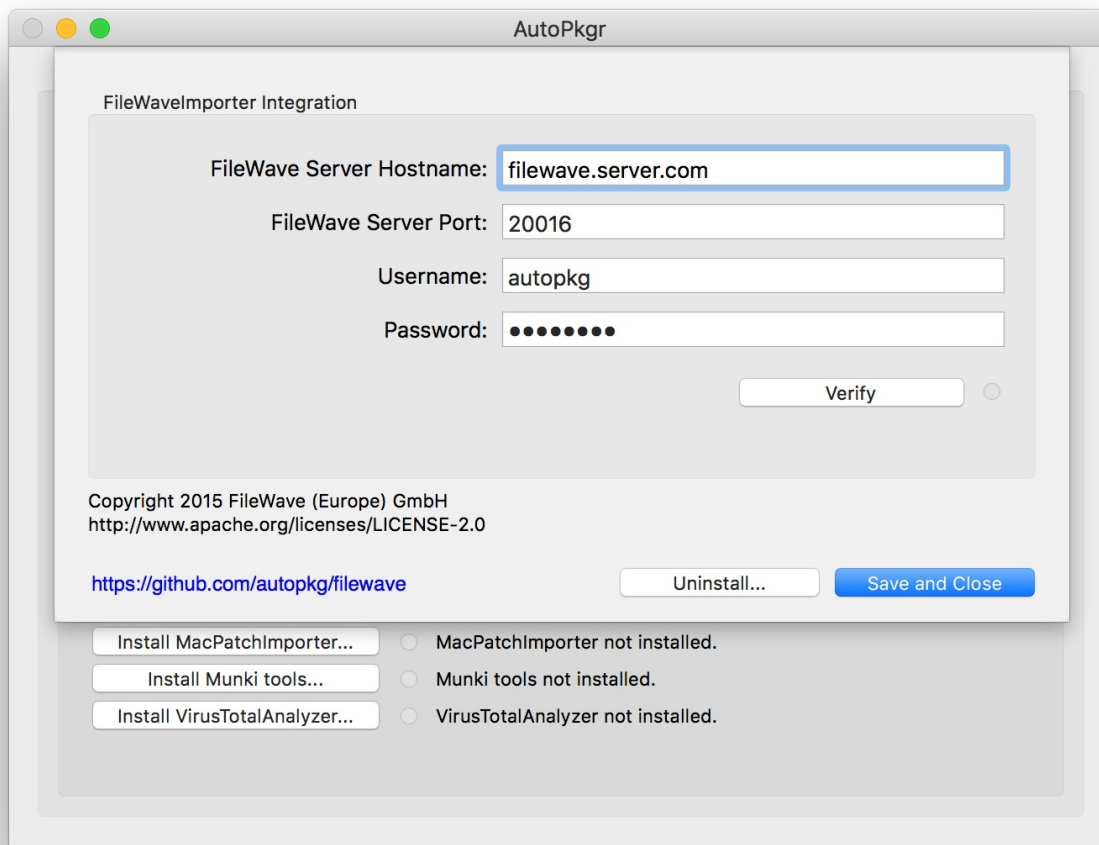
User settings

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:

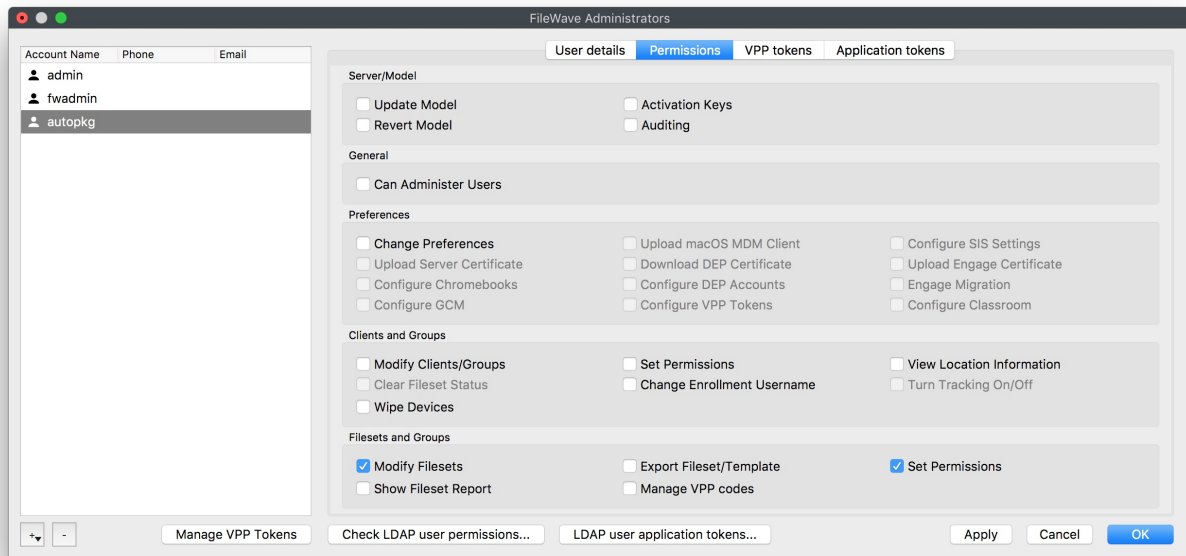
- login
- System

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.