

# Filesets - A Closer Look

Contents, Properties, Settings, Revisions and Dependencies

- [Filesets View](#)
- [Desktop File Level Filesets](#)
- [Desktop PKG and MSI Filesets](#)
- [Verification](#)
- [Windows Registry](#)
- [Apple VPP Apps](#)
- [Enterprise Filesets and Documents \(iOS only\)](#)
- [Apple Profile Filesets](#)
- [Android Apps](#)
- [Fileset Properties](#)
- [Dependencies](#)
- [Revisions](#)

# Filesets View

Creating a Fileset and deploying to clients can be relatively simplistic, e.g. PKG, MSI or VPP. However, custom configuration through properties, provides tremendous flexibility.

The standard Fileset view provides some basic details:

Dashboard	Name	Size	Version	Files	Modified	ID	Comment	VPP Token
Clients	Android Grammar					690		
Filesets	APK - German Lite	4 MB	1	1		605		
Associations	APK - LearnEnglish Grammar	10.1 MB	1	1		607		
Imaging	APK - FileWave Client	7.9 MB	2	1		427		
IOS Inventory	EMCO - MSI Package Builder Enterprise	23.4 MB	2	42		431		
License Management	FileWave Engage Windows Client 10.0.0.14548264	29.7 MB	2	11		407		
Inventory Queries	FileWave OSX Client 10.0.0.1454826	49 MB	2	88		250		
	FWWinClientUpgrade_10.0.0.14548264	28.6 MB	3	4		421		
	IOS App - Google Earth		0		Modified	689		

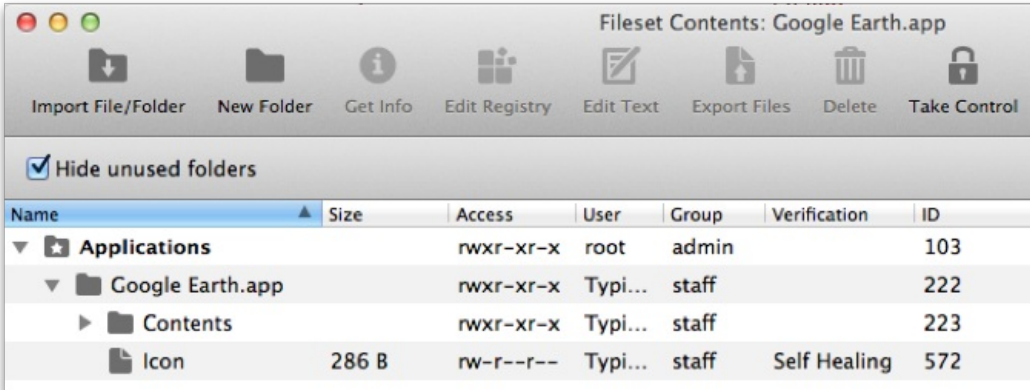
Name	Fileset Title
Size	Fileset size, as stored on FileWave Server and Boosters or Clients when transferred.
Version	Each update to a Fileset will increment its version number after the Model Update
Files	Total number of files stored within the Fileset
ID	A unique identifier used by FileWave to track each Fileset
Comment	User customisable text for viewing in FileWave Admin software only
VPP Token	Apple Volume Purchase Program token name
Default Revision	Indicates the default revision to be deployed, where a Fileset contains multiple revisions
Revision Count	Highlights the amount of revisions within each Fileset.

Each Fileset, though, has additional, customisable features, which vary depending upon the Fileset type, e.g. Apple VPP, Windows Registry or File Level, as covered in the other pages of this chapter.

# Desktop File Level Filesets

Contents of a file level Fileset will be delivered to devices, with matching permissions and ownership. The entire directory structure of the Fileset will be copied, with any directories created if not yet existing.

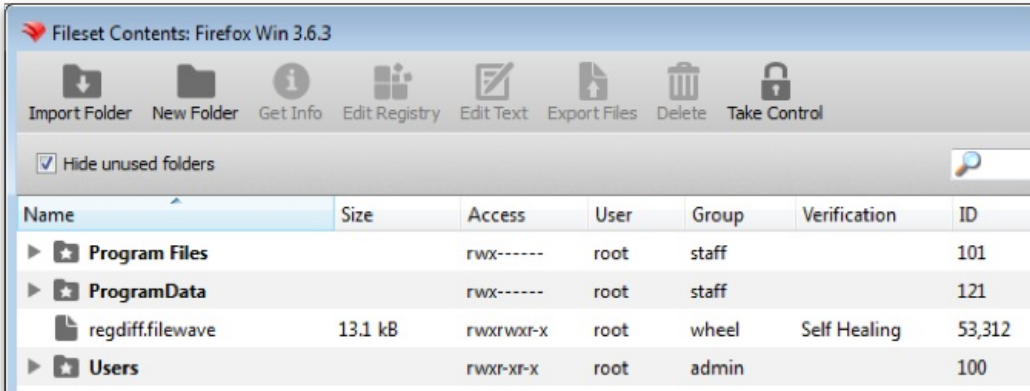
## Example 1



The screenshot shows a window titled "Fileset Contents: Google Earth.app". It has a toolbar with buttons: Import File/Folder, New Folder, Get Info, Edit Registry, Edit Text, Export Files, Delete, and Take Control. Below the toolbar is a checkbox labeled "Hide unused folders" which is checked. The main area contains a table with the following data:

Name	Size	Access	User	Group	Verification	ID
▼ Applications		rwxr-xr-x	root	admin		103
▼ Google Earth.app		rwxr-xr-x	Typi...	staff		222
▶ Contents		rwxr-xr-x	Typi...	staff		223
Icon	286 B	rw-r--r--	Typi...	staff	Self Healing	572

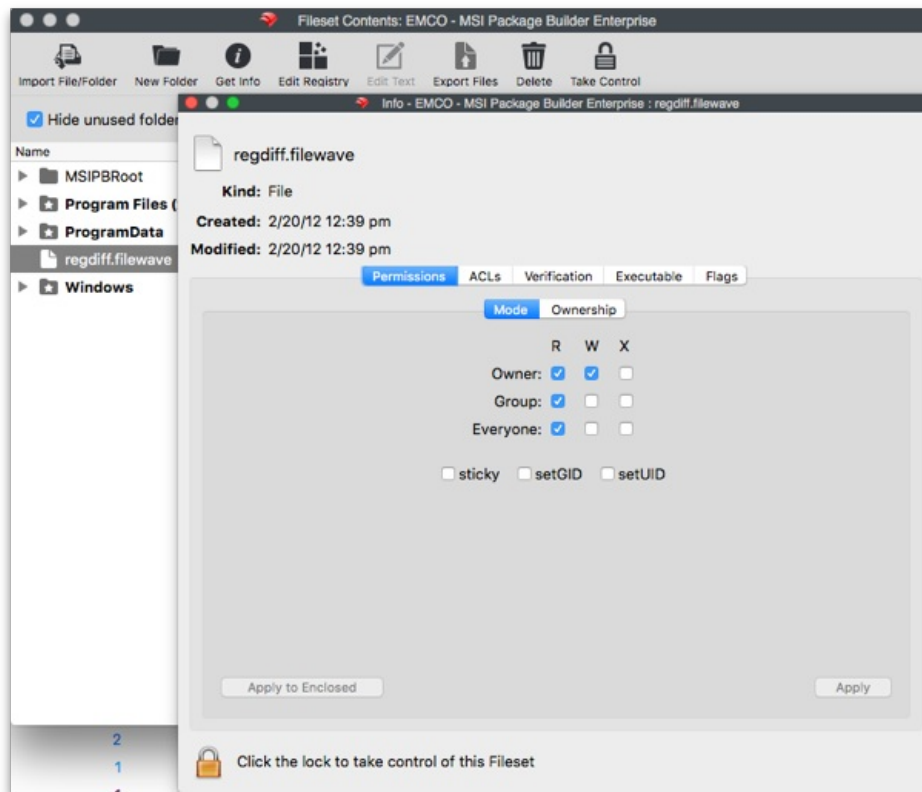
## Example 2



The screenshot shows a window titled "Fileset Contents: Firefox Win 3.6.3". It has a toolbar with buttons: Import Folder, New Folder, Get Info, Edit Registry, Edit Text, Export Files, Delete, and Take Control. Below the toolbar is a checkbox labeled "Hide unused folders" which is checked. The main area contains a table with the following data:

Name	Size	Access	User	Group	Verification	ID
▶ Program Files		rwx-----	root	staff		101
▶ ProgramData		rwx-----	root	staff		121
regdiff.filewave	13.1 kB	rw-rw-r--	root	wheel	Self Healing	53,312
▶ Users		rwxr-xr-x	root	admin		100

Additional items may be included by way of either the New Folder or Import Folder buttons. Files and folders may also be added by way of drag and drop. Other attributes may be observed and altered through the Get Info button, when an item is selected.



Beyond ownership and permissions, further options are available also, for example, ACLs, Verification and file execution (scripts and EXE files). Verification has its own page in this chapter and is important to understand.

A selected file's contents may be altered using the Edit Text option. This is particularly useful for scripts, but can apply to many file types, as set within the Preferences > Editor tab.

- ✓ Files may be selected and exported from the Fileset to the local device.



# Desktop PKG and MSI Filesets

## What

Special Filesets may be created for PKG and MSI installers. These will automatically instal the software on deployment activation.

## Why

It is usual for Admins to just simply require PKGs or MSIs to instal on activation. To simplify the process, FileWave can automatically create a special Fileset to handle this process.

## How

Using the New Fileset option within the FileWave Central Admin App, it is possible to choose PKG or MSI Fileset. From the pop-up view, select the installer and FileWave will create this auto installing Fileset.

More easily, if a PKG or MSI installer file is dragged into the FileWave Central Fileset view, the same automated Fileset should be created, without the need to use the New Fileset view.

Dragging the installer file onto a desired Fileset Group will cause this new Fileset to be created within that group.

Consider renaming the installer before upload, such that the version number is not included. Instead, add the name and version number to the revision name. The below section on revisions should make this clear why this would be a good idea.

## Information

Opening a PKG or MSI Fileset will show the contents similar to the below:

Name	Size	Access	User	Group	Verification	ID	Modification Date
usr		rwxf-rf-x	root	wheel		304	
local		rwxf-rf-x	root	wheel		305	
etc		rwxf-rf-x	root	wheel		1334	
FileWaveInstallers		rwxf-rf-x	root	wheel		1335	
FlatPackage-Firefox.pkg		rwxf-rf-x	root	wheel		791200	
Firefox.pkg	158.8 MB	rw-r--r--	Typical User ID 501	wheel	Ignore At Verify (Left Behind)	791201	2025/2/12 14:54

From creation, this special Fileset is expecting an installer of the named PKG or MSI from initial upload to the server.

If manually altering a PKG or MSI Fileset, to upload a newer version of an installer, it is therefore necessary that the replaced installer file MUST have the same name as the installer being replaced.

## Revisions

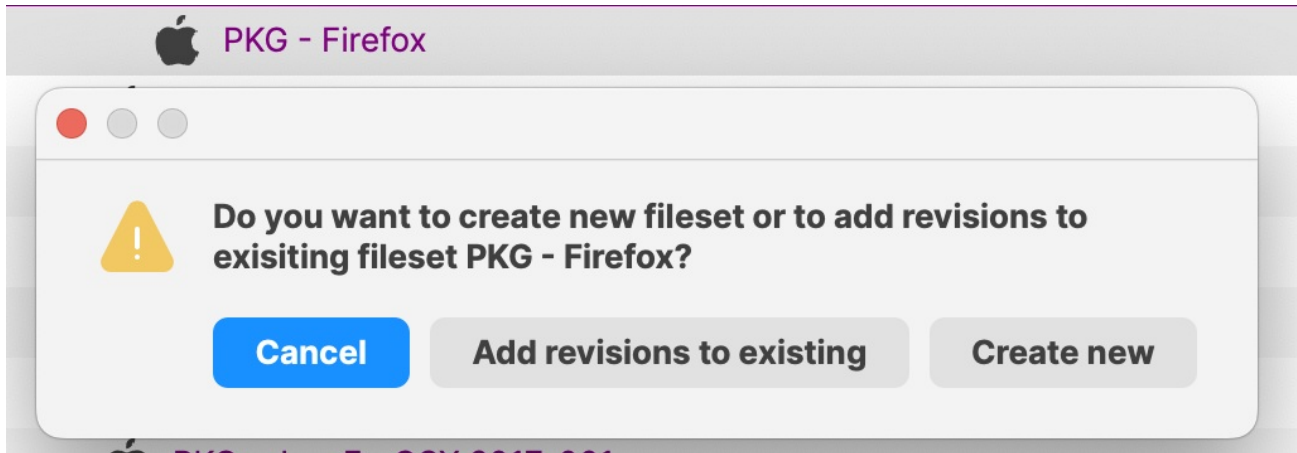
In some ways, revisions simplify the process of updating the Fileset with newer installers.

There are two ways to create a new revision:

- Drag and drop a new installer onto the existing Fileset
- Open the existing Fileset, chose the option to Manage Revisions and from the button to add a new revision, choose to duplicate

## Drag & Drop

When dragging a PKG installer onto a current PKG Fileset or an MSI onto a current MSI Fileset, a prompt similar to the below should be shown:



Choosing the option to 'Add revisions to existing' will automatically generate a new revision with this new installer. Assigning this new revision to devices or setting this new revision as the new default should then trigger this new installer to activate on devices.



After the new revision is created, consider opening the Fileset to set the name of the new revision, such that the name includes the version number of the installer.

## Duplication

When choosing to Manage Revisions from within the Fileset, there is the option to duplicate a revision of choice. On doing so, everything from the prior Revision can be copied over to this new revision.

Fileset Revision Management		
Name	Comment	ID
<default> (Initial Revision)		736784
PKG - Firefox 135.0		736786

+
-
Edit
Set as default
Cancel
Save

### New Fileset Revision

Name:

Comment:

☐ Empty revision

☒ Duplicate existing revision:

Cancel
OK

This is useful when there are multiple additional items added into a PKG or MSI Fileset. However, there is an important consideration when using this method.

As highlighted above, when manually replacing a PKG or MSI installer, the installer name with the Fileset must match. Hence, duplicating a revision, with the idea of replacing the current installer with a new version, whilst maintaining additional files, forces this requirement.

- Before creating the first Fileset for a given installer, rename the installer to exclude the version number before uploading. Instead, add the version number into the name of the revision instead.

- The name of the installer to be uploaded does not need to match, but the name of the installer within the Fileset must match. If the name of the installer file does not match before upload, 'copy' the name from within the Fileset, upload the new installer into the same folder as the original installer, delete the original installer and rename the new installer using 'paste'.

- If the name of the initial installer uploaded includes the version number, when manually altering the revision, since the name must match, the newly uploaded version would have to be misnamed with a version number that does not make sense.

- File Level Filesets (including PKG and MSI Filesets) can be exported for transfer to another FileWave server. They can also be

compressed and stored for future use or archived.



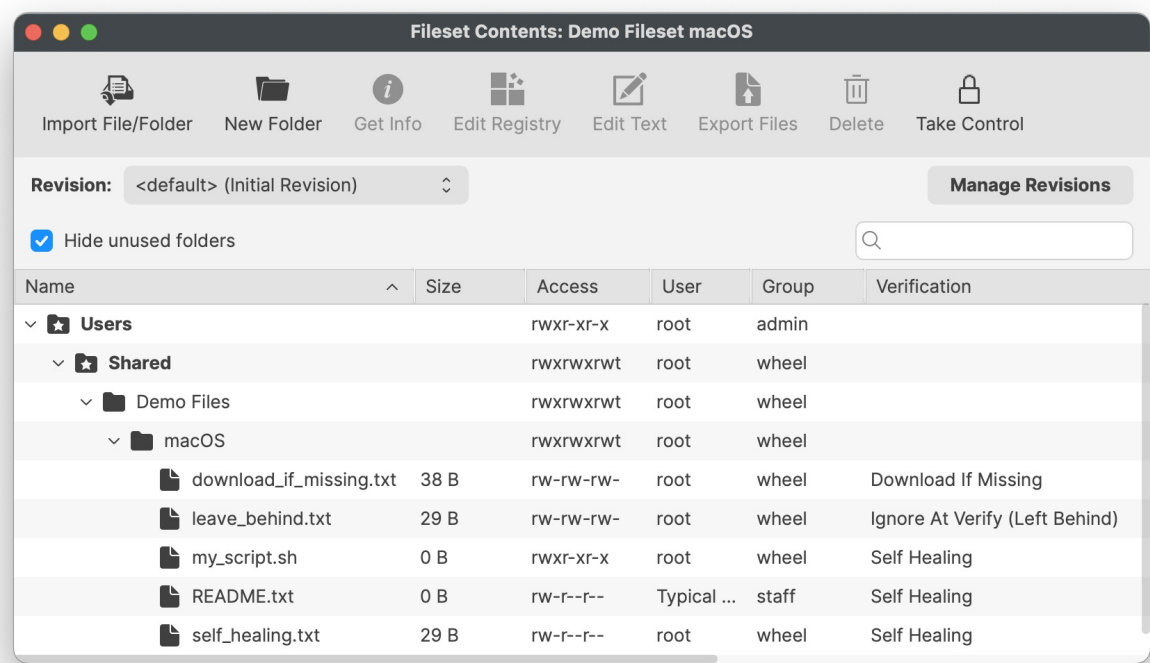
Most installers only handle upgrade, not downgrade. If there is a desire to rollback to an older revision, it is likely the currently installed version on the device will need to be deleted beforehand.

## Related Content

- [Fileset Revisions](#)

# Verification

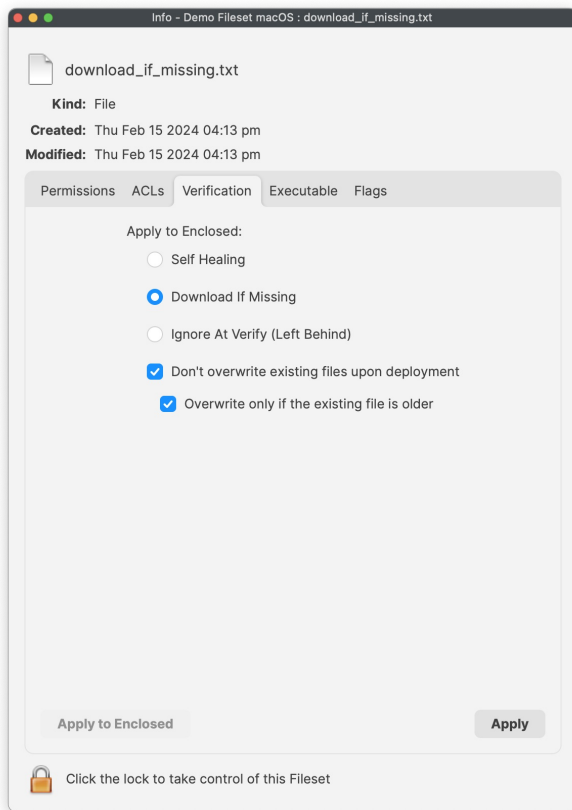
There are three primary verification settings. Each of these settings causes the related file(s) to behave differently once deployed.



Self Healing	Files set as self-healing will be replaced on devices with those within the Fileset, if different, or added if missing. The act of Verification re-confirms all included files.	On disassociation, Files included in the Fileset will be removed from the device
Download if Missing	Similar to self-healing, however, the file will only be added if the file is missing. After initial deployment, the file may be altered, without repair occurring at Verification	On disassociation, Files included in the Fileset will be removed from the device
Ignore At Verify (Left Behind)	Once delivered, the file may be altered.	On disassociation, delivered files will remain on the device.

A couple of additional options are available:

Don't overwrite existing files upon deployment	If enabled, only deliver the file if it does not exist.
Overwrite only if the existing file is older	A sub option to the overwrite option, only replace the current file on the device, if that existing file has an older timestamp than the matching file in the Fileset.

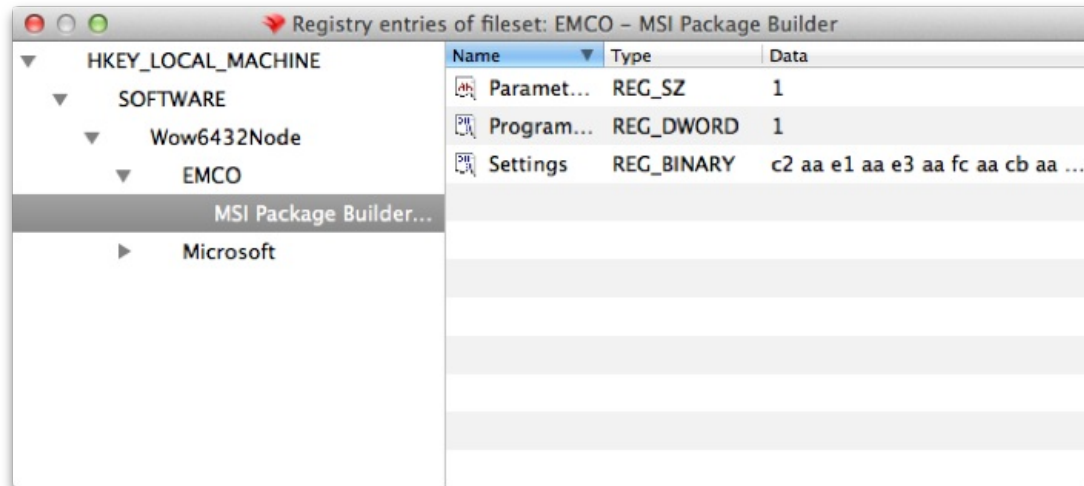


File comparison is by name and modification date.

- Any files that already exist between one Fileset and another, will not be transferred between server and client since they already exist as required; with the exception of the overwrite option on initial deployment.

# Windows Registry

Windows registry files may be included within a Filesets and edited within FileWave.



# Apple VPP Apps

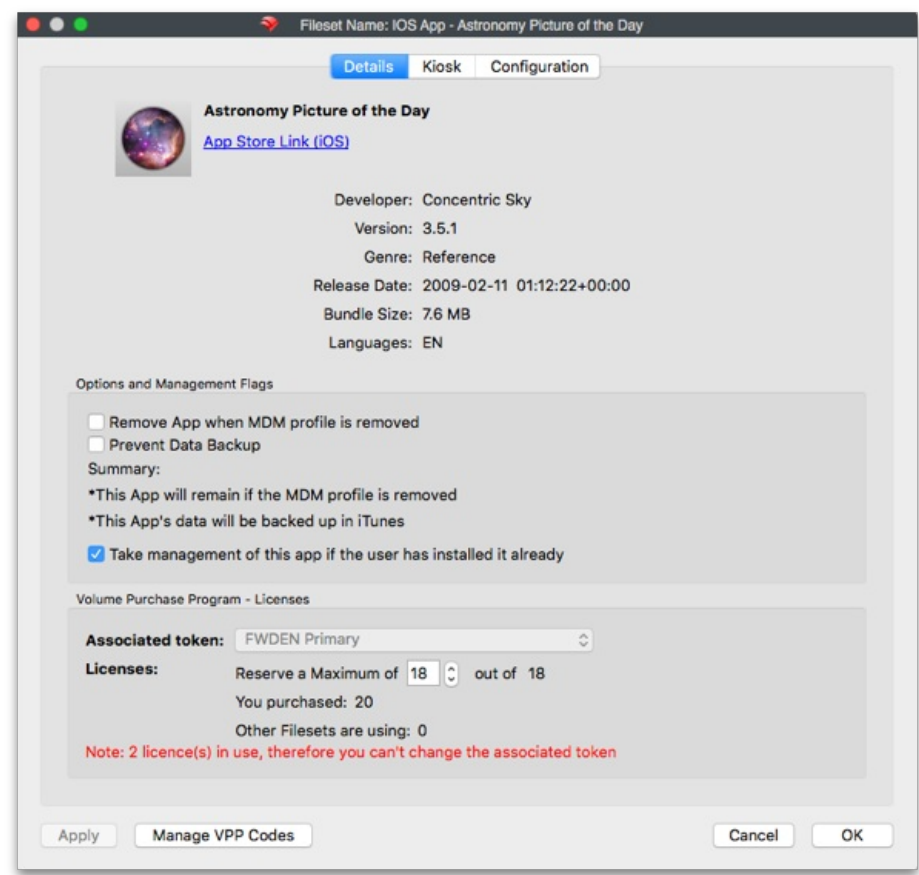
Unlike File Level Filesets, Apple VPP Filesets contain the details of the App from the App Store, including the link to the App, with 3 distinct tabs, Details, Kiosk and Configuration.

**i** FileWave does not store Apps from the App Store, devices will pull the Apps directly from the App Store on successful activation.

## Details

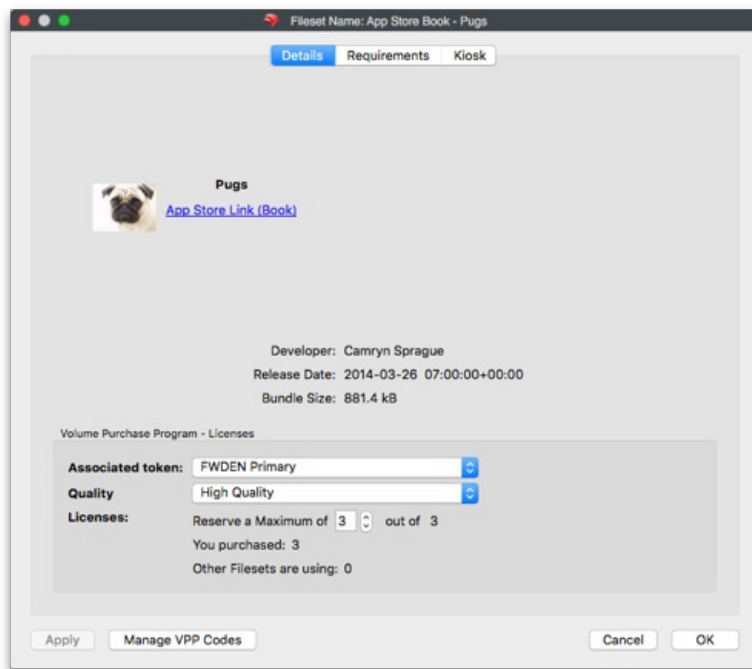
Amongst these are some key entries:

Remove App when MDM profile is removed	If the device has the MDM enrolment profile removed, this option will force Apps managed by MDM to be removed.
Prevent Data Backup	Disallow the App from backing up user data
Take management of this app if the user has installed it already	For Apps already installed, the MDM server will take over management of the App from the user. This allows VPP to auto update the App, amongst other things.



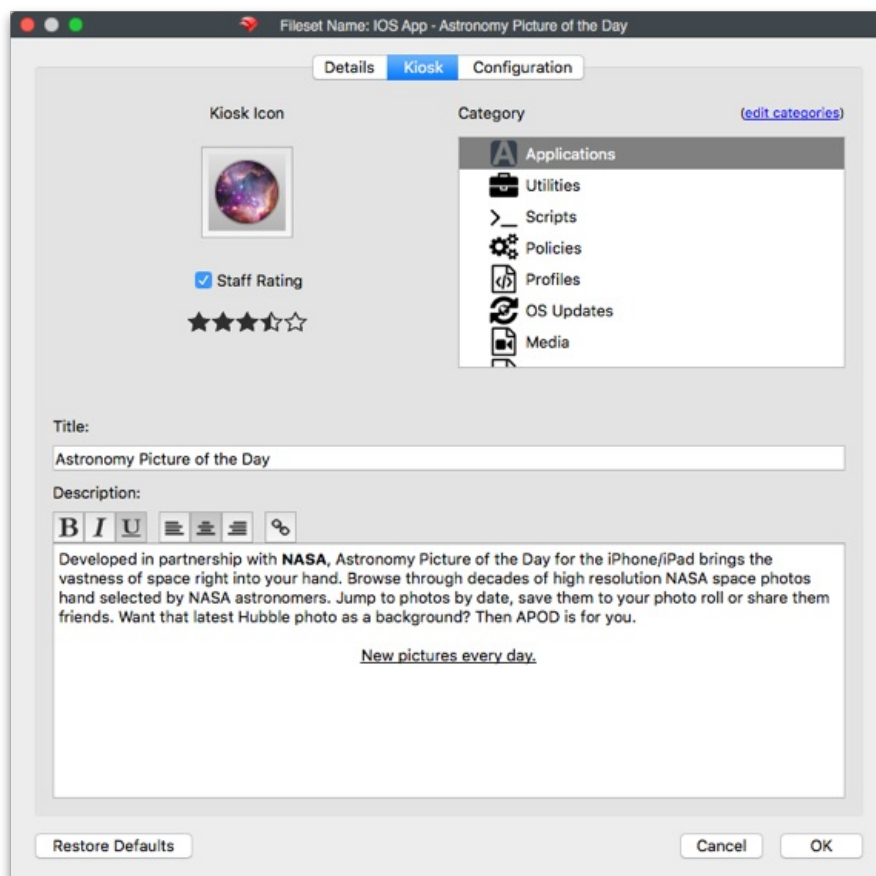
eBooks may also be delivered:





## Kiosk

Details to be displayed to the user through the FileWave Kiosk App, including App Store ratings, description, etc. Text may be updated, categories added or removed and icons replaced.



## Configuration

Enhanced App configuration may be applied, where developers have chosen to offer such features, by way of a preference manifest in plist format.

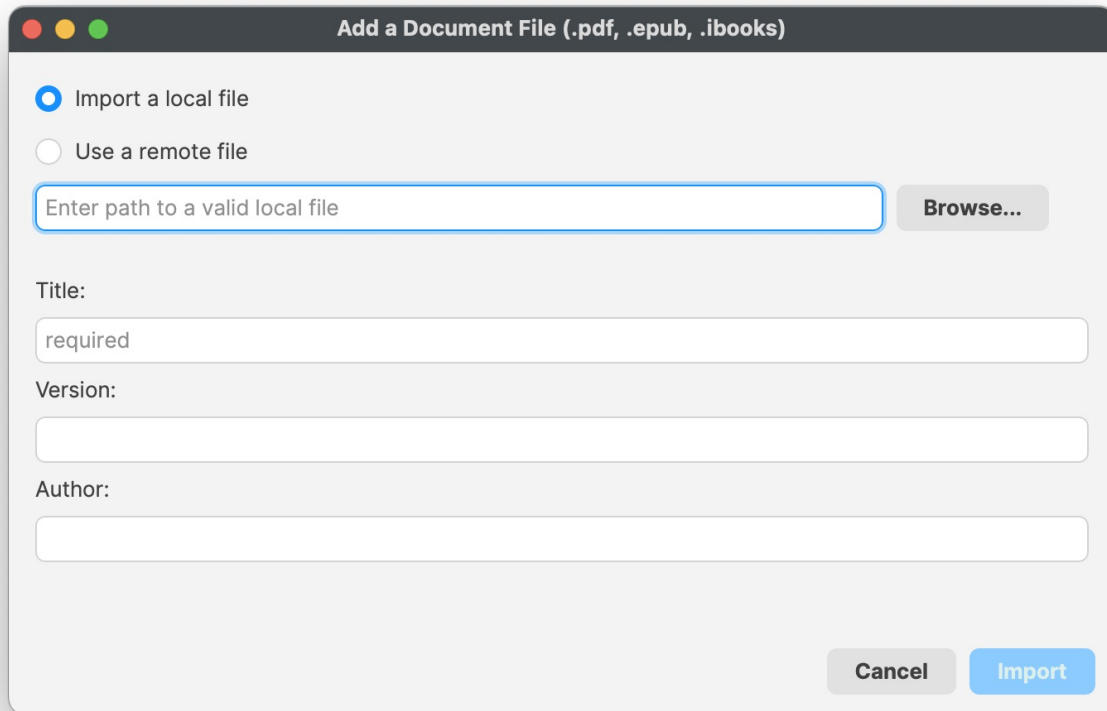
<https://developer.apple.com/library/ios/documentation/General/Reference/InfoPlistKeyReference/Articles/AboutInformationPropertyListFiles.html>

Suppliers of Apps should provide details if they exist.

# Enterprise Filesets and Documents (iOS only)

Apps developed outside of the App Store may also be delivered to devices, where developer requirements are met. Documents may also be delivered to devices.

These Apps or documents may be stored either locally on FileWave Servers or remotely, with a URL link provided.



The image shows a macOS-style dialog box titled "Add a Document File (.pdf, .epub, .ibooks)". It features two radio buttons: "Import a local file" (selected) and "Use a remote file". Below the radio buttons is a text input field with the placeholder "Enter path to a valid local file" and a "Browse..." button. Further down are three text input fields labeled "Title:", "Version:", and "Author:". The "Title:" field contains the text "required". At the bottom right are "Cancel" and "Import" buttons.

**Add a Document File (.pdf, .epub, .ibooks)**

☒ Import a local file  
☐ Use a remote file

Enter path to a valid local file **Browse...**

**Title:**  
required

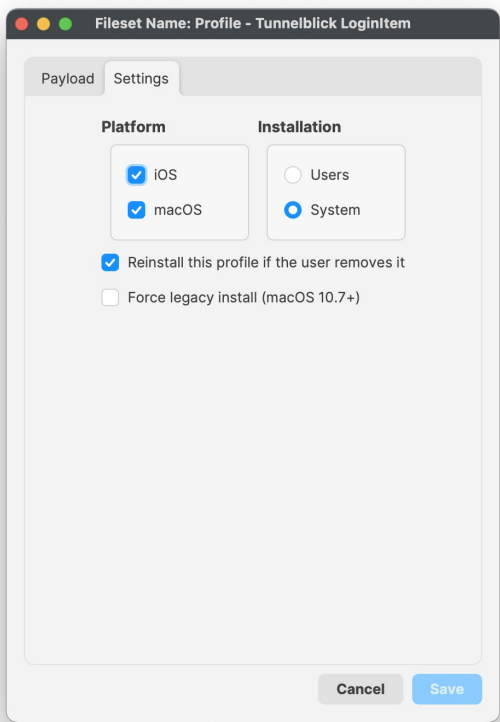
**Version:**

**Author:**

**Cancel** **Import**

# Apple Profile Filesets

Profiles utilise Apple’s defined payloads, allowing management or configuration of User and System. FileWave presets User or System correctly where only one option is available for included payloads. Some payloads, though, could be either User or System as shown in the Fileset Settings:



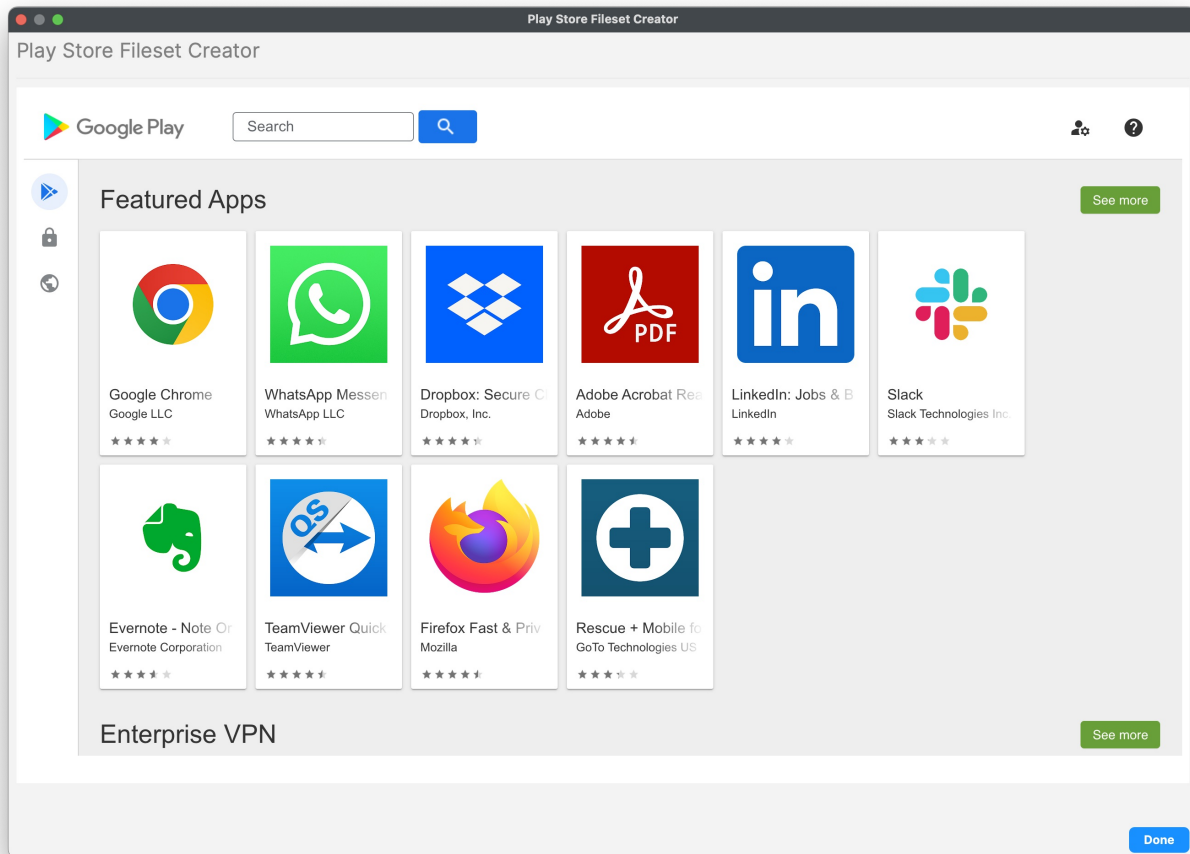
User	Only set when a managed user logs in
System	Payload settings applied regardless of user logged in.

**i** Apple allows the management of any amount of directory users for a single device. However, only one local user will be managed. This user will be the first user after enrolment of the device.

Profiles are stored directly within the Fileset and delivered to devices after MDM communication has been established.

# Android Apps

Android Filesets are created from either the public or private Play Store. Additionally, Web Apps may be considered.



Three tabs are available for a Play Store App Fileset: Configuration, Permissions and Managed Properties. The later will only contain management options, where the developer has chosen to include options.

Fileset Name: Play Store App - Google Chrome: Fast & Secure 1

ConfigurationPermissionsManaged Properties

Force installed ▾

UnspecifiedPreinstalledForce installedBlockedAvailable

Application Install Policy

Specify the app installation policy on devices. Choices are:

- \* Unspecified - on most devices, defaults to Available
- \* Preinstalled - automatically installed; user can remove it
- \* Force installed - automatically installed; user cannot remove it
- \* Blocked - blocked from installation, even if otherwise available; if the app was installed under a previous policy, it will be uninstalled
- \* Available - available to install at users behest

Unspecified ▾

Auto Update Mode

Specify the auto update mode for this app. Choices are:

- \* Unspecified - defaults to Default but does not clutter your installed policy
- \* Default - detects updates within 24 hours, updates when device is charging, not being used, and connected to an unmetered network
- \* Postponed - apps will not be updated within 90 days of the device detecting a new version
- \* High Priority - app update should be triggered within 60 seconds of detection (or connection, if the device is off or not connected to a network)

CancelOK



Configuration and Permissions should be reviewed, as should any any management properties where they exist, before assignment.

# Fileset Properties

Properties exist for most Filesets, but options displayed will vary depending upon Fileset type.

File Level Filesets include:

Properties	Additional features, including Reboot (force sub-option), priority and an option to alter the verification for the entire Fileset. Priority offers additional control over the order of Fileset activation.
Requirements	Settings defining the OS and hardware specific necessities. Only if devices match these, will the Fileset be downloaded and activated.
Dependencies	Discussed in greater details elsewhere. In essence, the Fileset will only activate if another defined Fileset installs first.
Delete Files	A Defined list of files to be removed during activation
Kiosk	Details shown to users via the Kiosk Self-Service menu and system tray item.

Example, File Level Requirements:

Fileset Name: Demo Fileset macOS

Revision: <default> (Initial Revision)

Manage Revisions

Properties

Requirements

Dependencies

Delete Files

Kiosk

☒ Platform

☐ Windows

☒ macOS

☐ Architecture

☒ Intel

☒ 64 bit

☒ Apple Silicon

☒ 32 bit

☐ Min. Memory

0

^

MB

☒ System Version

macOS

☐ 10.7.x

☐ 10.8.x

☐ 10.9.x

☐ 10.10.x

☐ 10.11.x

☒ 10.12.x

☐ 10.13.x

☐ 10.14.x

☐ 10.15.x

☐ 11.x

☐ 12.x

☐ 13.x

☐ 14.x

☐ XP

☐ Vista

☐ 7

☐ 8

☐ 8.1

☐ 10.0

☐ 11.0

☐ 2008 R2

☐ 2012

☐ 2012 R2

☐ 2016

☐ 2019

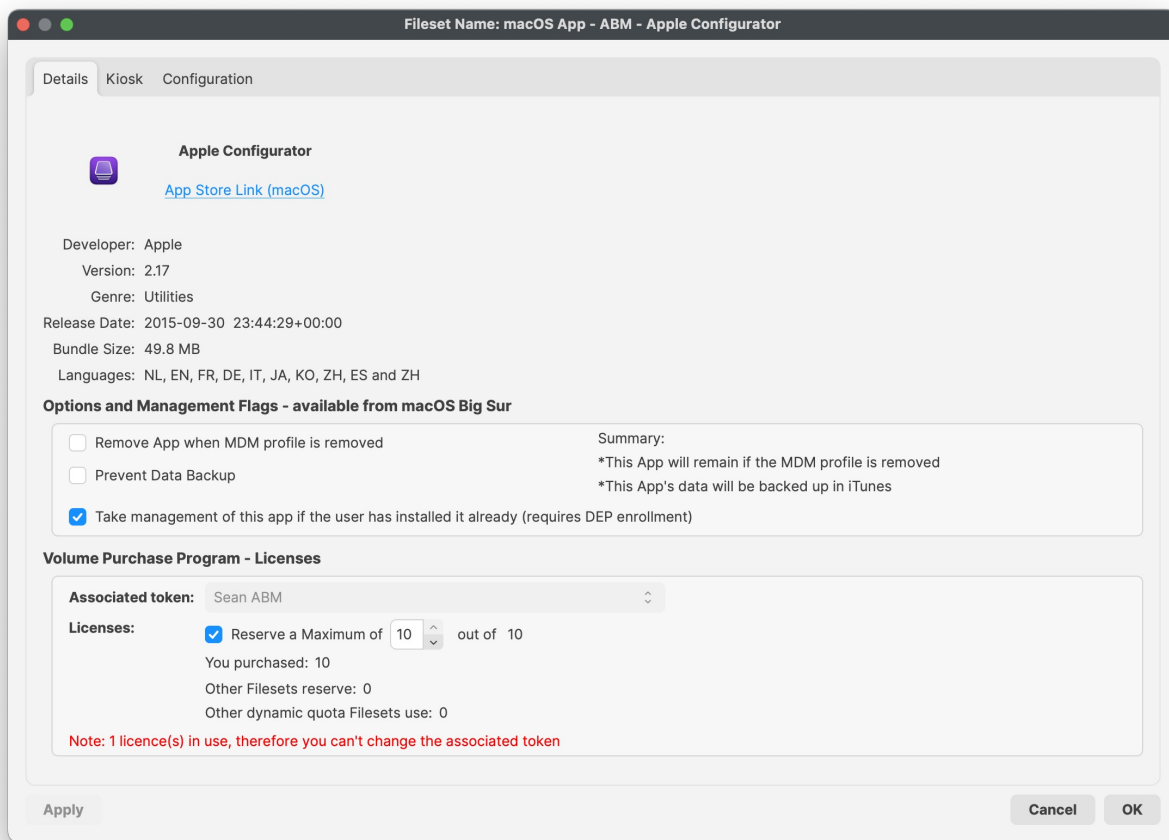
☐ 2022

☒ Use/install on newer OS versions

☐ Evaluate requirements on change and uninstall active Fileset if they failed

Cancel

OK



Example, Apple Profile Kiosk:




Fileset Name: Profile - Tunnelblick LoginItem


Dependencies

Kiosk

Kiosk Icon





☐ Staff Rating





Category


[\(edit categories\)](#)

 Profiles

 OS Updates

 Media

 Documents

 Staff Notice


Title:


Description:


**B**


*I*

U









Restore Defaults

Cancel

OK

⚠ Profiles may only be a dependency in one direction, to prevent unexpected delay of Fileset installations.

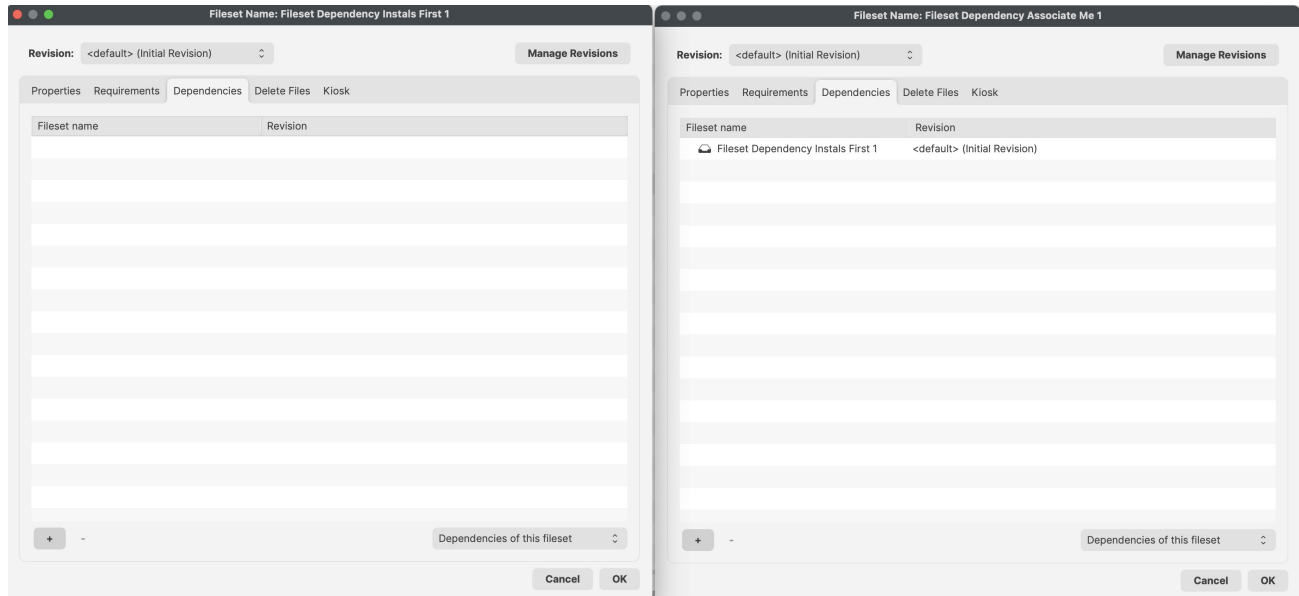
FileWave has no control over the MDM protocol, but must rely upon Apple's process for delivery of MDM items. For this reason, using dependencies, an MDM Fileset, e.g. Profile or VPP App, may be defined to instal after any other non-MDM Fileset type has been activated, but does not allow other Filesets to be reliant on an MDM Fileset.

# Dependencies

Dependencies offer a way to associate multiple Filesets with reliance upon other Filesets to be installed first.

From the below example image, 2 Fileset Dependency Properties are displayed.

- Left image: the Fileset which will instal first (should not be associated with devices relying upon this dependency).
- Right image: the Fileset that will be associated and therefore instal afterwards. It can be seen that the Fileset on the left is shown as a dependency of the Fileset on the right.

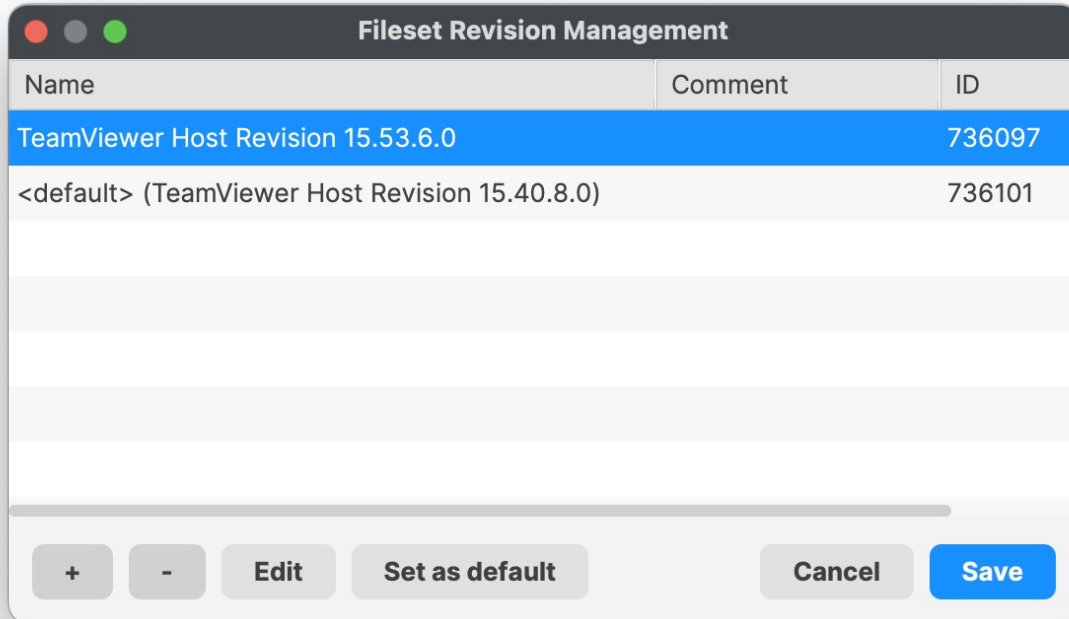


Multiple Filesets may be used as dependencies or be the dependent. Dependents of a Fileset may also be a dependency of another Fileset, creating a chain of Fileset installations.

# Revisions

Previously, updating one version of a software, for example, to a newer version, would require: a new Fileset created, new association or deployment generated and then the prior association removed. Revisions simplifies this process.

To prevent associations or deployments being altered, a Fileset becomes a container of multiple revisions.



The version associated with devices may be altered within the Fileset. This allows for efficient update of Filesets on devices, simplified process between testing and mass assignment, whilst at the same time providing an easy method to roll back if found to be necessary; depending upon Fileset Type.

**i** Behind the scenes, swapping from one revision to another is the same as actually swapping between two different Filesets. All contents will be reconsidered and any scripts ran, e.g. a post-uninstaller script will run if included in the Fileset Revision that is being unassigned.