

What makes Deployments better than Associations?

What

Deployments are new, and will replace Associations entirely eventually. But, does this make them better? Why yes, indeed it does! However there are exceptions where Associations still may be required.

Deployments

First consider deployments.

When/Why

Why Deployments rather than Associations? There are three primary improvements with deployments over associations:

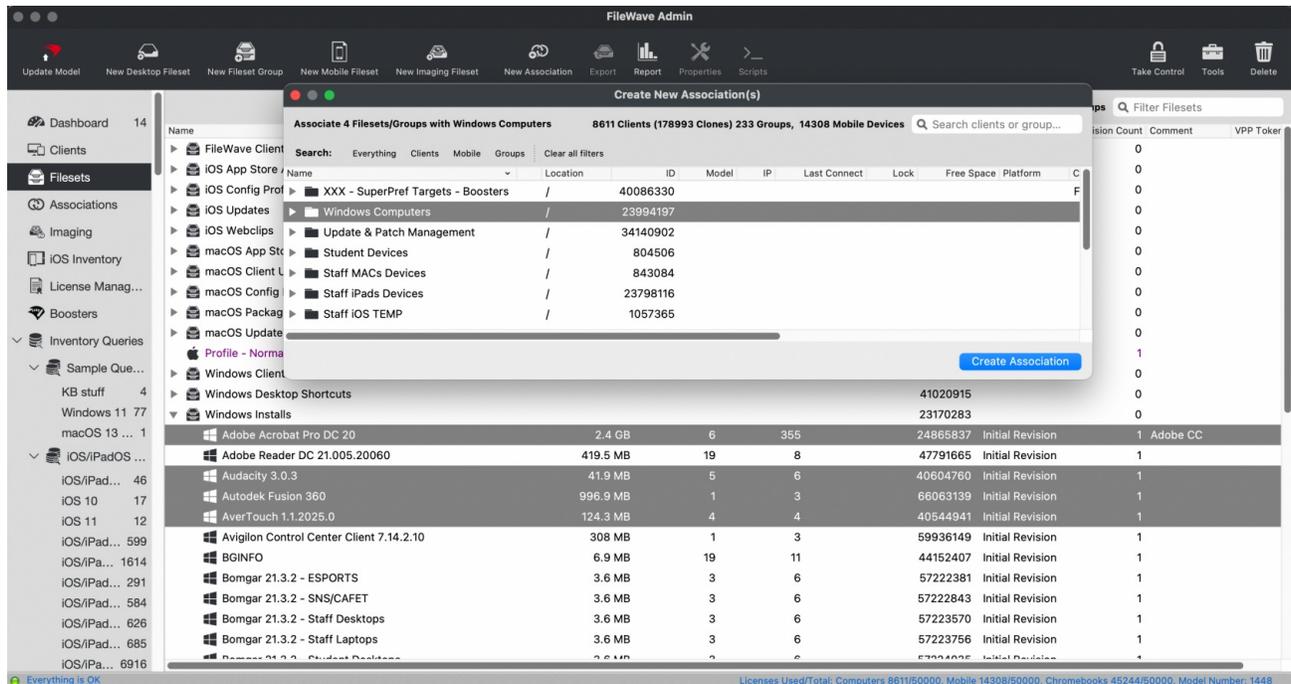
1. The ability to specify Exclusions in a deployment
2. New targets/payloads can be added to existing deployments and will inherit the options
3. Neater Fileset grouping

Exclusions allow you to specify a big target group like “All Windows”, and EXCLUDE from the deployment a subset, perhaps the “Staff Windows” group as an example

Once an Association is converted to a Deployment, the Association will currently still show in FileWave Central, but will no longer be editable as an Association; the Deployment is now managing the assignment of any Filesets included.

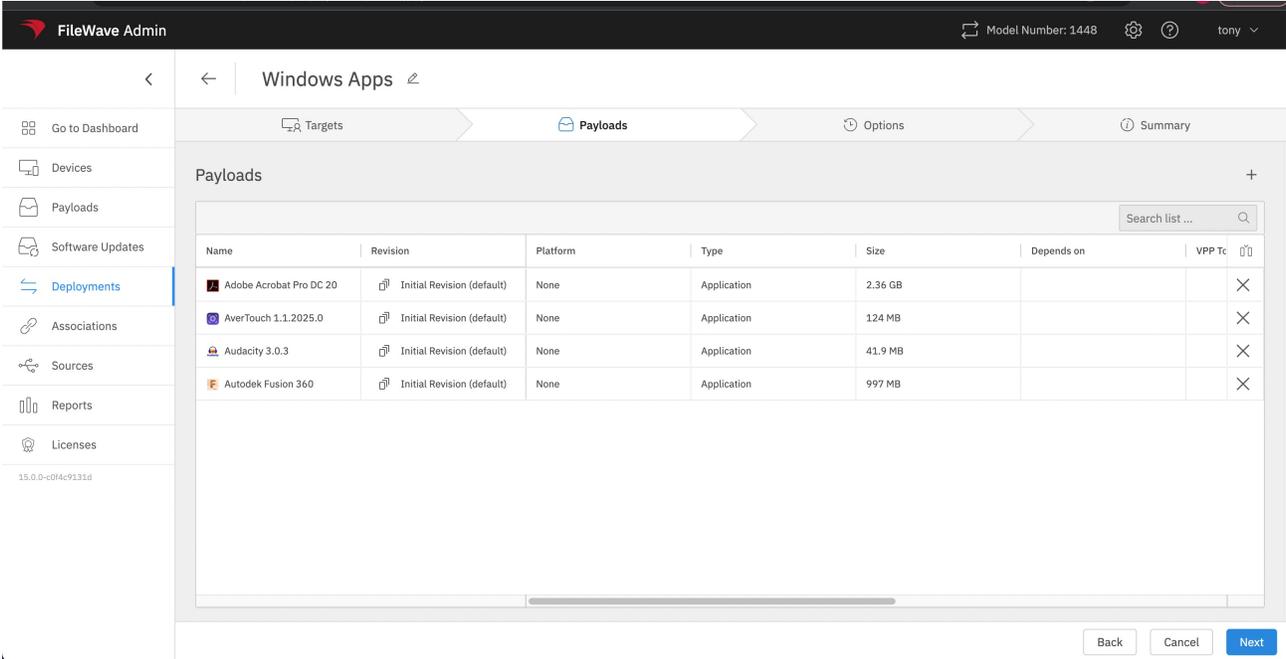
How

Let’s look at a case study. If we were to create associations for the following 4 apps to the Windows Computers group:



Then that will result in 4 associations, each with their own properties. And if we want to switch them all to Kiosk, then we must find/edit them together or individually. If we decided to then add the BGInfo payload as well, we’d have to create another association, and edit that one too.

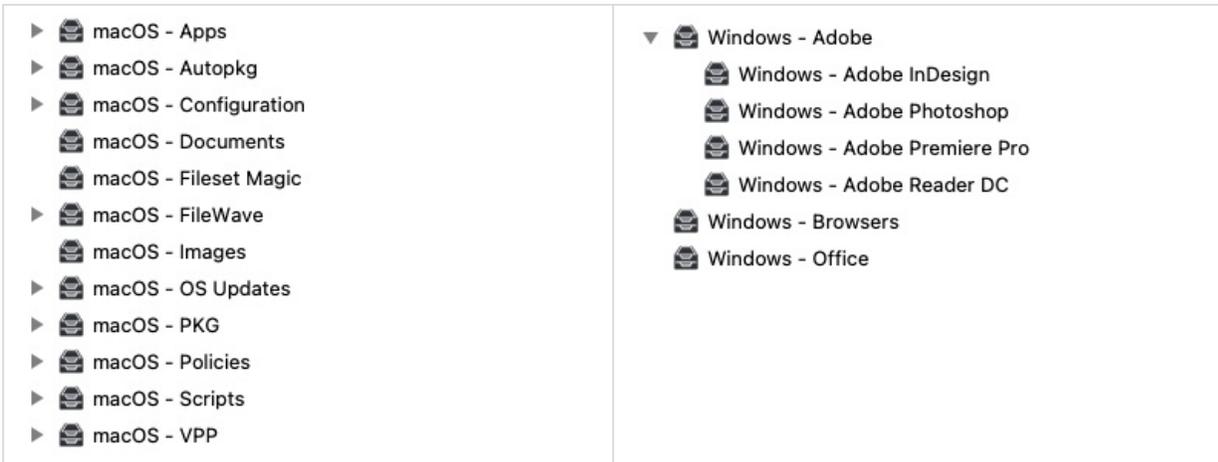
But, the same exact scenario with a deployment is only one entry, with multiple parts:



Therefore, we only need to edit these in one place and one time to change them all to Kiosk. This makes them more efficient, and if we want to add BGInfo to this deployment, then we can do so -- with that installation now inheriting the same rules.

Fileset Grouping

Deployments do not impact Fileset Grouping. Perhaps it is nice to group Filesets by type or by Vendor:



Deployments make it easy to keep Fileset structure, with the assignment of the Fileset contained within the Deployment.

Without Deployments, it would be typical to have something like:

- ▼  Windows - Adobe
 -  Windows - Adobe InDesign
 -  Windows - Adobe Photoshop
 -  Windows - Adobe Premiere Pro
 -  Windows - Adobe Reader DC
- ▼  Windows - Browsers
 -  Windows - Brave Browser
 -  Windows - Firefox
 -  Windows - Google Chrome
- ▼  Windows - Default Apps
 -  Windows - Adobe Reader DC - Latest
 -  Windows - Google Chrome - Latest
 -  Windows - Office
- ▼  Windows Kiosk
 -  Windows - Brave - Latest
 -  Windows - Firefox - Latest

The Default Apps Group, although desirable, has fragmented the Filesets. The latest versions of Adobe Reader DC and Google Chrome are no longer in their respective Fileset Groups. With the addition of multiple default and Kiosk Apps, this fragmentation issue is compounded and makes it harder to keep a neat Fileset structure and to find Filesets.

Digging Deeper

For those who already efficiently use Fileset/Payload groups in associations, you may be asking “Isn’t that the same thing?”. It is, almost...even if you are already efficient with this, Deployments still bring exclusions and the ability to easily find them (by name). So, even then there is an advantage.

Associations

How about Associations?

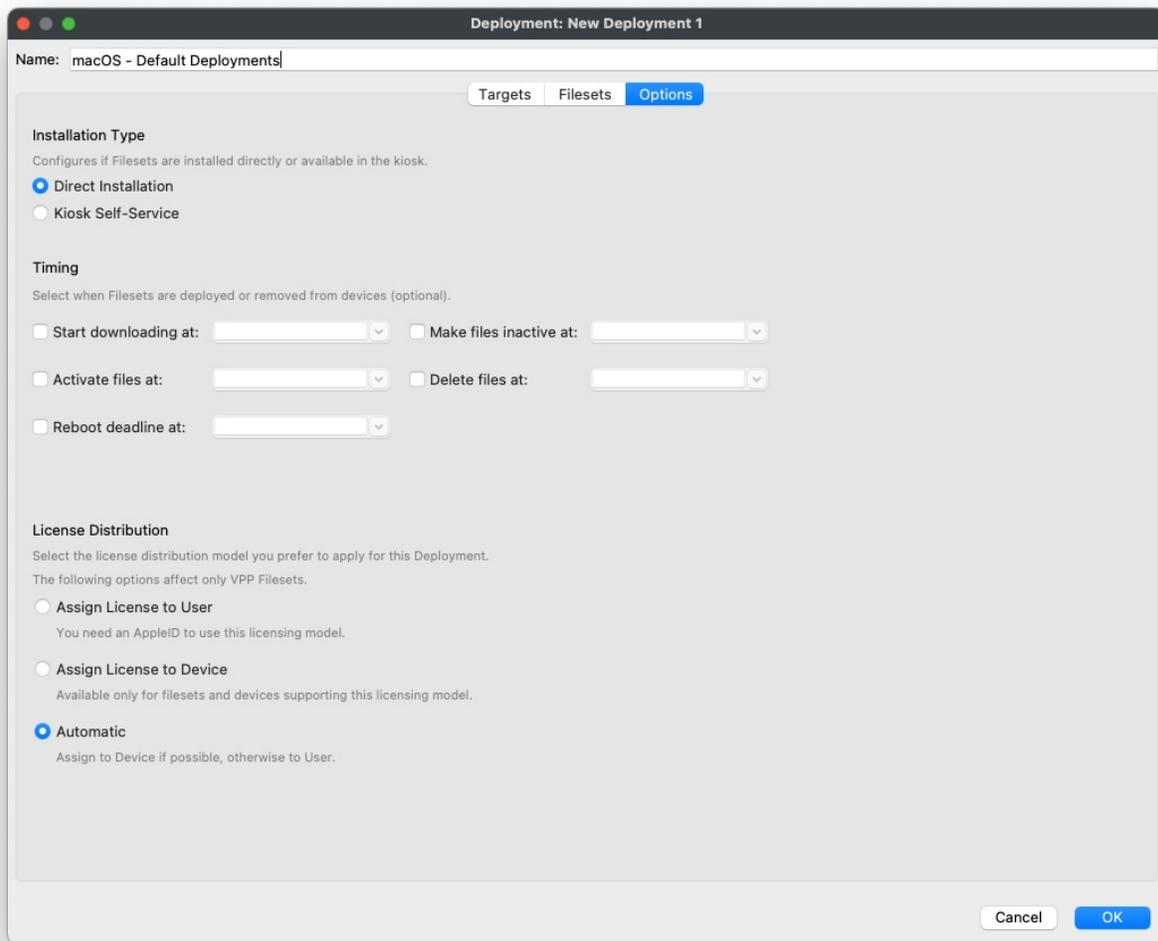
Associations stand out due to a key feature when working with Fileset timings. Imagine one of the Filesets is chosen to be set for Deletion or to be made Inactive for a period of time, but to be made Activate again later.

 It is not possible to change the timings for just one Fileset within a Deployment and so that Fileset will need to be removed from the Deployment.

 Associations allow the same Fileset (but differing Revision) to be associated to the same Client Group.

Fileset Timings

Deployment timing options show as:



All options apply to all containing Filesets. If there was a requirement to alter one Fileset, to perhaps make it temporarily disabled, with Deployments it would involve either:

- A temporary Deployment would be created to move the Fileset into, returning the Fileset back once re-activation is approved
- A permanent new Deployment is created and this one Fileset will now only live within this one new Deployment thereafter
- Another Deployment will be created, but the new Deployment will be 'closer' to the target device(s) and set as inactive, until re-activation is required and this new Deployment may then be deleted

Associations are handled individually, as such the current association timing may be changed without any other concerns.

Revisions

Each Fileset may have multiple Revisions, but why is this beneficial? Consider the question 'How and, more importantly, when is one Revision going to replace another?'

- Associations may have differing Revisions of the same Fileset Associated to the same group of devices
- Deployments can only handle a single assignment of a Fileset, regardless of Revisions

Kiosk

By supplying multiple revisions to the Kiosk, users can have the ability to choose between Revisions. With Deployments, a new Deployment would have to be created to allow for each alternate Revision, whilst with Associations, all chosen Revisions may be associated to the same Client Group.

Timed Revision Updates

It is often desirable to transition between two Revisions at a chosen time, for example, out or hours. Achieving this with Associations is much neater. The same group that already has the current Association will receive the Association to the new revision. Within each Association, the time would be altered, one set as Delete and the other set as Activate. When that point in time is reached, the magic happens and the Revision on the devices is replaced.

As an example, below are 2 Associations of the macOS EyeBrowse App, both to the All macOS group, but the current version 1.0 is set for Deletion at the same time that version 2.0 is set for Activation.

Fileset	Client/Group/Clone	Revision	Type	Start Download	Become Active	Start Passive	Delete Files
macOS - EyeBrowse	All macOS	<default> (1.0)	Standard	-	-	-	27/07/2023 19:00
macOS - EyeBrowse	All macOS	2.0	Standard	-	27/07/2023 19:00	-	-

Again, with Deployments, a new Deployment would need to be created, just for this one new Revision. Actually 2 new Deployments!

Why 2 new Deployments? A single Deployment may only handle one Fileset (regardless of Revision). The original Deployment will have no timings set, nor should it. As such one Deployment needs to be created for the Deletion and another Deployment for the Activation.

Once the target Date and Time have been reached, the additional Deployments themselves could be removed, with the Fileset being reinstated, back into the original Deployment.

If you do this with many Revisions, then the quantity of Deployments becomes unwieldy, with differing Deployments handling differing Revisions.

For this same example, the Eyebrowse App is contained within the Deployment:

As such, new Deployments would need to be created to handle the timed Revision changeover.

Original Deployment (EyeBrowse Removed)	Deletion Deployment	Activation Deployment

Neatness

This is a fundamental consideration of the choice between Associations and Deployments.

On the one hand, Deployments are neater. Filesets can be grouped, sensibly by either vendor supplied, Fileset contents type, etc. however seen fit. The Deployment then handles the link between the Filesets and the Clients. Associations causes Fileset grouping to be dictated by the Association instead, fragmenting the Fileset structure.

On the other hand, Associations keep things neat, where Fileset Timings are concerned. One single Client Group can handle all Revisions for any single Fileset or a single Fileset Association's timings can be altered, preventing the need to build out lots of Deployments.

Conclusion

Yes Deployments are better, but currently only in some instances. Each assignment of Filesets to Clients should be considered, with merits given and choices made appropriately, be that Association or Deployment.

Like to see examples in action? Check out the Foundry Onboarding section on [Associations and Deployments](#), which contains a comparison video.

