

Fileset Revisions

- [Fileset Revisions Overview](#)
- [Associating a Fileset Revision](#)
- [Editing Filesets that have Revisions](#)
- [Managing Revisions](#)
- ["Default" Revisions](#)
- [Creating a New Revision](#)
- [Editing Fileset Revisions](#)
- [Removing/Deleting a Revision](#)

Fileset Revisions Overview

What

Fileset (payload) revisions allow you maintain two (or more) different versions of the same fileset within one FileWave fileset object.

When/Why

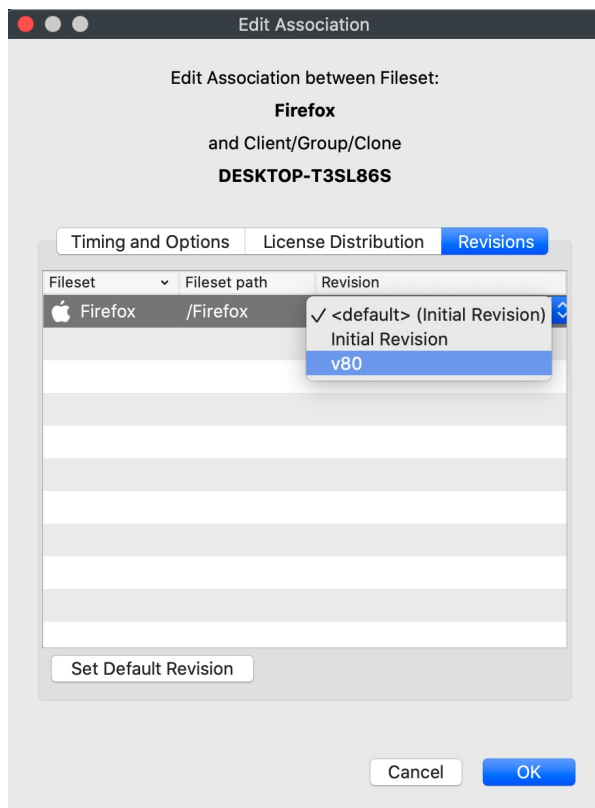
The above description might not knock your socks off, but we are pretty excited about this, and think you will be too! Consider this scenario:

We are currently deploying Firefox (v79) to all of the devices in the environment. But v80 of Firefox is now out, and we want to upgrade. Previously we had to either duplicate the FileWave fileset and change the content of it, or create a brand new fileset from scratch. Then, when we had to remove the association for v79 from our test devices and add a new association for v80. And, assuming all was good, we then had to remove the old association for all devices, and replace it with the new one. Finally, if we were worried about the possible interruption of service by removal of the old app, we might have had to play a bit of a shell game with the timing to make sure customers weren't impacted by the upgrade.

Not anymore!

How

Fileset revisions allow you put version 80 of Firefox in the very same fileset in which you had v79. Then, rather than removing associations, you would just change the revision the association uses. And, we take care all of that complicated fileset timing stuff for you. In subsequent articles (linked below), we'll go into all of the specifics, but here is a sneak preview of setting a revision in an association:



You may be thinking to yourself: "This sounds complicated, and I don't want to use revisions." There are two responses: First, you absolutely don't have to use them if you don't want to. If you choose to ignore revisions, then everything will continue to work exactly the way it has for you in the past. Rest assured though, Fileset revisions aren't actually complicated, and you may find they really make things easier for you.

Related Content

- [Associating a Fileset Revision](#)
- [Editing Filesets that have Revisions](#)

- Managing Revisions

Associating a Fileset Revision

What

When you attempt to associate a fileset that has revisions, you'll notice a slightly different workflow, where you are asked to select a revision to associate. (Associating a fileset that has no revisions functions as it always has)

When/Why

Assume our same Firefox fileset, containing two revisions for v79 and v80. When we create a new association for this fileset, we need to tell the system which revision (v79 or v80) that we want to use.

How

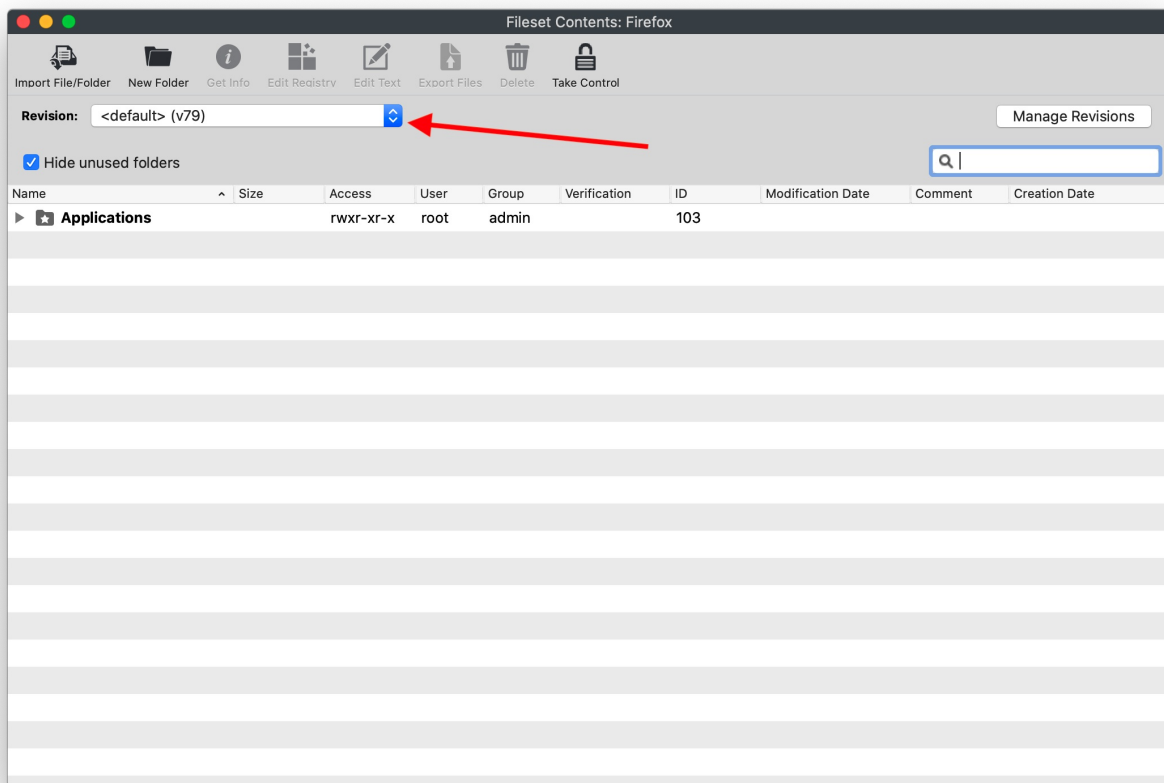
Selecting the revision we want to do is very simple, since the FileWave admin just asks us which one we want to use as you'll see below:

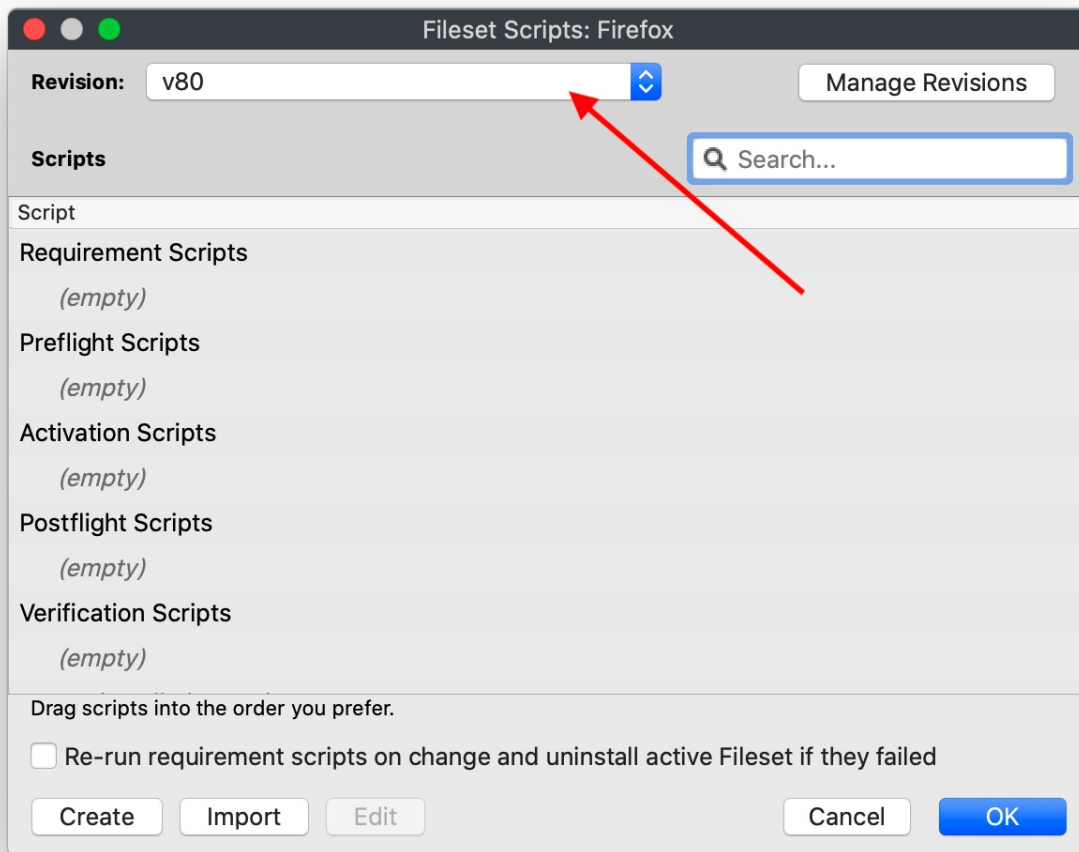


You may ask yourself, "What is the 'default' option I see in that menu?" That is a great question and we'll talk about that super-helpful option in another article you'll find linked below.

Related Content

- ["Default" Revisions](#)
- [Associating a Fileset Revision](#)
- [Editing Filesets that have Revisions](#)
- [Managing Revisions](#)





Switching which fileset you are editing in either window is as simple as changing the value in the drop-down. And, if you just want to do filesets the same way you always have, you can safely ignore that drop-down altogether.

Related Content

- [Associating a Fileset Revision](#)
- [Editing Filesets that have Revisions](#)
- [Managing Revisions](#)

Managing Revisions

What

There are quite a number of things we can do to "manage" revisions. We can create new revisions, edit them, delete them, and set default revisions.

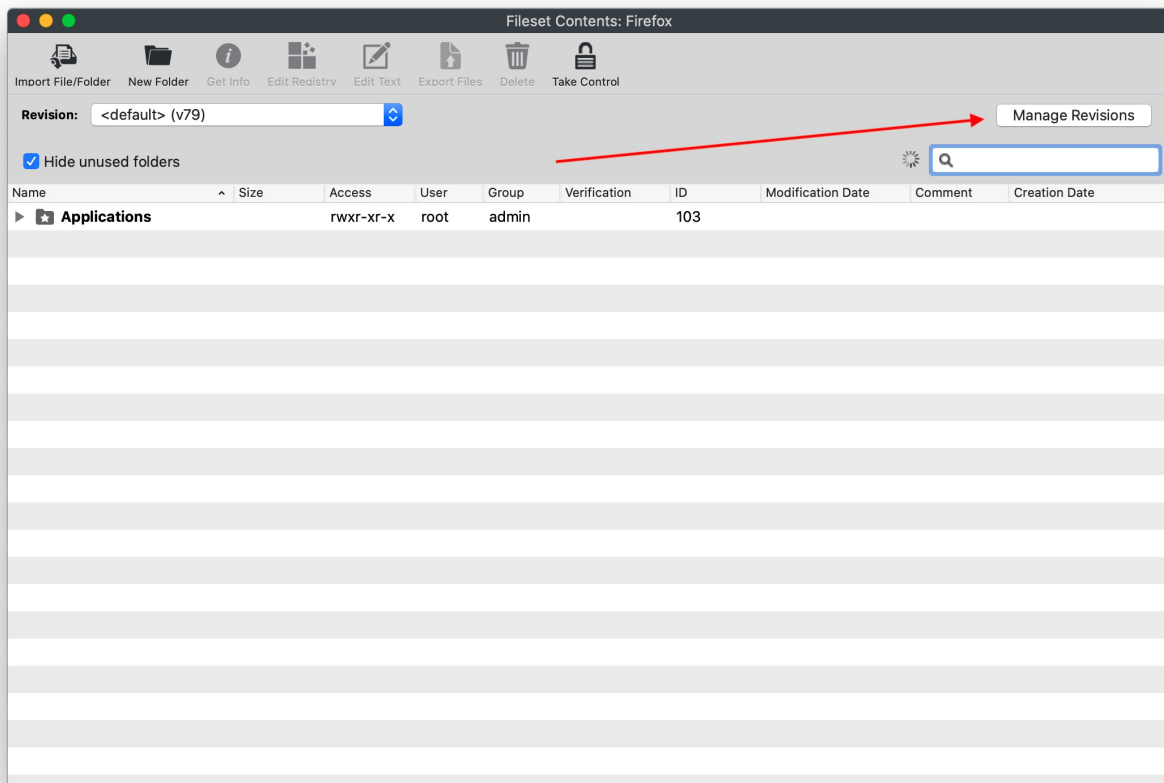
When/Why

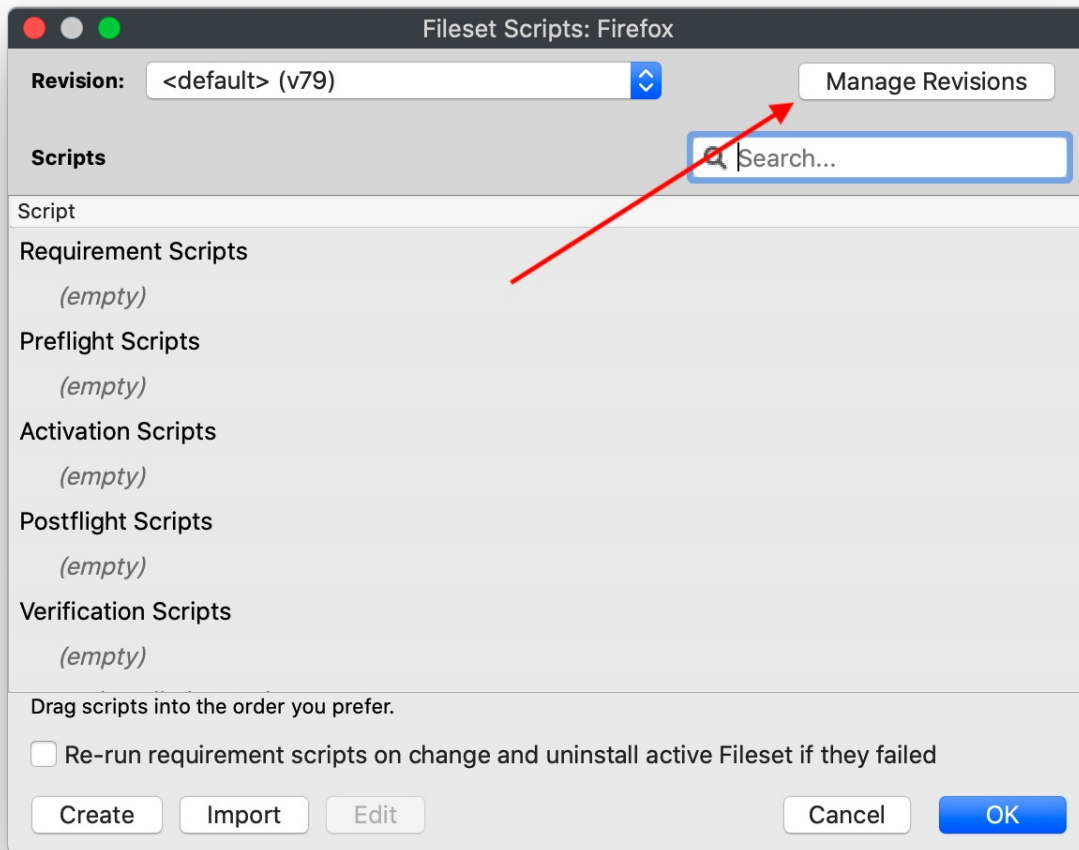
Our management options breakdown as follows:

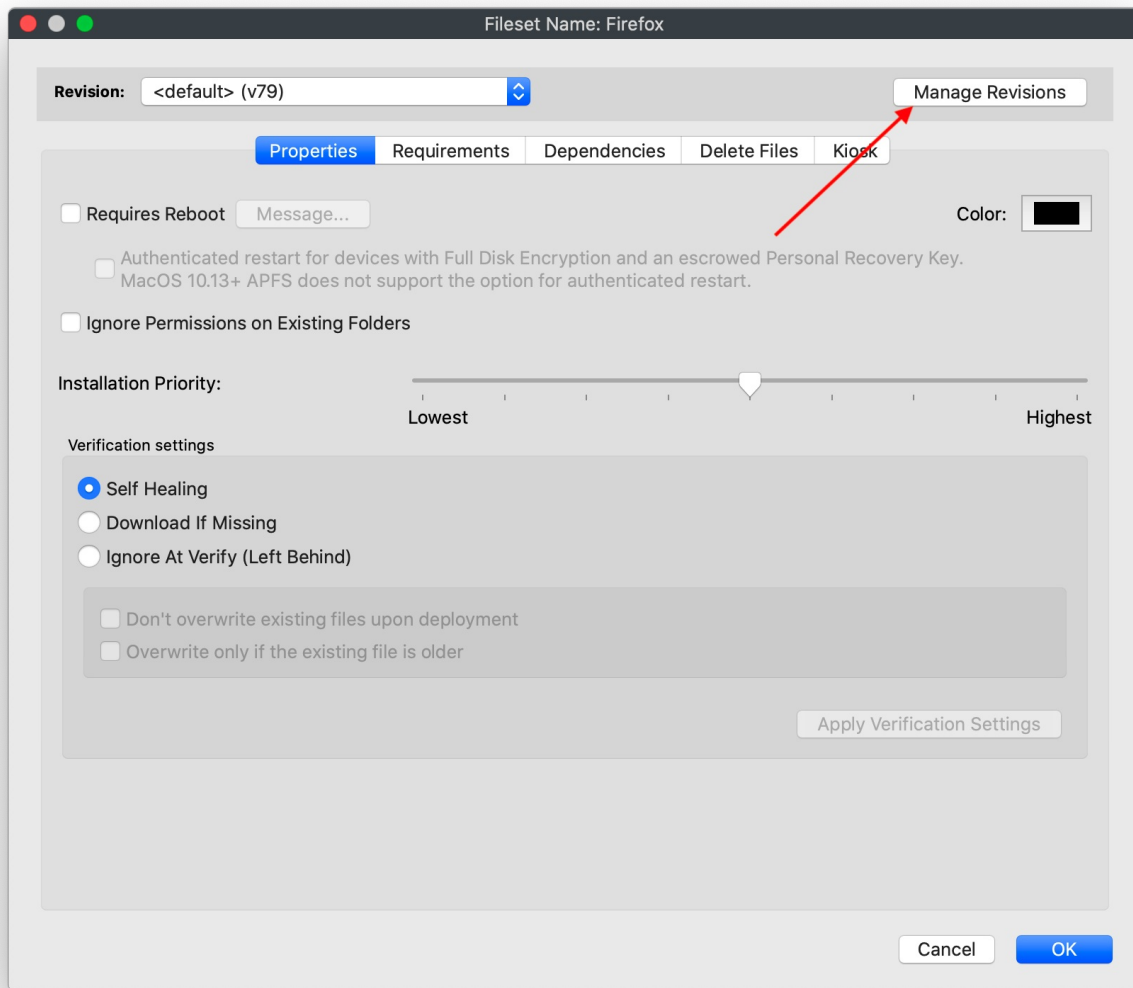
- Editing revisions (different from editing the contents of the revision) allows us to change the names and descriptions of revisions
- Creating them allows us to create additional revisions in the current fileset
- Deleting a revision, as the name implies, allows us to remove a revision from a fileset
- Setting a revision as the "Default" allows us to easily deal with associations for testing and production in different ways

How

The thing that all of the options listed above have in common is they are all accessed in the same way, through the "Manage Revisions" button in the Fileset Editor, the Script Editor, or the Fileset Properties window, as shown below:







Each of the management options is detailed in the articles linked below.

Related Content

- ["Default" Revisions](#)
- [Creating a New Revision](#)
- [Editing Fileset Revisions](#)
- [Removing/Deleting a Revision](#)

"Default" Revisions

What

What are default revisions? An excellent question indeed. Default revisions are a way of marking a specific revision of a Fileset as the "default", regardless of what it is called. Keep reading, and you'll see how very useful this is, and how our great developers try to make your job easier on an everyday basis.

When/Why

We are going to use our example again of v79 and v80 of Mozilla Firefox. Assume for a moment that we have v79 assigned to all devices right now in two associations as shown below::

The screenshot shows the FileWave Admin interface. On the left is a sidebar with navigation links: Dashboard, Clients, Filesets, Associations, Imaging, iOS Inventory, License Ma..., Boosters, and Inventory Q... The main area is titled "7 Clients (22 Clones) 18 Groups, 2 Mobile Devices". It features a search bar and a table of filesets. The table has columns for Name, Size, and Version. The filesets listed are: Windows Get Image Date (464 B, 1), Testing Empty (70 B, 4), Test FW Shell (0 B, 1), Profile - Tik Tok Blocked (1), Profile - Test Block (1), OS - Windows, and OS - macOS. Below the fileset table, there are filters for Association Type, Software Update, License Distribution, and From Deployment. The bottom section shows a table of associations with columns for Fileset, Client/Group/Clone, Start Do, Become Active, Start Passive, Delete Files, ID, Type, Is So, Lik, Rebo, Deg, Ex, and Revision. The associations shown are for Firefox, Beta Testers, and Production, all associated with revision v79.

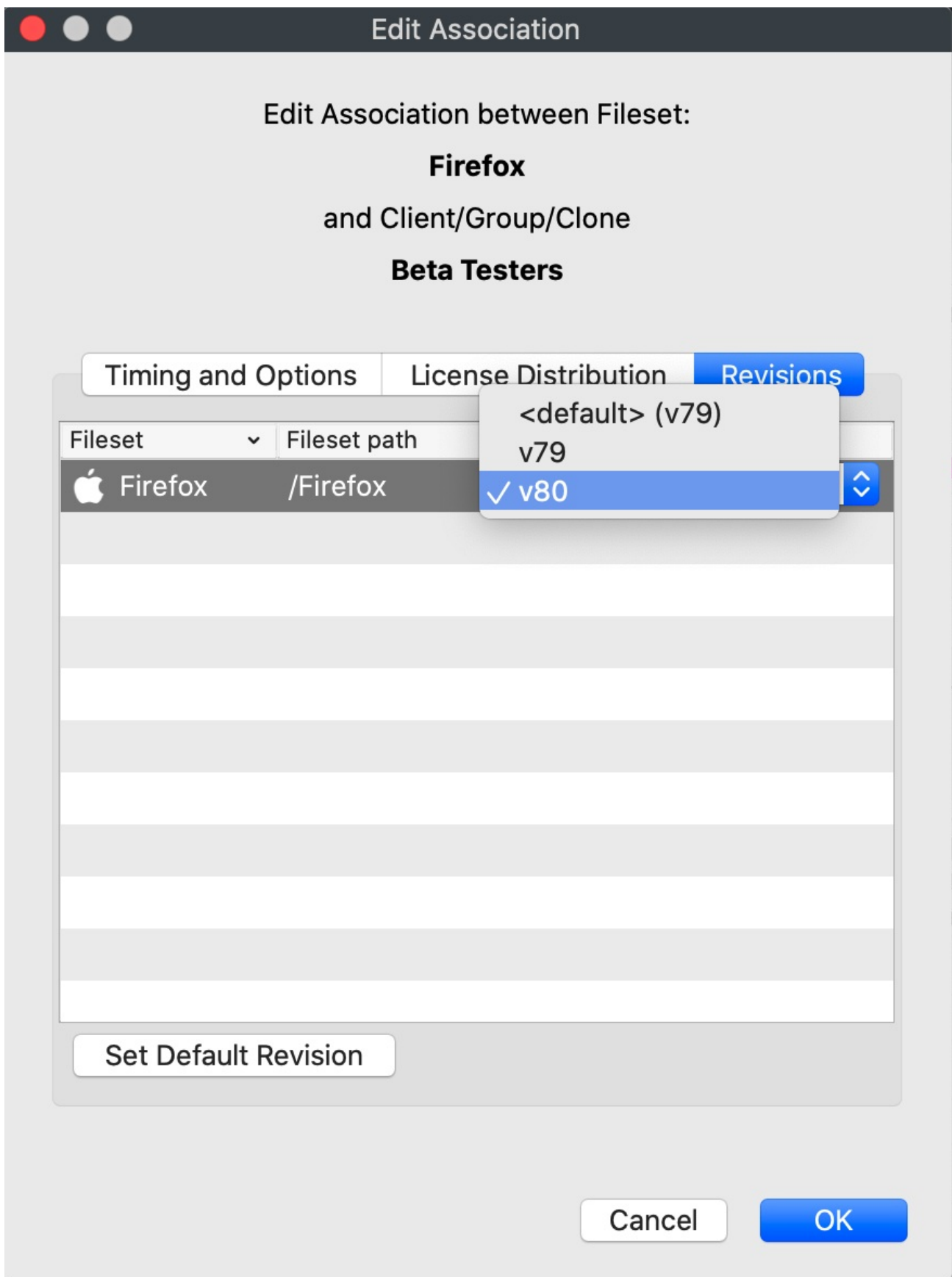
Name	Size	Version
Windows Get Image Date	464 B	1
Testing Empty	70 B	4
Test FW Shell	0 B	1
Profile - Tik Tok Blocked		1
Profile - Test Block		1
OS - Windows		
OS - macOS		

Fileset	Client/Group/Clone	Start Do	Become Active	Start Passive	Delete Files	ID	Type	Is So	Lik	Rebo	Deg	Ex	Revision
Firefox	Beta Testers					3347	Standard	-					v79
Firefox	Production					3348	Standard	-					<default> (v79)

If you'll look closely, you'll see that the "Production" group has an association to the revision <default> (v79) and the "Beta Testers" group is associated to v79. Effectively, this is exactly the same for both groups...v79 is installed. But, let's take a look below at what happens when we want to upgrade to v80 in the environment.

How

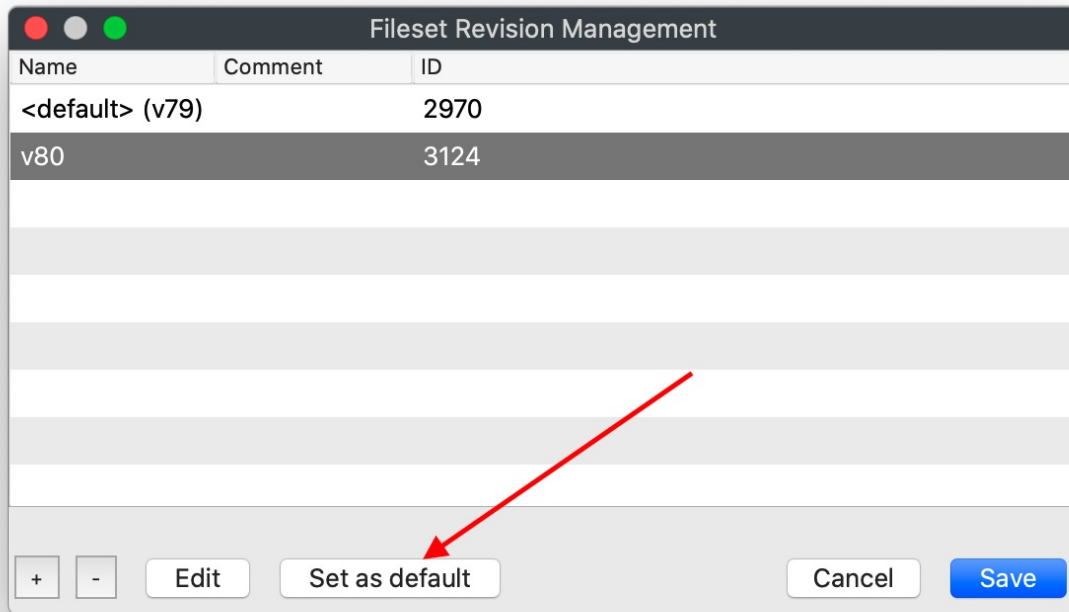
We aren't just going to assign v80 to all devices right away...as always, we want to test first, so with our patch testers group we'll edit their association as follows:



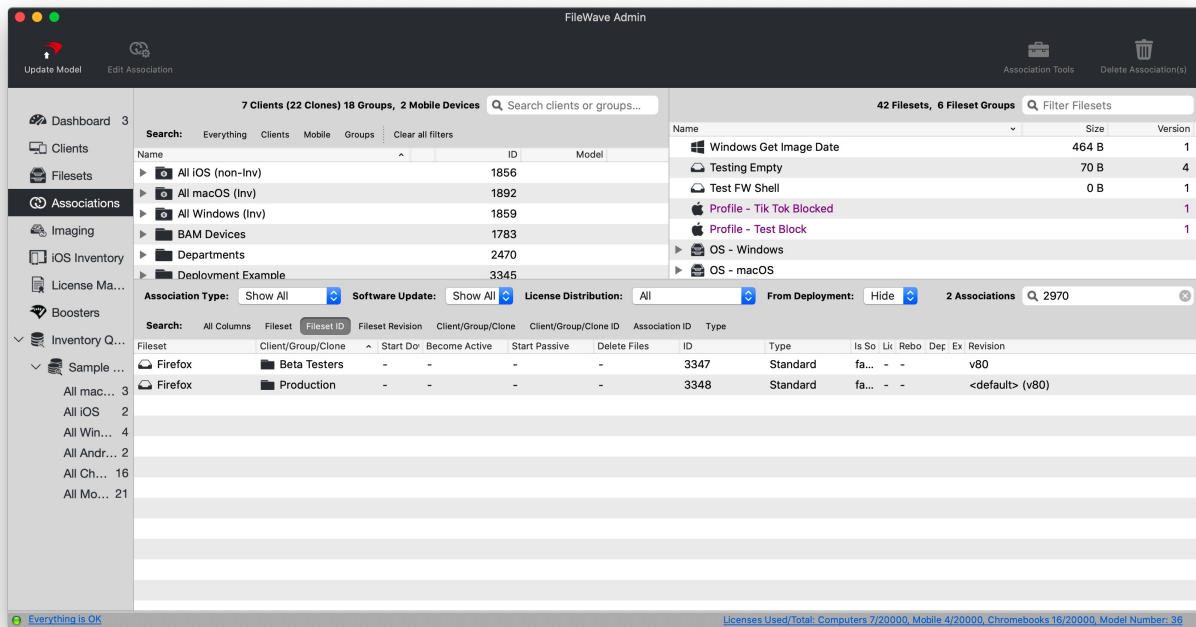
Once we save this association and update the model, all patch testers will get the new version. No deleting of the association or creating a new one is required.

Now, assume that all testing goes well, and we are ready to release to production. Now, we could edit the association for "production", but assume for a second that we had ten such associations...would we want to edit all ten of them? In a word, no. But, if you remember, the association above to production was not v79, but rather Default v79...which means the association is just to the "default" revision of the fileset.

So, we can simply edit the properties of the fileset itself and change the "default" to v80 instead.



Once we save that change, and update the model, you'll see the change reflected in the association:



And, upon next check-in, all "Production" devices will upgrade to v80.

- Remember, we take care of the timing of this transition between versions...so you don't have to worry about the transition timing.

Related Content

- [Managing Revisions](#)
- ["Default" Revisions](#)
- [Creating a New Revision](#)
- [Editing Fileset Revisions](#)

- [Removing/Deleting a Revision](#)

Creating a New Revision

What

Creating a new fileset is the same as it always has been, but in some way, shape or fashion we need the ability to create new fileset revisions before we can use them.

When/Why

Typically, we'll want to use fileset revisions whenever we expect to continually upgrade an application...so, in most cases basically. In this case, we'll look at how to create a v81 revision of Firefox.

How

Adding a new revision is done from the "Manage Revisions" view by clicking the plus icon:



Note that in this case, we chose "Empty" because we don't want the files from the previous revision. But, if we had custom scripts or anything else like that, we would probably duplicate the previous revision instead. Once the revision is created, you edit it like you would any fileset.

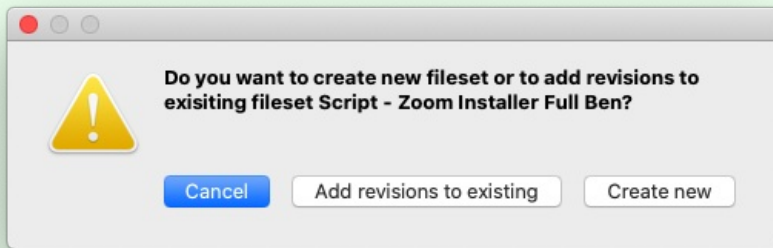
When creating a new revision, you can create it as empty, duplicate an earlier revision entirely, or duplicate only the properties



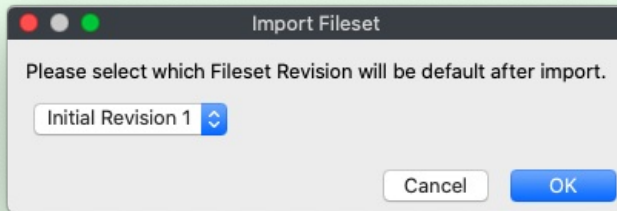
✓ Duplicate everything
Duplicate properties only
Duplicate content only

of that revision:

Fileset Revisions can also be created by drag and drop operation from the file system onto an existing Fileset. You'll be prompted about making a new Fileset, or adding a revision as shown:



If adding a revision, you will also be prompted about setting the default revision:



Related Content

- [Managing Revisions](#)
- ["Default" Revisions](#)
- [Creating a New Revision](#)
- [Editing Fileset Revisions](#)
- [Removing/Deleting a Revision](#)

Editing Fileset Revisions

What

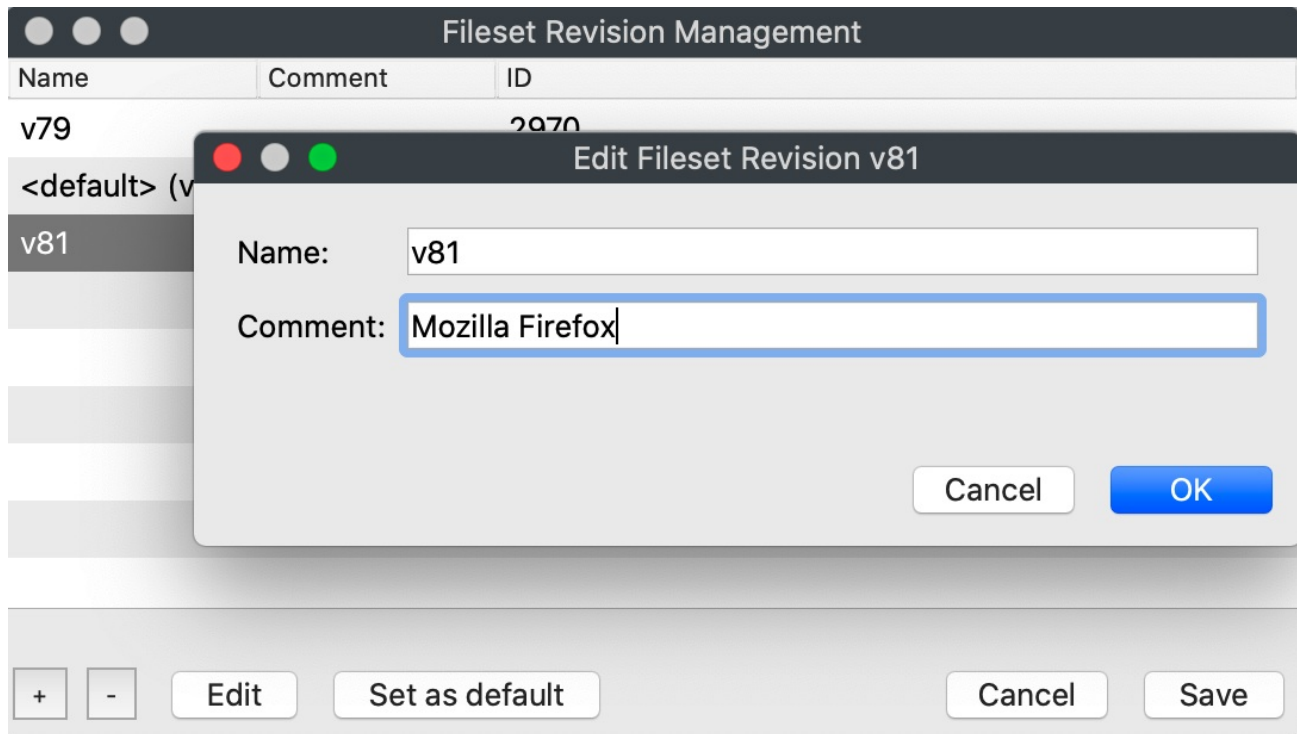
Revisions themselves have some properties that can be edited: namely the name and the description of the revision.

When/Why

Editing the properties of a revision is also done from the "Manage Revisions" window...we'll use it if we want to edit the name. For instance, every time the first revision is created, it is called "Initial Revision"...which is helpful to know, but not very descriptive. Below is how we change that.

How

First, we'll go into the "Manage Revisions" view for a particular fileset, then choose "Edit" with a revision selected:



There are only two fields to set, but setting them descriptively does help later when creating associations.

Related Content



url 

- [Managing Revisions](#)
- ["Default" Revisions](#)
- [Creating a New Revision](#)
- [Editing Fileset Revisions](#)
- [Removing/Deleting a Revision](#)

Removing/Deleting a Revision

Removing/Deleting a Revision (v14+)

What

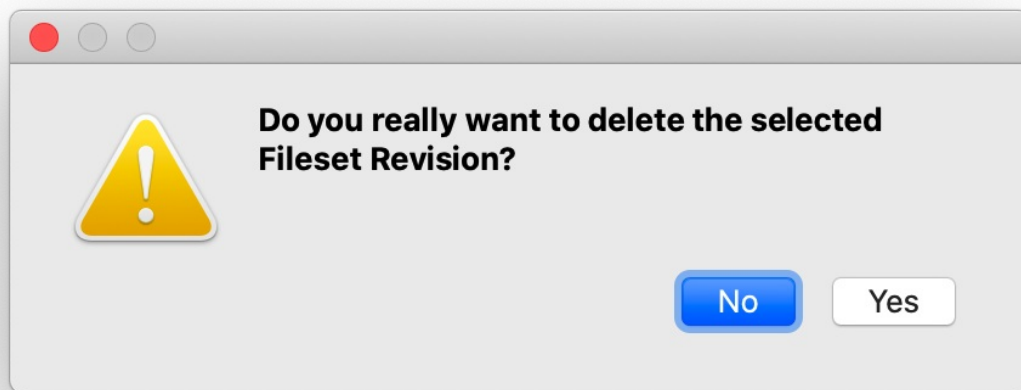
We can create revisions, so of course we want to be able to delete them as well.

When/Why

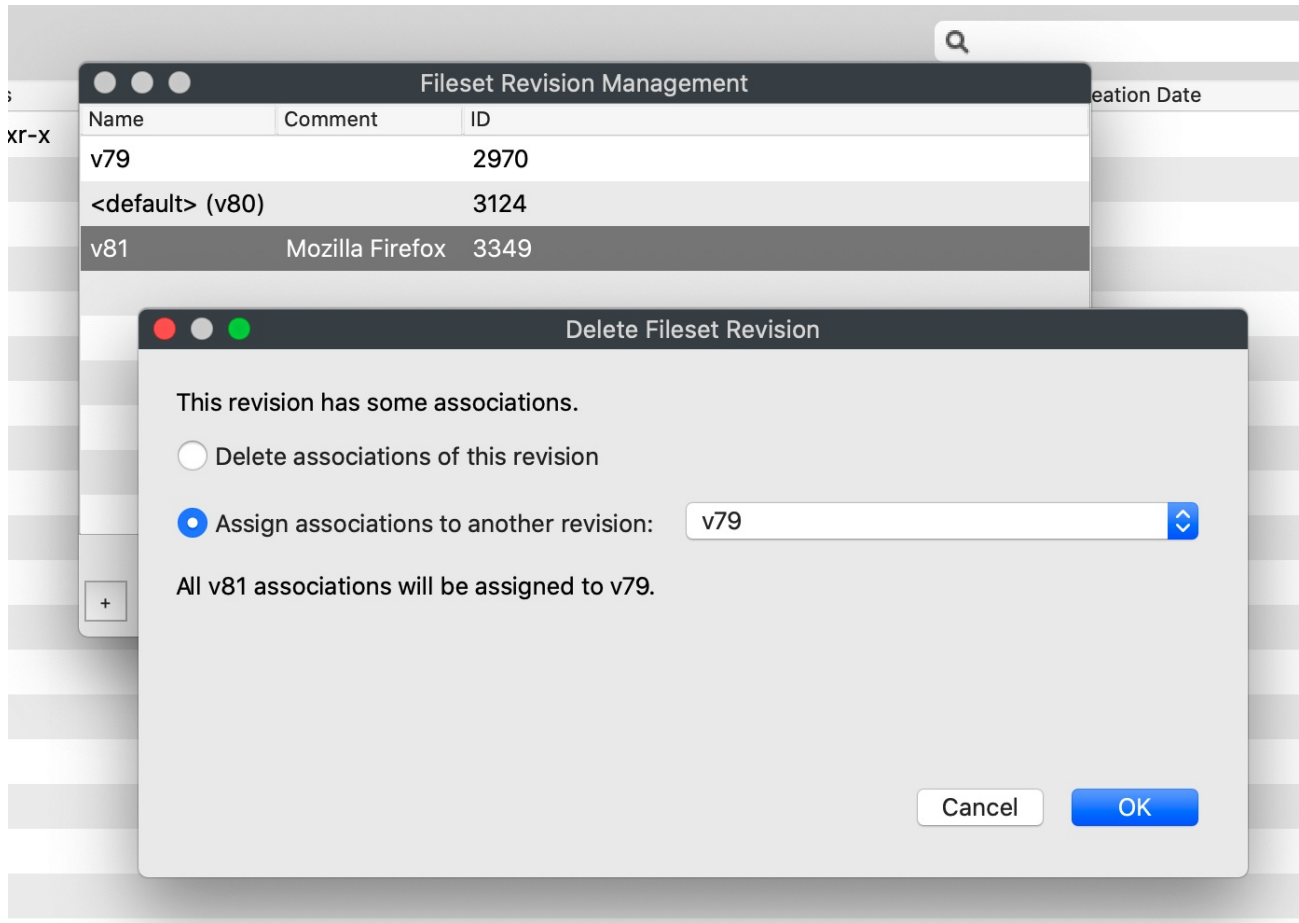
We'll usually want to delete revisions whenever the revision is no longer pertinent. We would probably always leave the N-1 version of our fileset, but if we have older revisions, we would certainly want to clean up to save on disk space.

How

If we go into the "Manage Revisions" view, we can select a revision and hit the minus sign to remove it. If no associations exist for that revision, we'll simply get a prompt to confirm:



But, if there is an association, we can choose to remove the association outright, or switch it to another revision:



Related Content



- [Managing Revisions](#)
- ["Default" Revisions](#)
- [Creating a New Revision](#)
- [Editing Fileset Revisions](#)
- [Removing/Deleting a Revision](#)