Un-install Filesets

What

There will likely come a time when software installed is no longer required. What options are there for removing installed software.

Options

Self-Healing

The verification settings ‘Self Healing’ and ‘Download if Missing’ ensure that any files included within the Fileset are removed on disassociation from the devices, so naturally files will be removed.

MSI

Windows MSI Filesets are unique. They provide an un-install option for developers and FileWave benefits from this feature, if enabled per Fileset. On disassociation, the MSI un-installer feature may be triggered.

PKG/EXE

Unlike MSI installers, PKG and EXE installers have no built-in un-installer and the files installed are not part of the Fileset, just the PKG or EXE itself can be set with verification settings. In this instance, if the developer of the software does not offer an additional PKG/EXE or script designed to remove the software, it is usual to self-build out some kind of script to remove those items installed.

Apple VPP & Android Play Store

Disassociation of these Filesets triggers a command to remove the Application

Other Files

When software is opened by the user, additional files can be created, which are not part of the original software or included in any of the above installer types. If desired to remove these, then again some kind of self-build method would be required.

Why

Removing software and supporting files keeps devices clean and helps ensure the users are productive, using chosen software and making the sharing of files is simplified; along with other reasons, like security, etc.

When

Identifying and building un-installers is one part of this process, but when the un-installer runs is key. It may seem obvious that the un-installer should run when Filesets are no longer associated, but it is not quite that clear cut.

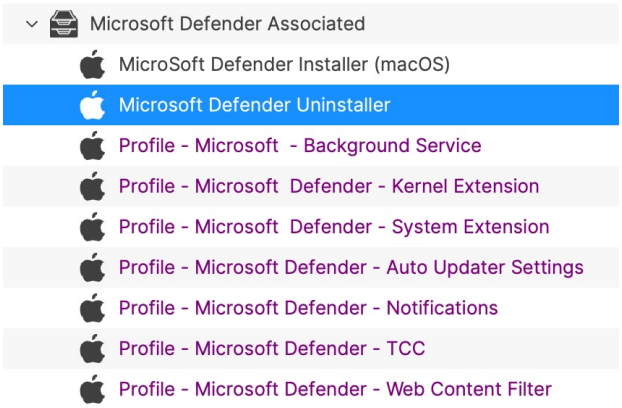
For example, consider the idea of writing a script to remove an application. Typically, a pre or post un-installer script could be set within a Fileset. It may seem natural to add this script within the same Fileset as the installer of the software. However, this can raise concerns.

Over time, developers will provide updates to software. Where autoupdating of software is prevented, new Filesets or Fileset Revisions will be built to push out these updates. Self-Healing handles swapping between Fileset or Revisions, since any files that match between the two differing Filesets remain untouched. However, any containing pre or post un-installer scripts will run at this time, which is likely undesirable.

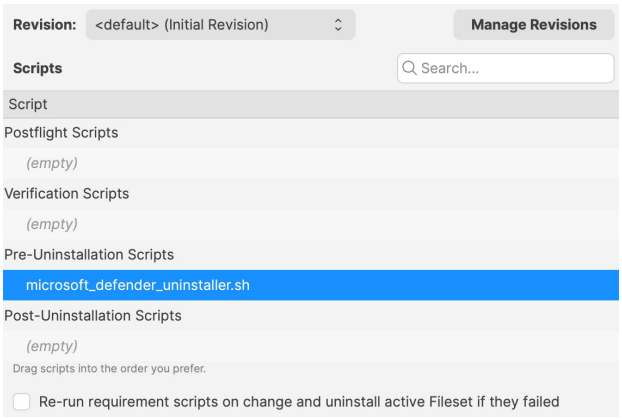
How

There is an installer Fileset, a method to uninstall the Fileset, so how to make sure these run only when desired. One such method is the use of Fileset Groups.

Using an example, the following diagram shows a Fileset Group for Microsoft Defender, which includes an Installer Fileset, some necessary Profiles and an Un-installer Fileset. Association will be with this Fileset Group.



The un-installer is scripted:

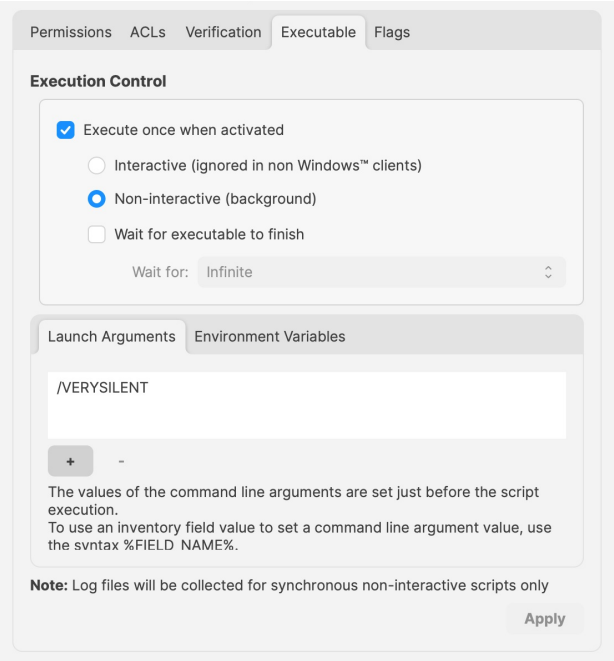


Since the Fileset Group is associated with devices, the installer can easily be updated. Either a new Fileset can be dropped into this group and the current Installer Fileset removed or the Installer could have an alternate Revision added and swapped over. However, when removing the Fileset Group association with devices, this will trigger this Pre-Uninstaller Script and only at this point in time will the software be removed.

PKG/EXE Un-installers

Where developers supply un-installers, pre-packaged in an installer PKG, building out auto running PKG Filesets would require the association to this un-installer only be assigned when disassociation occurs from the installer. Planning this becomes very complex. Instead, the PKG can be added to an empty Fileset and an un-installer script can be used to trigger the PKG file, akin to the above Microsoft Defender example.

Similarly, EXE installers are usually triggered from the Fileset Executable options:



EXEs built to un-install software however have the same issue as PKG Filesets. Pre or Post Un-installer Scripts yet can be used instead though. Again, upload the un-installer EXE into an Empty Fileset and add an un-installer script inside this same Fileset to trigger this EXE; creating a separate un-installer Fileset that can be used in a Fileset Group as demonstrated above.

Revision #1
★Created 24 July 2024 12:40:45 by Sean Holden
Updated 24 July 2024 13:02:19 by Sean Holden