

macOS Erase and Install Fileset (Erase Optional)

Description

Upgrading or refreshing macOS from erasing requires the macOS Install App. The following provides a Fileset to handle either of these cases



Historically, two KBs were provided to handle each of these options. This newer Fileset allows for activation of either from one Fileset. Certain fundamental changes have been made, for example, the hidden upgrade file in `/var/db` is no longer required.

Ingredients

- Custom Field – "macos_instal_flag"
- Full Apple macOS Installer App
- Provided Recipe
- Provided Custom Field

Fileset:

↓ macOS

Custom Field:

↓ macOS

Full installer

Confirm the installer app is the 'Full' installer before continuing. If the app is relatively small e.g. 50MB, re-download the Install macOS Monterey.app. A full installer may be downloaded through Terminal on macOS 10.15+ devices. Example to download the 12.0 full installer to the Applications directory:

```
softwareupdate --fetch-full-installer --full-installer-version 12.0
```

Typically Apple only provides the latest point release of each Major OS version: 10.13.6, 10.14.6, etc. For a full list of available updates, use the following command:

```
softwareupdate --list-full-installers
```

Directions

Fileset

1. Download and import the provided Fileset
2. Download the desired version of the macOS Installer App
3. Add the macOS Installer App into the same folder as the .placeholder fie; the .placeholder file may be removed if desired
4. Select the Instal macOS App > Get Info > Verification, select Ignore At Verify and Apply to Enclosed
5. Edit the Environment Variables for both the Activation Script and the Requirements Script

Fileset Contents

Fileset Contents: upgrade or erase macOS startosinstall template - add install macOS App						
<div> <div> <div>Import File/Folder</div> <div>New Folder</div> <div>Get Info</div> <div>Edit Registry</div> <div>Edit Text</div> <div>Export Files</div> <div>Delete</div> <div>Take Control</div> </div> <div> Revision: <default> (Initial Revision) <div>Manage Revisions</div> </div> <div> <input checked="" type="checkbox"/> Hide unused folders <div></div> </div> </div>						
Name	Size	ID	Access	User	Verification	Group
usr		230	rwxr-xr-x	root		wheel
local		231	rwxrwxr-x	root		wheel
etc		232	rwxrwxr-x	root		wheel
FileWaveInstallers		666	rwxrwxr-x	root		wheel
macos_installer		487394	rwxrwxr-x	root		wheel
.placeholder	0 B	487862	rw-r--r--	root	Self Healing	staff
var		227	rwxrwxrwx	root		wheel
scripts		228	rwxrwxr-x	root		wheel
412644		472116	rwxrwxr-x	root		wheel
startosinstal_m1&intel.sh	3 kB	487861	rwxr--r--	root	Self Healing	wheel
startosinstall_requirements.sh	2.6 kB	487404	r-x-----	root	Self Healing	wheel

Environment Variables

The Environment Variables must be edited, providing appropriate values

- M1 devices provide an additional complication when choosing to script the installation of macOS Instal Apps. In this instance, a local macOS Administrator Username and Password must be passed to the script. The two Variables 'local_admin' and 'admin_pass' should have their Values adapted respectively. Intel device has no such consideration and will ignore these values if set.

Requirement Script – statosinstall_requirements.sh

Key values that require editing (in bold):

startosinstall_requirements.sh

Kind: File

Created: Wed May 25 2022 04:24 pm

Modified: Wed May 25 2022 04:24 pm

PermissionsACLsVerificationExecutableFlags

Execution Control

☒ Execute at requirements step

☐ Interactive (ignored in non Windows™ clients)

☐ Non-interactive (background)

☒ Wait for executable to finish

Wait for: Infinite

Launch Arguments

Environment Variables

Variable	Value
api_key	REPLACE_ME
device_id	%filewave_id%
local_admin	REPLACE_ME

Reset

Reset All

The values of the environment variables are set just before the script execution.
To use an inventory field value,use the syntax %FIELD_NAME%.
For instance: MY_VAR: foo-%asset_tag%

Note: environment variable names are case insensitive in Windows

Note: Log files will be collected for synchronous non-interactive scripts only

Apply

Click the lock to take control of this Fileset

Activation Script – startosinstal_m1&intel.sh

Key values that require editing (in bold):

startosinstal_m1&intel.sh

Kind: File

Created: Wed May 25 2022 10:32 pm

Modified: Wed May 25 2022 10:32 pm

PermissionsACLsVerificationExecutableFlags

Execution Control

☒ Execute once when activated

☐ Interactive (ignored in non Windows™ clients)

☒ Non-interactive (background)

☒ Wait for executable to finish

Wait for: Infinite

Launch Arguments

Environment Variables

admin_pass	REPLACE_ME
api_key	REPLACE_ME
device_id	%filewave_id%
installer_name	Install macOS Monterey
local_admin	REPLACE_ME

Reset

Reset All

The values of the environment variables are set just before the script execution.
To use an inventory field value,use the syntax %FIELD_NAME%.
For instance: MY_VAR: foo-%asset_tag%

Note: environment variable names are case insensitive in Windows

Note: Log files will be collected for synchronous non-interactive scripts only

Apply

Click the lock to take control of this Fileset

- api_key – Copy a Token (base64) from a chosen FileWave Administrator's 'Application tokens'

- admin_pass – password for a local admin on the target macOS device (Apple M1 only)

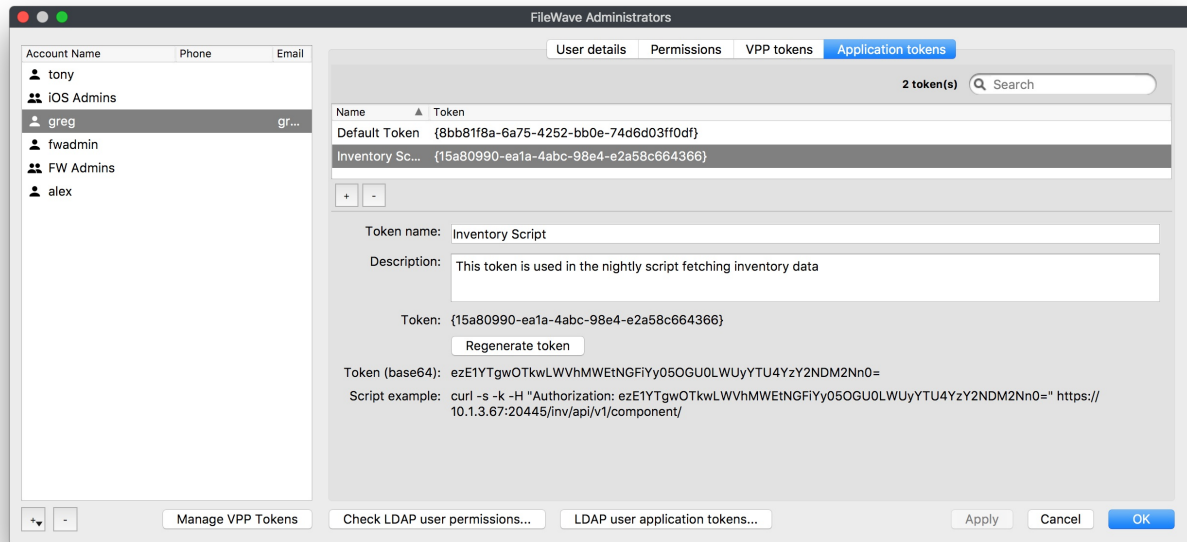
- local_admin – username for a local admin on the target macOS device (Apple M1 only)

- local_admin – username for the matching local admin as password on the target macOS device (Apple M1 only)
- installer_name – the name of the App added, e.g. 'Install macOS Big Sur', 'Install macOS Monterey'
- api_key – Copy a Token (base64) from a chosen FileWave Administrator's 'Application tokens'

'installer_name' in Fileset is also set as REPLACE_ME, but the value is left as 'Install macOS Monterey' as an example in the Fileset image.

Application Token Location

The FileWave Administrator's Application Token (base64) can be found under FileWave Central -> Assistants -> Manage Administrators.



Custom Field

Custom Field has 4 options:

- NA – default value, the Fileset will not do anything with this setting
- ERASE – the device will have the OS erased and install the provided version of macOS
- INSTAL – the device will upgrade to the provided version of macOS
- FAILED – the device will update the Custom Field flag to FAILED if something unexpected occurs. No further attempts to run the Fileset will action anything whilst this is still the case

Either 'Assign to all devices' or select appropriate devices to associate the Custom Field.

- ✓ Consider making Smart Group associations based upon the Custom Field value

Activation

Create a Smart Group based upon macos_instal_flag value being either ERASE or INSTAL



Set the associated Custom Field for a device to one of these two values depending upon the experience you would desire.



Selecting a group of devices to associate and/or alter the Custom Field value, will action that new value for all devices currently within that group. The video shows an example of setting the Custom Field value as INSTAL, with the 2 devices within the macOS 10.13 group.

Using a group in this way will have no impact on the Custom Field values for devices leaving or joining the group afterward.

Since it is likely a subsequent installation attempt will be desired upon failure (once the reason for failure has been addressed, e.g. not enough disk space), a Smart Group could be set with the Custom Field value of FAILED if desired.

To re-run a failed attempt, after addressing the reason for the failure (e.g. freeing up disk space), reset the Custom Field value to the appropriate ERASE or INSTAL value and choose to 'Reinstall Fileset'

Notes

That seemed quite complex, why?

Whilst the Fileset is associated it may trigger an action. When a device is erased, for example, any reference to the installation taking place is now lost. The consequence of this is that the device will attempt to reinstall the Fileset over and over again. Custom Fields are rescuing the process, by ensuring that once the process has been completed, there will be no re-attempt to upgrade or erase the device again since the association will no longer be in place.

By creating a FAILED Custom Field value, devices are also automatically reporting any such failure, and queries or groups may be pre-built to observe and address any such experience.

By setting the macOS Installer App as leave behind, if the installation has not been completed, but is no longer associated, this will ensure the installer remains on the device for subsequent attempts. Additionally, since Apple automatically removes the macOS Installer App after an upgrade, if the Fileset were still associated there would be no re-attempt to download the installer.

Filesets that error will automatically re-attempt installation as standard. The requirement script is designed to report an exit code of 210 to overrule this. Even if the Requirement Script fails, it will report success and will neither continue download/activation nor will it re-attempt without intervention. In this case, the client will update the Custom Field instead to report the failure. The script log should show the output of the command.



Example log message:
/usr/local/etc/macos_instal.log

|main|CUSTOM|CLIENT|startosinstall_requirements Exiting. Flag set as:

In this example, the Custom Field Flag reported has no value. This should either mean the Custom Field is not associated with the device or the script has failed to read the value. If the latter, check the Environment Variables to ensure they are correct.

🕒Revision #4

★Created 16 July 2023 03:03:31 by Josh Levitsky

✍Updated 17 October 2024 16:41:02 by Andrew Kloosterhuis