Microsoft Windows

Microsoft Windows is a widely used operating system known for its user-friendly interface and compatibility. With versions like Windows 10 and Windows 11, it offers a customizable experience, supports various tasks, and provides a vast software ecosystem. Windows is trusted for its versatility, regular updates, and broad application support, making it a popular choice for both work and personal use.

- Active Directory Join (Windows)
- Adding A Printer for All Users (Windows)
- Create local admin accounts on Windows
- Deploy SSL Certificates (Windows)
- Installing Windows Fonts
- Local Group Policy Object Utility (Windows EXE)
- Rename Windows Hostname based FileWave Client Name
- Storing the BitLocker volume keys using a Custom Field
- Upgrade Windows 10 and 11
- Windows 11 Compatible Devices
- Windows 11 support in FileWave 14.7+
- Notify Users with a dialog (Windows)
- Using Native Windows Tools to Troubleshoot

Active Directory Join (Windows)

Description

This Fileset is designed to bind Windows computers to a Directory structure. By associating this Fileset the binding process can be automated.

Ingredients

- Active Directory domain
- Windows 10 or 11 filewave client
- FileWave Admin

Directions

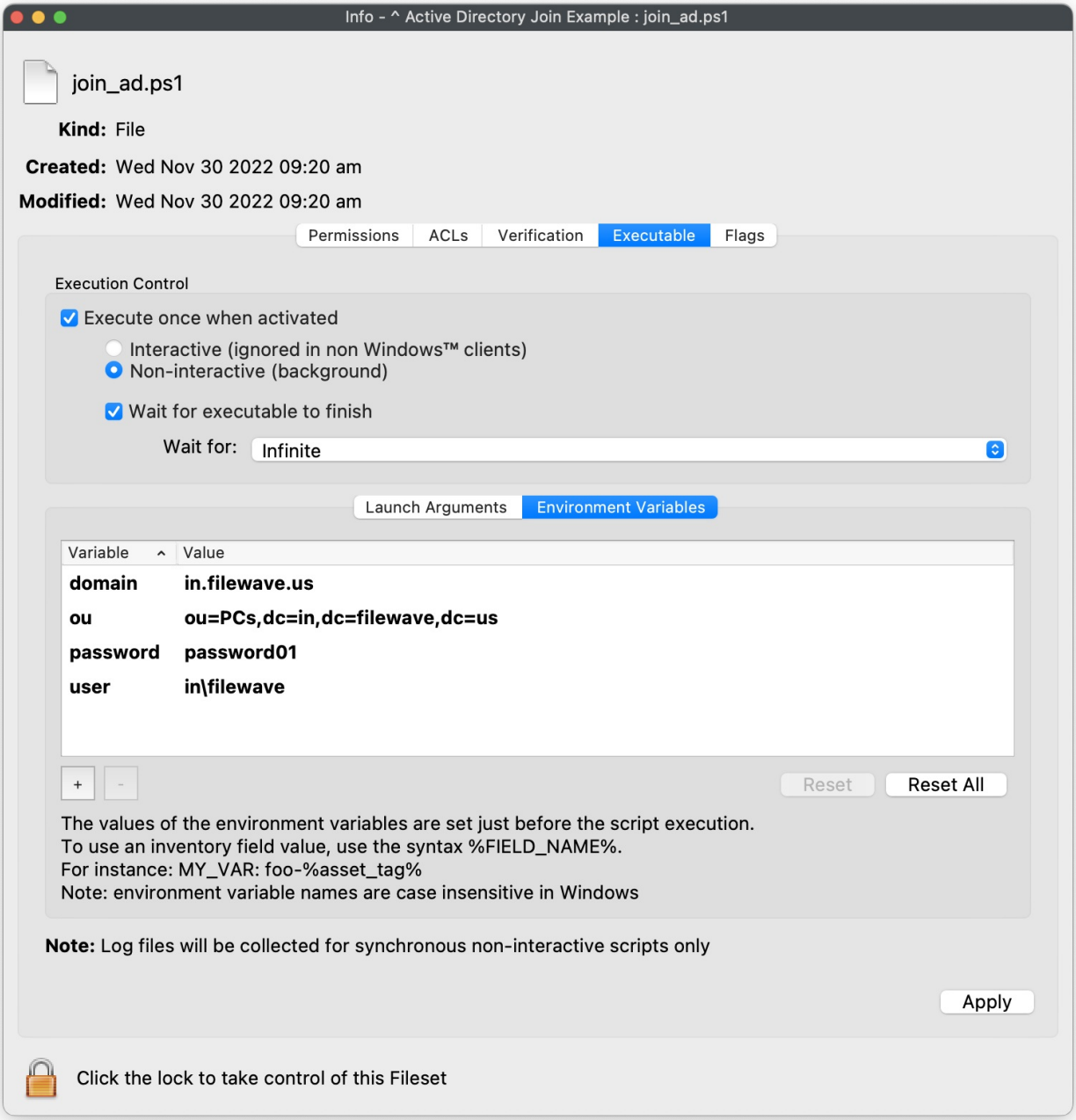
1. Download the Active Directory join fileset template: Active Directory Join.fileset.zip
2. Unzip and import the fileset into FileWave Admin.
3. Open the Fileset, highlight the join_ad.ps1 script and choose Get Info > Executable > Environment Variables.
4. Modify these variables to reflect the Active Directory environment:

```
user
password
domain
ou
```

⚠ THIS SCRIPT WILL FORCE THE MACHINE TO RESTART. IF THAT IS NOT THE BEHAVIOR THAT IS DESIRED REMOVE THIS LINE FROM THE JOINDOMAIN.PS1 FILE:
Restart-Computer -Force

❗ THIS SCRIPT WILL DELETE ITSELF ONCE IT HAS RUN ON THE CLIENT MACHINE LOCALLY.

Example:



For the user, please use full path like
e.g. "domain\username"

Save changes and associate the Fileset to either Windows 10 or 11 client machines!

Adding A Printer for All Users (Windows)

The task at hand seems simple enough...install a printer for a Windows user for a printer on a Print Server. In our example, I'll use a print share called BigDill on a print server named Arkone.

Easy, right? A quick web search for "powershell add printer" takes me to the add-printer cmdlet, and it is pretty easy to use for a print server. The command looks like this (don't use this example!):

```
import-module printmanagement
add-printer -ConnectionName \\arkone\BigDill
```

I tested it locally outside of FileWave in the PowerShell ISE and it worked fine. I ran the command, and it added the printer for me. So, I created a fileset for the exact same thing and tried it out. The result: nothing whatsoever. The script seemed to run fine, but the user logged in didn't see a new printer!

So, why did this happen? For two important reasons:

1. The Add-Printer cmdlet is great, but it ONLY adds a printer for the current user
2. When FileWave runs a script, it is always run under the context of the System account

When I investigated further by opening a command prompt as system, I found that in fact my fileset had run fine, and added the printer, but only for the system account.

> Testing Scripts for Use in FileWave
> Review our KB Script Best Practices which demonstrates the use of psexec to run scripts on Windows as if they were run through FileWave (as System User and 32bit).

So, a little more research was required, and PowerShell in this instance is not the answer. Instead, we are going to use a command-line utility in a batch file called printui.exe. PrintUI can be used in many ways: Microsoft PrintUI Documentation

We won't get into all of the options of this command here, but printui can add a printer globally for all users using the /ga command line option (/gd is a global delete if you happen to want to add a removal script as well). So our new batch file (Activation Script) code looks like this:

```
@echo off
printui /ga /n\\arkone\BigDill
exit 0
```

And our results, in this case, were excellent...the printer is added for every user at their next login. (Given this, you may want to make this a reboot fileset)

And, for completeness' sake, if we wanted to add a post-uninstallation script to "clean-up" if this fileset were removed, we could do:

```
@echo off
printui /gd /n\\arkone\BigDill
exit 0
```

Related Content

- Script Best Practices

Create local admin accounts on Windows

Description

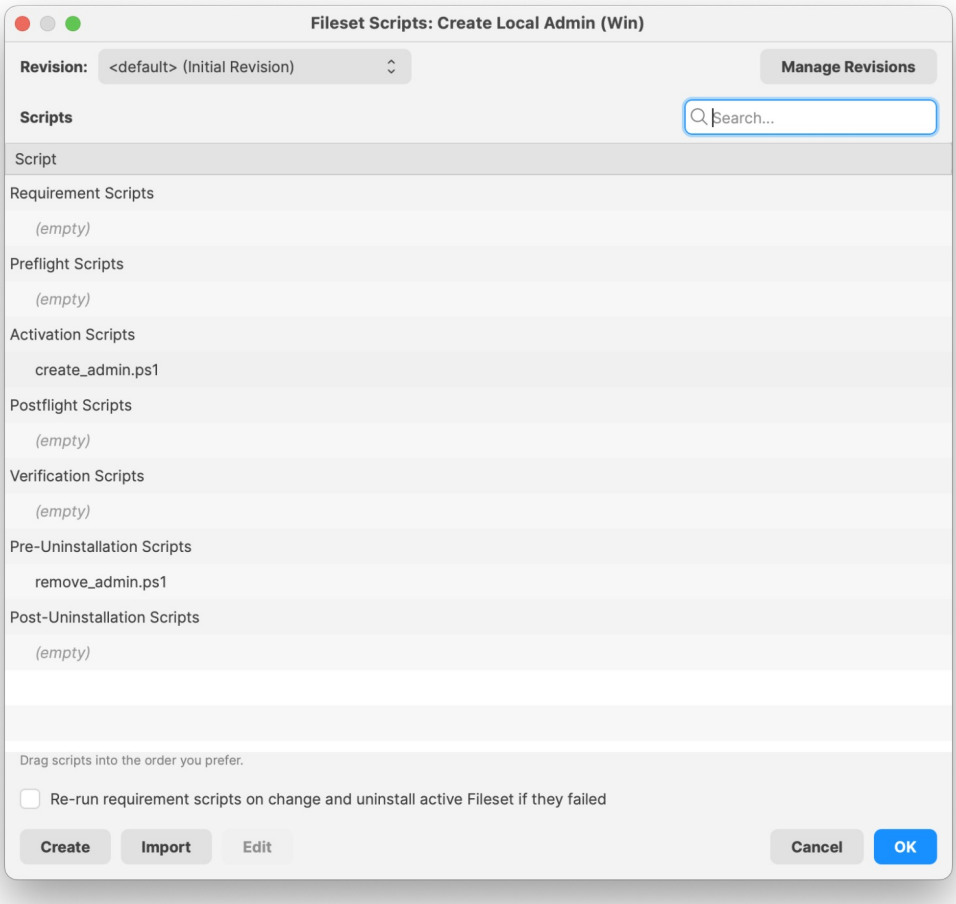
Need to manage local admin accounts on your Windows devices? FileWave has you covered. Below is the recipe with Fileset to create your local admin account with username, password and full name. In addition, if needing to remove the local admin account, there is a removal script included.

Ingredients

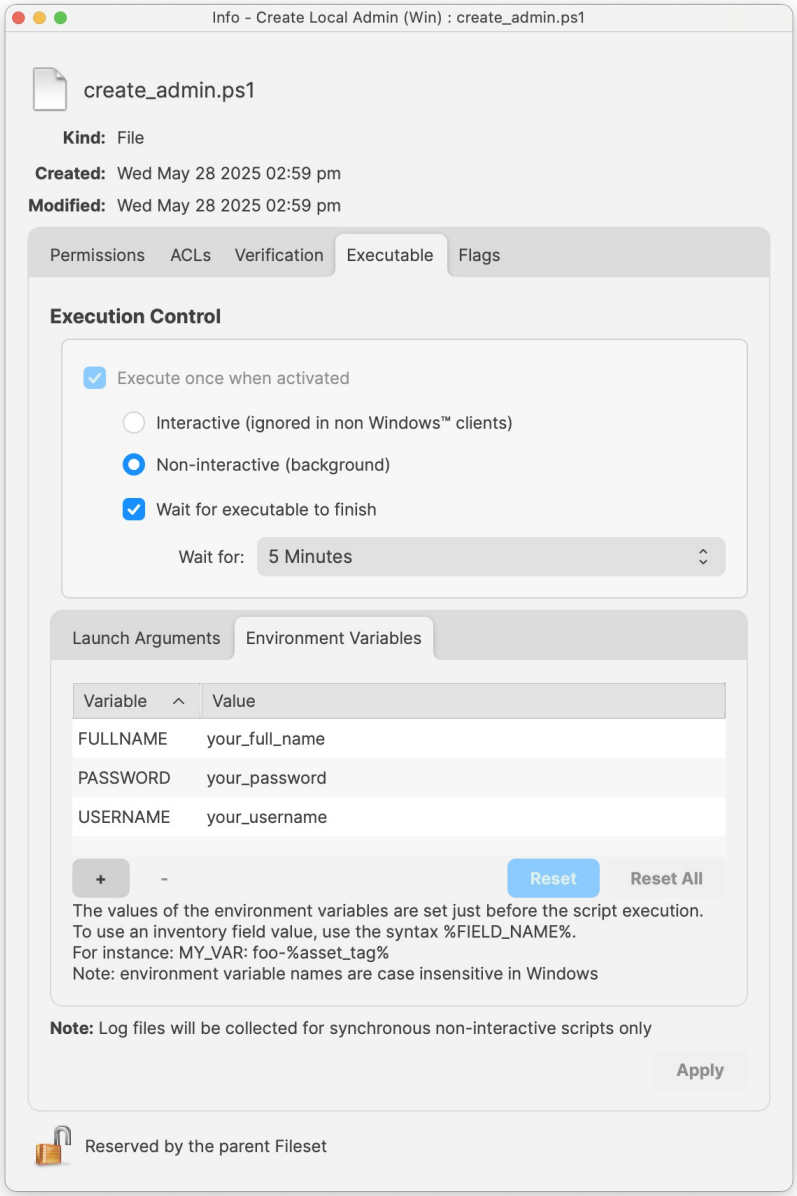
- FW Admin
- Create Local Admin (Win).fileset.zip

Directions

1. Download and unzip the Fileset
2. Import into your server via FileWave Admin
3. Highlight the Fileset and select "Scripts" for this Fileset
4. Select the create_admin.ps1 to open the script properties

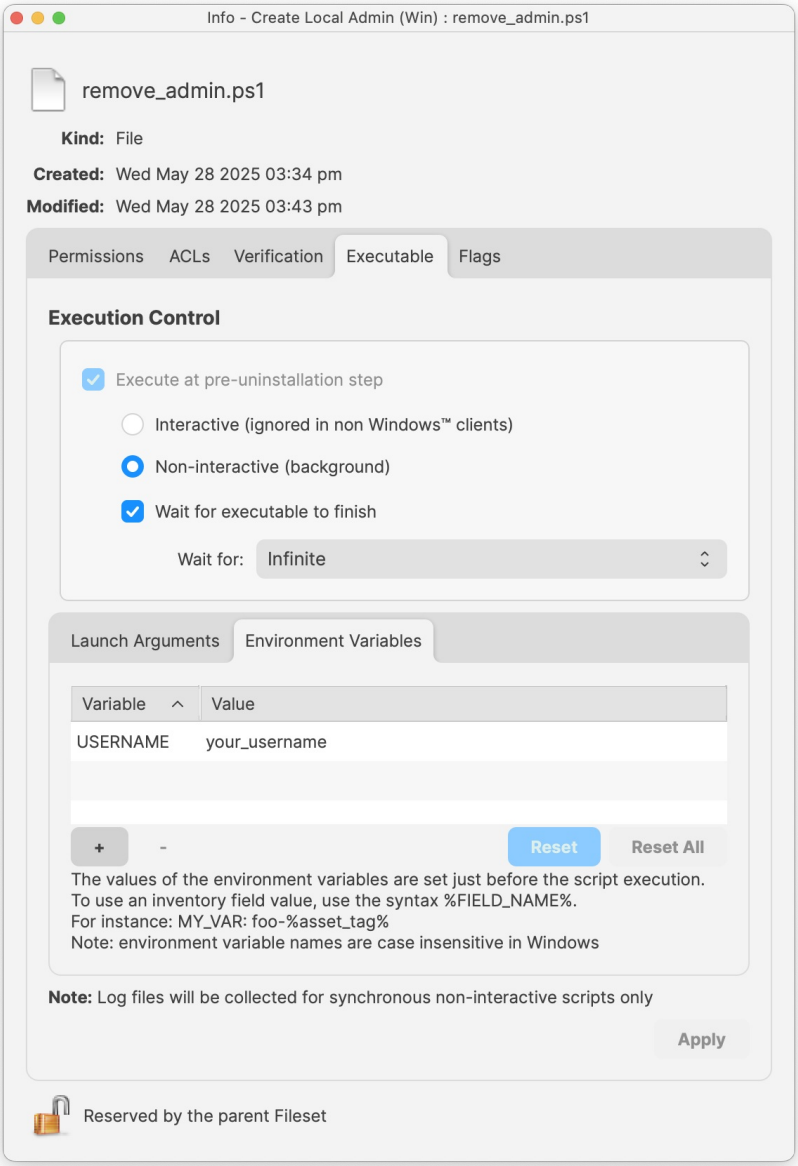


5. Enter in your local admin Username, Password and Full Name for the desired account



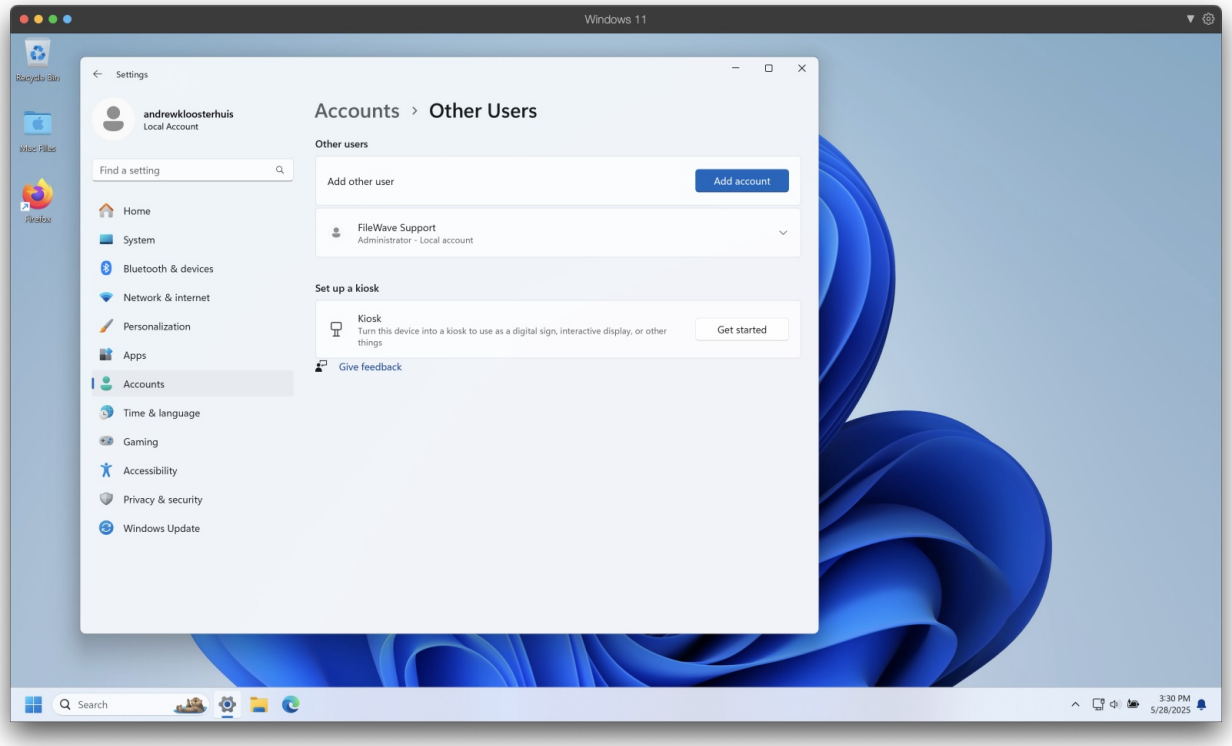
6. Repeat the process for the remove_admin.ps1 and enter in the desired Username to remove

> ⚠ The remove_admin.ps1 script environment variable needs to match the Username found on the machine or in the create_admin.ps1. If it does not match it will not successfully remove the admin account.



7. Close the Script window to Save
8. Assign to a test device
9. Perform a Model Update to deploy

Confirmation of local admin created, you may open the Windows Settings > Accounts > Other Users to view the newly created local admin account.



Notes

Both scripts will output their executed tasks for detailed logging. These logs may be found in:

```
C:\ProgramData\FileWave\Logs\
```

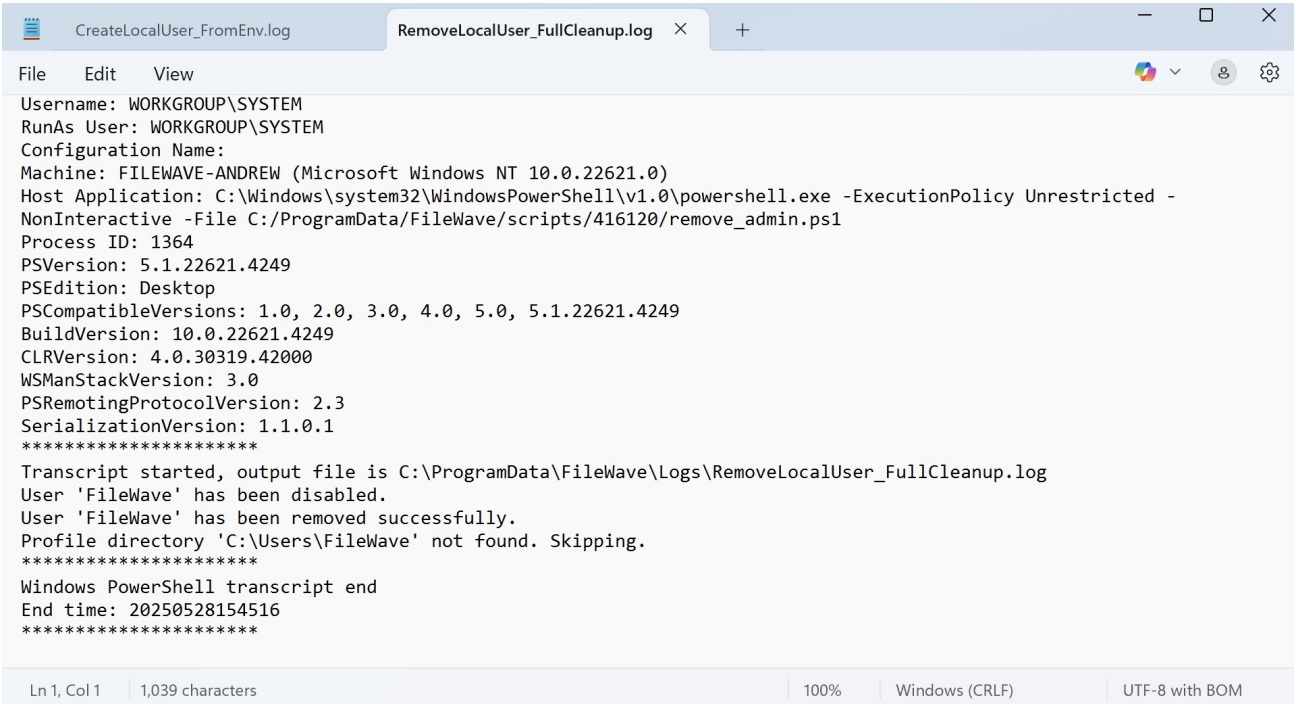
The create_admin script log will be labeled: CreateLocalUser_FromEnv.log

CreateLocalUser_FromEnv.log

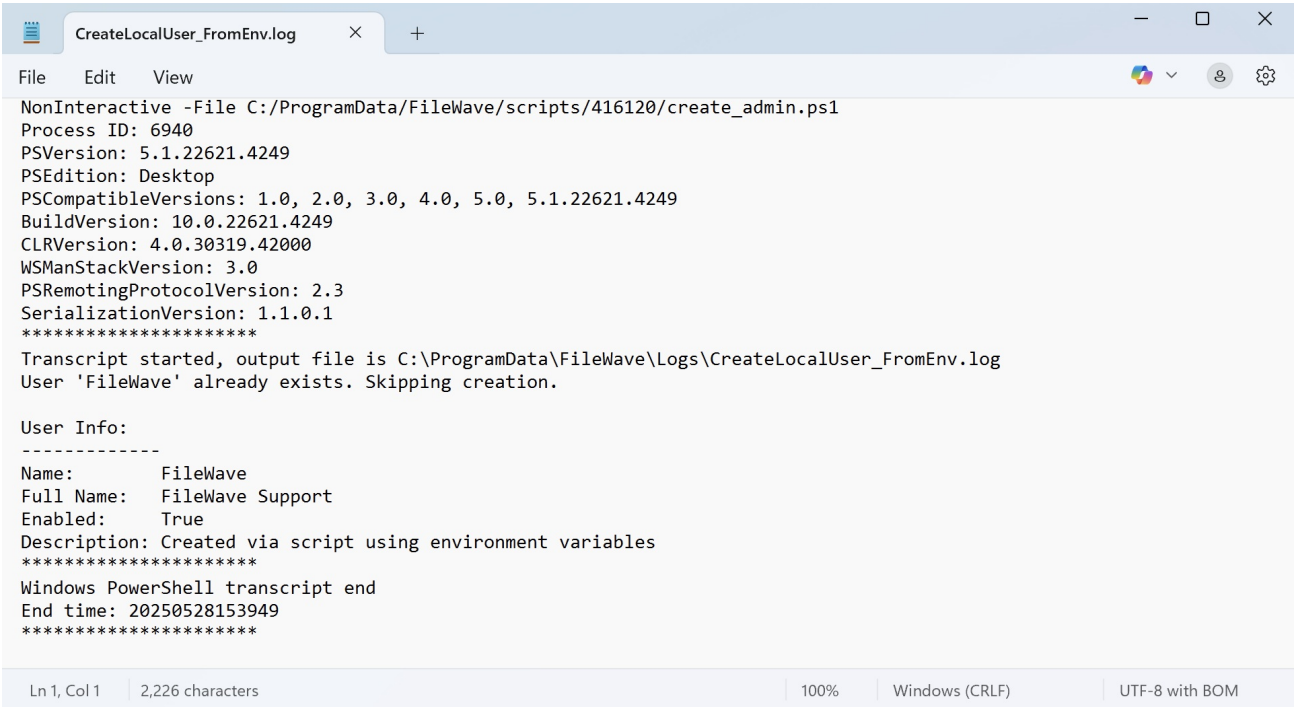
File Edit View

```
PSVersion: 5.1.22621.4249
PSEdition: Desktop
PSCompatibleVersions: 1.0, 2.0, 3.0, 4.0, 5.0, 5.1.22621.4249
BuildVersion: 10.0.22621.4249
CLRVersion: 4.0.30319.42000
WSManStackVersion: 3.0
PSRemotingProtocolVersion: 2.3
SerializationVersion: 1.1.0.1
**********************
Transcript started, output file is C:\ProgramData\FileWave\Logs\CreateLocalUser_FromEnv.log

User 'FileWave' created.
User 'FileWave' added to 'Administrators' group.

User Info:
-------------
Name:        FileWave
Full Name:   FileWave Support
Enabled:     True
Description: Created via script using environment variables
**********************
Windows PowerShell transcript end
End time: 20250528152957
**********************
```

The remove_admin script log will be labeled: RemoveLocalUser_FullCleanup.log



The create admin script will skip if there is an username already exists.



Related Content

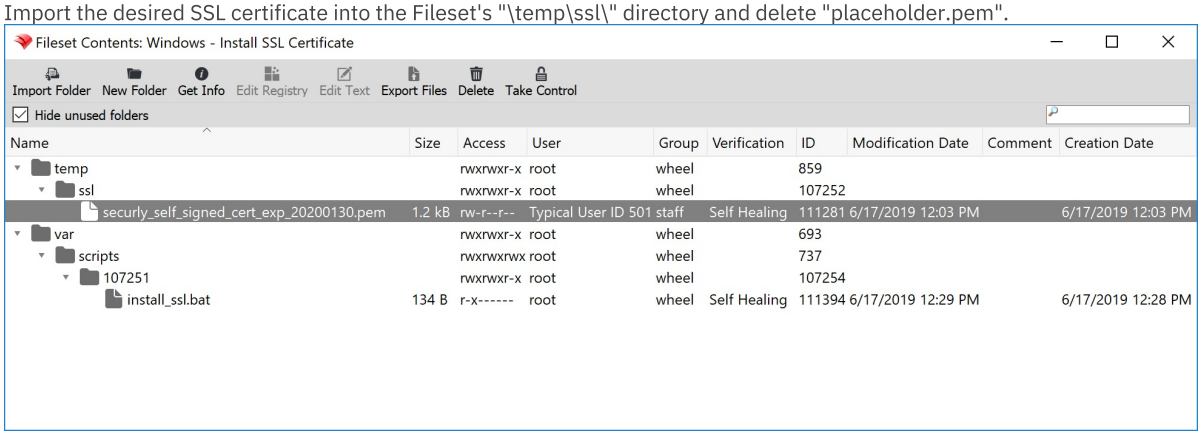
- How to Create Local User Accounts on macOS

Deploy SSL Certificates (Windows)

Deploy SSL certificates to Windows "Trusted Root Certification Authorities" certificate store for use in secure services such as web content filters.

Step-by-step guide

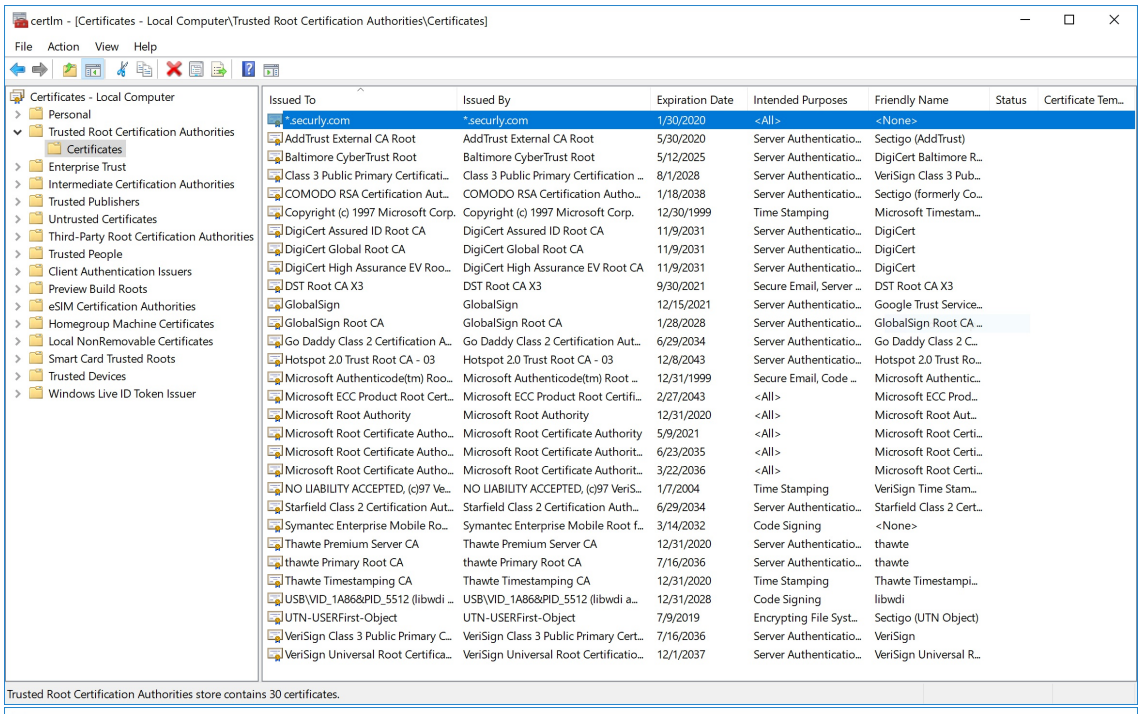
- Download, unzip, and import the following Fileset Template into FileWave Central - Windows - Install SSL Certificate.fileset.zip
- Import the desired SSL certificate into the Fileset's "\temp\ssl\" directory and delete "placeholder.pem".



- Select the "install_ssl.bat" file, click "Edit Text", and replace "placeholder.pem" with the full file name of the newly uploaded certificate.

> This script will add the desired SSL certificate to the Windows "Trusted Root Certification Authorities" certificate store for both the Local Machine (-enterprise) and the Current User (-user).

- Associate and deploy the Fileset to a test machine and verify the installation on the Windows machine.
  - Open "certlm.msc" for the Local Machine (-enterprise) Certificate Store or "certmgr.msc" for the Current User (-user) Certificate Store via Windows Run dialog (Win + R) or Command Prompt.
  - Navigate to "Trusted Root Certification Authorities>Certificates" and verify the name of the newly added certificate.



> Additional Information
> More information and options for the "certutil" program can be found on Microsoft Docs.

Related articles

- APNs Certificate Creation & Renewal on Windows Computers

Installing Windows Fonts

Description

Windows font installation is not as simple as adding files to a folder. As well as copying files, the registry requires editing. The following Fileset will add Fonts to Windows systems.

> TTF
> The Fileset has been tested with TTF on Windows Pro 10, 1803 and 1903

Registry Editing

> This Fileset edits the Windows registry. Follow instructions carefully to ensure only the required Font files exist in the suggested folder.

Information

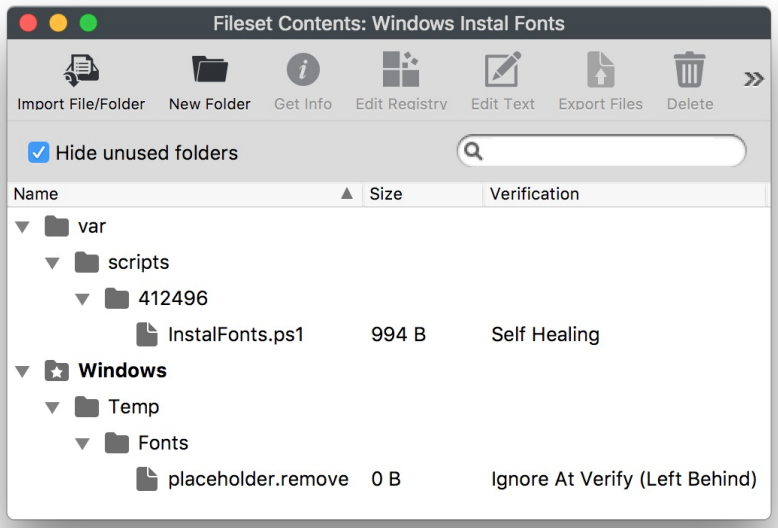
Download and import the Fileset, then ensure to edit appropriately, as per the directions, for desired fonts

[Windows Instal Fonts.fileset.zip]

Description

The Fileset consists of:

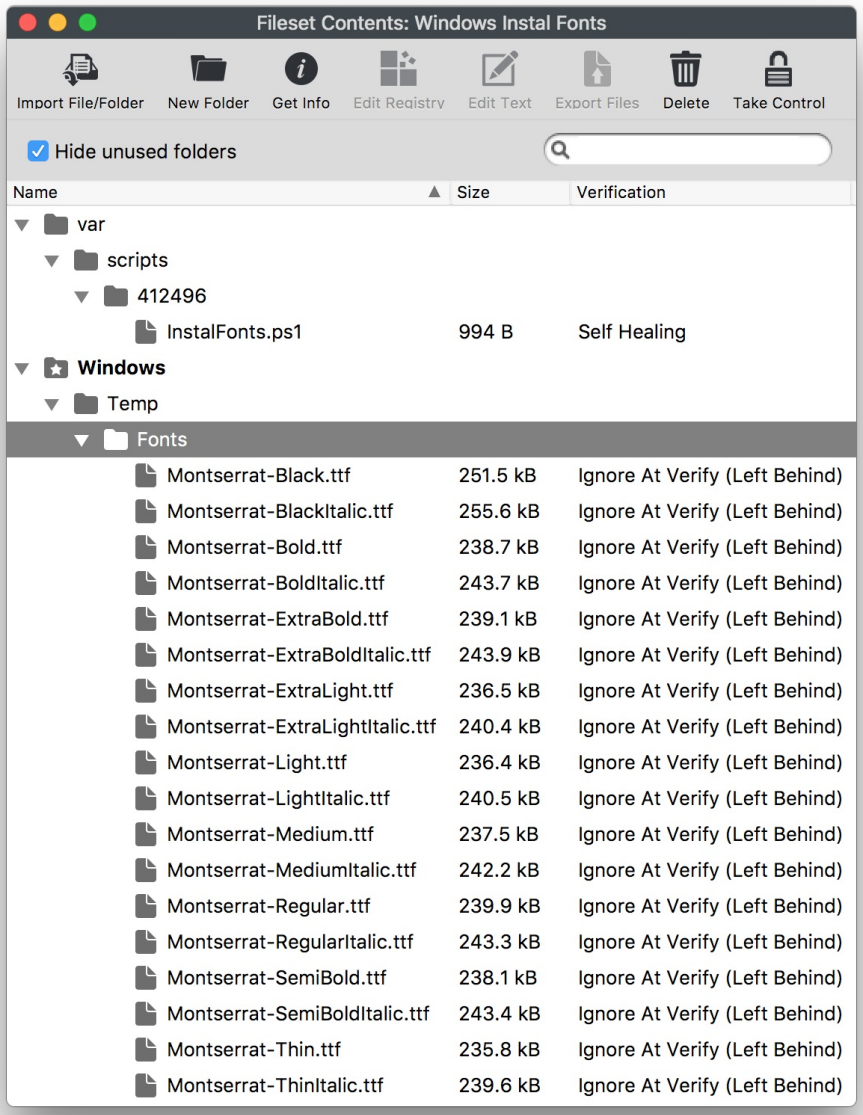
- An Activation Script
- A folder to provide the fonts; with a placeholder file



Adding Fonts

- Select the Fonts in a macOS Finder or Windows Explorer Window
- Drag them into the Fileset folder Windows > Temp > Fonts
- Select the Fonts folder in the Fileset, choose Get Info > Verification and set Ignore At Verify; Apply to Enclosed
- Remove the 'placeholder.remove' file

Example, Monsterrat Font:



Deployment

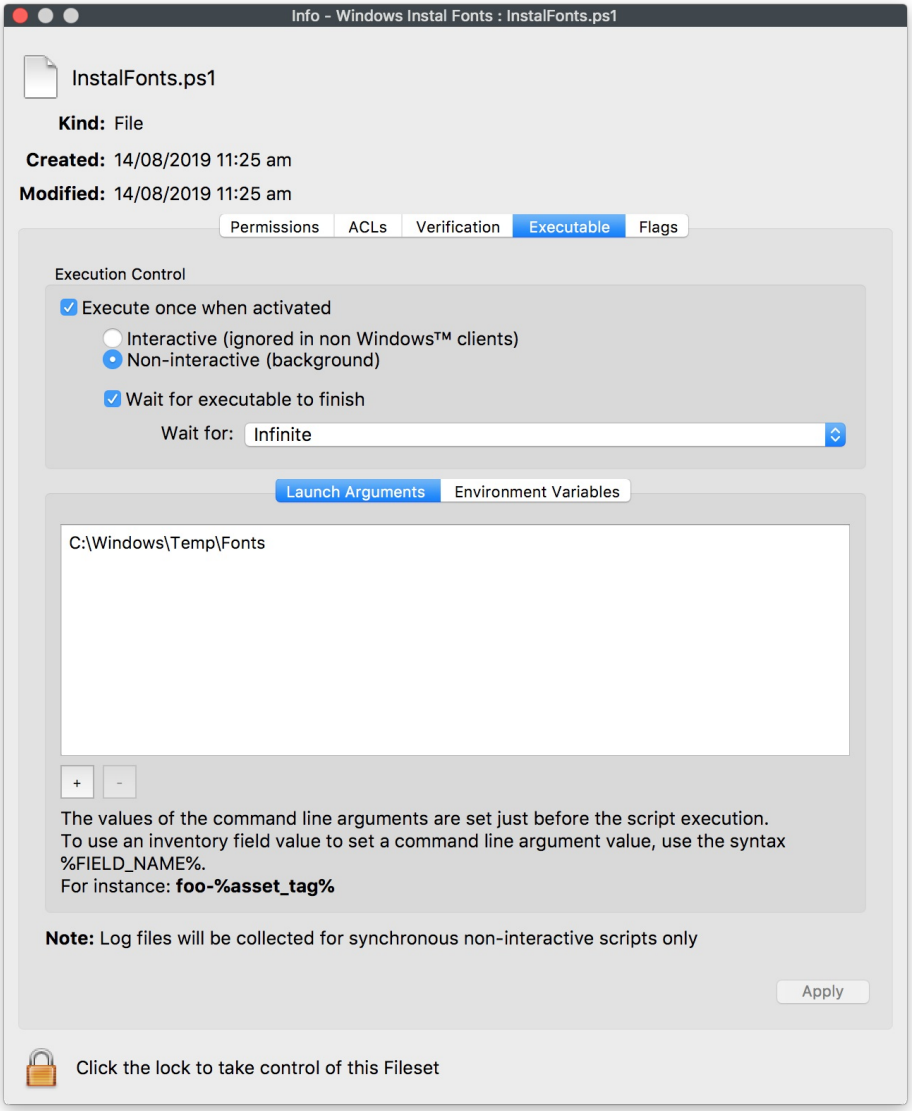
On Association and Activation, the script will:

- Copy ALL files from C:\Windows\Temp\Fonts to C:\Windows\Fonts (if they do not already exist)
- For each file copied, a registry entry will be created (if one does not yet exist)
- Finally the script will remove each file from the temporary directory: C:\Windows\Temp\Fonts

Changing Temporary Folder

Should a different temporary folder be desirable:

- Drag the fonts to the desired location in the Fileset
- Change the Launch Argument to match: InstalFonts.ps1 > Get Info > Executable



As always, test on appropriate devices before deploying en masse.

Local Group Policy Object Utility (Windows EXE)

What

Local Group Policy Object Utility or LGPO.exe is a new command-line utility to automate the management of local group policy. It replaces the no-longer-maintained LocalGPO tool that shipped with the Security Compliance Manager (SCM), and the Apply_LGPO_Delta and ImportRegPol tools.

When/Why

Many of the features help import required settings to your devices meeting organization policies for security compliance.

Features:

- Import settings into local group policy from GPO backups or from individual policy component files, including Registry Policy (registry.pol), security templates, and advanced auditing CSV files.
- Export local policy to a GPO backup.
- Parse a Registry Policy (registry.pol) file to readable "LGPO text" directly to the console or redirected to a file which can edited and imported into local policy.
- Build a new Registry Policy (registry.pol) file from "LGPO text".
- Enable group policy client side extensions for local policy processing.

How

The zip file contains the LGPO installer:

LGPO.zip LGPO.exe v3.0 - Local Group Policy Object utility

LGPO.exe has four modes:

1. Import and apply policy settings;
2. Export local policy to a GPO backup;
3. Parse a registry.pol file to "LGPO text" format;
4. Build a registry.pol file from "LGPO text".

To apply policy settings, use the LGPO.exe commands below, where one or more of the following (each of which can be repeated):

```
/g path                              import settings from one or more GPO backups under "path"
/m path\registry.pol     import settings from registry.pol into machine config
/u path\registry.pol      import settings from registry.pol into user config
/s path\GptTmpl.inf     apply security template
/a[c] path\Audit.csv     apply advanced auditing settings; /ac to clear policy first
/t path\lgpo.txt              apply registry commands from LGPO text

/e <name>|<guid>       enable GP extension for local policy processing; specify a GUID, or one of these names:
              **  "zone" for IE zone mapping extension
              **  "mitigation" for mitigation options, including font blocking
              **   "audit" for advanced audit policy configuration

/boot                                reboot after applying policies
/v                                       verbose output
/q                                       quiet output (no headers)
```

Some example GPO policies are listed below to use. Be sure you are updating the correct path and names for your GPOs.

To create a GPO backup from local policy:

```
LGPO.exe /b path [/n GPO-name]

/b path                        Create GPO backup in "path"
/n GPO-name           Optional GPO display name (use quotes if it contains spaces)
```

To parse a Registry.pol file to LGPO text (stdout):

```
LGPO.exe /parse [/q] {/m|/u} path\registry.pol
```

```
/m path\registry.pol         parse registry.pol as machine config commands
/u path\registry.pol          parse registry.pol as user config commands
/q                                        quiet output (no headers)
```

To build a Registry.pol file from LGPO text:

```
LGPO.exe /r path\lgpo.txt /w path\registry.pol [/v]

/r path\lgpo.txt              Read input from LGPO text file
/w path\registry.pol     Write new registry.pol file
```

Related Content

- Microsoft Download Security Compliance Toolkit and Baselines

Digging Deeper

- Microsoft Security Compliance Toolkit Documentation

Rename Windows Hostname based FileWave Client Name

Description

This Fileset will automatically rename the Windows Hostname of your Clients whenever you rename a Client via the FileWave Admin.

With this Fileset associated, the FileWave Client Name will have the utmost precedence. For example, if the Windows Hostname is changed directly from Windows settings, Active Directory, or any other method, the Fileset will change the hostname back to what is listed as the FileWave Client Name.

You must provide a FileWave Client Name that conforms to the Windows naming convention. Please also avoid using spaces, underscores, or any other special characters that will not be directly translated in the Windows NETBIOS name. There is no name checking logic in this script.

Good name examples:

- FILEWAVE-PC
- TCE-3453234
- PRINT-SRV

Bad name examples:

- Jerry's Desktop Computer
- GO_BEARS!
- THISNAMEISMUCHLONGERTHAN15CHARACTERS

Ingredients

- FileWave Client 13.2.3+
- Windows - Rename Windows Hostname (No Domain).fileset.zip
- Windows - Rename Windows Hostname (Domain Joined).fileset.zip

Updated Filesets using PowerShell

- Windows - Rename Windows Hostname (No Domain) PowerShell.fileset.zip
- Windows - Rename Windows Hostname (Domain Joined) PowerShell.fileset.zip

Machines not joined to Active Directory

1. Unzip and Import "Windows - Rename Windows Hostname (No Domain)" Fileset into FileWave Admin.
2. Associate Fileset to one or all devices.
3. Rename the device via FileWave Admin.
4. Update Model
5. Wait ~2 minutes (tickle time) for Fileset to activate.
    1. Future name changes will apply with ~2 minutes. This interval is "hardcoded" and is when Preflight scripts will launch.
6. A pop-up will notify the user of reboot, and the device will reboot in 60 seconds.
    1. If you want to customize the reboot prompt or timeout, edit the following bold text in the "rename_windows_hostname_nodomain.bat" script:
        1. %windir%\System32\shutdown.exe /r /t 60 /c "Computer renamed to %fwClientName%. Rebooting in 60 seconds." /f /d p:4:1
7. The machine will be renamed based on FileWave Client Name upon reboot.
8. FileWave Client communication will not be affected.

Machines joined to Active Directory

1. Unzip and Import "Windows - Rename Windows Hostname (Domain Joined)" Fileset into FileWave Admin.
2. Modify the "rename_hostname_domain.bat" script to update Active Directory username/password in bold:
    1. echo $user = "DOMAIN\USERNAME" > %~dp0rename.ps1
    2. echo $pass = ConvertTo-SecureString "PASSWORD_HERE" -AsPlainText -Force >> %~dp0rename.ps1
3. Associate Fileset to one or all devices.
4. Rename the device via FileWave Admin.
5. Update Model
6. Wait ~2 minutes (tickle time) for Fileset to activate.
    1. Future name changes will apply with ~2 minutes. This interval is "hardcoded" and is when Preflight scripts will launch.
7. A pop-up will notify the user of reboot, and the device will reboot in 60 seconds.
    1. If you want to customize the reboot prompt or timeout, edit the following bold text in the "rename_windows_hostname.bat" script:
        1. %windir%\System32\shutdown.exe /r /t 60 /c "Computer renamed to %fwClientName%. Rebooting in 60 seconds." /f /d p:4:1
8. The machine will be renamed based on FileWave Client Name upon reboot.
9. FileWave Client communication will not be affected.

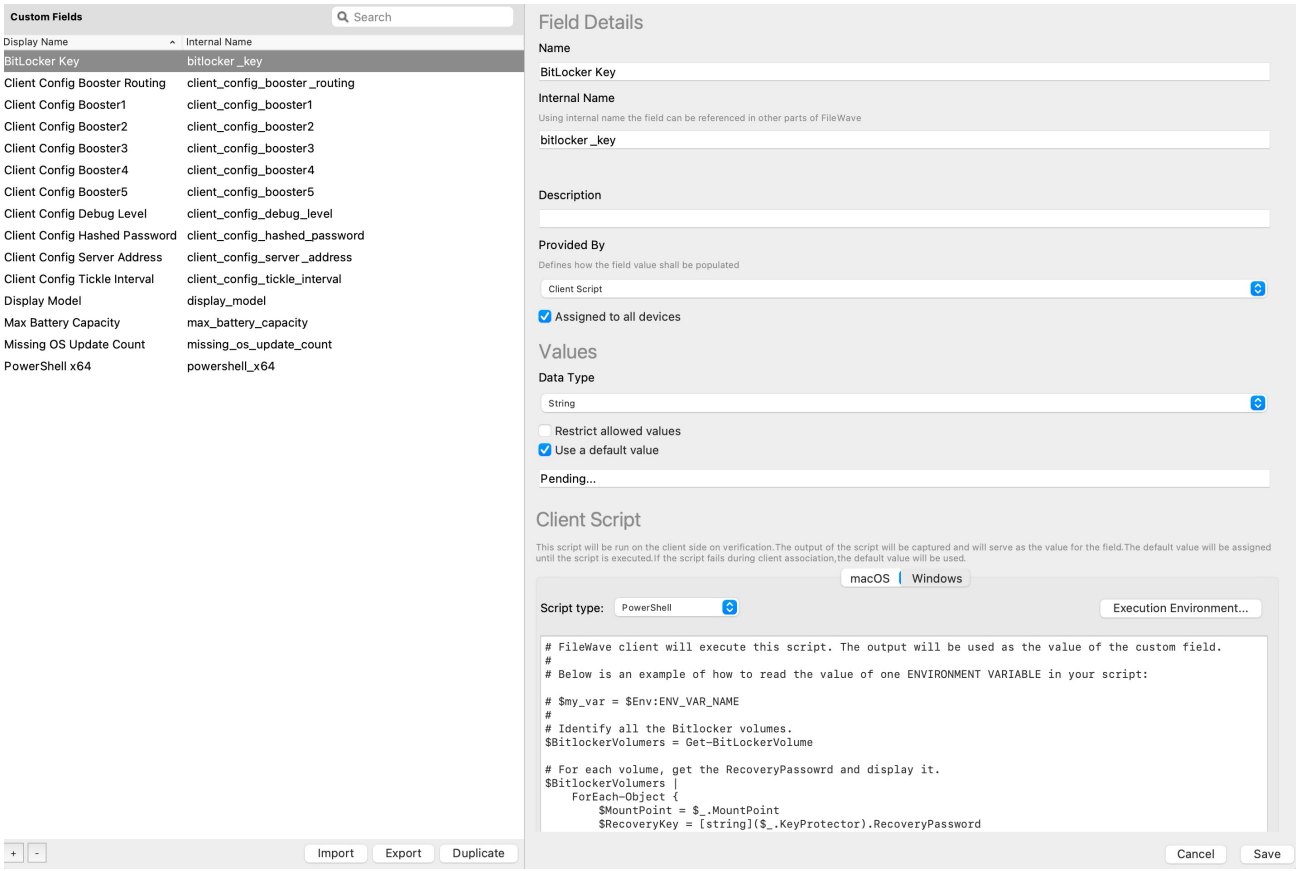
Sync Computer Name
With FileWave 13.2.3+, FileWave will automatically disable "Sync Computer Name" when a client is manually renamed (right-click>Rename) via the FileWave Admin. This Fileset is made possible by a result of this change.

Storing the BitLocker volume keys using a Custom Field

Use a FileWave Custom Field to store the volume keys for your BitLocker volumes. This can be helpful if you don't have another way to escrow the volume keys. The Custom Field outlined in this article will get the volume key for every volume so if there is an encrypted C: and D: you would see both reported by this field.

Adding the Custom Field

1. Download the following Custom Field export: BitLocker Key Custom Field.customfields
2. Import the downloaded file into "FileWave Admin>Assistants>Custom Fields>Edit Custom Fields>Import".
3. Save changes within Custom Fields dialog.
4. Associate Custom Field with desired Windows devices via "right-click>Edit Custom Field(s) Associations".
    1. A Windows-based Smart Group is very helpful to quickly associate Custom Field
    2. Smart Group criteria: "Client OS Platform [equals] Windows"



Here is the script from the Custom Field:

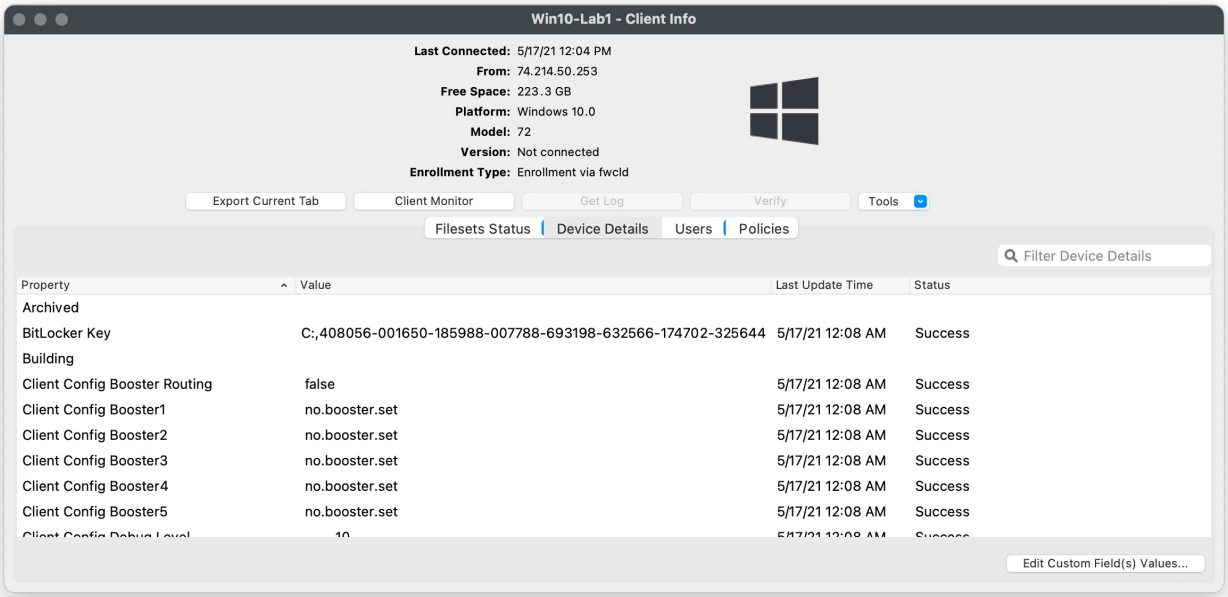
```
# FileWave client will execute this script. The output will be used as the value of the custom field.
#
# Below is an example of how to read the value of one ENVIRONMENT VARIABLE in your script:

# $my_var = $Env:ENV_VAR_NAME
#
# Identify all the Bitlocker volumes.
$BitlockerVolumers = Get-BitLockerVolume

# For each volume, get the RecoveryPassowrd and display it.
$BitlockerVolumers |
    ForEach-Object {
        $MountPoint = $_.MountPoint
        $RecoveryKey = [string]($_.KeyProtector).RecoveryPassword
        if ($RecoveryKey.Length -gt 5) {
            Write-Output ("$MountPoint,$RecoveryKey")
        }
    }
```

```
exit 0
```

Assigning the Custom Field to devices

1. Save changes within Custom Fields dialog.
2. Associate Custom Field with desired Windows devices via "right-click>Edit Custom Field(s) Associations".
   - A Windows-based Smart Group is very helpful to quickly associate Custom Field
   - Smart Group criteria: "Client OS Platform [equals] Windows"
3. Alternatively you could assign the field to all devices since only Windows devices will run the script.

Results



Related articles

- Securing FileWave Server on the Internet for Remote Device Management

Upgrade Windows 10 and 11

Description

Although Software Updates are available as standard catalogue, Feature Updates are not. The following method may be used to update Windows devices using Feature Updates, e.g. 1909, 21H1. Last tested upgrading 20H2 to 21H1

Ingredients

- Latest Windows 10 ISO or Windows 11 ISO – If downloading from a Windows 10 device, please use the steps linked here to access the appropriate ISO download page.
- Following Fileset Recipe:

↓ Windows
Windows - Feature Upgrade.fileset.zip

Directions

Test manually launching the ISO on one or more typical example machines. Not only will this provide an idea of how long the update may take, but the installer may highlight additional criteria required when pushing the Fileset. Depending upon setup and desired options, differing arguments can be supplied to the Fileset to meet requirements.

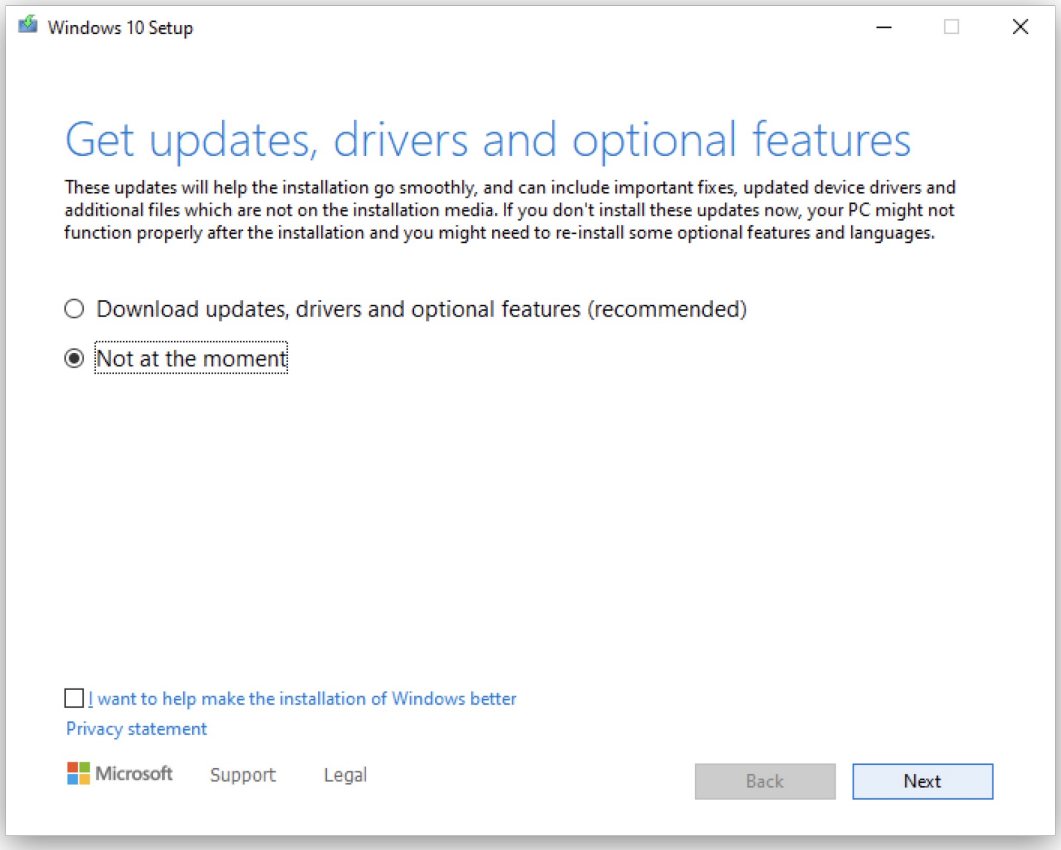
https://docs.microsoft.com/en-us/windows-hardware/manufacture/desktop/windows-setup-command-line-options

> Microsoft may change these options with differing versions. For example, there is a new argument /EULA which has been introduced for Windows 11, whilst the /DynamicUpdate argument has additional options available since Windows 10, 2004.

Examples could include Dynamic Updates or Compatibility (which have been included in the provided Fileset)

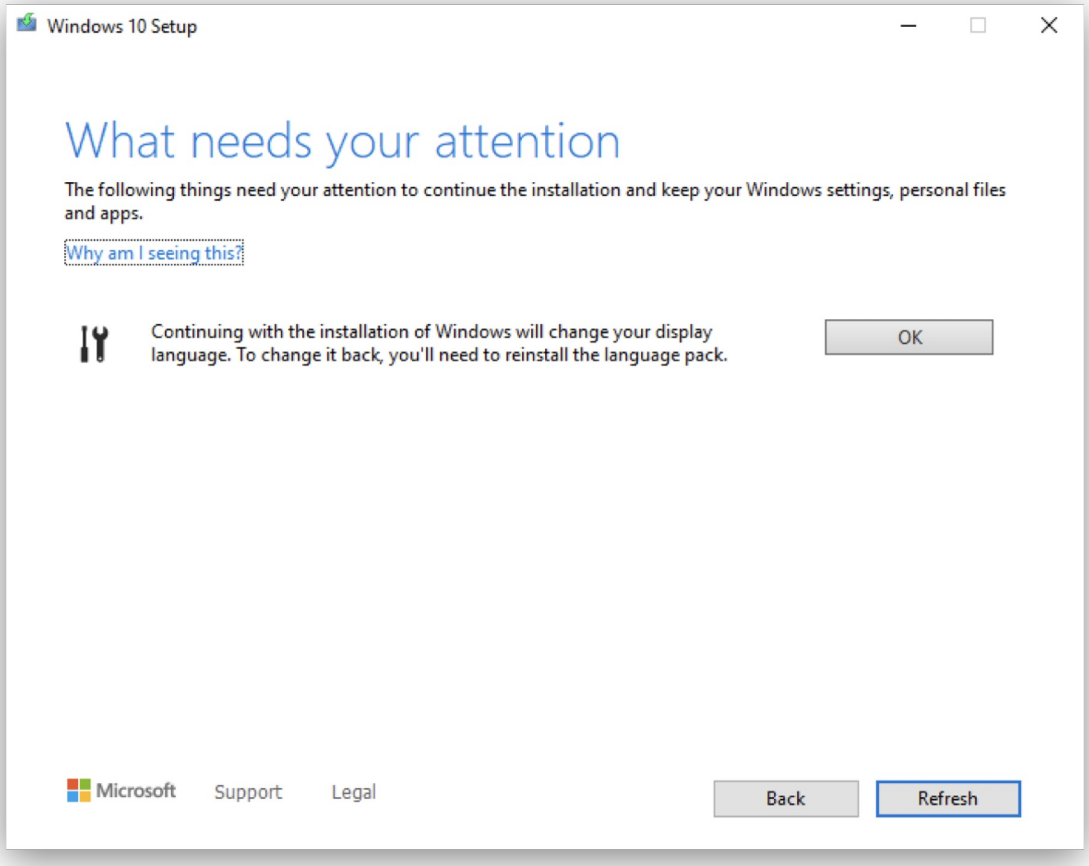
Dynamic Updates (DynamicUpdate):

https://docs.microsoft.com/en-us/windows-hardware/manufacture/desktop/windows-setup-command-line-options#dynamicupdate



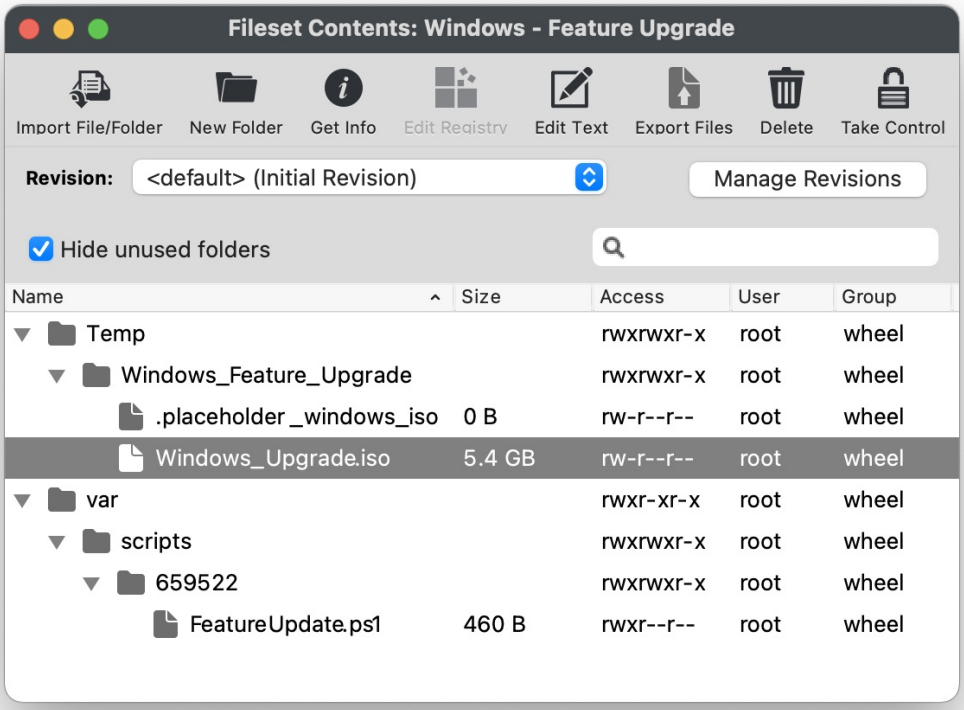
Compatibility (Compat):

https://docs.microsoft.com/en-us/windows-hardware/manufacture/desktop/windows-setup-command-line-options#compat

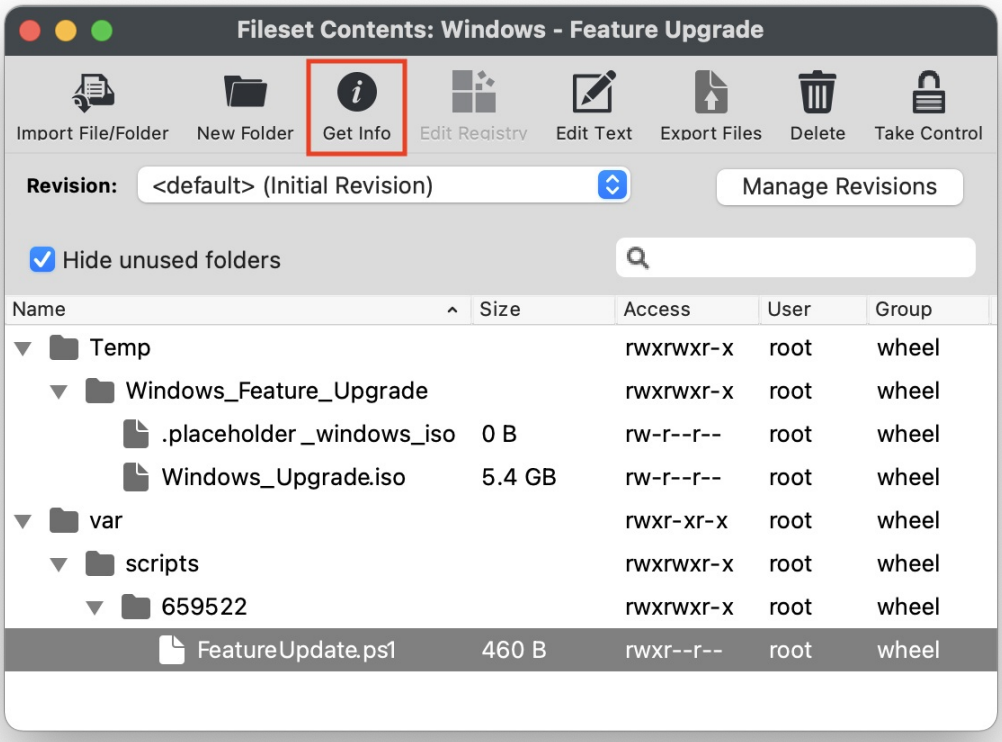


⚠ Defining Dynamic Updates is optional (Microsoft default values will be applied if not supplied), but where warnings are received, if Compatibility is not defined, the Fileset will fail to instal the update.

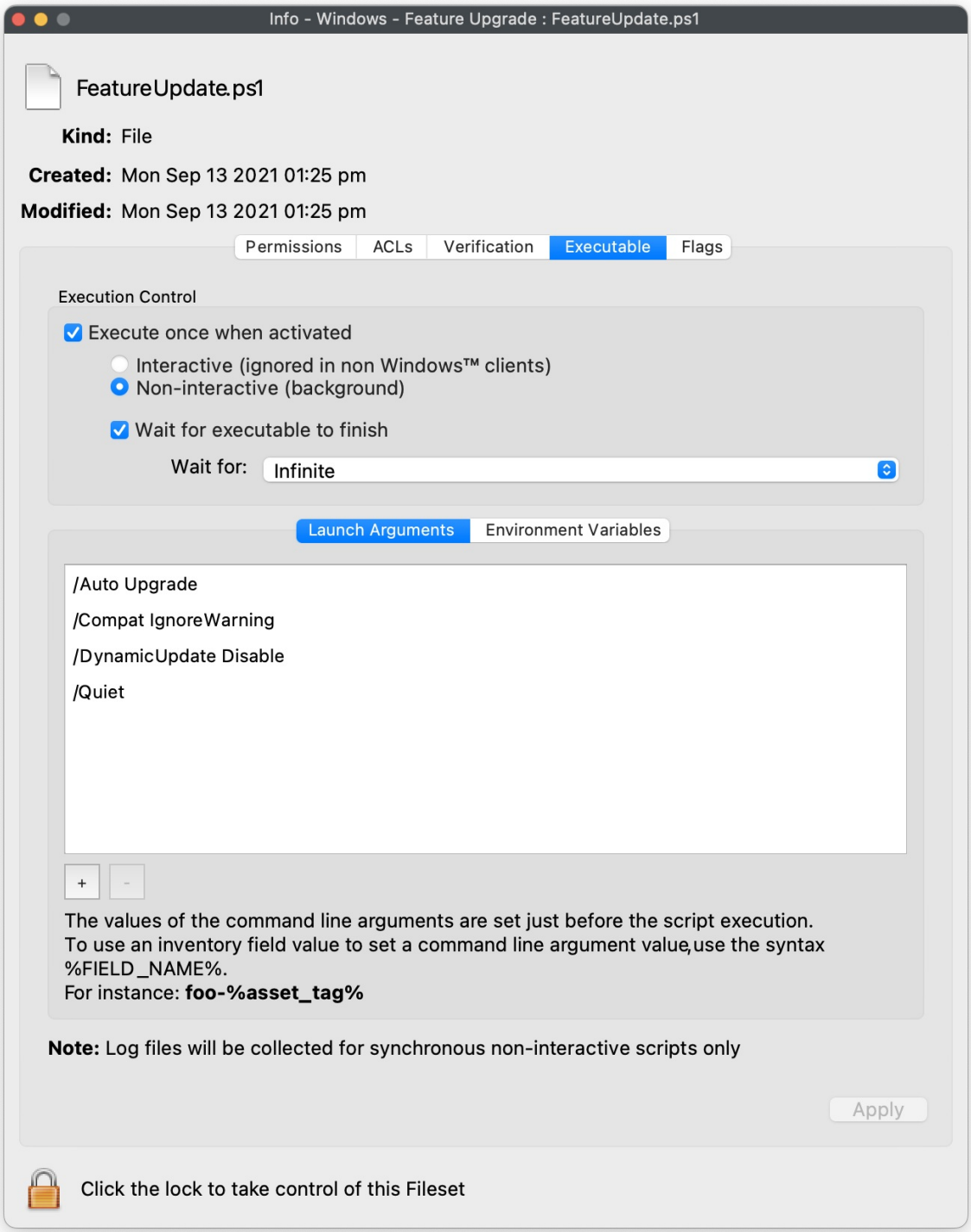
1. Upload the provided Fileset using FileWave Admin
2. Upload the downloaded ISO into the same folder as the Placeholder within the Fileset; approximately 5GB in size. (The .placeholder_windows_iso file may be removed). Ensure the ISO has the same name as the below screenshot: 'Windows_Upgrade.iso"



3. Select "FeatureUpgrade.ps1" from the Fileset contents and click "Get Info" from the top menu bar.



4. Select the "Executable" tab within the "Get Info" window.



5. Modify the "Launch Arguments" as desired and click "Apply" to save changes. Remove any unwanted or add any additional arguments from Microsoft's above KB. Please consider the following:

   - Launch Arguments as a bare minimum:
     - /auto upgrade
     - /quiet

   - Launch Arguments for Windows 11:
     - /auto upgrade
     - /quiet
     - /noreboot
     - /eula accept
     - /dynamicupdate disable
     - /copylogs C:\Temp\win11upgradelogs

   - IMPORTANT NOTE: Compat mode will be required if warnings prevent the installer from completing.

6. Associate Fileset to machines and wait patiently for the upgrade to complete.

Testing

When testing, consider disabling the Reboot option in the Fileset properties, such that the Windows interface is still available during

the initial process of the upgrade. Windows Task Manager will show "Modern Host Setup" process whilst upgrade is in progress.

Timings will vary depending upon chosen options, device usage and network bandwidth. It could take 30-40 minutes or more before the device shows the Windows Update blue screen; as the update prepares the device and other possible updates. It should also be expected that the device may reboot multiple times.

> It could also be possible to have the ISO available via a network mount and adapt the script to mount the shared ISO rather than pushing the ISO via FileWave.

User Experience

Due to the nature of how Microsoft Feature Updates work, there can be a substantial amount of time between the launch of the update and the device continuing to the Blue Updates Screen. Where the Reboot option is selected for the Fileset, this wait will not commence util the user accepts the update. As such the FileWave user prompt may be on the screen for a lengthy period of time. If the reboot option is not selected, although the user will not be impacted by this preliminary stage of the Feature Upgrade, once completed, the user will suddenly be dropped out of their user session, without warning, for the installation and reboots to take place.

Windows 11 Compatible Devices

Description

Microsoft have provided their list of supported Windows 11 requirements:

https://www.microsoft.com/en-gb/windows/windows-11-specifications

Including links to subcategories, for example processor compliance:

https://docs.microsoft.com/en-gb/windows-hardware/design/minimum/windows-processor-requirements

The variety of machines that could be either complaint or non-compliant is vast. The recipe here allows for a scripted method to confirm the status of compliance and is based upon Microsoft's Readiness PowerShell script, details of which are highlighted in the following documentation:

https://techcommunity.microsoft.com/t5/microsoft-endpoint-manager-blog/understanding-readiness-for-windows-11-with-microsoft-endpoint/ba-p/2770866

Two of the methods provided are edited versions of the original supplied Microsoft script. One is a straight forward Custom Field, whilst the other uses a more advanced method to achieve the same result. The script for both methods will provide an output of Pass or Fail in the Custom Field value. Please choose as desired.

Custom Field values may be added to the Client View:

Unaltered Version

Unaltered version of the Microsoft supplied readiness script. Output will include all text as dictated by Microsoft. As a Custom Field, this information can be lengthy, but inventory Queries may be configured to identify the word 'Fail'.

Ingredients

- Following Custom Field

↓ Windows
Windows 11 Readiness Unaltered.customfields

Directions

1. Download the provide Custom Field: 'Windows 11 Readiness Unaltered'
2. Open the Custom Field Editor: FileWave Admin > Assistants > Custom Fields > Edit Custom Fields
3. Select Import and choose the downloaded Custom Field from step 1
4. Change Name if desired
5. Save

Example failed value:

```
{"returnCode":1,"returnReason":"TPM, Processor, ","logging":"Storage: OSDiskSize=98GB. PASS; Memory:
System_Memory=4GB. PASS; TPM: TPMVersion=False. FAIL; Processor: {AddressWidth=64; MaxClockSpeed=2494;
NumberOfLogicalCores=4; Manufacturer=GenuineIntel; Caption=Intel64 Family 6 Model 70 Stepping 1; }. FAIL;
SecureBoot: Capable. PASS; ","returnResult":"NOT CAPABLE"}
```

Simplified Method

The information output by the default script is lengthy and can be considered as inappropriate as a single Custom Field value. This method alters the script, which when used as a Custom Field will return either Pass or Fail. However the details of why it failed will not be provided.

Ingredients

- Following Custom Field

↓ Windows
Windows 11 Readiness.customfields

Directions

1. Download the provide Custom Field: 'Windows 11 Readiness'
2. Open the Custom Field Editor: FileWave Admin > Assistants > Custom Fields > Edit Custom Fields
3. Select Import and choose the downloaded Custom Field from step 1
4. Change Name if desired
5. Save

Advanced Method

Since the hardware of the device will rarely change, it is unnecessary to have the Custom Field script run on every inventory. Additionally, the information output by the default script is lengthy and can be considered as inappropriate as a single Custom Field value. The following method involves building an Administrator Custom Field and the script will be added as a Fileset instead. This Fileset will update the Custom Field value when ran, the details will be stored in a local log file on the device, yet the Custom Field will merely show Pass or Fail once the script has ran on a Windows device.

As a Fileset, the script will run only once without intervention, preventing the script from unnecessarily running over and over again.

Ingredients

- Administrator Custom Field with an internal name of 'windows_11_compatible'

↓ Windows
Windows 11 Compatible.customfields

- Following Fileset:

↓ Windows
Windows 11Compliance.fileset.zip

Directions

Custom Field

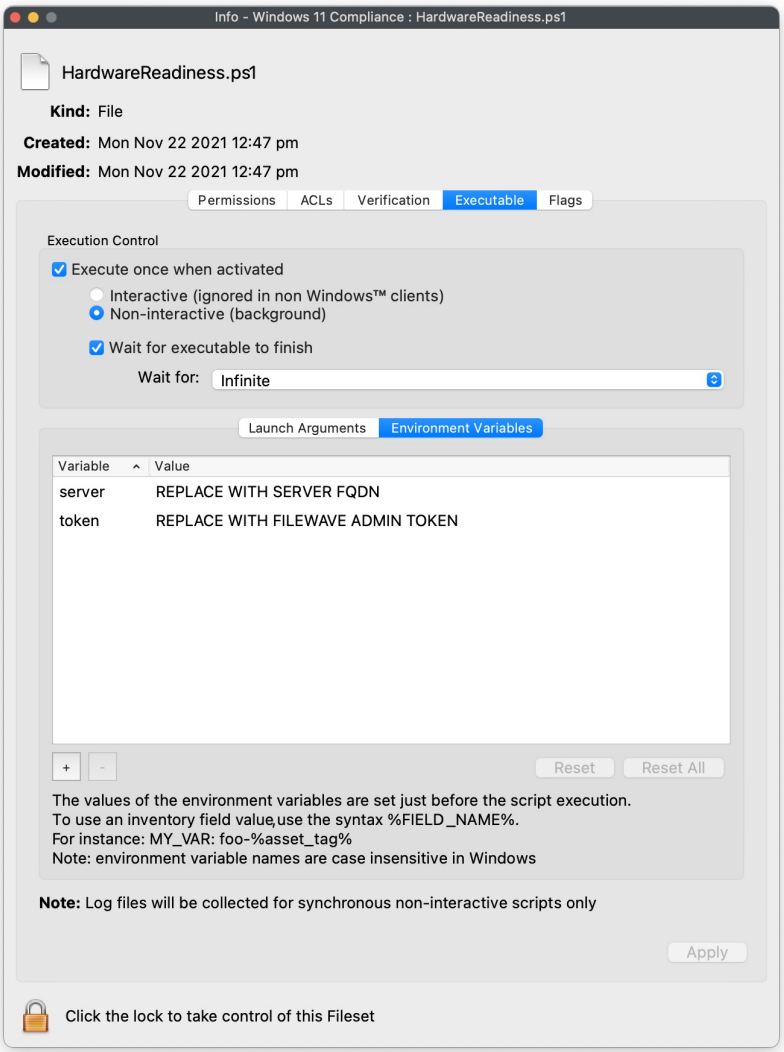
1. Download the provided Custom Field: 'Windows 11 Compliance'
2. Open the Custom Field Editor: FileWave Admin > Assistants > Custom Fields > Edit Custom Fields
3. Select Import and choose the downloaded Custom Field from step 1
4. Change Name as desired, but ensure the Internal Name is not altered and association is to all devices
5. Save
6. Once configured, the Fileset may then be associated and pushed to devices

Fileset

1. Download the provided Fileset
2. Edit the Fileset's script Environment Variables (details below)
3. Associate to devices for testing and then once satisfied push to all devices

Fileset Editing

- Open the Fileset, select the script and choose Get Info
- Select the Executable tab and then Environment Variables
- Replace the Values as appropriate



- The 'value' for the 'server' variable should be replaced with the name of the server as seen in Preferences > Mobile of the Admin console
- The 'value' for the 'token' should be replaced with a chosen Admin token from: Assistants > Manage Administrators > (Chosen Account Name) > Application Tokens. Copy the 'Token (base64)'

Additional Information

The Fileset will use the FileWave API to report back the current status of the device's compatibility during Fileset activation. If devices are addressed to change their compatibility status, it is possible to run a 'Reinstall Fileset' which will cause the API to update the current information, refreshing the Custom Field.

The full output of the script will be available in the script log, accessible from the right click menu item of a Fileset's script status view from Client Info (local network between Admin device and selected machine is required). A failure example:

```
{"returnCode":1,"returnReason":"TPM, Processor, ","logging":"Storage: OSDiskSize=98GB. PASS; Memory: System_Memory
=4GB. PASS; TPM: TPMVersion=False. FAIL; Processor: {AddressWidth=64; MaxClockSpeed=2494; NumberOfLogicalCores=4;
Manufacturer=GenuineIntel; Caption=Intel64 Family 6 Model 70 Stepping 1; }. FAIL; SecureBoot: Capable. PASS; ","re
turnResult":"NOT CAPABLE"}
```

Self-Signed Certs

The Fileset Activation Script 'HardwareReadiness.ps1' must be edited to allow for Self-Signed Certificates. The following section should have the mentioned lines updated to remove the leading hashes. After removal it should look like the following:

```
#####################################################
# Beginning of ammendment for FileWave Custom Field report

# REMOVE HASHES FROM FOLLOWING 12 LINES IF USING A SELF-SIGNED CERTIFICATE
add-type @"
using System.Net;
using System.Security.Cryptography.X509Certificates;
public class TrustAllCertsPolicy : ICertificatePolicy {
public bool CheckValidationResult(
ServicePoint srvPoint, X509Certificate certificate,
WebRequest request, int certificateProblem) {
return true;
}
}
```

```
"@
[System.Net.ServicePointManager]::CertificatePolicy = New-Object TrustAllCertsPolicy
```

> The client must be able to reach the server on port 443 to be able to post the API update back to the server.

Result

The Custom Field for the Simplified and Advanced methods actually provides 3 possible values:

- NA – Default value
- Fail — One or more items failed the check
- Pass – All items passed the check and the device is ready for Windows 11

Notes

These options are by no means the only options available. The script could be used within an Upgrade Fileset for Windows 11, for example, and the script may run prior to confirm if the device satisfies the requirements. However, requirement scripts should only be used where they will eventually become true, to prevent them from running forever and being a constant draw on the server.

Windows 11 support in FileWave 14.7+

What

Windows 11 was released by Microsoft on October 5, 2021.

When/Why

As an administrator, it is important to know if this new OS is supported, and about the end of life of older versions of Windows.

How

The FileWave client running on Windows 11 is fully supported. We will also continue to support Windows 10 because it is still a supported OS by Microsoft.

- Windows 11 is recognized in Filewave
- Windows Server 2022 is recognized in Filewave
- All Windows 10 versions (e.g. 20H1, 20H2, 21H1 and 21H2) are recognized and classified as a "Codename"

Related Content

- FileWave Downloads

Notify Users with a dialog (Windows)

Description

The provided Fileset is an example of notifying users, in particular here, a message regarding Fileset status when downloading and installing new Filesets.

The Fileset is designed to:

- Create a continual running service that monitors Fileset changes
- Where Fileset changes occur, begin monitoring the FileWave Client log file
- If a number of preset text strings are found in the log file, send this to the Notification Centre
- Lastly, where another preset text is found, stop monitoring the log file

The service has been built to be actioned automatically by the user logging in. Where Filesets are disassociated, each has a pre-uninstallation script to ensure the services should also be removed.

Ingredients

- Provided Fileset:

↓ Windows


Directions

For the example provided:

- Download the necessary provided Fileset
- Upload using FileWave Admin
- Associate to the appropriate devices
- 'Update Model'.

Fileset scripts may be modified for personal preference. In each Fileset there is a script that is actioned by the local computer service. The scripts are using a pattern match. The pattern matching may be edited as required, removing or adding appropriately.

Windows

Locate the "BallonTipSwitchWatcher.ps1" file within the Fileset and choose to edit. In the following code block snippet from this script, the switch statement is pattern matching text. In the provided example the script is looking for lines that contain any one of the following:

- Model version
- Downloading Fileset
- Done activating
- Activate all

Where found, the 'ShowBalloonTipInfo' function is being used to prompt the user:

BallonTipSwitchWatcher.ps1

```
$changeAction = Get-Content C:\ProgramData\FileWave\FWClient\fwcld.log -tail 1 -wait | ForEach-Object {
switch($_) {
{ $_ -match "Model version" -or $_ -match "Downloading Fileset" -or $_ -match "Done activating" -or $_ -match
"Activate all" } { ShowBalloonTipInfo ("FileWave: ",$_.split("|")[4]) }
```

The second part of the switch statement is causing the script to exit. The pattern match this time, is any line that contains:

- Installation

BallonTipSwitchWatcher.ps1

```
{ $_ -match "Installation" } { break }
```

Notes

The above provides an example of notifying users, using a service. However, with some adaptation messages could be sent in other ways at alternate times to users.

Related Content

- Notify Users with a dialog (macOS)

Using Native Windows Tools to Troubleshoot

What

Things break, thank goodness! If they didn't all IT engineers would be out of work. But what do you do when things break? Specifically, if a FileWave client on a Windows device isn't working right, what tools are available to help? Turns out there are tons of options!

When/Why

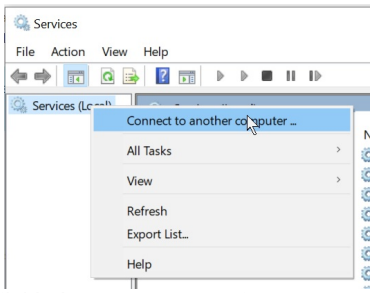
We'll turn to other tools whenever we need an alternate method of confirming things with remote machines. This list isn't meant to be all-inclusive, but is a great "starter pack" of tools you can use!

> Note that access to remote Windows tools is always reliant on being logged in with a MSFT account that has rights to perform those actions.

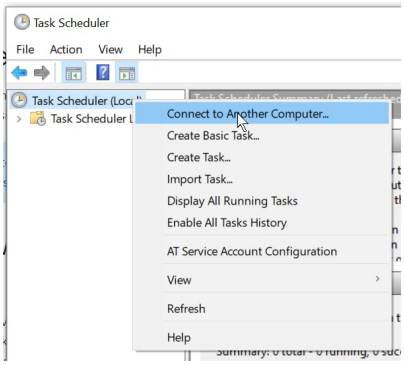
How

Here is a list of super-helpful tools:

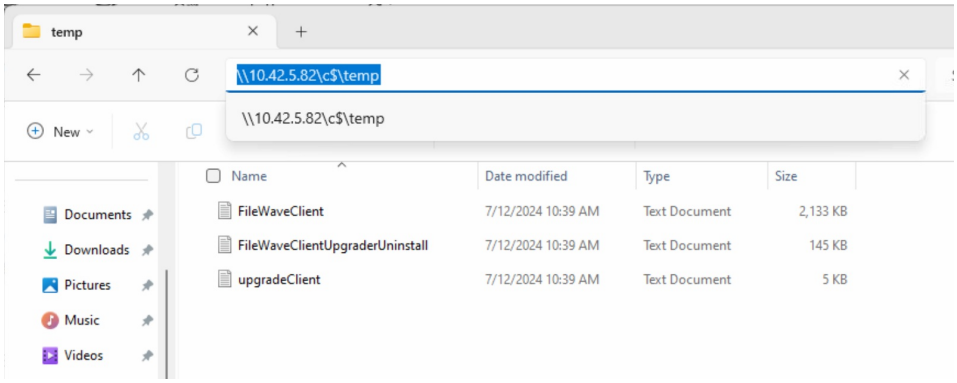
- Services App: Everyone knows the services app, but did you know you can use it against remote machines too? Just right-click on Services and choose to connect to another computer (by name or IP). Great for checking, starting and stopping services.



- Task Scheduler: Task scheduler is also available for remote connection:

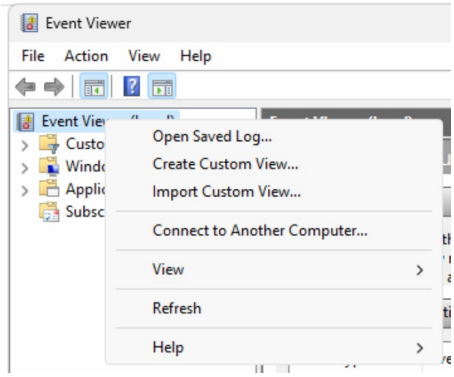


- The remote file system: Yep, you read that right, you can actually browse the remote machine's files. In explorer, just navigate to \\ip(or name)\c$. Great for looking at log files and generally just checking file status.

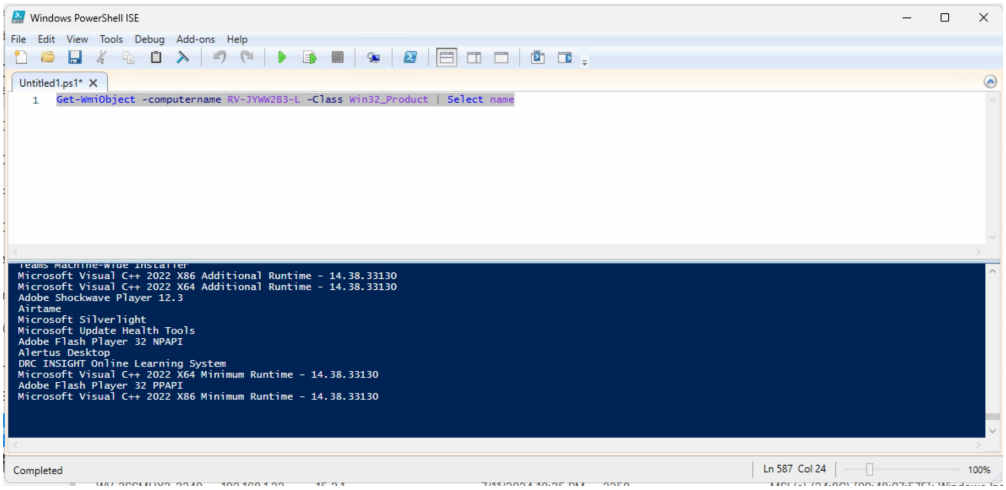


- Event Viewer: You can also remotely connect to Event Viewer, which is an excellent tool for MSI troubleshooting in

particular:



- PowerShell: PowerShell is a fantastically helpful tool, and you can do almost anything with it. For instance, you can't use the Add/Remove Programs tool to remotely access another machine, but you can use PowerShell to remotely connection and see what is installed:

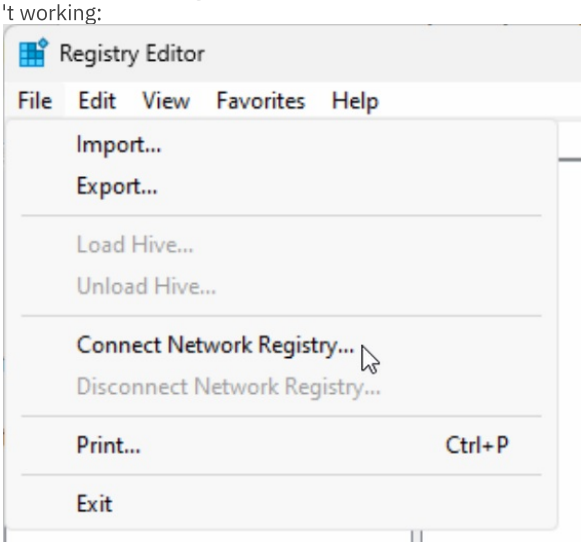


- PSTools: PSTools aren't native to Windows, but they are an excellent source for a wide range of utilities, PSEXEC in particular. (Link below about using PS Tools)

- Tasklist and Taskkill: These are old school DOS commands, but are super helpful to see running processes on remote machines (and to kill them if necessary. Do tasklist /? or taskkill /? for instructions:

```
C:\Users\filewave3>tasklist /S 10.45.6.60

Image Name                     PID Session Name        Session#    Mem Usage
========================= ======== ================ =========== ============
System Idle Process              0 Services                   0          8 K
System                           4 Services                   0      5,576 K
Secure System                  108 Services                   0     29,532 K
Registry                       152 Services                   0     29,732 K
smss.exe                       632 Services                   0      1,128 K
csrss.exe                      884 Services                   0      4,812 K
wininit.exe                    972 Services                   0      6,232 K
services.exe                   668 Services                   0     14,372 K
LsaIso.exe                     996 Services                   0      3,728 K
lsass.exe                     1040 Services                   0     32,092 K
svchost.exe                   1180 Services                   0     36,620 K
WUDFHost.exe                  1220 Services                   0     17,960 K
fontdrvhost.exe               1264 Services                   0      3,572 K
svchost.exe                   1356 Services                   0     21,880 K
svchost.exe                   1400 Services                   0     10,568 K
svchost.exe                   1448 Services                   0     22,268 K
svchost.exe                   1544 Services                   0      9,248 K
```

- Regedit: Yes, even regedit can connect to remote registries. For FileWave specifically this is helpful to change client preferences if client monitor isn't working:



Related Content

- PSExec as a Helper in Troubleshooting
- Script Best Practices
- Using PowerShell to Remotely Check the Windows FileWave Client Status