

Reset Windows device through a script (FileWave Recipe)

What

AutoPilot assigns devices to your organization. This, coupled with the ability to "reset" a Windows 10 or 11, device allows you to be able to "re-image" a Windows device without necessarily wiping it out. What is described in this article is a method to wipe a device which could be used for Autopilot, but can also be used independently of Windows MDM.

When/Why

As of FileWave v14.8.0, a [command to "reset" your Windows devices](#) will be included in FileWave itself, but that requires the device to be enrolled in MDM. In this Fileset in this article, we are providing you with a method of doing the device reset through a PowerShell command that does not require MDM. It goes without saying that this reset is destructive to data on the device, so appropriate caution should be utilized.

How

Wiping a device to reset it in the field

Windows Autopilot Reset - <https://docs.microsoft.com/en-us/mem/autopilot/windows-autopilot-reset> - takes the device back to a business-ready state, allowing the next user to sign in and get productive quickly and simply. Specifically, Windows Autopilot Reset:

- Removes personal files, apps, and settings.
- Reapplies a device's original settings.
- Maintains the device's identity connection to Microsoft Entra ID.
- Maintains the device's management connection to Intune.

The Windows Autopilot Reset process automatically keeps information from the existing device:

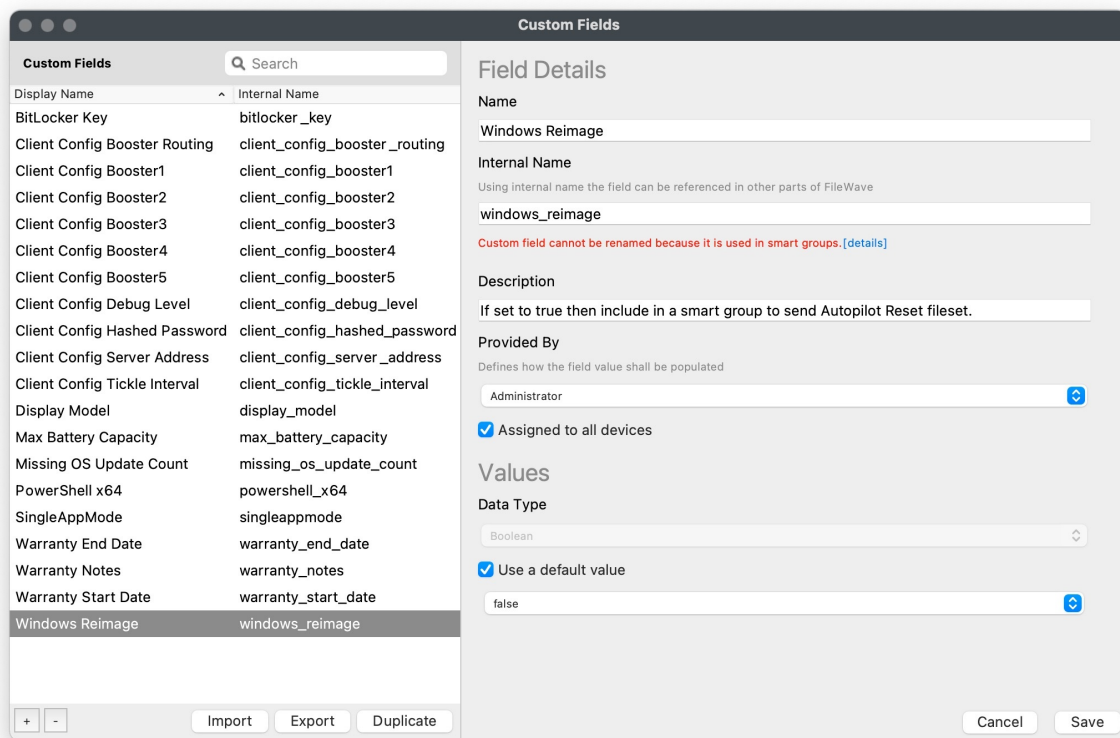
- Set the region, language, and keyboard to the original values.
- Wi-Fi connection details.
- Provisioning packages previously applied to the device
- A provisioning package present on a USB drive when the reset process is started
- Microsoft Entra Active Directory device membership and MDM enrollment information.

Windows Autopilot Reset will block the user from accessing the desktop until this information is restored, including reapplying any provisioning packages. For devices enrolled in an MDM service, Windows Autopilot Reset will also block until an MDM sync is completed. When Autopilot reset is used on a device, the device's primary user will be removed. The next user who signs in after the reset will be set as the primary user.

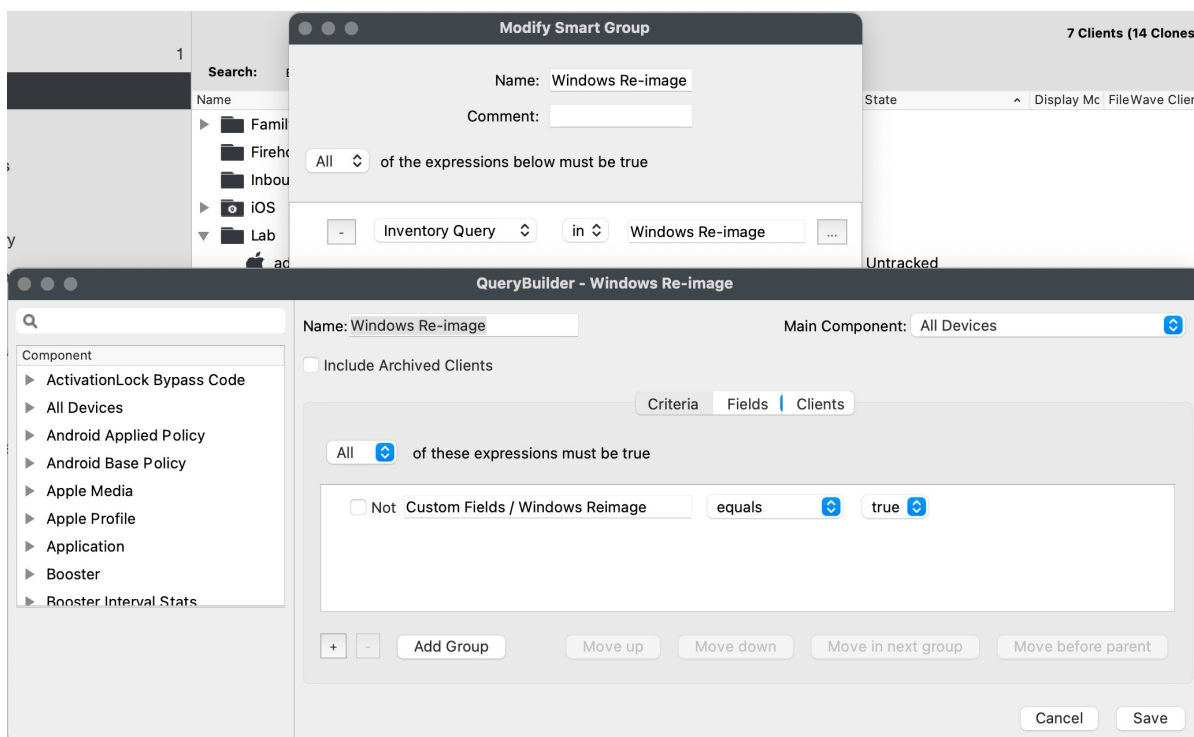
Initially FileWave does not directly issue the Autopilot Reset via MDM, but there is still a way to accomplish Autopilot Reset for a FileWave enrolled device. The below fileset will execute the above PowerShell. It will also enable the Windows Recovery Environment so that this can be successful. If using Windows 10 1703 or newer you can change the methodname to doWipeProtected so that the wipe will continue even if a user reboots in the middle of it.

Directions

1. Ensure that you have your device in Autopilot as outlined here: [Integrating with AutoPilot](#)
2. Create a custom field with the internal name of windows_reimage as seen below. The field should be Boolean and have a default value of "false".



3. Create smart group that looks for windows_reimage to be True as seen below.



4. Add this Fileset to your server. You can unzip it and then drag the Fileset into the Fileset window. Note that this Fileset uses the section of code below that can be edited to change "doWipeMethod" to "doWipeProtectedMethod" or to use any other method as outlined here but be sure to add "Method" to the one you want to use: <https://docs.microsoft.com/en-us/windows/client-management/mdm/remotewipe-csp>

```
# This part wipes the system
# https://docs.microsoft.com/en-us/windows/client-management/mdm/remotewipe-csp
# methodname can be doWipeMethod or doWipeProtected but the later needs Win 10 1703 or newer
$namespaceName = "root\cimv2\mdm\dmmap"
$class_name = "MDM_RemoteWipe"
$methodName = "doWipeMethod"

$session = New-CimSession

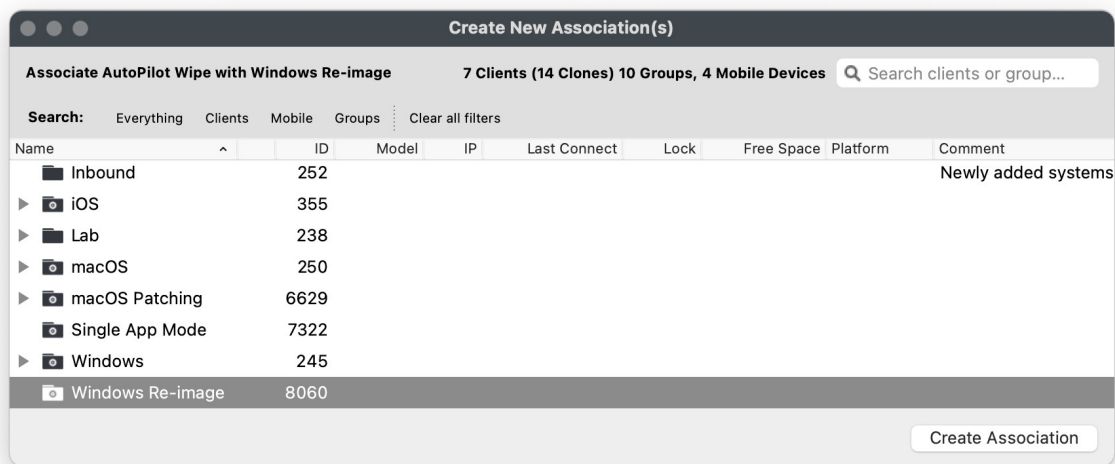
$params = New-Object Microsoft.Management.Infrastructure.CimMethodParametersCollection
```

```
$param = [Microsoft.Management.Infrastructure.CimMethodParameter]::Create("param", "", "String", "In")
$params.Add($param)
```

```
$instance = Get-CimInstance -Namespace $namespaceName -ClassName $className -Filter
"ParentID='./Vendor/MSFT' and InstanceID='RemoteWipe'"
$session.InvokeMethod($namespaceName, $instance, $methodName, $params)
```



5. Select the Fileset and click the Scripts button in the Native Admin. Right click on the Reset.ps1 script and pick Properties. You must change the first Environment Variable for that script to be the API token you want to use. You can get this from the Native Admin from Manage Administrators → Select an admin → Application Tokens. This token is used by the script to set the custom field for windows_reimage to false. If you don't update this then your device will be stuck in a loop of wiping once you enable it.
6. Associate the Fileset with the Smart Group that you created.



7. To wipe a device you will set the windows_reimage custom field to True. This will cause the device to appear in the Smart Group, and will cause the Fileset to be applied. The Fileset will set windows_reimage to be False while it runs, will enable Recovery Environment, and then will initiate a wipe.
8. Because the fileset sets windows_reimage to False the device leaves the smart group that would cause the AutoPilot Wipe Fileset to apply to it so it won't be caught in a re-image loop.

🔄Revision #7

★Created 15 June 2023 16:08:34 by Andrew Kloosterhuis

✍Updated 17 July 2023 01:34:17 by Josh Levitsky