

Troubleshooting

- [Clearing FileWave Client Certs](#)
- [FileWave Client notification for zsh running in the background](#)
- [FileWave Client Rename Behavior](#)
- [FileWave Client Status Check: How to ask the client what it is doing on macOS and Windows](#)
- [Using PsExec to Remotely Restart the FileWaveWinClient Service](#)
- [Understanding and Resolving Proxy Communication Issues in FileWave](#)
- [Using PowerShell to Remotely Check the Windows FileWave Client Status](#)
- [When does the inventory client run scans?](#)
- [PSEXec as a Helper in Troubleshooting](#)

Clearing FileWave Client Certs

In some situations, you may want to explicitly make the server clear and revoke a client certificate without deleting the client from FileWave, for instance if you are wiping a macOS client or reinstalling an IVS client.

From FileWave Central

For desktop clients, you can right-click the client and choose "Clear Certificate(s)".

Note that the current administrator needs to have write permissions on the clients, and will need to enter credentials.

From the Client

The first way is to use the client (13.1.1) certificate itself to authenticate, which is only possible if the certificate and its private key still exist on the client:

```
fwcld -clearCertificate [-serverHost <fwserver_address> -serverPort 20445]
```

This is the equivalent of the following command using curl (replace <fwserver_address> with the address of your FileWave server):

```
sudo curl --key /private/var/FileWave/client.key --cert /private/var/FileWave/client.crt -X POST
https://<fwserver_address>:20445/auth/client/clear_certificate
```

The client will then be unable to communicate with the server (until a new CSR is created). This command can be used in the activation script of a macOS reinstall fileset to make the server properly clear the old client certificate.

Note that the command above uses the client certificate itself to identify the client. In case the certificate's private key is already lost, there is an alternative where you can authenticate with an administrator's token rather than with the client certificate:

```
fwcld -clearCertificate -token <application_token> [-serverHost <fwserver_address> -serverPort <fwserver_port>]
```

- <fwserver_address>: The FileWave server address (optional)
- <fwserver_port> : 20445 by default
- <application_token>: An administrator application token, with write permissions on this client. Can be found in the Application Tokens tab of the Manage Administrators dialog (example: {1ca3fe82-a41d-8866-bd4d-f83f9f1a8dd5}).

Alternatively, you can also clear certificates en masse using the inventory superadmin token. In this case, you are allowed to clear the certificate of any client (obviously use with caution):

```
curl -X POST https://<fwserver_address>:20445/auth/client/clear_certificates -H 'Authorization:
<application_token>' -H 'Content-Type: application/json' -d '["<serial_1>", "<serial_2>", ...]'
```

Included in 13.1.1 and above are the options to clear with 'MAC address' or 'Device ID':

```
curl -X POST https://<fwserver_address>:20445/auth/client/clear_certificates?identifier=mac -H 'Authorization:
<application_token_base64>' -H 'Content-Type: application/json' -d '["<mac_address_1>", "<mac_address_2>", ...]'
```

```
curl -X POST https://<fwserver_address>:20445/auth/client/clear_certificates?identifier=device_id -H
'Authorization: <application_token_base64>' -H 'Content-Type: application/json' -d '["<device_id_1>", "
<device_id_2>", ...]'
```

The identifier parameter is optional, its default value is serial_number.

- <fwserver_address>: The FileWave server address.
- <serial_1>, ...: serial numbers of clients to revoke. Must match the serial_number field from inventory.
- <mac_address_1>, ...: MAC addresses or clients to revoke.
- <device_id_1>, ...: device ids of clients to revoke.
- <application_token_base64>: base64-encoded value of an administrator application token, with write permission on this client. Can be found in the Application Tokens tab of the Manage Administrators dialog (example: ezFjYTNmZTgyLWE0MWQtODg2Ni1iZDRkLWY4M2Y5ZjFhOGRkNX0=_).

A dict of lists of clients for: SUCCESS, NOT_FOUND and ERROR statuses is returned.

- SUCCESS: the client certificate was successfully revoked.
- NOT_FOUND: no certificate was found on the server for this client, or no such client was found. Maybe the certificate was

already revoked, or the client had no certificate yet, or the client had not reported its MAC address or serial number yet, if you passed a MAC address or a serial number.

- ERROR: an unexpected error occurred. Please check server logs for details.

Potential log entries

```
2019-06-12 7:12:02.481|main|FATAL|CLIENT|Unable to retrieve the contents of the cached custom field values: Error decrypting data
2019-06-12 7:12:02.833|main|INFO|CLIENT|CRL updated
2019-06-12 7:12:02.834|main|INFO|CLIENT|No certificate private key yet. Sending a certificate signing request to server my.FQDN.com.
2019-06-12 7:12:03.235|main|FATAL|CLIENT|Failed to send enrollment request (and CSR): error 400 a CSR for this client was already sent.
2019-06-12 7:12:03.235|main|INFO|CLIENT|Falling back to no certificate.
```

FileWave Client notification for zsh running in the background

What

When I upgraded my FileWave Client to 15.3.0 or newer I saw a popup saying "zsh is now running in the background".

When/Why

This happens once when first going to version 15.3.0 or higher of the client on macOS. There is nothing to do. This popup is because the FileWave client needs permission to run zsh for scripting. It can be ignored and development is going to look at ways to suppress it so that in the future it should not show, but it should only happen 1 time on each machine and then even on newer versions of the clients you should not see it because it should already have the permission.

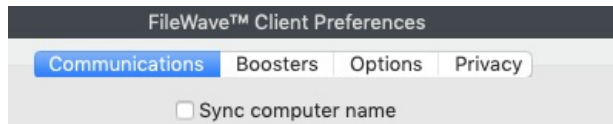
Related Content

- [FileWave Version 15.3.1](#)

FileWave Client Rename Behavior

Renaming a desktop client in the admin console will now change the Client Name inventory field to match the new name entered and also change the client name "sync" setting in the client's preferences. This ensures that the device is able to connect properly with FileWave instance without duplicated entries. (The Device Name inventory field is not modified)

So, when a client is renamed, the "Sync Computer Name" settings is turned off:



Further Details

- The FileWave Client uses its name as part of the identity of the client (see: [How the FileWave Client Communicates](#) for more)
- The FileWave Client has a configuration ([FileWave Client Configuration Settings](#)) to "Sync Computer Name" that can be changed one at a time (Via Client Monitor) or en masse ([Creating a Superprefs Fileset](#)).
- Renaming devices in the FW Admin (r-click → rename) has a different behavior depending on the device (see below)

Behavior (Prior to FileWave v13.2)

- iOS - If you rename an iOS 9.3+ Supervised device in the admin, we send a rename command to the device. No disruption to communication
- macOS / Windows
 - If Sync Name is checked - Client name is new name, Device name is old name. Disruption to communication; Client is still checking in with the old name, and will appear as a new device.
 - if Sync Name is NOT checked - Client name is new name, Device name is old name. No disruption to communication, but there is a discrepancy between names.
- Android / Chromebook - Names in admin are for reference only, and has no effect on communications.

FileWave Client Status Check: How to ask the client what it is doing on macOS and Windows

What

The FileWave client, available for both macOS and Windows, allows administrators to verify its status and understand the tasks it's currently handling. This can be accomplished by executing a specific command in the Terminal application on macOS, or the Command Prompt on Windows. This article provides instructions on how to use this feature for troubleshooting purposes.

When/Why

This feature is especially useful when you have direct access to the device and need to determine what the FileWave client is currently working on. It enables you to view the current status of the FileWave client, the filesets in inventory, and other crucial information.

You might find this feature beneficial when:

- You're diagnosing issues related to the application of filesets or profiles.
- The FileWave client is behaving unpredictably and you want to determine its current state.

How

Follow the steps below to check the status of the FileWave client:

macOS

1. Open the Terminal application on the macOS device.
2. Execute the following command: `/usr/local/sbin/FileWave.app/Contents/MacOS/fwclld -s`

Windows

1. Open the Command Prompt on the Windows device.
2. Depending on the architecture of your Windows OS, execute the appropriate command:
 - For Windows 64bit or ARM, run: `"C:\Program Files (x86)\FileWave\fwclld.exe" -s`
 - For Windows 32bit, run: `"C:\Program Files\FileWave\fwclld.exe" -s`

After running the command, you will see output similar to the example below:

```
*****
**FileWave Client Status**
*****
User ID: 11354
Current Model Number: 660

Filesets in Inventory:

1. Fileset Mac MDM OS Update - macOS Monterey 12.6.3 12.6.3, revision ID 10148, ID 10148, revision ID 10148,
version 1 - Apple MDM OS Update is not supported (0)
2. Fileset Mac MDM OS Update - Safari 16.3, revision ID 10149, ID 10149, revision ID 10149, version 1 - Apple MDM
OS Update is not supported (0)
3. Fileset FileWave_macOS_Client_14.10.2_df52a47c77, revision ID 10169, ID 10169, revision ID 10169, version 1 -
Active (0)
4. Fileset Profile - TeamViewerHost Allow Standard User, revision ID 10173, ID 10173, revision ID 10173, version 1
- Handled via MDM (0)
5. Fileset Profile - Microsoft Defender - Kernel Extension, revision ID 10190, ID 10190, revision ID 10190,
version 1 - Handled via MDM (0)
6. Fileset Profile - Microsoft Defender - Notifications, revision ID 10191, ID 10191, revision ID 10191, version 1
- Handled via MDM (0)
7. Fileset Profile - Microsoft Defender - Web Content Filter, revision ID 10192, ID 10192, revision ID 10192,
version 1 - Handled via MDM (0)
8. Fileset Profile - Microsoft Defender - TCC, revision ID 10193, ID 10193, revision ID 10193, version 1 - Handled
via MDM (0)
9. Fileset Profile - Microsoft Defender - Data Acceptance, revision ID 10194, ID 10194, revision ID 10194, version
1 - Handled via MDM (0)
10. Fileset Profile - Microsoft Defender - System Extension, revision ID 10195, ID 10195, revision ID 10195,
version 1 - Handled via MDM (0)
```

```
11. Fileset Profile - Microsoft - Background Service, revision ID 10196, ID 10196, revision ID 10196, version 1 -  
Handled via MDM (0)  
12. Fileset MS Defender macOS, revision ID 10197, ID 10197, revision ID 10197, version 1 - Active (0)  
Filesets not meeting requirements:
```

```
Worklist:
```

The output provides the following information:

- User ID and Current Model Number where User ID is really the ID number of the device in FileWave
- Filesets currently in the device's inventory, along with their status
- Filesets not meeting requirements
- Current worklist

This information can assist you in understanding the tasks that the FileWave client is processing and aid in troubleshooting any issues.

Related Content

- [FileWave Log File Locations](#)
- [Using PowerShell to Remotely Check the Windows FileWave Client Status](#)

Using PsExec to Remotely Restart the FileWaveWinClient Service

What

Using PsExec to remotely restart the "FileWaveWinClient" Windows service allows you to remotely manage the FileWave client on Windows devices. This can be useful in situations where the client is not functioning properly and needs to be restarted in order to resolve the issue.

When/Why

There may be a variety of reasons why you would need to remotely restart the FileWaveWinClient service on a Windows device. Some common reasons include:

- The service has stopped functioning properly and needs to be restarted in order to resolve the issue
- The service needs to be restarted in order to apply a configuration change
- The service needs to be restarted as part of a troubleshooting process

How

To use PsExec to remotely restart the FileWaveWinClient service, you will first need to download and install PsExec on the device from which you will be initiating the restart. PsExec can be downloaded from the Microsoft TechNet website (<https://docs.microsoft.com/en-us/sysinternals/downloads/psexec>).

Once you have PsExec installed, you can use the following command to remotely restart the FileWaveWinClient service:

```
psexec \[remote device] -u [username] -p [password] net start "FileWaveWinClient"
```

Replace [remote device] with the hostname or IP address of the remote device, and [username] and [password] with the appropriate credentials for the remote device.

Related Content

- Microsoft TechNet: PsExec (<https://docs.microsoft.com/en-us/sysinternals/downloads/psexec>)

Digging Deeper

In addition to using PsExec to remotely restart the FileWaveWinClient service, you can also use the "net" and "sc" command-line tools to query and change Windows services.

To query a service using "net", you can use the following command:

```
net start [service name]
```

This will display the status of the specified service.

To start or stop a service using "net", you can use the following commands:

```
net start [service name] net stop [service name]
```

To change the startup type of a service using "net", you can use the following command:

```
net start [service name] [startup type]
```

Valid startup types include: boot, system, auto, demand, disabled

To query a service using "sc", you can use the following command:

```
sc query [service name]
```

This will display detailed information about the specified service, including its status, startup type, and binary path.

To start or stop a service using "sc", you can use the following commands:

```
sc start [service name] sc stop [service name]
```

To change the startup type of a service using "sc", you can use the following command:

```
sc config [service name] start=[startup type]
```

Valid startup types include: boot, system, auto, demand, disabled

Keep in mind that you will need to have the appropriate permissions on the remote device in order to use these commands. You can also use PsExec to execute these commands remotely, as described in the previous section.

Understanding and Resolving Proxy Communication Issues in FileWave

What

This article addresses the common communication issues encountered with FileWave, especially when using proxy servers like Lightspeed, Securlly, and others. These issues often arise during periods of high network traffic, such as the re-deployment of devices within an organization. This article is meant to offer one possible reason for devices behind proxies could experience communication issues trying to talk to the FileWave Server.

When/Why

Communication problems with proxy servers tend to happen more frequently during peak network traffic times, such as the summer months when educational institutions are re-deploying devices for the new school year. The problem can manifest either through the proxy server being unable to handle the increased load or through port exhaustion if all network ports are in use. Understanding when and why this occurs will aid in prevention and troubleshooting.

How

To alleviate communication issues related to proxy servers, follow these steps:

1. Assess the Situation: Check if the proxy server is overwhelmed with traffic or experiencing port exhaustion. Look for signs such as slower response times or connection failures.
2. Bypass Filtering for Apple Devices: For organizations utilizing Apple devices, it is essential to bypass filtering for the IP range 17.0.0.0/8 as per Apple's guidance. This will prevent inspection of Apple traffic, which could otherwise lead to problems.

For example, you may configure this in your proxy settings:

```
# Bypass filtering for Apple IP range
Allow 17.0.0.0/8
```

3. Consider Additional Public IPs: If port exhaustion is an issue, contemplate adding an additional public IP to expand the available network ports.
4. Monitor and Adjust: Continuously monitor the situation and adjust configurations as needed to ensure smooth operations during peak times.
5. Work with FileWave Support: Bring any issues to [Customer Technical Support](#) so that you don't have to investigate alone. There may also be the possibility of a FileWave Server issue that needs to be resolved.

Understanding the underlying architecture of the proxy server, along with FileWave's communication protocols, can be instrumental in troubleshooting and resolving these issues. Being proactive by preparing the network for expected traffic surges and adhering to recommended practices (like Apple's guidance for bypassing filtering) can prevent these challenges from occurring in the first place. Regular monitoring and adaptive strategies will ensure a resilient and responsive network environment.

Related Links

- [Use Apple products on enterprise networks - Apple Support](#) - Official instructions from Apple
- [Default TCP and UDP Port Usage](#) - A detailed guide on configuring network settings within FileWave

Using PowerShell to Remotely Check the Windows FileWave Client Status

What

The FileWave Client on Windows is like any other software service...the service can be impacted by computer uptime, user interference, crashes, etc. This article gives you a way to INDEPENDENTLY check that a list of devices has the FileWave Client, and it is in working order (or not).

When/Why

Will you need this frequently? Unlikely, but all the same, it is a great tool for sweeping a network to look for devices and confirm the FileWave client (for Windows only). The code here does make some assumptions about your environment, but those are called out below.

How

So, you think the FileWave client may be broken or missing on some endpoints? Wouldn't it be great if you could verify that remotely rather than having to confirm the devices by hand. The following allows you to do just that.

```
#import a list of computers
$mypath=$MyInvocation.MyCommand.Path
$mypath=Split-Path $mypath -Parent
try {
    $computers=Get-Content $mypath\computers.txt -ErrorAction Stop
} catch {
    #no computers.txt file found
    write-host "`nTo use this utility, a text file called computers.txt must exist in the same location as the
script. The file should contain one computer name or IP per line"
    break
}

foreach ($computer in $computers) {
    $online=$false
    try{
        #try to resolve the name
        $online = Resolve-DnsName $computer -quicktimeout -ErrorAction Stop
        $online = $true
    }catch{
        #Catching errors...machine offline
        $online= $false
    }

    if (!$online) {
        #device not online...show it in UI so that we see progress, but don't write it to the results file since
it isn't actionable
        write-host "$computer, Not online"
    } else {
        #device online, so let's just see if the service is there
        $fw_service=""
        try{
            #Getting service ...sometimes device might not allow collection (if RPC is unavailable for instance)
            $fw_service = Get-Service -ComputerName $computer -Name 'FilewaveWinClient' -ErrorAction Stop
            $fw_service=$fw_service.Status
        }catch{
            #Catching errors...no filewave service
            $fw_service="no"
        }

        if ($fw_service -eq "no") {
            #no need to look further since we either can't talk to RPC, or there is no FW service
            write-host "$computer, No FW Service or RPC unavailable"
            Add-Content -Path $mypath\output.txt -Value "$computer, FW Needs Installed or RPC Unavailable"
```

```

    } else {
        #fw is there as a service, so let's return status, version, and server address
        try {
            #using C$ share, which won't require winrm
            $TargetPath = "\\$computer\C$\Program Files (x86)\FileWave\fwclld.exe"
            $fw_version = [System.Diagnostics.FileVersionInfo]::GetVersionInfo($TargetPath)
            $fw_version = $fw_version.ProductVersion
        } catch {
            #Catching errors
            $fw_version="version not readable"
        }

        #get fw server address from registry
        try {
            #read server address from registry
            #we need remote registry turned on to read, but we'll turn it back off
            #note this does not account for an environment where remote-registry is on by default...if so,
comment out the remote registry lines
            Get-Service -ComputerName $computer -Name RemoteRegistry | Set-Service -StartupType Manual -
PassThru| Start-Service

            $Reg = [Microsoft.Win32.RegistryKey]::OpenRemoteBaseKey('LocalMachine', $computer)
            $RegKey= $Reg.OpenSubKey("SOFTWARE\WOW6432Node\FileWave\WinClient")
            $fw_server = $RegKey.GetValue("Server")

            #turn remote registry back off
            Get-Service -ComputerName $computer -Name RemoteRegistry | Set-Service -StartupType Disabled -
PassThru| Stop-Service

        } catch {
            #Catching errors...no registry
            $fw_server="server address not readable"
        }

        #write the output
        write-host "$computer, $fw_service, $fw_version, $fw_server"
        Add-Content -Path $mypath\output.txt -Value "$computer, $fw_service, $fw_version, $fw_server"

    }
}
}

```

Assumptions made in the above code:

1. There is a text file called computers.txt in the same location as the Powershell script
2. That computers.txt file contains a computer name or IP per line (name is better if you have dynamic DNS)
3. That the Powershell itself is running from a Domain Admin account...this avoids any credential related issues
4. It is assumed that WinRM is not enabled in your environment (if it is this code could easily be made more elegant)

Note that this script could easily be modified to look at other services, to authenticate differently, and to take remediation. As provided, it simply provides a list of results of device name, FW service status, FW client version, and FW server address assigned. All very useful information for troubleshooting. PSEXEC is highly recommended for taking corrective action.

Related Content

- [Script Best Practices](#)
- [Using PsExec to Remotely Restart the FileWaveWinClient Service](#)
- [FileWave Client Status Check: How to ask the client what it is doing on macOS and Windows](#)

Digging Deeper

If you want a resource to pre-sweep the device list for devices that are online separately, you can use the following. A refined list will just make the above script run a bit faster.

```

#Let's just look for a list of devices online

#import a list of computers
$mypath=$MyInvocation.MyCommand.Path

```



```
$mypath=Split-Path $mypath -Parent
$computers=Get-Content $mypath\online_test.txt

foreach ($computer in $computers) {
    $online=$false
    try{
        #try to resolve
        $online = Resolve-DnsName $computer -quicktimeout -ErrorAction Stop
        $online = $true
        write-host $computer
    }catch{
        #Catching errors...machine offline
        $online= $false
    }
}
```

When does the inventory client run scans?

Inventory

The FileWave Client runs an inventory scan at startup, every time you execute a verify on the client, and every 24 hours.

PSEXec as a Helper in Troubleshooting

What

The PS Tools from Microsoft (from SysInternals) are a terrifically powerful tool to help you troubleshoot when all else fails. In this article we'll look at how you can use PSEXec to help troubleshoot an ill-behaving FileWave Client.

When/Why

From time to time, things don't work right. None of us would be employed if this weren't the case, so let's look on the bright-side of that! But what to do if a FileWave client on a Windows device is misbehaving, and you can't communicate through normal FileWave channels? PSEXEC to the rescue.

How

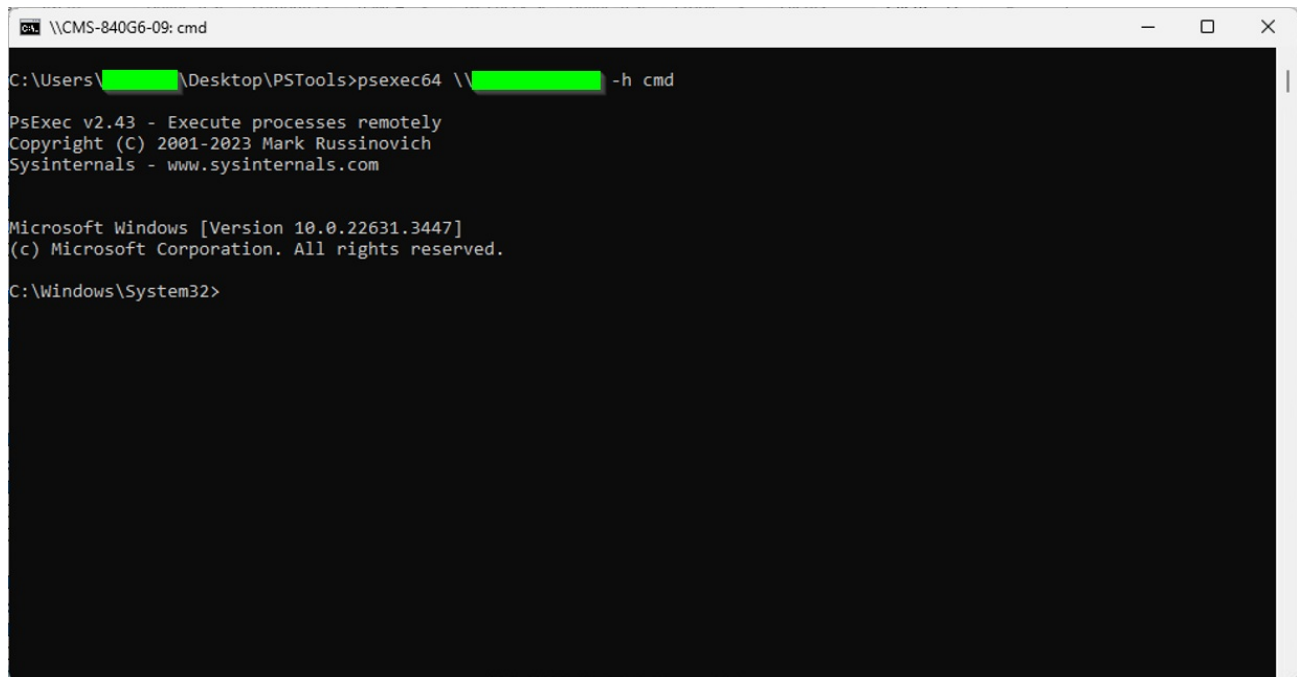
Assumptions made in the following:

- 1) You have download PSTools, and unzipped
- 2) That you launched a cmd prompt as a domain admin user (makes credentials issue easier to deal with)
- 3) That you have changed directory into the directory where PSTools is located

We'll start by simply connecting to the remote computer by name in an interactive PSEXEC shell:

```
psexec64 \\computername -h cmd
```

You'll end up in a shell like the below ('exit' will allow you to leave that shell)



```
\\CMS-840G6-09: cmd
C:\Users\████████\Desktop\PSTools>psexec64 \\████████ -h cmd
PsExec v2.43 - Execute processes remotely
Copyright (C) 2001-2023 Mark Russinovich
Sysinternals - www.sysinternals.com

Microsoft Windows [Version 10.0.22631.3447]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>
```

Now, what's remarkable about this is that shell is running as your domain admin account, and you can do anything on it you can do from the command line. This article isn't meant to be a Windows CLI primer, but the following are some examples of things we could do if we assume we have a device that isn't reporting in correctly:

1. Check the FileWave Client Service:

- ```
sc query filewavewinclient
```

```
SERVICE_NAME: filewavewinclient
 TYPE : 10 WIN32_OWN_PROCESS
 STATE : 4 RUNNING
 (STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
 WIN32_EXIT_CODE : 0 (0x0)
 SERVICE_EXIT_CODE : 0 (0x0)
 CHECKPOINT : 0x0
 WAIT_HINT : 0x0
```

## 2. Stop the FW Client Service

- `sc stop filewavewinclient`

## 3. Restart the FW Client Service

- `sc start filewavewinclient`

## 4. If the service won't start or stop, maybe we need to kill it forcefully by:


### 1. Looking for the client process

- `tasklist | findstr fwcl`

```
C:\Windows\System32>tasklist | findstr fwcl
fwcl.exe 16264 Services 0 29,660 K
```

### 2. And then killing it by PID

- `taskkill /PID 16264 /F`

 Note that this same procedure can be very helpful to clear up a misbehaving Windows Update agent. When Windows Update hangs, the service itself usually won't stop. Taskkill /SVC | find wuauclt will identify the proper task to stop to correct this. (A reboot is also corrective for this, but obviously impacts the use of the device)

## 5. Check the FW Client Log for entries from today

- `type c:\programdata\filewave\fwclient\fwcl.log | findstr mm-dd`

- (where mm-dd is today's date such as 05-16)

## 6. Get the IP of the workstation

- `ipconfig`

## 7. Restart the device (which is obviously destructive to any existing user)

- `shutdown -r -t 0 -f`

## 8. Determine if there are other users logged in

- `quser`

```
C:\Windows\System32>quser
USERNAME SESSIONNAME ID STATE IDLE TIME LOGON TIME
bctowery console 2 Active none 5/14/2024 12:04 PM
```

## 9. Get the last boot time

- `wmic path win32_operatingsystem get lastbootuptime`

```
C:\Windows\System32>wmic path win32_operatingsystem get lastbootuptime
LastBootUpTime
20231208021432.923243-300
```

# Related Content

- [PS Tools](#)
- [Great reference for Windows CLI commands \(SS64\)](#)