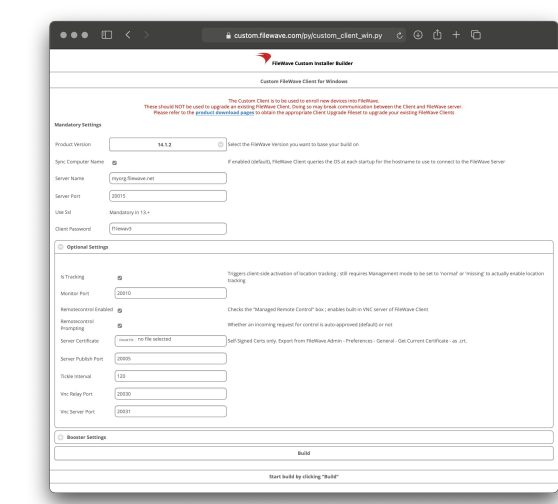


# Windows Client Pre-Requisites

All you need to do to enroll a Windows Client is to deploy a customized FileWave Client MSI to your machines. We typically recommend using an existing tool capable of deploying a MSI such as Group Policy. This customized MSI can also be "baked" into a Windows image that can be deployed via FileWave's Imaging Virtual Server (IVS) so that freshly imaged Windows machines will automatically enroll into FileWave. Lastly, the FileWave Client MSI can be manually distributed and installed on any Windows machine that has local Administrator privileges.


If your organization uses Microsoft Entra ID and your users authenticate using Microsoft Entra ID credentials into their Windows machines, please consider enrolling your Windows machines into FileWave via Microsoft Entra ID. This will also allow for Windows MDM management within FileWave. Learn more on our [Windows MDM](#) article.



## Generating a custom FileWave Client MSI

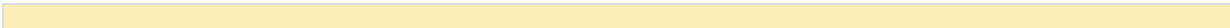
1. Open the [FileWave Customer Installer Builder](#) for Windows.
2. Fill out the settings accordingly.
3. Click the "Build" button and wait for the automatic download.
4. Extract ZIP and install the customized FileWave Client MSI.

Mandatory Settings
Product Version = Your FileWave Server Version
Sync Computer Name = Windows Hostname will be FileWave Client Name (recommended)
Server Name = Fully Qualified Domain Name of your FileWave Server (required)
Server Port = 20015 (do not modify)
Client Password = Password used to modify client preferences remotely (not used in FW 16.0+)



**Note:** The default Server Port setting above is 20015. However, SSL is now required, and the system will automatically use port 20017 instead when 20015 is entered. Do not manually set the port to 20017. Always enter 20015, and the system will handle the SSL port change for you.

Optional Settings
Is Tracking = Is Location Tracking Enabled for Windows Clients
Monitor Port = Port used for FileWave Client Monitor - 20010 (do not modify but also this port is no longer used as of FileWave 16.0.0)
Remotecontrol Enabled = Screen-sharing enabled for Windows Clients
Remotecontrol Prompting = Whether or not to Prompt the end-user before starting screen-sharing session
Server Certificate = Self-Signed Certificate only; not required for CA-signed certificate
Server Publish Port = 20005 (do not modify)
Tickle Interval = Idle time for Windows Clients before checking for new Model Update (do not modify)
Vnc Relay Port = 20030 (do not modify)
Vnc Server Port = 20031 (do not modify)



## Deploying the FileWave Client with Group Policy (Startup Script)

If you have no other way to easily deploy the FileWave Client to Windows systems, but they are all tied to an Active Directory then leveraging Group Policy (GPO) can allow you to rapidly deploy the client with very little effort.

### Customize the Script

- Save the script shown below and save it to Deploy-FileWave.ps1 on your desktop.
- You will want to make sure you edit it to point at the correct server location for the MSI installer that you created from <https://custom.filewave.com> to use for the install and make sure that the computer has permission to access the file.
- Open \<your-domain>\SYSVOL\<your-domain>\Policies\{YourGPO\_GUID}\Machine\Scripts\Startup in File Explorer (or use the GPMC “Show Files...” button on the Startup Scripts pane) to put the script in your domain.
- The machine’s computer account (DOMAIN\COMPUTER\$) must have read permission on \\server\share\FileWaveCustom.msi. If it doesn’t, the copy step will fail—so be sure your share ACLs allow Domain Computers or the specific computer objects.

### Create FileWave Client Group Policy

- Open Group Policy Management in the Active Directory.
- Create a GPO and navigate to Computer Configuration >> Policies >> Windows Settings >> Scripts (Startup/Shutdown) >> Startup
- Double-click Startup, then click the PowerShell Scripts tab.
- Click Add..., then Browse... and pick Deploy-FileWave.ps1 from the Startup folder.
- Click OK to close out all dialogs and link the GPO where you need it. (Please test against 1 machine first)

### Enable "Turn on Script Execution" Policy (This policy can be combined into above GPO)

- Open Group Policy Management in the Active Directory
- Navigate to Computer Configuration >> Policies >> Administrative Templates >> Windows Components/Windows PowerShell >> Turn on Script Execution: Set to Enabled and click OK.

### ADDITIONAL INFORMATION

The script checks if the FileWave Client (fwcmd) is running, if not it will copy the installer from the share, install and generate a log file under C:\temp. If anything goes wrong (missing share access, MSI failure, etc.), you’ll see errors in C:\temp\Deploy-FileWave.log, trimmed and rotated down to the last 100 KB on each reboot.

Deploy-FileWave.ps1 script:

```
<#
.SYNOPSIS
    Deploys the FileWave agent via a Group Policy startup script, always logging output.
.DESCRIPTION
    Checks whether the FileWave process is running; if not, copies the MSI from the network share
    to a local temp directory and installs it in silent mode. All verbose, warning, and error output
    is logged to C:\temp\Deploy-FileWave.log, trimmed to the last 100 KB at each run.
.NOTES
    - Verbose output is forced by setting $VerbosePreference = 'Continue'.
#>

# Force verbose output so all Write-Verbose messages appear in transcript
$VerbosePreference = 'Continue'

# Define directory and log file
$destDir = 'C:\temp'
$logFile = Join-Path -Path $destDir -ChildPath 'Deploy-FileWave.log'
$maxBytes = 100KB

# Ensure the log directory exists
if (-not (Test-Path -Path $destDir)) {
    try {
        New-Item -Path $destDir -ItemType Directory -Force | Out-Null
    } catch {
        Write-Error "Failed to create log directory '$destDir': $_"
        Exit 1
    }
}

# Trim existing log to last 100 KB
if (Test-Path -Path $logFile) {
    try {
        $fs = [System.IO.File]::Open($logFile, 'Open', 'ReadWrite')
        if ($fs.Length -gt $maxBytes) {
```

```

        $buffer = New-Object byte[] $maxBytes
        $fs.Seek(-$maxBytes, 'End') | Out-Null
        $fs.Read($buffer, 0, $maxBytes) | Out-Null
        $fs.SetLength(0)
        $fs.Write($buffer, 0, $maxBytes)
    }
    $fs.Close()
} catch {
    Write-Warning "Failed to trim log file: $_"
}
}

# Start transcript logging
Start-Transcript -Path $logFile -Append
Write-Verbose "Transcript started at '$logFile'."

$ProcessName = 'fwcld'
Write-Verbose "Checking for process '$ProcessName'..."

# Check if FileWave agent is running
try {
    Get-Process -Name $ProcessName -ErrorAction Stop | Out-Null
    Write-Verbose "Process '$ProcessName' is currently running."
    $isRunning = $true
} catch {
    Write-Verbose "Process '$ProcessName' not found. Proceeding with installation."
    $isRunning = $false
}

if (-not $isRunning) {
    Write-Verbose "Deploying FileWave agent..."

    # Ensure destination directory exists for MSI
    if (-not (Test-Path -Path $destDir)) {
        Write-Verbose "Directory '$destDir' does not exist. Creating directory..."
        try {
            New-Item -Path $destDir -ItemType Directory -Force | Out-Null
            Write-Verbose "Successfully created directory '$destDir'."
        } catch {
            Write-Error "Failed to create directory '$destDir': $_"
            Stop-Transcript
            Exit 1
        }
    } else {
        Write-Verbose "Directory '$destDir' already exists."
    }

    # Define installer paths
    $installerSharePath = '\\server\share\FileWaveCustom.msi'
    $installerLocalPath = Join-Path -Path $destDir -ChildPath 'FileWaveCustom.msi'
    Write-Verbose "Installer source: '$installerSharePath'."
    Write-Verbose "Installer destination: '$installerLocalPath'."

    # Copy the MSI installer
    Write-Verbose "Copying installer from network share to local directory..."
    try {
        Copy-Item -Path $installerSharePath -Destination $installerLocalPath -Force
        Write-Verbose "Successfully copied installer to '$installerLocalPath'."
    } catch {
        Write-Error "Failed to copy installer: $_"
        Stop-Transcript
        Exit 1
    }

    # Install MSI and capture logs
    $installLog = Join-Path -Path $env:TEMP -ChildPath 'FileWaveInstall.log'
    $msiArgs = "/i `"$installerLocalPath`" /qn /norestart /l*vx `"$installLog`""
    Write-Verbose "Starting MSI installation. Command: msiexec.exe $msiArgs"

    $process = Start-Process -FilePath 'msiexec.exe' -ArgumentList $msiArgs -Wait -PassThru

```

```
if ($process.ExitCode -eq 0) {
    Write-Verbose "Installation completed successfully (exit code 0). Log: '$installLog'."
} else {
    Write-Error "MSI installation failed (exit code $($process.ExitCode)). See '$installLog'."
    Stop-Transcript
    Exit $process.ExitCode
}

Write-Verbose 'FileWave installation process finished.'
} else {
    Write-Verbose "Skipping deployment; '$ProcessName' already running."
}

# Stop transcript logging
Stop-Transcript
Write-Verbose "Transcript stopped. Log written to '$logFile'."
```

---

🕒Revision #18

★Created 12 June 2023 20:39:01 by Josh Levitsky

✍Updated 21 April 2025 15:02:23 by Josh Levitsky