FileWave General Info

- FileWave Components
- How does FileWave work?
- Default TCP and UDP Port Usage
- Allow External Devices to Connect to the FileWave Server and Boosters
- Naming conventions and acronyms
- Scripting
    - Execute macOS scripts as Console User
    - macOS 10.15+ and zsh shell for scripting
    - Referencing Launch Arguments in Scripts
    - Script Best Practices
    - Scripting Languages supported in FileWave
    - Using PsExec to Test PowerShell 32bit Scripts on Windows with FileWave

- Technical Support Tools
- What languages does FileWave support?
- Troubleshooting
    - Apple MDM Troubleshooting
    - Backup Procedures for FileWave Hosted Servers
    - FileWave Error Codes
    - FileWave Log File Locations
    - How to Restart FileWave Components
    - Resolving Network Issues with FileWave Server or Boosters on macOS when using Carbon Black EDR Extension
    - What is Compatibility Mode?

- Rufus - Creating bootable USB drives

FileWave Components

In this section, we will describe the key FileWave components:

FileWave Server

The FileWave Server is the central repository hosting every file to be delivered to Clients. It consists of five processes and a web server. The first process interacts with logged-in Administrators. The second process services incoming requests from Clients and Boosters. The third process interacts with a directory server through LDAP. The fourth process communicates with Apple and Microsoft software update servers to download the current lists of available software updates. The fifth process is the Postgres database service for Inventory and MDM. Finally, the web server is the FileWave MDM Server; it handles Mobile Device Management (MDM) components. Detailed information on setting up the FileWave server is covered in Chapter 2 of this manual.

FileWave Central application

The FileWave Central application is the primary interface to the FileWave Server. The FileWave Admin displays different views that give a representation of the FileWave Server's database. These views are the Dashboard, Clients, Filesets, Associations, Imaging, optional Classroom, iOS Inventory, License Management, Boosters, and Inventory Queries views. FileWave Admin also acts as the unified management console for creating and administering FileWave administrator accounts; network imaging for the Imaging Appliance; managing Apple DEP and VPP associations; system software updates for iOS 9+, OS X (macOS) and Windows; and overall management of all devices and Filesets. Multiple instances of the FileWave Admin application can be in use at the same time with specific devices, Groups and Filesets assigned to various administrator accounts. Detailed information on configuring and using the FileWave Admin application is in Chapter 2 of this manual.

FileWave Anywhere

The FileWave Web Console is an Inventory tool designed to help with quick FileWave inventory references for specific clients in your server. Within the Web Console you will be able to view all devices currently enrolled, their Filesets, installed applications, users who have logged in, what groups they are apart of, and in the case of MDM enrolled Apple devices the command history. For more information please visit the page linked here.

FileWave Client (macOS and Windows)

The FileWave Client has two processes, fwcld and fwGUI. The first runs as a Launch Daemon on macOS and as a service on Windows. This means it runs in the background without any user interface. The client starts automatically after being installed and each time the computer boots. The fwcld process always runs with root (Mac) or local system (Win) privileges to allow for maximum access by any management operations. The second process, fwGUI, handles user interaction with the client, such as asking the client to quit open applications and informing them of the status when activating Filesets that require rebooting. The fwGUI process is what provides the Kiosk / self-service functionality. The Imaging Virtual Server (IVS) contains a modified version of the fwcld for reporting its status back to the FileWave Admin. Chapter 4 of this manual covers the installation and configuration of the FileWave client.

Filesets

FileWave's patented Fileset technology provides the ability to distribute applications, content, and management settings at the file level. While FileWave supports distribution of the standard .pkg and .msi packages, its capability to distribute individual files, application bundles, content, and management profiles allows for a level of granular control missing from other client management solutions. Filesets can be distributed to clients and cached for activation at a later date; a process that provides maximum scalability and control over the deployment cycle.

When a Fileset is distributed, it is protected from network outages. If there is an interruption in the transmission, FileWave will resume the distribution as soon as the network is restored. Filesets can also be modified after distribution. If any portion of the Fileset is modified by the administrator, only that specific portion of the Fileset is sent out to the associated clients. This process greatly reduces the network traffic. Another feature is the ability to deploy content and roll back to the previous version of that item if there is a problem with the deployed item. Self-healing functionality allows a Fileset to automatically repair itself if the end user deletes a portion of the payload. Chapter 5 of this manual covers the creation, configuration, distribution, and management of Filesets.

Self-service Kiosk

FileWave's self-service Kiosk provides the ability to allow end users access to content with their own device. In a BYOD deployment, you could post institutionally owned applications, documents, and updates for the end users to install at their convenience. In most of the deployment models, you can assign custom application sets to Groups as needed. Users do not need to be local administrators in order to install applications or content. End users can be provided with new applications, updates, documents, and other key content needed. The end user also has the option of un-installing that same content to free up space as needed. Use and configuration of the Kiosk is covered in Chapter 4.

Booster

The FileWave Booster is designed to act as a Fileset caching device for computer clients assigned to it as well as as to handle all Client-Server communications. Unlimited Boosters are allowed, regardless of license count or type. The FileWave Boosters allow administrators to increase the speed and scale of the Server's distribution of Filesets to Clients as well as offloading the overhead for constantly opening sockets for Client communications. When a set of Clients are connected to a Booster, their total network load on the Server will be roughly equivalent to a single Client connecting directly to the Server from that location. The use of Boosters can benefit remote sites with bandwidth constraints by providing a focused, local target for Clients as well as a single point of distribution from upstream.

Boosters are designed to work with Windows, OS X (macOS), and Android clients. iOS clients do not have the ability to use a Booster for cached Filesets, but they can utilize a Mac caching server, part of macOS that runs just fine on a Mac mini.

Imaging Virtual Server (IVS)

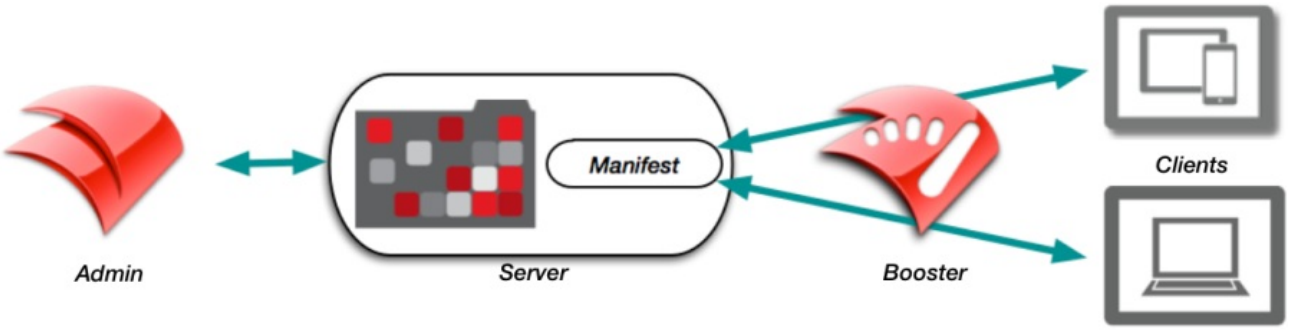
The FileWave Imaging Virtual Server is a standalone Linux container (CentOS) that you can download from the Support site and run on any device that supports a Virtual Machine application, such as VMware™. The IVS provides NetBoot and PXEboot services. Storage for network images for Mac and Windows, as well as Windows Drivers images is now on the FileWave server. FileWave Admin provides the management console for associating network images with designated client computers.

Dashboard

FileWave provides an integrated Dashboard displaying a snapshot of the current status of the FileWave infrastructure. The Dashboard can be "torn off" to run on a separate display, and you can copy the URL of the Dashboard to provide to another systems administrator for viewing on their own device, including on a tablet. The information posted includes the status of all major services, such as DEP, VPP, and LDAP; account sync status; server performance status; and server licenses; plus much more.

How does FileWave work?

FileWave is a combination of tools and services integrated through a common administrative application front end. Since the FileWave Admin application is multi-platform, using Apple's macOS and Microsoft Windows, a systems administrator is not limited to a single platform for day-to-day lifecycle management. The FileWave basic workflow involves the 'push-pull' interaction between the FileWave Admin, FileWave server, and FileWave clients.



A FileWave administrator creates a Fileset which resides on the FileWave Server. Filesets contain applications, images, profiles, books, settings, or other content are associated with client devices. The FileWave Client is sent a Manifest that identifies a new Fileset. The Client then requests the Fileset, that may be cached at a FileWave Booster in order to provide better scalability. A basic FileWave configuration consists of a single administrator connecting to a FileWave Server to manage and maintain a set of clients. Multiple administrators may be in use, as well as Boosters to decrease network load by distributing Filesets closer to the client systems as well as, with FileWave handling all Client-Server communications, with the exception of inventory. Each of the major components is described in the following section.

To learn more you can review the Evaluation Guide or video our video based intro course FileWave Foundry: Onboarding Videos

Default TCP and UDP Port Usage

FileWave software uses the below-listed TCP/IP ports. These are default settings and may be configured to listen on different ports if required. Consider FileWave Server should not have IPv6 enabled for the best experience.

Port Testing

Please consider downloading the FileWave Port Testing macOS/Windows utility to confirm communication of Google Cloud Messaging, Apple Push Notifications and connectivity between device network(s) and Server/Boosters.

The following may be run from the server to confirm Apple, Microsoft, and FileWave services:

Server Command Line

```
sudo /usr/local/filewave/python/bin/python /usr/local/filewave/django/manage.pyc check_connections
```

TeamViewer Ports

TeamViewer has an additional set of ports to consider:

https://community.teamviewer.com/English/kb/articles/4139-ports-used-by-teamviewer

FileWave Server Ports

> MDM default port is now 20445 as shown throughout this KB. On older versions of FileWave, this was 20443. To confirm the defined port, check the Port setting in FileWave Central > Preferences > Mobile > MDM Server > Port

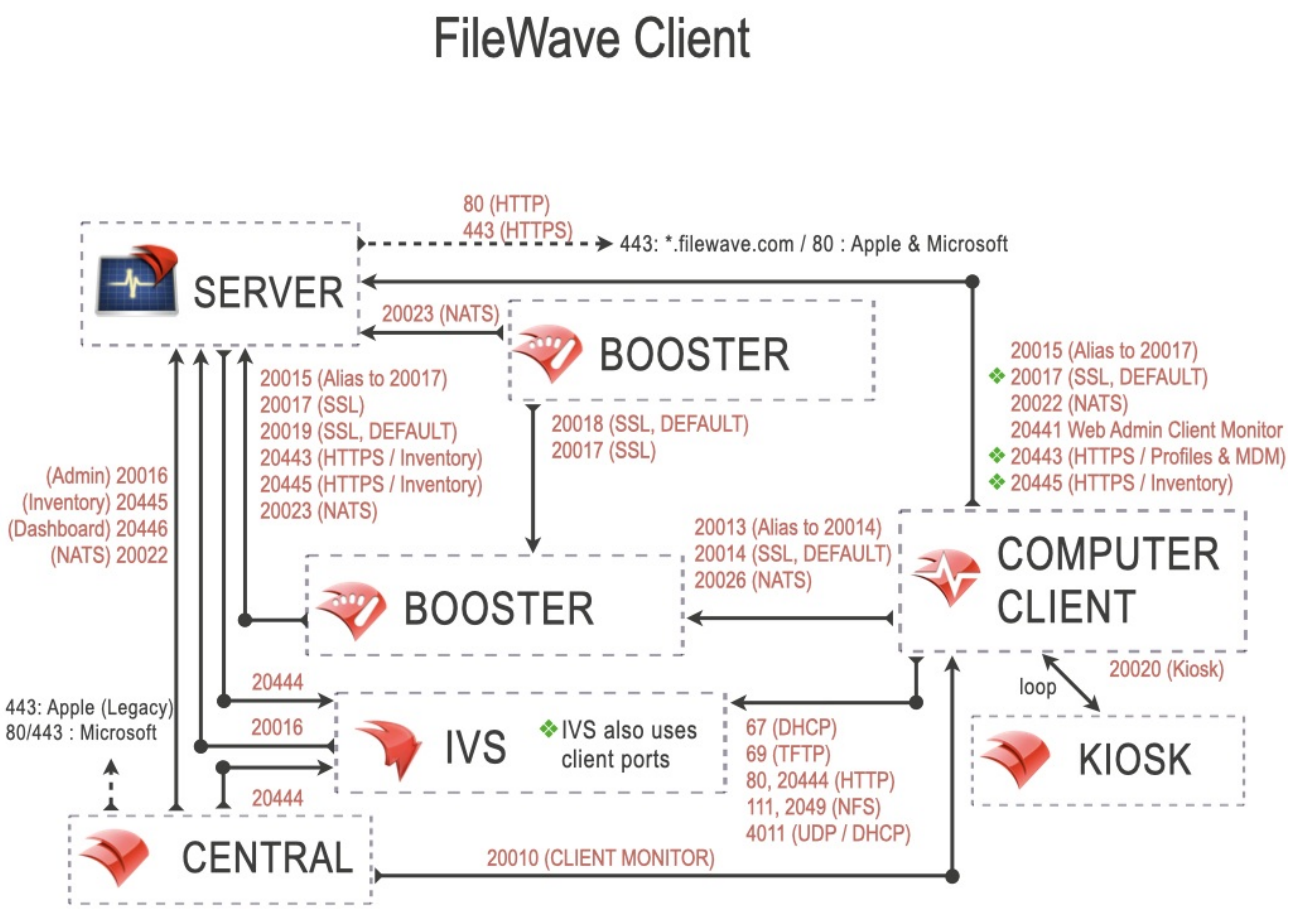
Server Ports	Service	Protocol	Server In/Out	Description
80	HTTP	TCP	Outgoing	FileWave Software Updates (apple.com & microsoft.com) ***
443	HTTPS	TCP	Outgoing	FileWave License Server (fwks.filewave.com & logstash.filewave.com) FileWave Software Updates (apple.com) *** FileWave/TeamViewer Session Server (rcs.filewave.com) FileWave/TeamViewer Push Notification Server (fwpn.filewave.com)
443	HTTPS	TCP	Incoming	FileWave Anywhere API endpoints - Anywhere API (v2 API)
20015	Proprietary	TCP	Incoming	FileWave Client to Server; Legacy but should be used as the port by the Agent. SSL traffic will run on 20017. (Server does not listen to this port in FileWave 15.4+)
20016	SSL	TCP	Incoming	FileWave Central to Server
20017	SSL	TCP	Incoming	FileWave Client to Server: Secure
20019	SSL	TCP	Incoming	Booster to Server: Priority Traffic
20022	SSL**	TCP	Incoming	FileWave Central to Server: NATS FileWave Client to Server: NATS IVS to Server: NATS
20023	SSL**	TCP	Incoming	FileWave Booster to Server: NATS
20124	SSL	TCP	Incoming	FileWave Server JSON Websockets (JWT) Websocket connections for NATS SERVER used between FW Anywhere and FW

				Server. (Added 14.10.0)
20441	Proprietary	TCP	Incoming	FileWave Client to Server: Remote Client Monitor
20443	HTTPS	TCP	Incoming	FileWave Client to Server: Profiles Booster to Server: Inventory/Discovery API endpoints - Command Line API (v1)
20445	HTTPS	TCP	Incoming	FileWave Client to Server: Inventory Booster to Server: Inventory/Discovery API endpoints - Command Line API (v1)
20446	HTTPS	TCP	Incoming	FileWave Central and FileWave Anywhere to Dashboard

* NATS includes: Remote Control Publishing, Remote Control Routing, device renaming, revoking device certificates, push notifications
** Only encrypted when compatibility mode is disabled
*** Also used by FileWave Central to vendor Software Update Servers.

FileWave Client Ports

Client Ports	Service	Protocol	In/Out	Description
443	HTTPS	TCP	Out	FileWave Kiosk (*.filewave.cloud)
20010	Proprietary	TCP	In	FileWave Central to Client: Client Monitor: macOS, Windows & Android APK

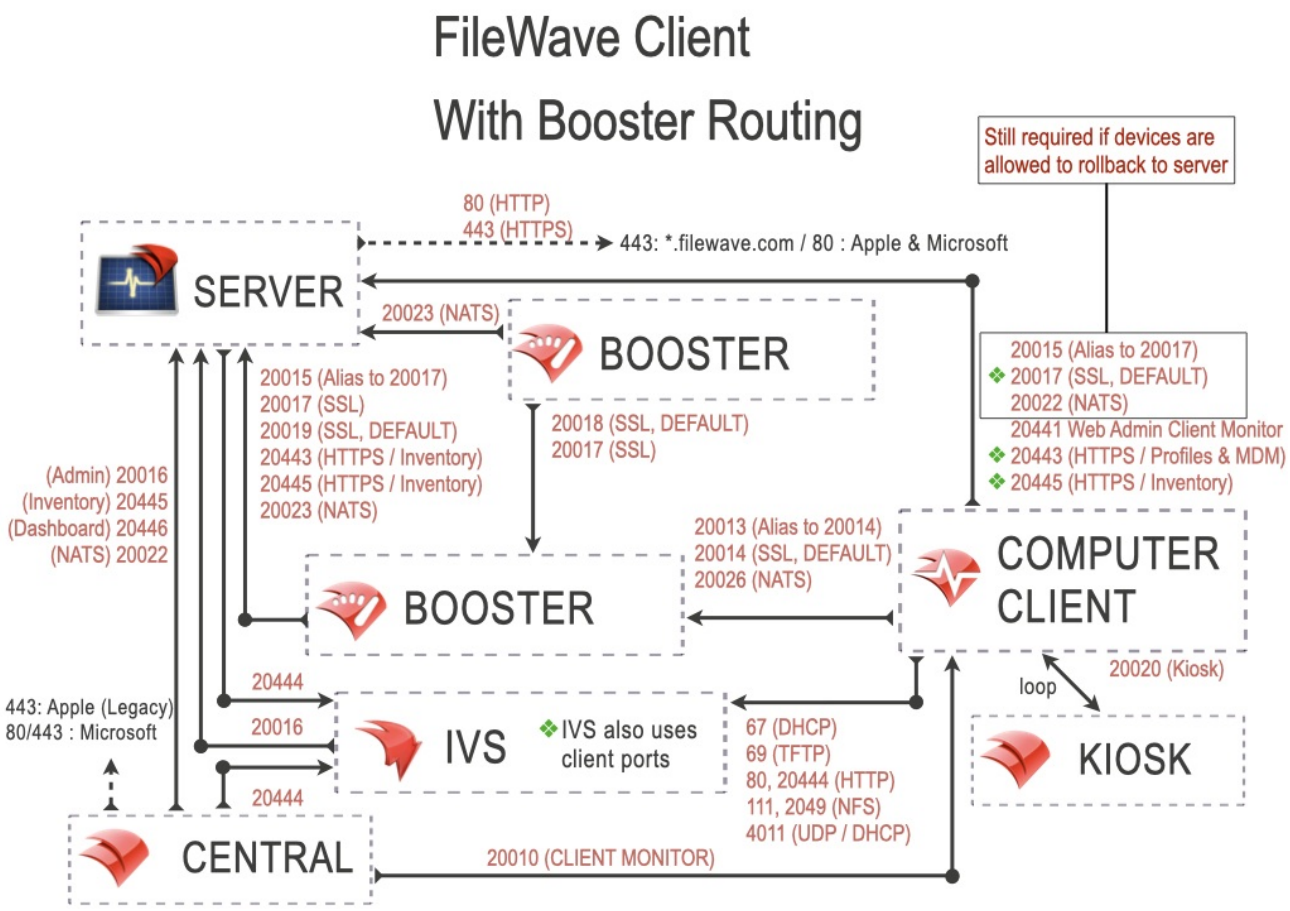


FileWave Booster Ports

Booster Ports	Service	Protocol	Server In/Out	Description

20013	Proprietary	TCP	Incoming	FileWave Client to Booster; legacy (Booster does not listen to this port in FileWave 15.4+)
20014	SSL	TCP	Incoming	FileWave Client to Booster: Secure (Booster Priority fallback)
20018	SSL	TCP	Incoming	Booster to Booster: Priority Traffic
20026	SSL ††	TCP	Incoming	FileWave Client to Booster connections using NATS Server

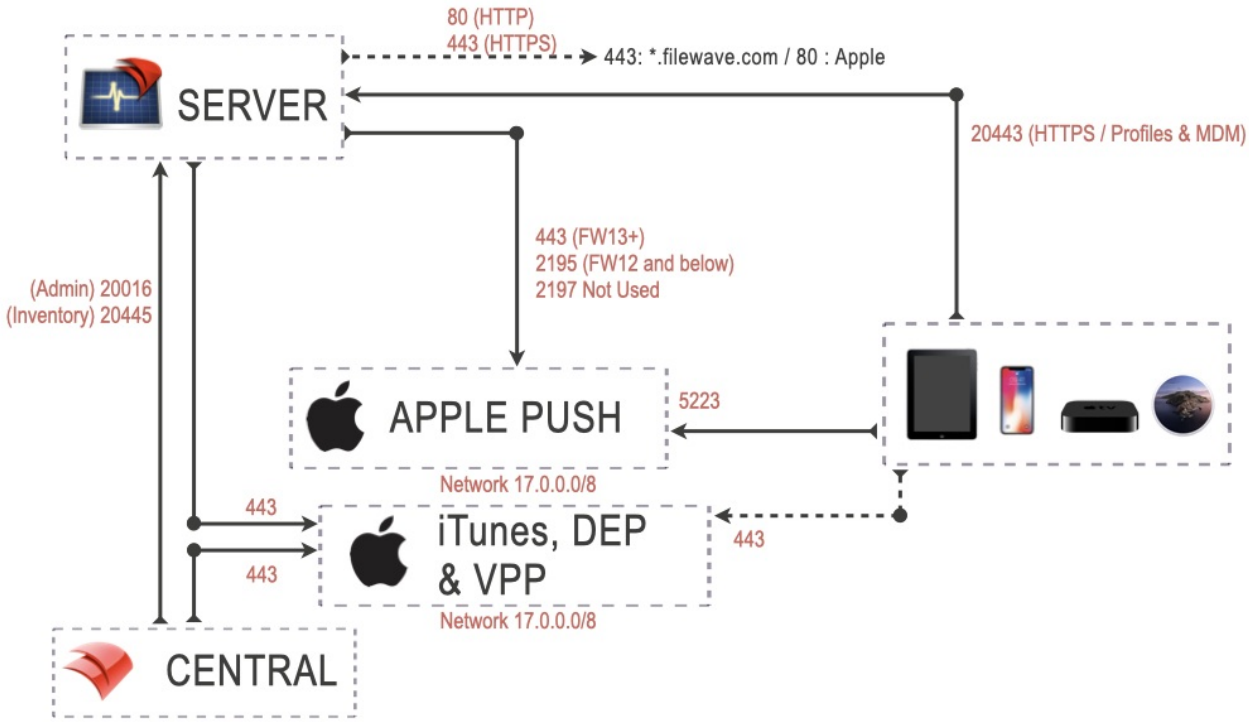
† NATS includes: Remote Control Publishing, Remote Control Routing, device renaming, revoking device certificates, push notifications
†† Only encrypted when compatibility mode is disabled



Apple MDM Ports

Apple MDM Ports	Service	Protocol	Server In/Out	Description
443	HTTPS	TCP	Outgoing	FileWave Server to Apple's servers (17.0.0.0/8) FileWave Admin to iTunes, DEP & VPP (17.0.0.0/8) Device to iTunes, DEP & VPP (17.0.0.0/8)
5223	APNS	TCP	Outgoing	FileWave Server to Apple's servers (17.0.0.0/8)
20443	HTTPS	TCP	Incoming	Device to Server: Profiles & MDM
20445	HTTPS	TCP	Incoming	FileWave Central to Server

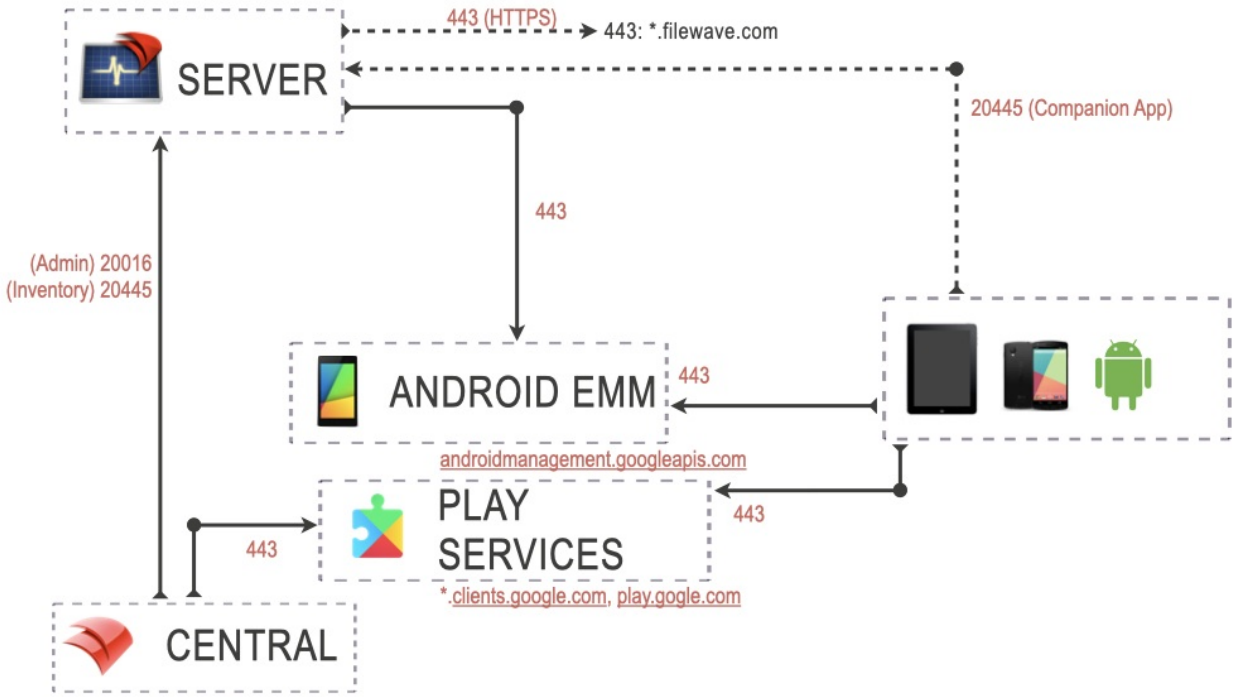
Apple MDM



Android EMM Ports

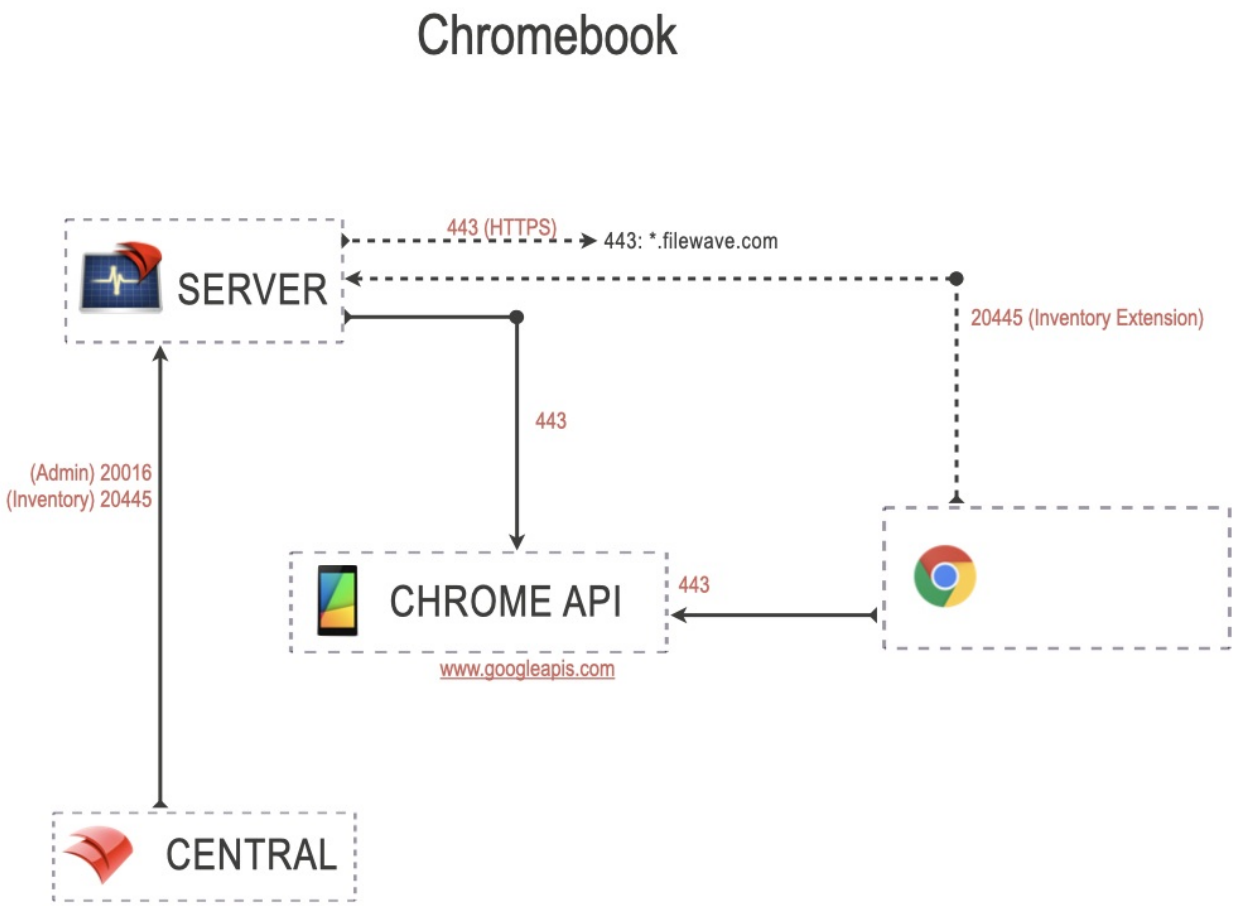
Android EMM Ports	Service	Protocol	Server In/Out	Description
443	HTTPS	TCP	Outgoing	Server to EMM commands (androidmanagement.googleapis.com) Device to Activation servers (*.clients.google.com) Device to Play Store (play.google.com) EMM commands (androidmanagement.googleapis.com) FileWave Central to Play Store (play.google.com)
20016	SSL	TCP	Incoming	FileWave Central to Server
20445	HTTPS	TCP	Incoming	FileWave Central to Server: Inventory Companion App to Server: Location Tracking

Android EMM



Chromebook Ports

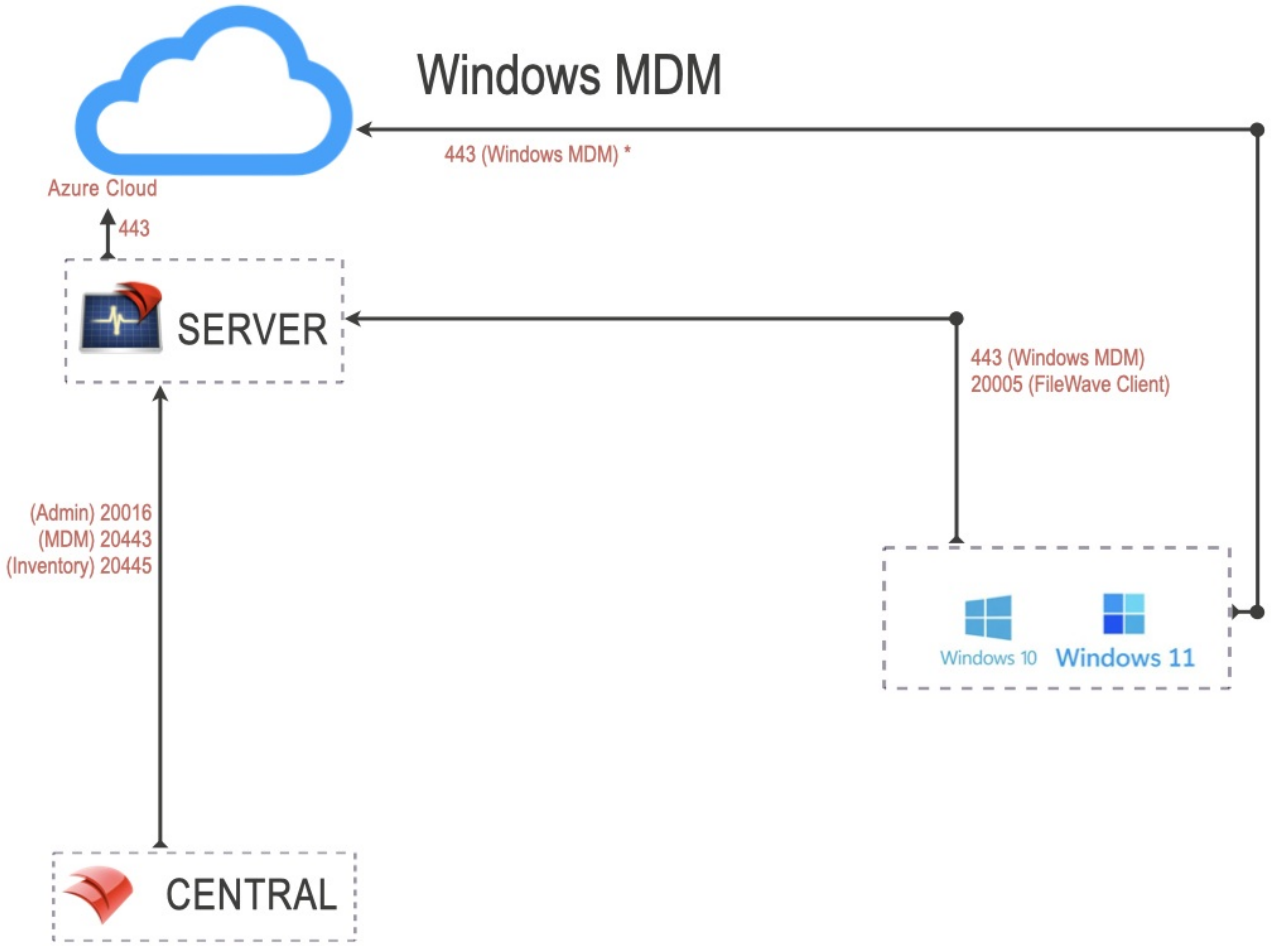
Chromebook Ports	Service	Protocol	Server In/Out	Description
443	HTTPS	TCP	Outgoing	Server to Chrome API Chromebook to Chrome API (www.googleapis.com)
20016	SSL	TCP	Incoming	FileWave Central to Server
20445	HTTPS	TCP	Incoming	FileWave Central to Server Chromebook Inventory Extension to Server



Windows MDM Ports

Windows MDM Ports	Service	Protocol	Server In/Out	Description
443	HTTPS	TCP	Incoming	Device to Server (Enrollment URL)
443	HTTPS	TCP	Outgoing	Server to Windows MDM (*.azure.com) Device to Windows MDM (*.azure.com)

NOTE: The FileWave client will also be installed and all previously listed FileWave client ports are required.



FileWave IVS Ports

IVS (Imaging) Ports	Service	Protocol	Server In/Out	Description
67	DHCP	UDP		Client to IVS‡‡
69	TFTP	UDP		Client to IVS‡‡
80	HTTP	TCP		Client to IVS
111	NFS	TCP/UDP		Client to IVS‡
4011	DHCP	TCP/UDP		Client to IVS: UEFI PXE‡
2049	NFS	TCP/UDP		Client to IVS‡
20015	Proprietary	TCP	Incoming	IVS to Server (Removed in FileWave 15.4+)
20016	SSL	TCP	Incoming	IVS to Server
20017	SSL	TCP	Incoming	IVS to Server: Secure
20022	SSL	TCP	Incoming	Imaging Server to FileWave Server NATS
20443	HTTPS	TCP	Incoming	IVS to Server: Inventory
20444	HTTPS	TCP	Incoming	Server to IVS Client to IVS FileWave Anywhere interface (Appliance only)
20445	HTTPS	TCP	Incoming	IVS to Server: Inventory

‡ TCP/IP & UDP
‡‡ UDP only

Allow External Devices to Connect to the FileWave Server and Boosters

Now that most Organizations are supporting E-Learn and/or remote work environments, you may want to open up your firewall to allow devices outside of your network to connect to the FileWave Server and Boosters.

> If you have multiple Boosters, not all have to be opened up externally. We recommend roughly one Booster for around every 2000 Clients. If you'd like to create additional Boosters and keep your existing Boosters internal, you can find information about setting up a new Booster here: Booster installation

DNS Settings

In order for devices to connect on an external network, the FileWave Server and Boosters will first need to be set up with a Fully Qualified Domain Name (FQDN) that can be resolved outside the network.

> Server and booster hostnames using local domains (.local, .corp etc.) or IP addresses will not be able to allow devices to connect outside of its private network. And, changing the hostname can not be taken lightly because this change on the server will cause Clients and mobile devices to stop communicating with the server. See more: Root Trusted Certificate

You can confirm your internal DNS settings by running the following commands on a computer inside your network.

Replace 'myserver.domain' with your Server/Booster FQDN.

This will use your devices current DNS settings

```
nslookup myserver.domain
```

Then run the command doing an external lookup (on a device outside of the network).

```
nslookup myserver.domain 8.8.8.8
```

If nslookup doesn't resolve to the correct IP (or not at all), you will need to make DNS changes...best to contact your Network Administrator to get this set up.

Inside, then an Outside Example:

```
$ nslookup server.filewave.com
Server:         192.168.1.1
Address:        192.168.1.1#53

Non-authoritative answer:
Name:   preview.filewave.com
Address: 10.1.10.45

$ nslookup server.filewave.com 8.8.8.8
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
Name:   preview.filewave.com
Address: 52.38.32.242
```

See how inside we got a 10.1.10.45 address, and looking outside we got a 52.38.32.242 IP address? This server's FQDN is setup inside and outside properly.

Network Changes

Once DNS is correct, the next thing you'll need to do is configure your firewall to allow external incoming traffic to your Server and Booster(s) on the following ports:

Server: 20017, 20022, 20441, 20443 and 20445

Boosters: 20026 and 20014

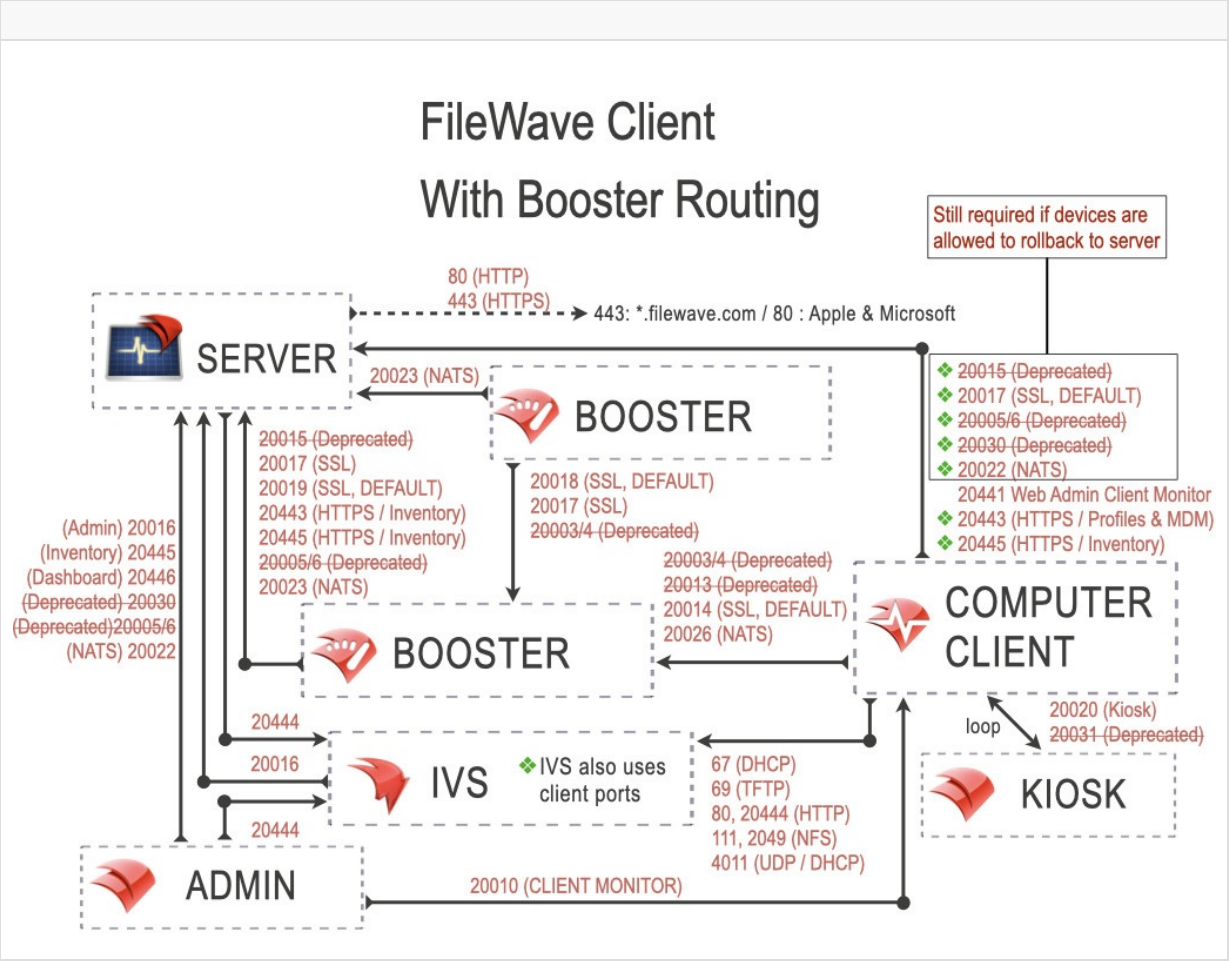


Figure 1.1 - Network with ports

If you'd like to know more about what these ports are used for, you can find that information here: Default TCP and UDP Port Usage

Testing the Network for External Communication

Once the changes are in place on the network, you will want to test the connection to be sure devices can communicate with the server.

First, download our Port Tester on a computer connected to an external network. You can find the Port Tester here: https://supportresources.filewave.com.

Once it's installed, input the Server/Booster's FQDN in the Hostname field. You can toggle the switch next to the port numbers to autofill the ports for the Server or Booster or input them yourself.

Select 'Go' and if all the necessary ports come back successful, you're all set!

Connecting your Clients to the External Boosters

If you don't already have your Clients pointing to the Boosters you selected to make available outside the network, you will need to create a Superprefs Fileset to deploy these changes.

You can learn more about creating and deploying Superprefs here: Creating a Superprefs Fileset

Related Content

- Securing FileWave Server on the Internet for Remote Device Management

Naming conventions and acronyms

What

FileWave Knowledge Base articles will sometimes use acronyms, and occasionally we may miss giving detail or context, but we try to always use the full name once or explain the term. This will be a list of terms you can refer to when unsure.

Alphabetical list of terms:

A

- ABM - Apple Business Manager is a web-based portal designed for businesses and organizations to manage Apple devices and services. It provides tools for device deployment, app distribution, and configuration management. ABM allows businesses to streamline device procurement, deploy custom apps, enforce security policies, and centrally manage Apple devices for improved efficiency and productivity.
- ADE - Apple Device Enrollment is a program that simplifies the deployment and management of Apple devices within organizations. It allows for easy enrollment of devices into a mobile device management solution, automating configuration and saving time in the setup process.
- Admin - When you see FileWave Admin used it is frequently referring to FileWave Central and FileWave Anywhere, but some older documentation may only be referring to FileWave Central.
- Administrator - A user that may log into the FileWave Server via the Admin application.
- Anywhere API (v2) - The second RESTful API that is used by FileWave Anywhere and usually commands are on TCP 443 unless you have changed the port that FileWave Anywhere operates on.
- AppleSeed - AppleSeed is an invitation-only program offered by Apple that allows selected users to test pre-release software and provide feedback to Apple's engineering teams. Participants in the AppleSeed program get early access to upcoming software updates and can help identify and report bugs, ensuring a more stable and polished user experience before the software is released to the general public.
- Archive - To archive a device is to remove it from active monitoring. The device remains in the database with its last reporting information intact; but the device is no longer counted as an active Client. Archiving a Client frees up one Client license.
- ASM - Apple School Manager is a web-based portal provided by Apple for educational institutions to manage Apple devices and services. It allows schools to easily deploy and manage Apple devices, distribute apps and books, create Apple IDs for students and staff, and configure settings for a seamless educational experience.
- Associations - An Association is made between a Fileset and a Client or Client Group and represents the link between the two objects. Time-based attributes can be assigned to the Association. Associations are how distributions are made. You can also make associations between images and clients, licenses and Filesets, and VPP users and devices.
- Attributes - Properties of files that specify how the files are treated once the FileWave Client activates them.
- Azure - Now called "Microsoft Entra". It is a Directory Service from Microsoft.

B

- Booster - A Linux, macOS, or Windows system running software that caches software deployments for macOS and Windows clients.

C

- Clients - A Client represents one computer with the FileWave Client software installed or a mobile device that has been enrolled.
- Client Group - A client Group is a container of like Clients and/or Client Groups.
- Client State - The current condition of a client device as reported to the Admin. The states are: Normal, Missing, Not Tracked, or Archive. A Normal device is fully accessible by the FW and the location is being tracked. A Missing device has been reported as stolen or not where it belongs and tracking is active. Not Tracked means that the device is monitored by FW Admin for all standard characteristics; but location tracking is disabled. An Archived device has been placed into stasis. It is no longer actively monitored by FW; but the last known device settings are available in Inventory.
- Clone - A Clone is an alias of a Client or Client Group that can exist in many Client Groups.
- Cloudv1 - Our original hosted server environment which is running in AWS with Linux based VMs.
- Cloudv2 - Our newer hosted server environment which is running in AWS with Docker/Kubernetes.
- Command Line API (v1) - The initial RESTful API for FileWave. Commands happen on TCP 20443 and 20445.

D

- DEP - Device Enrollment Program (DEP) is a service provided by Apple that enables organizations to streamline the initial setup and configuration of Apple devices. DEP allows devices to be automatically enrolled in a mobile device management (MDM) solution during the activation process, ensuring that they are pre-configured with the necessary settings and policies defined by the organization. DEP simplifies device provisioning, enhances security, and enables seamless device management for organizations deploying a large number of Apple devices.

F

- File - A File in a Fileset represents a file that will be delivered to a specific location for a FileWave Client. Files have attributes and permissions.
- Folder - Files in a Fileset can be organized into Folders (directories). A file is activated in the corresponding folder on the boot volume of the FileWave Client. Folders do NOT have attributes; they only have permissions.

- Fileset - A set of common files and/or folders (directories) meant for delivery to a FileWave Client with a wrapper that contains a detailed listing of the Fileset contents, including permissions and a checksum for each part (to facilitate non-corrupt delivery to clients). In FileWave Central it is called Fileset and in FileWave Anywhere it is called a Payload. The two terms Fileset and Payload are really the same item type.
- FileWave Admin - Now called FileWave Central.
- FileWave Anywhere - The web-based administration tool.
- FileWave Central - The native administration tool that runs on macOS and Windows.
- FileWave WebAdmin - Now called FileWave Anywhere.
- FW - FileWave the company.

G

- Geofencing - A way to define a boundary that a device should not enter or exit. Geofencing is used with Location Tracking for Android devices.
- Geolocation Tracking - Another way of referring to Location Tracking.

H

- Hosted Customer - Any customer where FileWave runs the server itself in either our Cloudv1 or Cloudv2 infrastructure.

I

- IVS - Imaging Virtual Server - A Linux Linux-based used for Windows imaging. The server sits on your network to capture and deploy images.

K

- Kiosk - The self-service portal to the FileWave server for a specified device. The Kiosk contains an Install pane with associated applications and content for that device/Apple ID, and in the case of OS X (macOS)/Windows computers, an Info pane with device configuration information and a Verify button for the user to initiate a request to the Server to verify, update, and repair any associated Filesets.

L

- Location Tracking - Used to locate devices and works with macOS, AppleTV, iOS, iPadOS, Windows, Chromebook, and Android. This is also sometimes referred to as Geolocation or Geolocation Tracking.

M

- Management Mode - In FileWave 11, we added a new client flag (for computer clients). It has two values: Managed (normal mode) and Inventory only. The latter setting allows you to have your client reporting data to FileWave, but will not be affected by any Filesets except for upgrade Filesets. Inventory only does consume a client license.
- Microsoft Entra - Entra (formerly named Azure) is a cloud computing platform and set of services provided by Microsoft. It offers a wide range of cloud-based services, including computing power, storage, networking, analytics, machine learning, and more. Microsoft Entra allows businesses to build, deploy, and manage applications and services using Microsoft-managed data centers spread across the globe. It provides scalability, reliability, and security, making it a popular choice for organizations seeking cloud-based solutions for their infrastructure, applications, and data storage needs.
- Model Update - The command that is issued to the FileWave Server to lock in all changes that have been made by an Administrator. During a model update, all modified Filesets are updated on the server, the Server model is incremented, and the automatic backup process stores the previous model. Filesets are activated based on their scheduled attributes the next time the Client checks in with the Server.

O

- On-Premise - The term used to refer to running a FileWave Server on your own network.

P

- Payload - A term for the item in FileWave that is a bundle of content that can contain files, policies, and profiles. In FileWave Central it is called Fileset and in FileWave Anywhere it is called a Payload. The two terms Fileset and Payload are really the same item type.
- Permissions - Properties of files and folders that specify the access rights of the files and folders. Permissions are set when the FileWave Client activates the files and folders in a Fileset. Self-healing also sets permissions during the verification phase.

R

- RESTful API - Another name for the Command Line API (v1) and you may hear it referred to as either name because originally there was a single API and it was RESTful. Now there are 2 different APIs and both are RESTful so it makes more sense to not call it the RESTful API.

T

- Time Attribute - A Time Attribute is a property of an Association that specifies the Time a FileWave Client executes an action.

V

- VPP - Volume Purchase Program (VPP) is an Apple program that allows businesses and educational institutions to purchase apps and books in bulk for distribution to their employees or students. It provides a centralized platform for organizations to buy, manage, and distribute apps, simplifying the process of app procurement and licensing for multiple devices.

W

- winget - A command-line tool and package manager for Windows operating systems. It allows users to discover, install, update, and uninstall software applications from the command line or through scripts. With Winget, users can search for available software packages from the Microsoft Store as well as third-party repositories. It simplifies the software management process by providing a centralized tool for handling installations and updates, making it easier to maintain a collection of software applications on Windows systems.

Scripting

Scripting

Execute macOS scripts as Console User

Description

By default, the FileWave Client executes scripts and tasks with elevated permissions (root on macOS). This addition to the start of a script will execute it as the currently logged-in user (also known as the console user).

Ingredients

- Text editor
- FileWave Central

Directions

1. Insert this near the start of your script, before the desired actions.

   ```
   #!/bin/zsh
   current_user=$(stat -f%Su /dev/console)
   current_user_id=$(id -u $current_user)
   ```

   Example:

   ```
   #!/bin/zsh
   current_user=$(stat -f%Su /dev/console)
   current_user_id=$(id -u $current_user)

   launchctl asuser $current_user_id sudo -u $current_user whoami
   echo "hello world" >> /Users/$current_user/Desktop/test.txt
   ```

macOS 10.15+ and zsh shell for scripting

Description

Apple announced changes to the default shell for macOS 10.15:

https://support.apple.com/HT208050

Information

For years now, Apple has appeared to avoid any tools that are covered by the GPL v3 Licence as well as remove any that were in use over time. Bash is one of these. In the early releases of 10.x Apple initially used tcsh shell as default, but soon moved to bash; so this is not the first time Apple has made a change of this kind.

In order to avoid newer versions of bash covered by GPL v3 licensing, it can be seen how old bash is on a macOS device:

```
$ bash -version
GNU bash, version 3.2.57(1)-release (x86_64-apple-darwin18)
Copyright (C) 2007 Free Software Foundation, Inc.
```

Run the same command on a modern Linux system, e.g. the FileWave Linux Server Appliance and you will see a much newer version:

```
$ bash -version
GNU bash, version 4.2.46(2)-release (x86_64-redhat-linux-gnu)
Copyright (C) 2011 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>

This is free software; you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
```

GPL v3 license has implications for Apple. zsh is licensed under MIT, which does not involve these same implications.

Considerations

Specifying the shell

The first line of a script should indicate which shell is used when a script runs.

For example, for bash this could typically be either of the following (this is known as the 'shebang'):

```
#!/bin/bash
```

```
#!/usr/bin/env bash
```

By providing this, the script will run from a shell of the specified type. Any scripts created should have this set. If this is not set, then the script will run in the shell type that is currently set for that shell's session. Apple's changes will have an impact on any scripts not specified by default. The best practice is to always add the shebang at the beginning of any script to ensure expected behavior.

Bash vs zsh

Bash and zsh are very similar, but there are some differences that may interfere with scripts that were written for bash but are then run as zsh. Scripts should therefore be analyzed for behavior if the shell type of the script is changed.

Examples

Arrays

The first item of an array differs when being referenced:

- bash reference of the first item in an array: index 0
- zsh reference of the first item in an array: index 1

There is also a difference in deleting items from an array. The following produces the same output from the same original array, but note differences on the item being indexed and the method of removal:

bash arrays

```
#!/bin/bash


myarray=("one" "two" "three")
echo ${myarray[@]}
echo "First item in array: "${myarray[0]}

echo "Remove first item in array, item 0..."
unset myarray[0]
echo ${myarray[@]}

exit 0

# Script output:
one two three
First item in array: one
Remove first item in array, item 0...
two three
```

zsh arrays

```
#!/bin/zsh

myarray=("one" "two" "three")
echo ${myarray[@]}
echo "First item in array: "${myarray[1]}

echo "Remove first item in array, item 1..."
myarray[1]=()
echo ${myarray[@]}

exit 0

# Script output:
one two three
First item in array: one
Remove first item in array, item 1...
two three
```

Variable Expansion

Word splitting on variable expansion differs between bash and zsh. For example, bash will print the following, one line per word, whilst zsh will print the whole variable as one line. Note also, that bash, by default, will not expand aliases when the shell is not interactive, unlike zsh.

bash variable expansion

```
#!/bin/bash

# Expand aliases
shopt -s expand_aliases

alias printvar="printf '%s\n'"
myvar='one two'
printvar $myvar

exit 0

# Script output:
one
two
```

zsh variable expansion

```
#!/bin/zsh

alias printvar="printf '%s\n'"
```

```
myvar='one two'
printvar $myvar

exit 0

# Script output
one two
```

zsh does have the ability to set an option to change this behavior, but consider converting to an array instead.

Apple has chosen zsh over bash since the overlap on script compatibility is high. However, as seen there can be differences and so it is prudent to check all scripts for behavior. The above are just examples; other differences may be experienced and will require addressing appropriately.

Referencing Launch Arguments in Scripts

Description

Scripts ran through FileWave have the option to supply 'Launch Arguments'. These are referenced from the script but are not included in the body of the script.

They may be supplied to any of the following:

- Fileset Scripts
- Custom Fields
- Policy Blocker Scripts

Often Admins feel that there is a limit of 9 'Launch Arguments' through FileWave, but the below will demonstrate this is not the case.

Information

The Launch Arguments, known as Positional Parameters, are referenced as follows:

	macOS/Linux	Windows Powershell	Windows Bat
First Argument	$1	`$args[0]`	%1
Second Argument	$2	`$args[1]`	%2
Third Argument	$3	`$args[2]`	%3

More may be added and each is referenced in turn by its positional place as in the above table.

Considerations

Certain shell types behave differently. This may particularly show when referencing the 10th or higher supplied argument.

> Although two solutions have been supplied, zsh is recommended to stay in line with Apple's policy: https://support.apple.com/en-gb/HT208050

bash and sh

The following example was hoped to print the first 3 positional parameters, followed by the 10th and 11th.

bash_test.sh

```
#!/bin/bash

echo $1
echo $2
echo $3
echo $10
echo $11

exit 0
```

However, if the character '1' was supplied as single argument the following would be observed when ran:

bash_test.sh

```
./bash_test.sh 1
1


10
11
```

The bash and sh shells are examples which treat $10 as ${1}0, $11 as ${1}1, etc.; returning the value of $1 and then appending the additional character (0 or 1 in this example).

Additional reference:

- https://www.oreilly.com/library/view/bash-cookbook/0596526784/ch05s07.html
- https://www.oreilly.com/library/view/bash-cookbook/0596526784/ch05s04.html

As such the above output is equivalent to:

	Description	Output
$1	Returns first argument	1
$2	Returns second argument	2
$3	Returns nothing, no third argument	
$10	Returns first argument, followed by the '0' character	10
$11	Returns first argument, followed by the '1' character	11

To ensure the correct positional parameters are referenced, the variables must be explicitly set to achieve the correct output.

The following shows the corrected script.

bash_test.sh

```
#!/bin/bash

echo $1
echo $2
echo $3
echo ${10}
echo ${11}

exit 0
```

Now when executed with 11 positional parameters, the expected output is displayed:

bash_test.sh

```
./bash_test.sh var1a var2b var3c var4d var5e var6f var7g var8h var9i var10j var11k
var1a
var2b
var3c
var10j
var11k
```

zsh

Not all shells act the same. An alternate to the above would be zsh. Taking the above example as a zsh script:

zsh_test.sh

```
#!/bin/zsh

echo $1
echo $2
echo $3
echo $10
echo ${11}

exit 0
```

Given the same 11 arguments, this would output:

zsh_test.sh

```
./zsh_test.sh var1a var2b var3c var4d var5e var6f var7g var8h var9i var10j var11k
var1a
```

```
var2b
var3c
var10j
var11k
```

Unlike bash and sh, zsh considers $10 to be the 10th argument, $11 the 11th argument and so on and so forth. Notice zsh may use either format.

Conclusion

Often admins are confused with output results when using 10 or more arguments. Armed with the above knowledge, this should assist structuring scripts.

Related Content

- Custom Fields
- Filesets / Payloads

Script Best Practices

Description

Tips and tricks for running Filesets with scripts

Don't put passwords in scripts

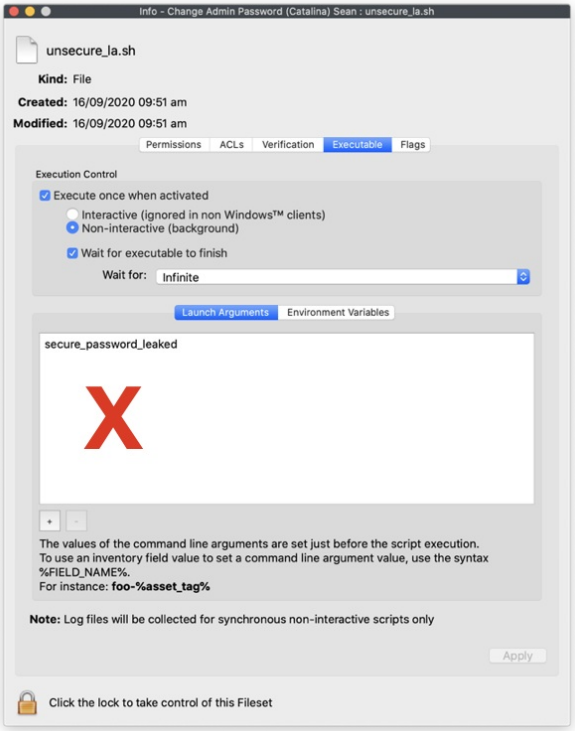
The scripts are stored locally on devices. For security reasons, usernames and passwords should not be included within the body of scripts.

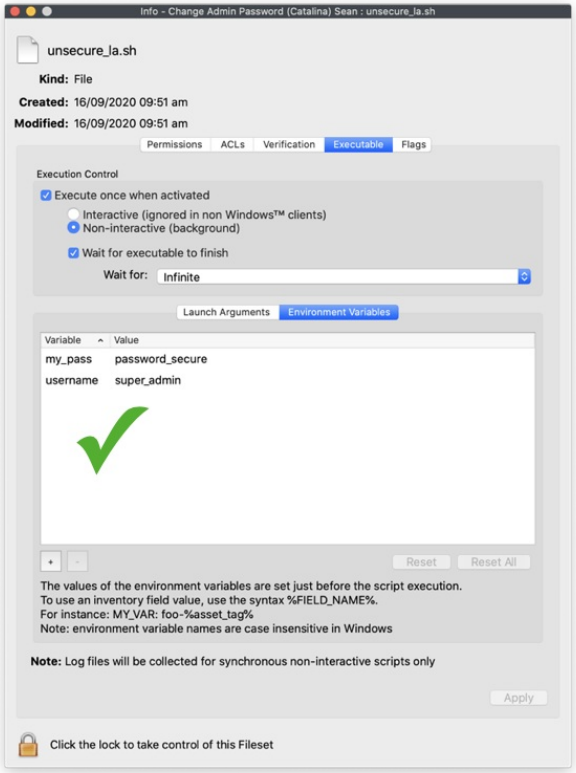
For example:

Example: password in command

```
somecommand -u "USERNAME_HERE" -p "PASSSWORD_HERE"
```

Additionally, DO NOT use Launch Arguments to provide passwords to scripts. Launch Arguments are visible in the process list during script execution. Instead supply the username and password as Environment Variables:





During script execution, the Launch Argument is seen:

Example: Visible Password

```
$ ps -ef | grep secure
    0 73010   155   0  9:51am ??         0:00.01 /bin/zsh /var/scripts/532417/unsecure_la.sh
secure_password_leaked
```

Using the example Environment Variables from the image, they would be addressed as:

OS	Script Type	Command
macOS	shell	`somecommand -u $username -p $my_pass`
Windows	Powershell	`somecommand -u $Env:username -p $Env:my_pass`
	Batch	`somecommand -u %username% -p %my_pass%`

macOS & Windows	Python	`import os`
		
		`os.getenv('username')`
		`os.getenv('my_pass')`

Keep Requirements Scripts Small

Requirements scripts are pulled from a fileset and sent before the remainder of the fileset.

It behaves this way because if a requirements script fails, there is no point in downloading and installing the remainder of the fileset.

> ⚠ A fileset whose requirements have failed will not even show up in the kiosk.

Where possible, avoid piping commands. This increases overhead on the scripts. If pipes are required, try to reduce the quantity of pipes. If nothing else, this makes the scripts easier to read.

```
$ time system_profiler SPHardwareDataType | grep "Model Identifier" | awk '{print $NF}'
MacBookPro11,4


$ system_profiler SPHardwareDataType | awk '/Model Identifier/ {print $NF}'
MacBookPro11,4
```

And other commands may achieve the same result more efficiently without the need to pipe.

```
$ time system_profiler SPHardwareDataType | grep "Model Identifier" | awk '{print $NF}'
MacBookPro11,4

real    0m0.192s
user    0m0.071s
sys     0m0.049s


$ time sysctl -n hw.model
MacBookPro11,4

real    0m0.004s
user    0m0.001s
sys     0m0.002s
```

Consider this for all scripts beyond just requirement scripts.

Leverage Dependancies and Scripts

If there is a script that several filesets will need, don't paste the same script into each one. Create an empty fileset with that script, and make the other filesets dependent upon it.

Log Script Output to the Client Log (macOS only)

Have a script that needs to write details steps to a log?

Want a quick status of the script.

Have a script write to the client log.

macOS / Linux

```
#!/bin/bash
exec 1>>/var/log/fwcld.log
exec 2>>/var/log/fwcld.log


... rest of script
```

You can then use Client Monitor, and pull and view the log as things are happening.

> Windows log file is locked such that additional appending may not take place

Testing Scripts

Scripts run by FileWave are run by root or System. As such, scripts should be tested using the same user context to prevent erroneous results. Many commands will yield the same result regardless, but this cannot be relied upon.

Windows

E.g. Run the following on a Windows 10 Professional system locally through Powershell as either user or 'Run As Admin' will see the following result:

```
 (Get-ItemProperty "HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion").EditionId
Professional
```

However, as a Custom Field running the same script, the result is surprisingly different:

```
 (Get-ItemProperty "HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion").EditionId
Enterprise
```

This is because Windows is providing a different answer based upon either the user running the script or may provide different responses based on 32-bit or 64-bit.

Take a look at Getting a CMD prompt as SYSTEM in Windows Vista and Windows Server 2008 for details about running scripts as System. Note, that by default, this will start an executable as 64-bit, for native 64-bit OS. However, the above example is because the FileWave fwcld process is calling the 32-bit version of PowerShell.

> PsTools: This relies on downloading and installing, onto the test machine, PsTools.

To mimic this experience, consider the guide to starting the CMD. When launching an executable, like PowerShell, the 32-bit version would need to be referenced. For example:

```
PSEXEC  -i -s -d C:\Windows\SysWOW64\windowsPowerShell\v1.0\powershell.exe
```

For CMD, the equivalent would be:

```
PSEXEC -i -s -d %windir%\SysWoW64\cmd.exe
```

Similarly, when attempting to run some commands, it may be necessary to ensure Windows is using the correct version of a binary with the 'sysnative' redirect. An example would be bitlocker's 'manage-bde.exe'. To use this in a Fileset, try the following:

```
C:\Windows\sysnative\manage-bde.exe -status
```

If you have a requirement to run a particular command through the 64 bit version of Powershell this can be achieved as follows:

```
If ( [IntPtr]::Size * 8 -ne 64 )
{
    C:\Windows\SysNative\WindowsPowerShell\v1.0\PowerShell.exe -File $MyInvocation.MyCommand.Path
}
Else
{
    # Add code here
}
```

Example

Create a new admin account

Two Fileset Environment Varaibles would be supplied. To add the user 'rstephens' with the password 'filewave'

Variable	Value
username	rstephens
password	filewave

Parameters may be supplied, that can then be added to the execution of Powershell from within the script:

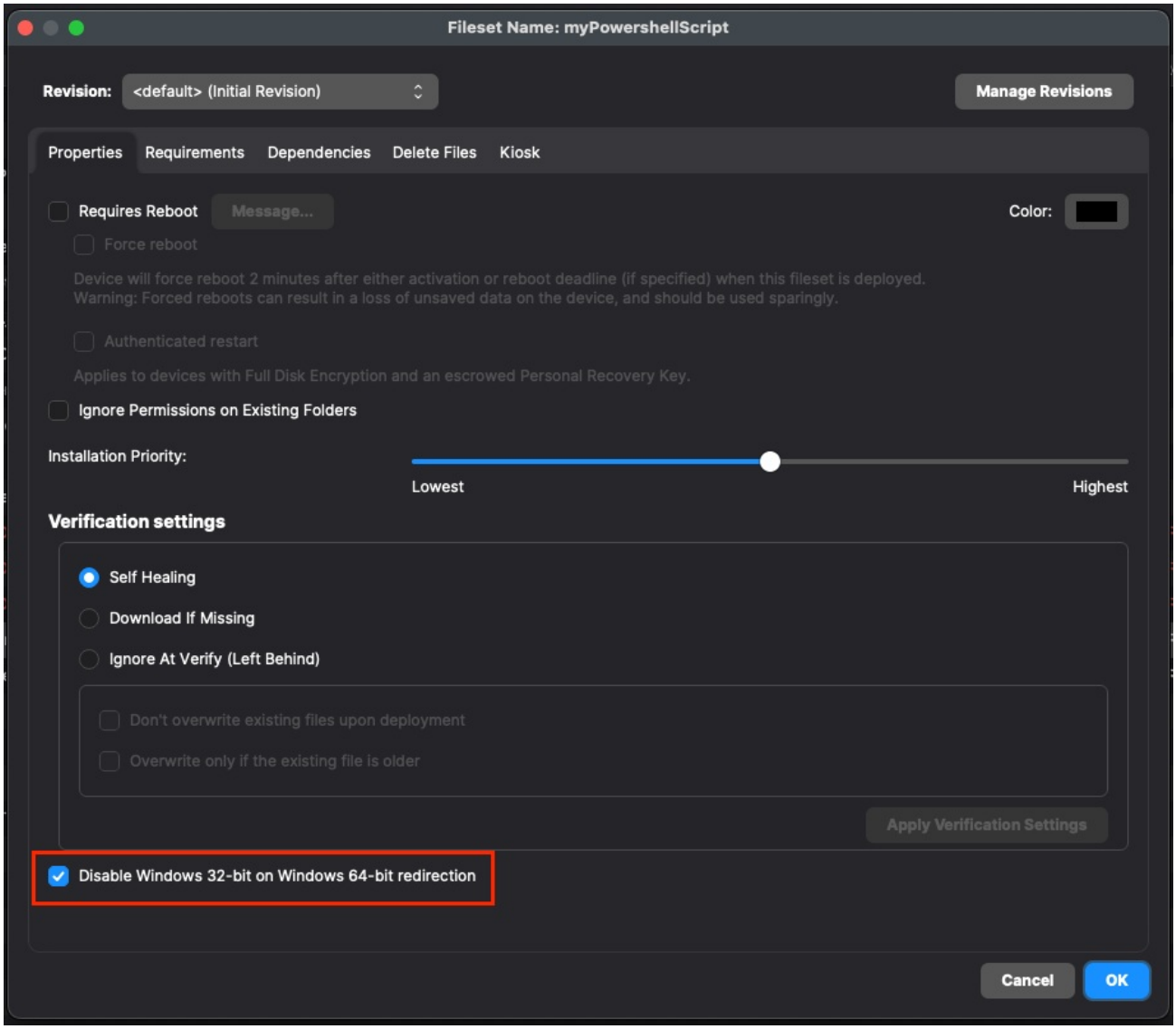
```
Param (
    [string]$MyUsername = $Env:username,
    [string]$MyPassword = $Env:filewave
)


If ( [IntPtr]::Size * 8 -ne 64 )
{
    C:\Windows\SysNative\WindowsPowerShell\v1.0\PowerShell.exe -File $MyInvocation.MyCommand.Path -MyUsername
$MyUsername -MyPassword $MyPassword
}
Else
{
```

```
    (New-LocalUser -AccountNeverExpires:$true -Password ( ConvertTo-SecureString -AsPlainText -Force $MyPassword)
-Name $MyUsername | Add-LocalGroupMember -Group administrators)
}
```

Troubleshooting PowerShell scripts

As a best practice, always check the "Disable Windows 32-bit on Windows 64-bit redirection" checkbox in the Properties tab for your fileset. This ensures that any scripts in the fileset will be run in a 64-bit session and built-in Windows executables triggered by those scripts will call the 64-bit versions. A common reason for why your script might not be performing the expected results could be due to 64-bit Only Modules or Cmdlets: Some PowerShell modules or cmdlets are available only for the 64-bit version of PowerShell. If a script relies on these 64-bit modules, it must run in a 64-bit shell.



macOS

On macOS, running commands as sudo is not necessarily the same as actually becoming root.

Root vs As Root

E.g. Run the following commands to evaluate the local variable $HOME, once using sudo and once as root.

```
$ whoami
auser
$ sudo echo $HOME
/Users/auser
$ sudo su -
$ whoami
root
$ echo $HOME
/var/root
```

Paths

Similarly, the paths used to locate executable files will differ, since FileWave is a service ran as root and is not the root account. On an example device:

User Account	Root Account	FileWave Client
% echo $PATH | tr ":" "\n" /opt/homebrew/bin /opt/homebrew/sbin /var/root/.cask/bin /usr/local/sbin /usr/bin /bin /usr/sbin /sbin	% echo $PATH | tr ":" "\n" /usr/local/bin /System/Cryptexes/App/usr/bin /usr/bin /bin /usr/sbin /sbin /Applications/VMware Fusion.app/Contents/Public /Library/Apple/usr/bin /var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/local/bin /var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/bin /var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/appleinternal/bin	/usr/bin /bin /usr/sbin /sbin

As such, consider always using the full path within a script to an executable, to be explicit, and ensure the executable is found.

For example, it can be seen from the above that homebrew is installed.

```
 % ls -al /usr/local/bin/brew
lrwxrwxrwx  1 root  _developer  28 Mar 23  2023 /usr/local/bin/brew -> /usr/local/homebrew/bin/brew
```

Running the following command would work as the user or root account, but would fail through FileWave, since the FileWave Client does not search /usr/local at all for executables:

`brew -v`

To ensure the script works and targets the correct brew, the full path should be entered:

`/usr/local/bin/brew -v`

Plist

It is common to see plist files edited with the 'defaults' command. However this command is unique when it comes to ownership and permissions of files. The 'defaults' command will both take ownership and change permissions of files when used to write to plist files:

```
$ whoami
root
$ ls -al /tmp/example_plist.plist
-rw-r--r--  1 rstephens  staff  66 Feb 28 10:03 /tmp/example_plist.plist
$ defaults write /tmp/example_plist Label example_plist
$ ls -al /tmp/example_plist.plist
-rw-------  1 root  wheel  66 Feb 28 10:05 /tmp/example_plist.plist
```

As such, ensure to add a repair to scripts to reset permissions and ownership after the command has been used or consider using the following command instead (Note the full path is required if /usr/libexec is not in the paths list:

```
/usr/libexec/PlistBuddy
```

Related Content

- Custom Fields
- Filesets / Payloads
- Fileset / Payload Script Exit Code Status

Scripting Languages supported in FileWave

Description

FileWave provides the ability to leverage certain scripting languages on macOS and Windows. This includes:

	macOS	Windows
Perl	✅	✅
Python	✅	✅
Shell	✅	
Bat		✅
PowerShell		✅

FileWave does not include these languages, they are either installed by the OS vendor or will need to be installed separately.

> Apple indicated they would be deprecating pre-installation of certain runtimes, for example: https://developer.apple.com/documentation/macos-release-notes/macos-catalina-10_15-release-notes

When testing any scripts locally prior to deployment through FileWave, it is imperative that they are tested in the same context/environment as if they were being ran by FileWave, as indicated in our Script Best Practices KB.

The below provides examples of how to instal Python on endpoints. This is one way to achieve this goal, but by no means the only way. Professional Services requests can be raised for items beyond standard FileWave Support if desired.

Ingredients

- FileWave Central
- Windows EXE

Installers available from either:

- https://www.python.org/ftp/python/
- https://www.python.org/downloads/windows/

Directions

macOS Python Installer

▼ macOS Python

Apple used to provide a version of Python pre-installed, however as noted, this was an old version and is now deprecated. It is possible to download Python installers, however Apple provide Python in the Xcode Command Line Tools. The following method demonstrates how the softwareupdate mechanism may be used to list (and therefore instal) the command line tools:

```
# touch /tmp/.com.apple.dt.CommandLineTools.installondemand.in-progress
# softwareupdate -l
Software Update Tool

Finding available software
Software Update found the following new or updated software:
* Label: Command Line Tools for Xcode-12.4
    Title: Command Line Tools for Xcode, Version: 12.4, Size: 440392K, Recommended: YES,
* Label: Command Line Tools for Xcode-13.2
    Title: Command Line Tools for Xcode, Version: 13.2, Size: 577329K, Recommended: YES,
* Label: Command Line Tools for Xcode-12.5
    Title: Command Line Tools for Xcode, Version: 12.5, Size: 470966K, Recommended: YES,
* Label: Command Line Tools for Xcode-12.5
    Title: Command Line Tools for Xcode, Version: 12.5, Size: 470820K, Recommended: YES,
* Label: macOS Big Sur 11.6.2-20G314
```

Note that more than one version may be returned. If Xcode is installed a version may already be in place, so care should be taken if Xcode is already installed. After installing the chosen version, the temporary file created may then be removed.

Instal Python Packages

Python scripts import packages, not all of which will be installed by default. Any additional packages will require installation. PIP may achieve this and may be used in a Fileset script; PIP is also installed when installing the Command Line tools (or Xcode)

It is possible to check pip with the following command, note it may require upgrading, but again take careful consideration if Xcode is installed:

```
# pip3 list
Package    Version
---------- -------
pip        20.2.3
setuptools 49.2.1
six        1.15.0
wheel      0.33.1
WARNING: You are using pip version 20.2.3; however, version 23.1.2 is available.
You should consider upgrading via the '/Library/Developer/CommandLineTools/usr/bin/python3 -m pip install --
upgrade pip' command.
```

For example, to instal pyobjc:

```
# Instal pyobjc
pip3 install pyobjc-core
pip3 install pyobjc-framework-Cocoa
pip3 install pyobjc-framework-Quartz
pip3 install pyobjc-framework-SystemConfiguration
```

> ⚠ In some instances, pip3 may fail. This can be due to the link created in:
> /Library/Developer/CommandLineTools/SDKs
> Remove and recreate the link to the correct installed version if need be.

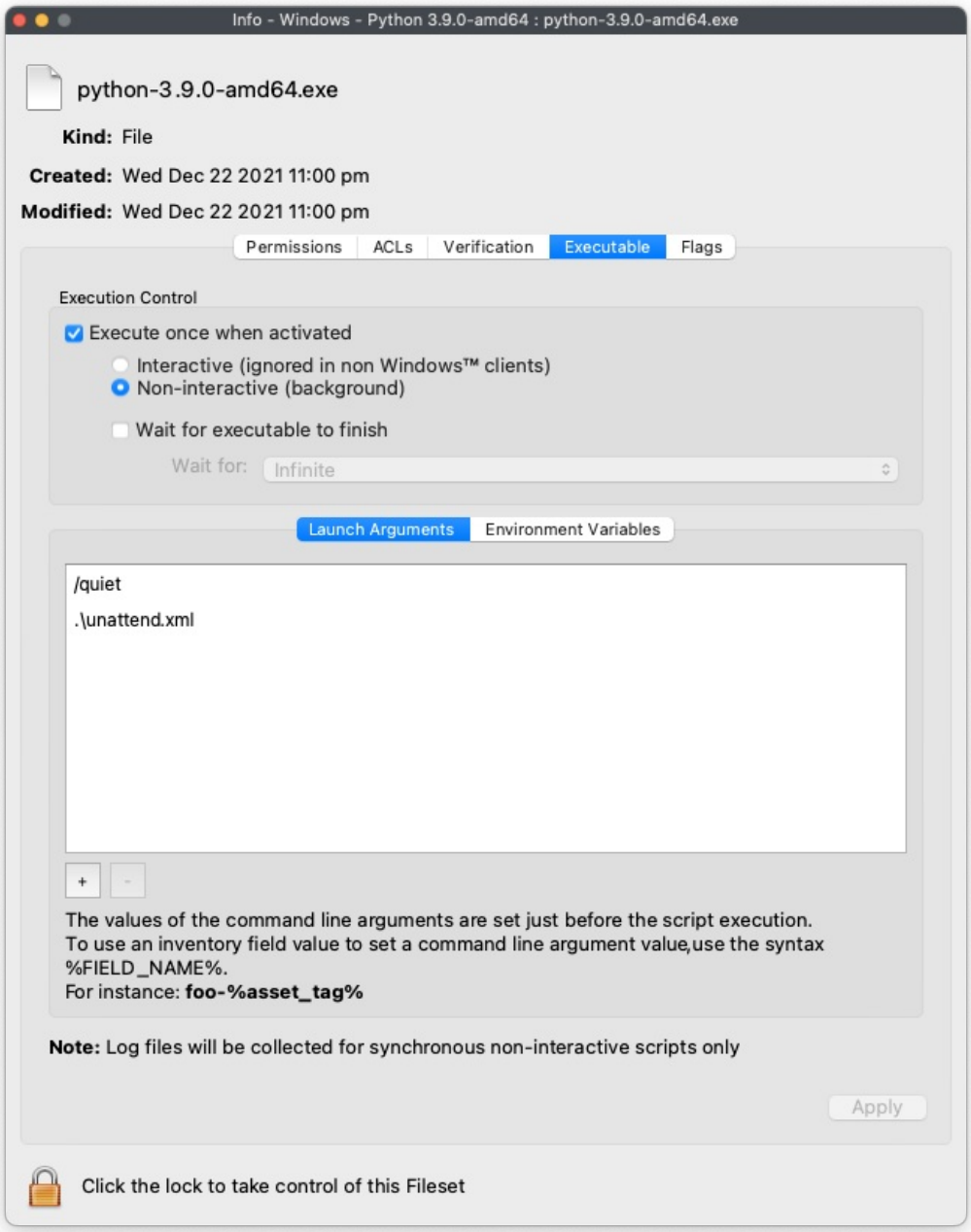
The above could be packaged into a single script which could:

- Touch the temporary file
- Run Software Update
- Obtain the desired version from the list
- Instal that version
- Remove the temporary file
- Remove the link if incorrect
- Use PIP to instal other desired packages
- Recreate the link if removed above

Windows Python Installer

▼ Windows Python

1. Download the appropriate python installer EXE
2. Create a new empty Fileset
3. Choose a location to store the installer and drag the EXE into this location
4. Highlight this EXE and from the Get Info window choose Executable
5. Tick the option to 'Execute once when activated'
6. Set two Launch Arguments as per the screen shot: /quiet and .\unattend.xml



Supporting File

An unattend.xml file may be used to pre-determine aspects of the installation, e.g. instal PIP during installation or set Paths

1. Use an editor to create a file called unattend.xml
2. Edit this file with the below details, editing to meet desired needs
3. Place this file in the Fileset within the same folder location as the EXE installer

An example could look as follows (see the Python documentation for a list and description of available options):

```
<Options>
<Option Name="InstallAllUsers" Value="1" />
<Option Name="TargetDir" Value="C:\Program Files\Python39" />
<Option Name="DefaultAllUsersTargetDir" Value="C:\Program Files\Python39" />
<Option Name="DefaultJustForMeTargetDir" Value="C:\Program Files\Python39" />
<Option Name="DefaultCustomTargetDir" Value="C:\Program Files\Python39" />
<Option Name="AssociateFiles" Value="1" />
<Option Name="CompileAll" Value="1" />
<Option Name="PrependPath" Value="1" />
<Option Name="Shortcuts" Value="1" />
<Option Name="Include_doc" Value="1" />
<Option Name="Include_debug" Value="1" />
<Option Name="Include_dev" Value="1" />
<Option Name="Include_exe" Value="1" />
```

```
<Option Name="Include_launcher" Value="1" />
<Option Name="InstallLauncherAllUsers" Value="1" />
<Option Name="Include_lib" Value="1" />
<Option Name="Include_pip" Value="1" />
<Option Name="Include_symbols" Value="1" />
<Option Name="Include_tcltk" Value="1" />
<Option Name="Include_test" Value="1" />
<Option Name="Include_tools" Value="1" />
<Option Name="LauncherOnly" Value="0" />
<Option Name="SimpleInstall" Value="0" />
<Option Name="SimpleInstallDescription"></Option>
</Options>
```

Edit the above example as desired, for example, set the target directory paths as desired.

Instal Python Packages

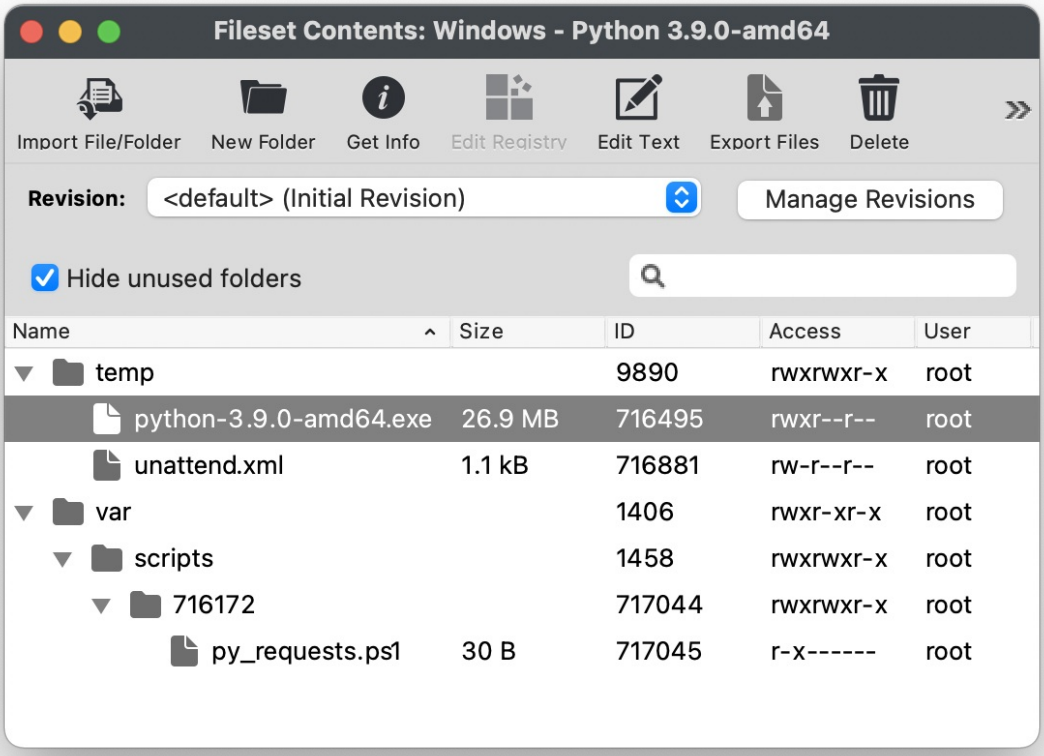
Python scripts import packages, not all of which will be installed by default. Any additional packages will require installation. PIP may achieve this and may be used in a post flight script, if the option above to include pip at installation was configured.

1. Create a PostFlight script within the Fileset
2. Edit the script to include any desired packages, e.g. to instal the 'requests' package:

```
python -m pip install requests
```

> The command 'pip list' may be used to list ALL additional packages installed.
> The command 'pip freeze' may be used to list packages installed by PIP.

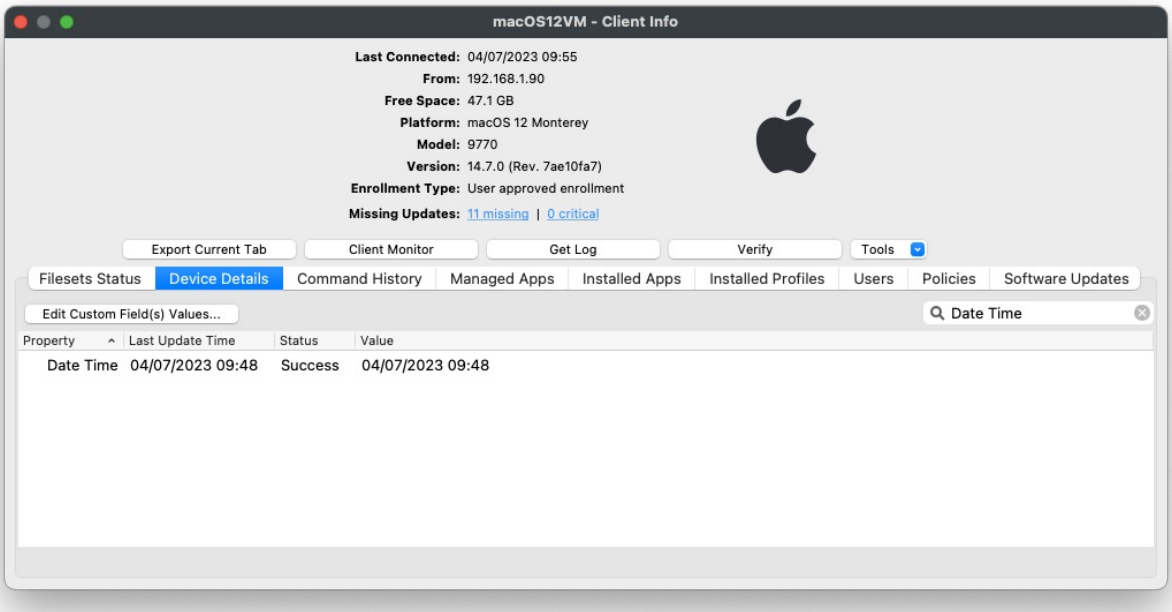
The final Fileset will may look something like:



> Packages, including PIP itself, may require updating over time.

Testing

The following download is a simple Python Custom Field for both macOS and Windows. It should report the date when ran.



↓ macOS/Windows

If desired, use the Custom Field Assistant window to Import this unzipped, Custom Field example, called 'Date Time'

Using PsExec to Test PowerShell 32bit Scripts on Windows with FileWave

What

FileWave is a Unified Endpoint Management tool that allows organizations to manage and deploy software and settings to macOS, Windows, iOS, iPadOS, tvOS, Chrome, and Android devices. When deploying PowerShell scripts through FileWave, it is important to test the scripts on a Windows device beforehand to ensure that they will function correctly when run through FileWave.

When/Why

FileWave runs all PowerShell scripts in 32bit PowerShell and runs them as SYSTEM. By testing the scripts on a Windows device using PsExec and executing the .ps1 script as SYSTEM with 32bit PowerShell, organizations can gain a better understanding of how the script will function when run through FileWave. This can help prevent potential issues and ensure that the scripts function as intended when deployed to devices.

How

1. Download PsExec from the Sysinternals Suite (https://docs.microsoft.com/en-us/sysinternals/downloads/psexec)
2. Open a Command Prompt or PowerShell window with Administrator privileges
3. Navigate to the folder where PsExec is located
4. Use the following command to run the PowerShell script as SYSTEM with 32bit PowerShell: `pPSEXEC -i -s -d C:\Windows\SysWOW64\windowsPowerShell\v1.0\powershell.exe -file "path\to\script.ps1"`

Example:

```
PSEXEC  -i -s -d C:\Windows\SysWOW64\windowsPowerShell\v1.0\powershell.exe -file "C:\test\testscript.ps1"
```

Related Content

- PsExec - Sysinternals | Microsoft Learn
- Script Best Practices

Digging Deeper

PsExec Switches

- `-i` : runs the program in the interactive mode (the user is prompted to confirm the action)
- `-s` : runs the program as the SYSTEM account
- `-d` : runs the program as a background process, without interaction
- `-accepteula` : automatically accepts the PsExec license agreement

Using 64bit PowerShell in a 32bit PowerShell script

Sometimes, certain tasks cannot be accomplished using 32bit PowerShell. To overcome this limitation, the following code can be used to run a 64bit PowerShell script within a 32bit PowerShell script.

```
If ( [IntPtr]::Size * 8 -ne 64 )
{
    C:\Windows\SysNative\WindowsPowerShell\v1.0\PowerShell.exe -File $MyInvocation.MyCommand.Path
}
Else
{
    # Add code here
}
```

This code block checks whether the processor architecture is 64-bit or not and runs the script in 64-bit mode and the main script can be run here. It is important to note that running the script in 64-bit mode should only be done when it is not possible to accomplish a task using 32bit PowerShell.

It is always recommended to test the script in a test environment to avoid any misconfigurations, and to make sure it is working as

expected.

Technical Support Tools

About

These are the FileWave Automated tools. We provide these scripts and automation tools to our customers to help them in their systems.

Tools Selection

Port Tester

The Port Tester (PT) is a utility to help verify that all the needed firewall ports are open. This tests both the server being able to get out and devices getting in. See the Default TCP and UDP Port Usage for the ports needed.

Windows - https://supportresources.filewave.com/downloads/Porttester_Win.zip
macOS - https://supportresources.filewave.com/downloads/Porttester_Mac.zip

Team Viewer

The Team Viewer (TV) Quick Support agent is a customized Team Viewer client. Download this client, launch it, and let support know the ID and Password it provides to allow them to remotely connect to your server and assist you.

Windows - https://supportresources.filewave.com/downloads/TeamViewerQS.exe
macOS - https://supportresources.filewave.com/downloads/TeamViewerQS.app.zip

Debug Uploader Tool

The Debug Uploader Tool (UT) is a utility that runs on the server that will collect the debug information and securely upload it to support.

Debug Uploader Tool - ut.filewave.com

What languages does FileWave support?

What

FileWave is a Unified Endpoint Management tool that allows users to manage and deploy updates to macOS, Windows, iOS, iPadOS, tvOS, Chrome, and Android devices. It is used by a variety of organizations, including educational institutions, corporations, and state and local governments around the world.

When/Why

It is important to know the languages that FileWave is natively in because it can help users understand the tool and its features more effectively. For example, if a user is more comfortable with a certain language, they may prefer to use the FileWave interface in that language. Additionally, knowing the native languages of FileWave can help organizations decide whether the tool is a good fit for their needs based on the language preferences of their staff or users.

How

FileWave is natively in:

- English
- French
- German
- Korean
- Japanese
- Traditional and Simplified Chinese.

Troubleshooting

Apple MDM Troubleshooting

This Knowledge base article will help you troubleshoot mdm with FileWave.

Before going deep into troubleshooting, make sure that you have got these steps correct:

1. Your FileWave server should have a fully qualified DNS name (this dns name is the one entered in the Admin Preferences->Mobile)
2. If for some reason you changed the Server DNS Name in Admin Preferences->Mobile, did you re-generate the certificate? If you did, then you have to trust the new certificate from the enrollment page (https://dns:20443/ios)
3. If the APN cert upload fails from Admin Preferences, make sure you followed the exact steps from step 1, as this can be caused of password-protected certificate
4. If all of the above are set and still have problems, you need to create an admin user account for debugging django:
   a. go to the FileWave server and type this command: "sudo fwcontrol mdm addadminuser" and follow the instructions
5. Another important log file is "/usr/local/filewave/log/filewave_django.log"
6. Make sure that your FileWave Admin displays "iOS/MDM Service OK" in the left lower corner in order to be able to manage your devices.

The following are some of the problems encountered before:

Enrolment Error (FileWave MDM Configuration is invalid):

The profile "Filewave MDM Configuration" is invalid. The MDM payload
"Mobile Device Management" contains an invalid topic

This is usually solved by re-generating the APN certificates because you have not generated them correctly.

CONNECTION PROBLEMS**:**

There are cases where ios devices fail to enroll and you get an error similar to this from sentry:

error
(61, 'Connection refused')
Request Method: PUT
Request URL: https://sscfilewave.co.sbmc:20443/ios/mdm_checkin
Exception Type: error
Exception Value:
(61, 'Connection refused')
Exception Location: /usr/local/filewave/python/lib/python2.7/socket.py in meth, line 222

This error is associated with a port "2195" being closed, you can verify by :

telnet gateway.push.apple.com 2195
Trying 17.172.239.89...
telnet: connect to address 17.172.239.89: Connection refused

the issue will be solved if the IT Admin opens port 2195 for FileWave.

5223 : IOS to apn server port:

port 5223 should be open for IOS clients to reach out to the APN server and receive push notifications.

For a list of all ports used, check this man-

Backup Procedures for FileWave Hosted Servers

What

This article details the backup procedures and policies for FileWave Hosted Servers. Understanding how and what data is backed up is essential for effectively managing and safeguarding your organization's devices and information.

When/Why

Backups are automatically performed daily for all FileWave Hosted Servers. This routine is crucial for disaster recovery, maintaining data integrity, and ensuring minimal downtime in unexpected data loss situations. The retention period for these backups is 30 days, ensuring a sufficient window for recovery if needed.

How

Backups are executed daily and stored securely in highly available AWS S3 buckets. The following paths are included in the backups, ensuring comprehensive coverage of both configuration and operational data:

- /usr/local/filewave/fwxserver/DB
- /private/var/log
- /usr/local/filewave/fwxserver/Data Folder
- /usr/local/filewave/instrumentation_data
- /usr/local/filewave/apache/conf
- /usr/local/filewave/apache/logs
- /usr/local/filewave/apache/passwd
- /usr/local/filewave/django/filewave
- /usr/local/filewave/conf
- /usr/local/filewave/certs
- /usr/local/filewave/fwcld
- /usr/local/filewave/ipa
- /usr/local/filewave/log
- /usr/local/filewave/media
- /usr/local/filewave/tmp
- /var/log
- /usr/local/filewave/scripts
- /usr/local/etc
- /tmp
- /usr/local/filewave/nats
- /install/docker-entrypoint.sh
- /usr/local/filewave/tls
- /etc/filewave_init
- /etc/filewave_setup

It is important to note that while FileWave ensures the security and availability of backups, direct access to these backups by customers is not provided. Going to a backup would generally be based on scenarios such as database or file corruption or the loss of an AWS datacenter due to a disaster.

Related Links

- FileWave Server Backup and Restore for On-Premise

Digging Deeper

Backups are a critical aspect of data management and recovery strategies. They ensure that in the event of data loss, corruption, or disaster, operations can be restored with minimal impact. FileWave's Hosted Server backups are designed to provide a robust and secure safety net for your organization's device management infrastructure.

FileWave Error Codes

Server

Error	Context	Explanation	Solution
-8	During Database Verification	There are some orphaned objects in the database	The first thing to do is to run a DB compact. You can run it from Xserver monitor which is in /Applications/FileWave.This should solve the issue.If the compact is not fixing it, then we must be missing a certain type ofcleanup in that operation. Generally, this doesn't pose a problem. If you'dlike, you can stop the server and zip up the /fwxserver/Data Folder/*datand*idx files and post them to our ftp site.
-1	While doing a Model Update you see a blank window	error creating Fileset File: XXXXXX, folderID: YYYYYYY not found, database damaged, call FileWave Tech Support 889|0xb0513000|FATAL|Error: -1 when updating filesets during model update	This issue is fixed in FileWave version 4.1.1. If you are hitting this issue please upgrade Filewave Admin to 4.1.1.
14	Error 14 on file, found process: fwxserver/XXXXX exiting due to database error: 14 (Only Applicable to Server 3.7.4)	This is a soft database failure caused by a hard restart of the FileWave Server, it doesn't actually reflect a serious issue, but will cause the server to stop.	Upgrade to 3.7.5 to 4.0.X
` fwxserver conflicts with fw-mdm-server-10.1.1-1.0.x86_64 `	upgrading to FileWave 11.x from a previous version	This is normally caused by upgrading from a system that originally installed filewave with two packages: fw-mdm-server fwxserver	As of version 11, FileWave installs both servers with just the single fwxserver installer. To fix this simply remove the mdm component before updating the server. This will not remove any of your MDM data ` sudo yum remove -y fw-mdm-server `

Client / Admin

Error	Context	Explanation	Solution
-150		the file size downloaded to disk does not match the file size stored in the database of the FileWave Server	Delete this file from the Fileset and add a fresh copy from the Admin's hard disk
-125	Client downloading fileset	Booster does not have the file to serve to the client yet and so the client will try again later	Please wait
-13	fwgui is not running	On the client fwgui process is not running	Restart the filewave client from terminal : macOS / Linux ` sudo fwcontrol client restart `
-3	During Admin File Upload	On slow networks an upload may timeout.	
-1	Not in inventory	That comes from a client attempting to activate a fileset before it has downloaded it. After the model update, it adds the activation action back into the queue	Please wait for sometime as the client is still downloading the fileset and once it has finished downloading it will activate
2	reading file from disk	This error is due to a wrong offset request	Upgrade to 3.7.5 or 4.0.4 will solve this issue
15		This could happen if there is no or very less disk space left on the booster that the client is downloading the filesets from	Please check if the booster has enough free disk space. If the disk space is enough and still you are seeing this error contact support help.filewave.com
32	while trying to send file data XXXXXD (Where XXXXX is the file ID)	Error 32 means broken pipe in the network. Generally this error should resolve by itself if everything in the network is fine.Troubleshooting : check to ping from the booster/server to client and vice versaand check if the problem doesn't exist in the network	1. If you see this error for long time try to remove the association of this fileset with the client and than associate again. This should solve theproblem. 2. Update to latest Filewave 3.7.X or 4.0.X

Failed CRC Validation		A CRC check is a form of a checksum which is used to make sure data in files is the same on the client as on the server. The error "failed CRC validation" means that files on the client for whatever reason are being altered compared to what is on the server.	Please send the client log file from the client exhibiting this issue to support help.filewave.com
Kiosk Errors		See: VPP Kiosk Errors	

Booster

Error	Context	Explanation	Solution
Failed CRC Validation		A CRC check is a form of a checksum which is used to make sure data in files is the same on the client as on the server. The error "failed CRC validation" means that files on the client for whatever reason are being altered compared to what is on the server	Please send the booster log file from the booster exhibiting this issue to help.filewave.com

FileWave Log File Locations

The following lists the locations of log files, as well as some additional files used by FileWave across the FileWave family of products

FileWave Admin

FileWave Admin Logs

Details	File	Location

FileWave Admin Log
 	
Logs all FileWave Admin Connection Activity
	FileWaveAdmin.log, FileWaveAdmin.log.*	macOS
		
		`~/Library/Application Support/FileWave/FileWaveAdmin.log`
		
		Windows
		
		`C:\ProgramData\FileWave\FileWaveAdmin.log`

Client Logs
 	
Retrieved Client Logs
	ClientLog_$IP_$Port_$date.log	macOS
		
		`~/Library/Application Support/FileWave/Client Logs/`
		
		Windows
		
		`C:\ProgramData\FileWave\Client Logs\`

Server Logs
 	
Retrieved Server Logs
 	
FileWave Admin > Server > Get Logfile
	fwxserver_$timestamp.log	macOS
		
		`~/Library/Application Support/FileWave/Server Logs/`
		
		Windows
		
		`C:\ProgramData\FileWave\Server Logs\`

FileWave Admin Files

Details	File	Location

FileWave Admin Settings
 	
Settings for the local FileWave Admin App
	macOS
	• com.filewave.FileWaveAdmin.plist
	• com.filewave.admin.plist
	
	Windows
	• Registry
		macOS
		
		`~/Library/Preferences/`
		
		Windows
		
		`HKCU\Software\FileWave\FileWave Admin`

Exported Views
 	
Views saved from FileWave Admin:
 	
* Views > Export Current View
	Filesets Export ($date).txt	macOS
		
		`~/Library/Application Support/FileWave/Exports`
		
		Windows
		
		`C:\ProgramData\FileWave\Exports`

FileWave Booster

Booster Logs

Details	File	Location

Booster Log
 	
Global Booster activity
	fwbooster.log	macOS/Linux
		
		`/private/var/log/fwbooster.log`
		
		Windows
		
		`C:\ProgramData\FileWave\FWBooster\fwbooster.log`

NATS
 	
NATS Booster Logs
	macOS/Linux
	• nats-booster.err.log
	• nats-booster.out.log
	
	Windows
	• nats-booster.log
		macOS/Linux
		
		`/private/var/log/`
		
		Windows
		
		`C:\ProgramData\FileWave\FWBooster\NATS\nats-booster.log`

Discovery Log	macOS/Linux	macOS/Linux

Only exists when discovery configured and run	• fwdiscovery.log	`/private/var/log/fwdiscovery.log`

FileWave Client

FileWave Client Logs

Details	File	Location

Client Logs
 	
Global Client activity
	fwcld.log	macOS
		
		`/var/log/fwcld.log`
		
		Windows
		
		`C:\ProgramData\FileWave\FWClient\fwcld.log`

Kiosk Logs
 	
Kiosk application activity
	FWGUI.log	macOS
		
		`~/Library/Application\ Support/FileWave/FWGUI.log`
		
		Windows
		
		`C:\ProgramData\FileWave\FWGUI.log`

Fileset Script Logs
 	
Logs generated by Fileset scripts
	macOS
	
	$Fileset_ID/$script_name_from_fileset.log
	
	Windows
	
	$Fileset_ID\$script_name_from_fileset.log
		macOS
		
		`/var/log/fwcld/`
		
		Windows
		
		`C:\ProgramData\FileWave\log\fwcld\`

Custom Field Logs
 	
Logs generated by Custom Fields
	custom_field_script.$script_type.log
	e.g.
	• custom_field_script.ps1.log
	• custom_field_script.sh.log
		macOS
		
		`/var/log/fwcld/1/`
		
		Windows
		
		`C:\ProgramData\FileWave\log\fwcld\1\`

Fileset Blocker Script Logs
 	
Logs generated by Blocker Scripts
	blocker_script.$script_type.log
	e.g.
	• blocker_script.py.log
	• blocker_script.bat.log
		macOS
		
		`/var/log/fwcld/1/`
		
		Windows
		
		`C:\ProgramData\FileWave\log\fwcld\1\`

Installer (PKG / MSI) Logs
 	
Logs generated from PKG/MSI Filesets
	$Fileset_ID.log	macOS
		
		`/usr/local/etc/FileWaveInstallerLogfiles/`
		
		Windows
		
		`C:\ProgramData\FileWave\FileWaveInstallerLogfiles\`

FileWave Client Files

Details	File	Location

FileWave Client Settings
 	
Settings for the FileWave Client
	macOS
	• fwcld.plist
	
	Windows
	• Registry
		macOS
		
		`/usr/local/etc/`
		
		Windows
		
		`HKLM\SOFTWARE\Wow6432Node\Filewave\WinClient`

FileWave Client Preferences
 	
Preference file containing unique client details
	macOS
	• com.filewave.Client.plist
	
	Windows
	• client.ini
		macOS
		
		`/Library/Preferences/`
		
		Windows
		
		`C:\ProgramData\FileWave\`

FileWave Client Certificate
 	
Unique certificate & key per client
	• client.crt
	• client.key
		macOS
		
		`/var/FileWave/`
		
		Windows
		
		`C:\ProgramData\FileWave\FWClient\`

Trust Store
 	
Store for self-signed certificates
	*.crt	macOS
		
		`/private/var/FileWave/trust_store`
		
		Windows
		
		`C:\ProgramData\FileWave\FWClient\trust_store`

Cells highlighted in blue indicate files that are unique per client. These files should not be included when copying or migrating clients from one machine to another. To de-personalise a device, without removing the FileWave Client, some files would require editing, whilst others would need to be removed. If it was felt this was a requirement, consider contacting support to assist with this process.

FileWave Imaging Server (IVS)

IVS Logs

Details	File	Location
Django Imaging Server Logs Django logs for requests regarding Serial numbers, names etc. made by netbooted clients	filewave_imaging_server*.log	`/imaging/logs/`
Windows Image Upload Logs Captured Windows image uploads	fwadmin.log	`/imaging/logs/fwadmin.log`
Windows Image Upload Logs Captured Windows image uploads	fwadmin-dlog.log	`/var/log/fwadmin-dlog.log`

Messages Logs
 	
Netboot/PXE Queries & Responses,TFTP transfers, NFS Mounts
	dnsmasq Log	CentoS
		`/var/log/messages`
		Debian
		`/var/log/syslog`

Apache Imaging Server Logs Apache logs for requests regarding Serial numbers, names etc. made by netbooted clients	netboot_*.log	`/imaging/logs/`
Client Imaging Logs Client logs - indicating progress of imaging operation of netbooted clients	$Serial/$Mac-$Date	`/imaging/logs/`
FileWave Client Log IVS FileWave Client Log	fwcld.log	`/var/log/fwcld.log`

FileWave Server

FileWave Server Logs

Details	File	Location
Apache Logs Server Apache logs	• access_log, access_log.* • error_log, error_log.* • forensic_log • httpd.pid	`/usr/local/filewave/apache/logs/`
Apache Exporter Logs Server Apache Exporter Logs	• apache_exporter.out.log • apache_exporter.err.log	`/usr/local/filewave/log/`
Alert Manager Logs Server Alert Manager logs	• alertmanager.out.log • alertmanager.err.log	`/usr/local/filewave/log/`
FileWave Admin Audit Logs Audit logs from FileWave Admin	audit.log	`/usr/local/filewave/log/audit.log`
FileWave Admin Audit Logs Audit logs from FileWave Admin	fwaaudit-[date].txt	`/private/var/log/FWAdmin Audit/`
FileWave Dotenv file Environment variable like configs across services.	*.env	`/usr/local/etc/filewave/.env`
Django Logs Server Django logs	• filewave_django.log, filewave_django.log.* • filewave_django_vpp.log • filewave_django_classroom.log	`/usr/local/filewave/log/`
LDAP Logs Logs from LDAP	fwldap.log, fwldap.log.*	`/private/var/log/`
Software Update Logs	fwsu.log	`/private/var/log/fwsu.log`

Software Update logs		
FWX Process Logs Various fwx process logs	• fwxadmin.log • fwxother.log • fwxserver.log	`/private/var/log/`
Migration Logs Server migration logs	fwxserver-migration-*	`/var/log/fwxserver-migration-*`
Grafana Logs	• grafana.log • grafana.out.log	`/usr/local/filewave/log/`
Installer Logs Linux installer logs	install.log	`/private/var/log/install.log`
mtail Logs Server mtail logs	• mtail.out.log • mtail.err.log	`/usr/local/filewave/log/`
NATS Logs NATS logs	• nats-server.out.log • nats-server.err.log • nats-server-jwt.out.log • nats-server-jwt.err.log	`/usr/local/filewave/log/`
Web Admin Logs	• node_exporter.out.log • node_exporter.err.log • task_executor.log	`/usr/local/filewave/log/`
Postgres Exporter Logs	• postgres_exporter.out.log • postgres_exporter.err.log	`/usr/local/filewave/log/`
Postgres Database Logs	postgresql-$day.log	`/usr/local/filewave/fwxserver/DB/pg_data/pg_log/*.log`
Prometheus Logs	• prometheus_pushprox.out.log • prometheus_pushprox.err.log • prometheus.out.log • prometheus.err.log • redis_exporter.out.log • redis_exporter.err.log • redis.out.log • redis.err.log • redis.log	`/usr/local/filewave/log/`
FileWave Server Logs	request_errors.log	`/usr/local/filewave/log/`
SQL Logs	sql.log	`/usr/local/filewave/log/`
Update Controller Logs Removed in FileWave 14.10	• update_controller_access.log • update_controller.log	`/usr/local/filewave/log/`
Client Monitor	client-monitor.log	`/usr/local/filewave/log/`
FileWave Log Messages	task_executor.log	`/usr/local/filewave/log/`
Scheduler Log Messages	huey.log	`/usr/local/filewave/log/`

Additional Logging

All of the above will default to standard log level. There are 3 levels of logging available:

- 10 – Standard Log Level
- 99 – Debug Log Level
- 101 – Trace Log Level

The level of logging may be set as per our guide:

How to set FileWave Server components to debug mode

How to Restart FileWave Components

There may be times where you will need to restart all components within the FileWave server, or just a single component (postgres or apache). From your macOS or Linux server you can type "fwcontrol", which should give examples of fwcontrol usage.

macOS or Linux Server

You need to prefix commands with sudo to run them with elevated privileges.

At a command prompt:

```
sudo fwcontrol server stop
sudo fwcontrol server start
```

You can also accomplish the same end goal by performing a single command:

```
sudo fwcontrol server restart
```

It is a matter of preference, but some admins will prefer to execute a stop, then a manual start so that they can see all processes are indeed stopped.

Subcomponents can be individually stopped as follows:

```
sudo fwcontrol apache start|stop|restart

sudo fwcontrol postgres start|stop|restart

sudo fwcontrol scheduler start|stop|restart



sudo fwcontrol client start|stop|restart

sudo fwcontrol booster start|stop|restart
```

Troubleshooting

If you find that the fwcontrol control command is not found, you re-create the alias by inputting this command and then try the fwcontrol commands again:

```
alias fwcontrol='/usr/local/bin/fwcontrol'
```

Resolving Network Issues with FileWave Server or Boosters on macOS when using Carbon Black EDR Extension

What

FileWave has observed network issues when the Carbon Black EDR (Endpoint Detection and Response) extension is installed on a FileWave server or booster running on macOS. The issues can manifest as Boosters stopping to answer or respond, leading to disruption in device management workflows.

When/Why

The issue occurs when there is a high volume of network traffic and the Carbon Black EDR extension is inserted into the network stack. The extension's presence in the network stack seems to cause performance issues, which can result in network connectivity and communication problems.

How

If you experience network issues with FileWave when the Carbon Black EDR extension is installed, you can resolve the problem by removing the extension from the FileWave server or booster. This solution has been proven to resolve the issue in multiple cases. On a macOS system, you can use the following command in Terminal.app to list all kernel extensions:

```
systemextensionsctl list
```

The output will appear like this:

```
--- com.apple.system_extension.endpoint_security
enabled    active    teamID    bundleID (version)    name    [state]
*    *    7AGZNQ2S2T    com.vmware.carbonblack.cloud.se-agent.extension (3.7.2fc81/3.7.2fc81)
com.vmware.carbonblack.cloud.se-agent.extension    [activated enabled]
```

You should check the output of this command to determine if the Carbon Black EDR extension is present on your system. If you have concerns about the performance of the Carbon Black EDR extension in high-volume network traffic environments, it may be worth contacting Carbon Black's support team to discuss the issue further.

Related Content

- General Troubleshooting and Errors
- FileWave Log File Locations
- Booster Installation

Digging Deeper

Kernel extensions (KEXTs) are software modules that can be inserted into the macOS kernel to extend its functionality. They can be used to add new features, support new hardware, or modify the behavior of existing drivers. KEXTs run in kernel mode, which means they have the highest level of privilege and can access system resources directly.

However, KEXTs can also introduce stability and performance issues. Since they run in kernel mode, they can crash the system or cause conflicts with other KEXTs. In addition, they can potentially introduce security vulnerabilities if they're not properly designed or implemented.

The Carbon Black EDR extension is an example of a kernel extension that inserts itself into the macOS network stack. By doing so, it's able to monitor network traffic and detect security threats. However, in high-volume network traffic environments, the extension can cause performance issues, which can lead to disruptions in FileWave's device management workflows.

To manage kernel extensions on macOS, Apple provides the systemextensionsctl command. This command allows you to list, enable, disable, and uninstall extensions. If you're experiencing issues with a KEXT, you can use this command to disable or uninstall it to see if that resolves the issue.

In general, it's important to use kernel extensions with caution and only install those from trusted sources. If you're unsure whether a particular KEXT is necessary or safe to use, you should consult with the vendor or seek advice from a subject matter expert.

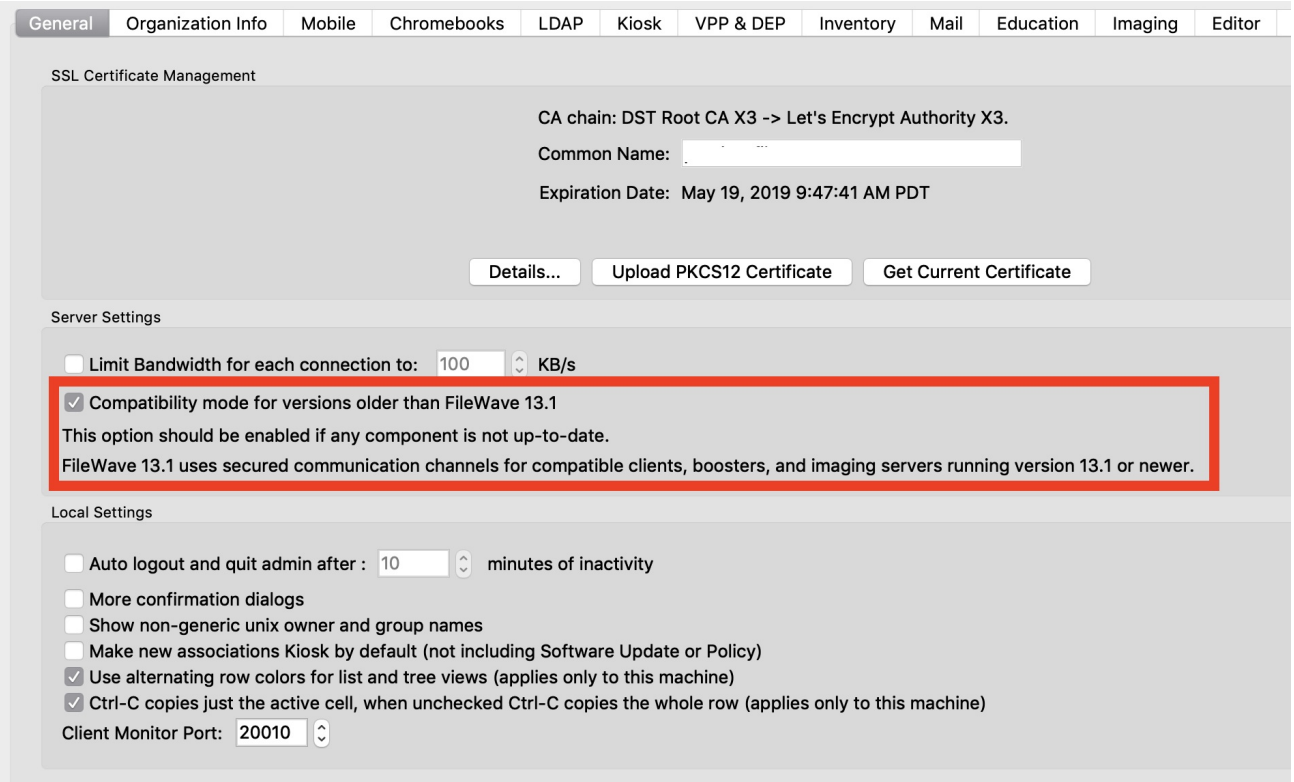
What is Compatibility Mode?

FileWave 13.1 introduced new security options and a mode to allow older clients to connect.

> Compatibility Mode was removed in FileWave 15.4.0 in favor of only using secure connections.

Problem

I don't know what compatibility mode is and what enable and disable do for me.



Environment

FileWave 13.1 introduces a new method of certificate-based security for communication between components (client, booster, server and IVS). Only 13.1 and greater components are able to generate and properly use certificates to communicate with other components using the new method. Therefore, if your server is running 13.1 but you have components that are older than 13.1 they can not generate the needed certificates to have the highest level of security, and will not be able to communicate together.

Resolution

Compatibility Mode Enabled

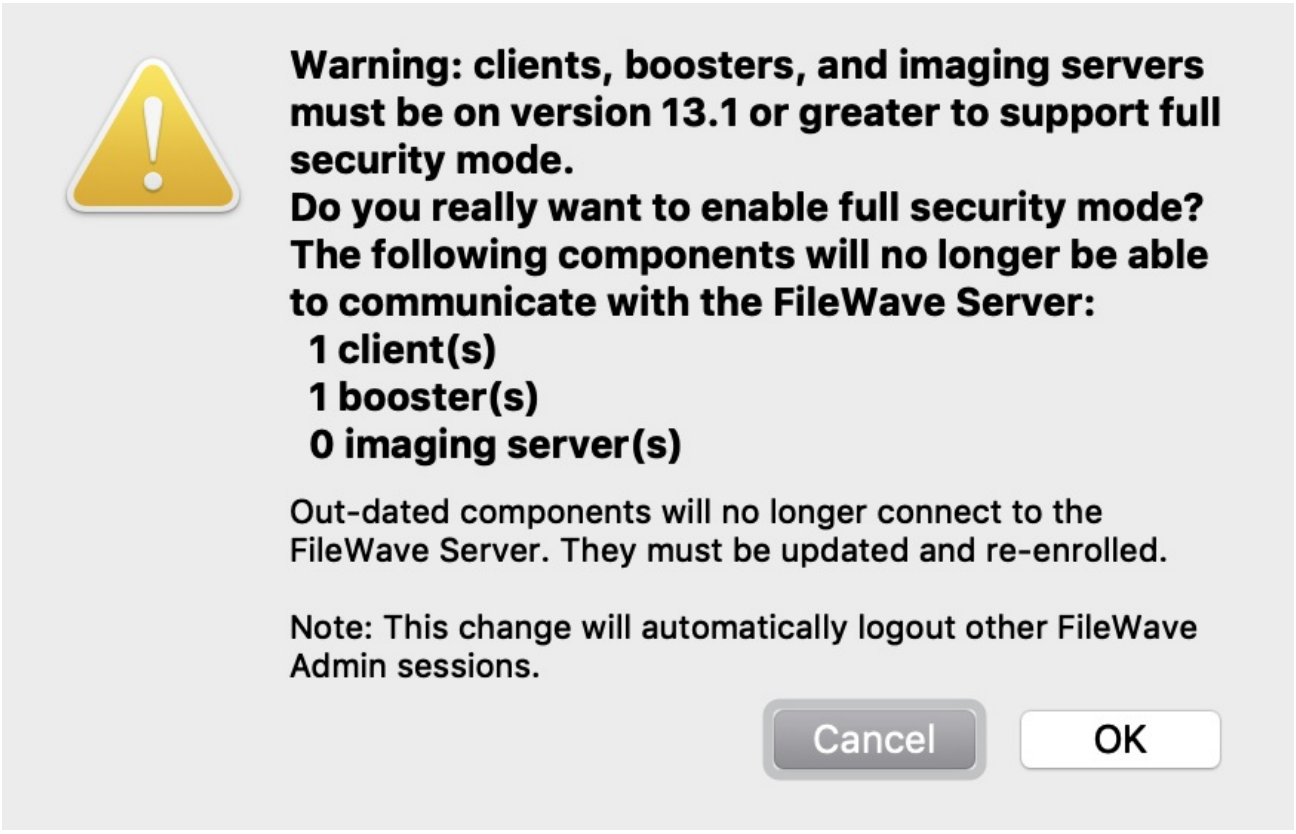
The server allows older clients, boosters, and IVS to communicate with the server with or without valid certificates

Compatibility Mode Disabled

The server will not allow any client, booster, or IVS to communicate with the server unless it has a valid and unique certificate. Boosters and clients are also checking peer certificates and will only communicate if the peer certificate is valid.

Additional Information

When you disable compatibility mode (uncheck the box in preferences) you will receive a warning of clients, boosters, and imaging appliances (AKA IVS), that may potentially be disconnected by you enabling this mode. If you get this warning, it is recommended that you cancel, and resolve the issue before compatibility mode is disabled.



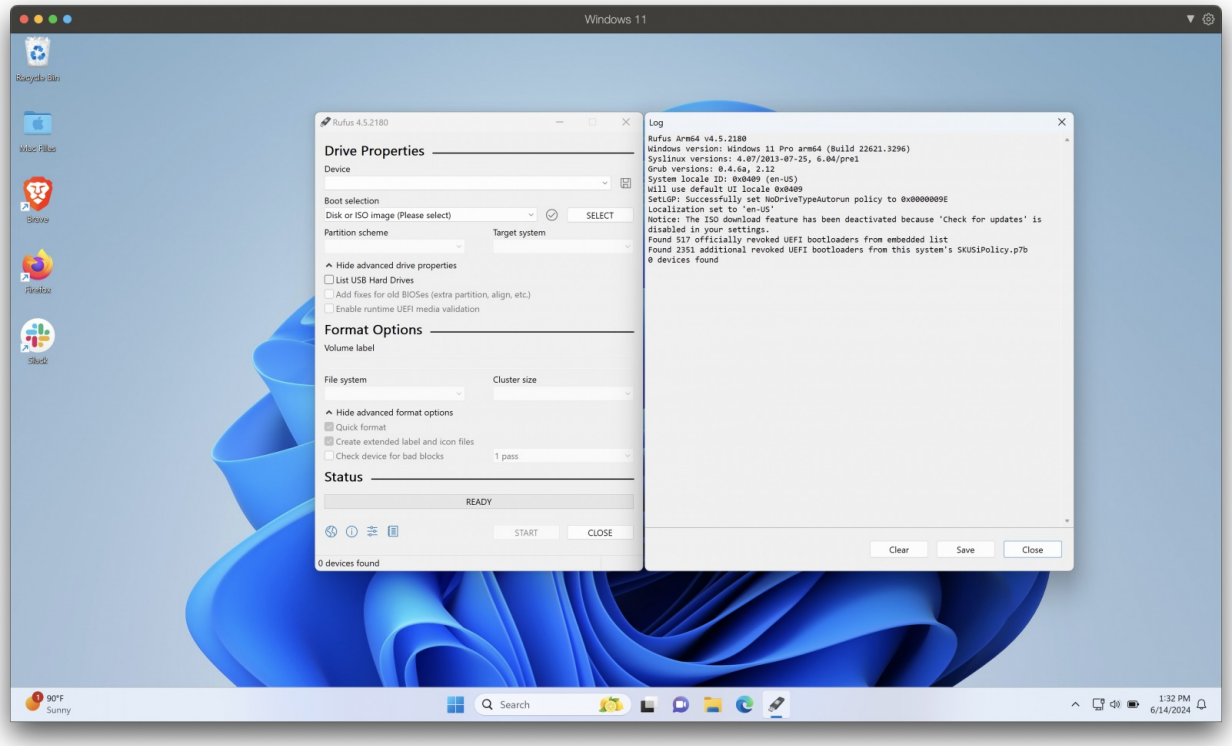
Related Content

- For approving devices see: Enrolling Computer Clients
- For approving boosters see: Booster installation

Rufus - Creating bootable USB drives

What

In this article, we will explain how to use Rufus, a free and open-source utility that helps format and create bootable USB flash drives. Rufus is widely used for creating installation media from bootable ISOs, such as Windows, Linux, and other operating systems. It is an essential tool for anyone needing to install an OS from a USB drive.



When/Why

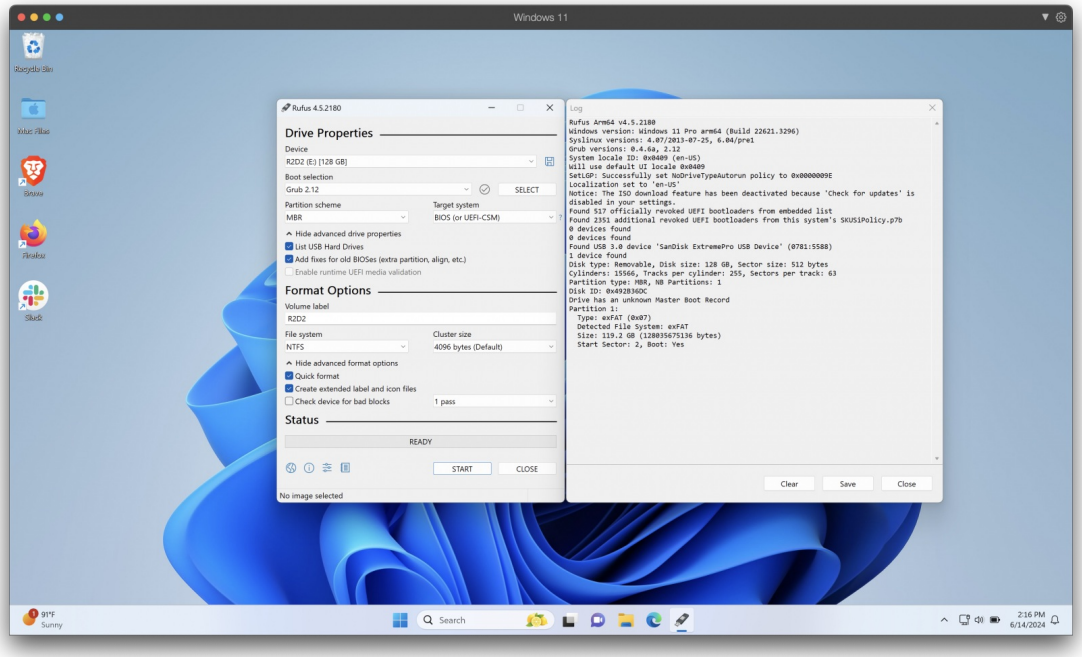
You would use Rufus in scenarios where you need to:

1. Create a bootable USB drive to install or repair an operating system.
2. Perform system recovery or troubleshooting on a computer that won't boot normally.
3. Install various versions of Windows or Linux from a USB drive instead of a CD/DVD.
4. Run a live version of an operating system for testing or recovery purposes.

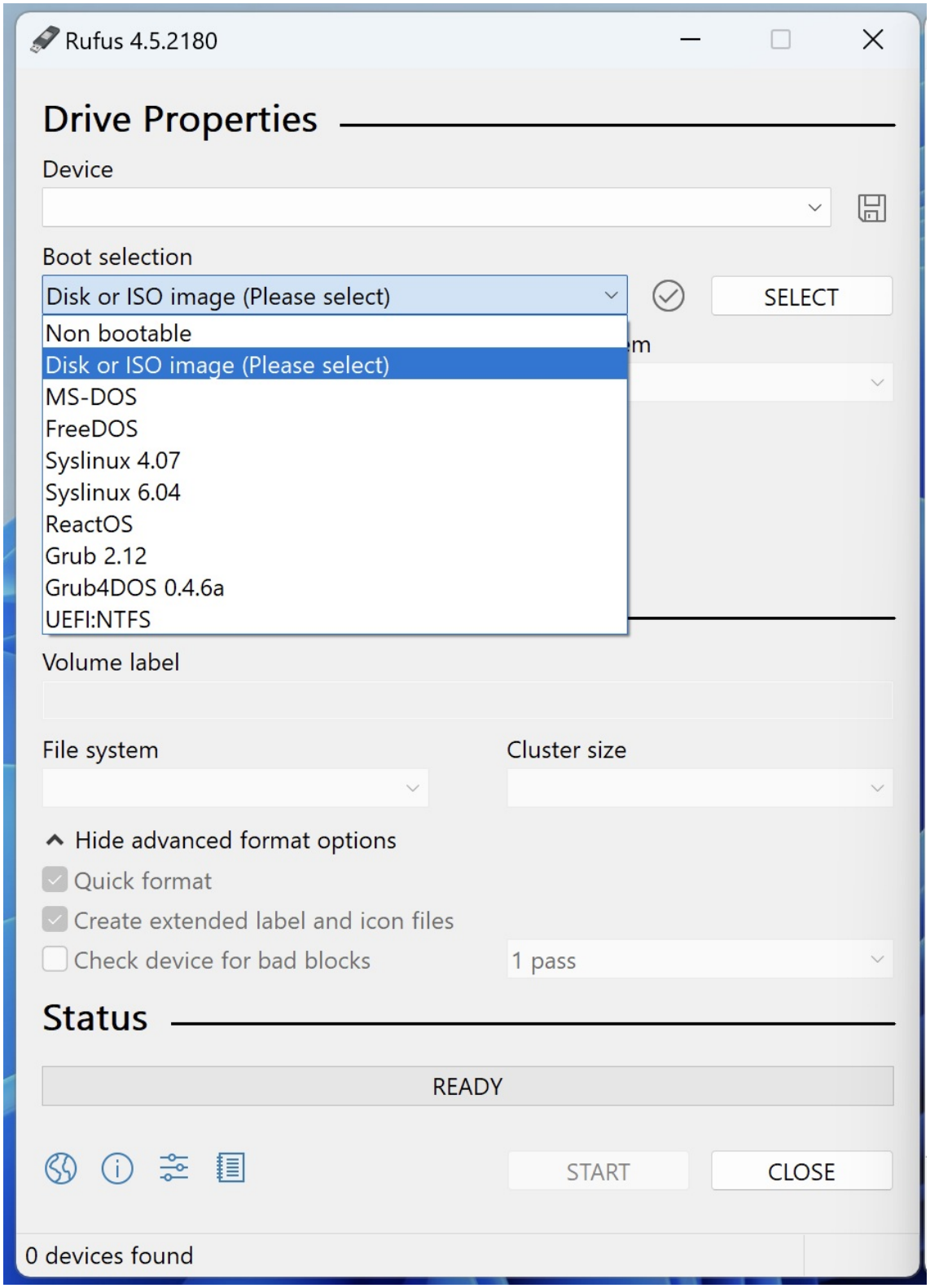
How

Step-by-Step Guide to Using Rufus

1. Download Rufus:
   - Visit the Rufus download page and download the latest version of the software.
   - Rufus is a portable application, so no installation is required. Simply run the downloaded executable file.
2. Prepare Your USB Drive:
   - Insert your USB flash drive into a USB port on your computer.
   - Ensure that the drive has no important data, as it will be formatted during the process.



3. Launch Rufus:
   - Open Rufus by double-clicking the executable file.
   - Rufus will automatically detect your USB drive.
4. Select Bootable ISO:
   - In the "Boot selection" section, click the "Select" button to choose the ISO file you want to use.
   - Navigate to the location of your ISO file and select it.



5. Configure Rufus Settings:
   - Partition Scheme: Choose "MBR" for BIOS or UEFI or "GPT" for UEFI only.
   - File System: Generally, NTFS is recommended for Windows installations, while FAT32 is used for other operating systems.
   - Rufus will adjust the settings automatically based on the selected ISO file.
6. Start the Process:
   - Click the "Start" button to begin the creation of the bootable USB drive.
   - Rufus will warn you that all data on the USB drive will be destroyed. Confirm to proceed.
   - Wait for Rufus to complete the process. This may take several minutes.
7. Completion:
   - Once Rufus has finished, you will see a "Ready" status.
   - Safely eject the USB drive from your computer.

Your bootable USB drive is now ready to be used for installing or repairing your operating system.

Related Content

- How to create a bootable USB drive for Windows

- Creating a bootable USB drive for Linux

Digging Deeper

Rufus offers several advanced features that can be very useful:

- Persistent Storage: Rufus supports persistent storage for some Linux distributions, allowing you to save changes and data between sessions.
- Bad Block Check: Rufus can perform a bad block check on your USB drive to ensure its integrity before creating the bootable media.
- UEFI and BIOS Compatibility: Rufus can create bootable USB drives that are compatible with both UEFI and BIOS systems, ensuring broad compatibility.

For more detailed information, troubleshooting, and FAQs, you can visit the official Rufus Wiki.