

# Automated Booster Configuration for Dynamic Network Environments

## Description

There are setups where automated Booster configuration is desirable; for example devices that may move in and out of the business or between offices. Covered here is an example method allowing clients to automatically adapt their Booster settings based upon certain criteria.

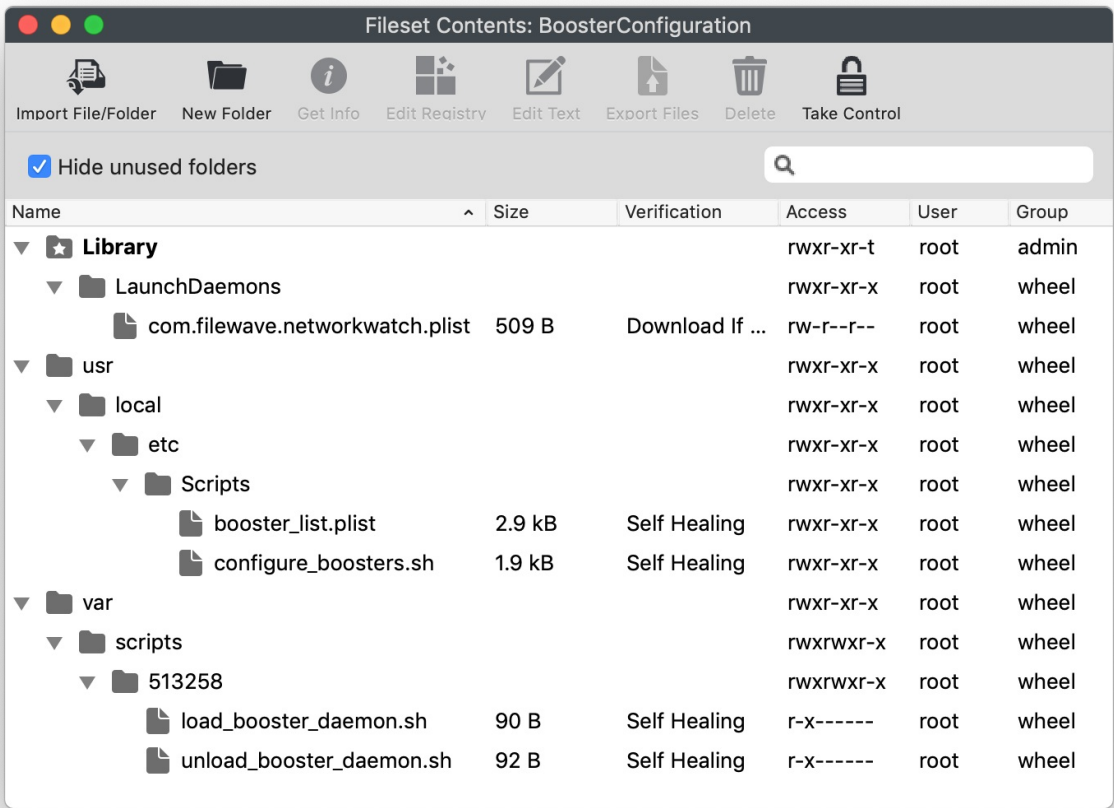
## Information

The following provides an example setup for macOS. Similar functionality could be achieved on Windows and some ideas of this are shown also in this KB.

Requirements:

- Provided Booster Configuration Fileset - [BoosterConfiguration.fileset.zip](#)
- Editing of 'booster\_plist.plist'


The Fileset contains the following items:




Name	Size	Verification	Access	User	Group
Library			rwxr-xr-t	root	admin
LaunchDaemons			rwxr-xr-x	root	wheel
com.filewave.networkwatch.plist	509 B	Download If ...	rw-r--r--	root	wheel
usr			rwxr-xr-x	root	wheel
local			rwxr-xr-x	root	wheel
etc			rwxr-xr-x	root	wheel
Scripts			rwxr-xr-x	root	wheel
booster_list.plist	2.9 kB	Self Healing	rwxr-xr-x	root	wheel
configure_boosters.sh	1.9 kB	Self Healing	rwxr-xr-x	root	wheel
var			rwxr-xr-x	root	wheel
scripts			rxrwxr-x	root	wheel
513258			rxrwxr-x	root	wheel
load_booster_daemon.sh	90 B	Self Healing	r-x-----	root	wheel
unload_booster_daemon.sh	92 B	Self Healing	r-x-----	root	wheel

- Activation Script: load\_booster\_daemon.sh
- Pre-Uninstallation Script: unload\_booster\_daemon.sh
- LaunchDaemon: com.filewave.networkwatch.plist
- Script: configure\_boosters.sh
- Plist file: booster\_list.plist


The Fileset will place the LaunchDaemon which is set to be triggered on load and any network change. Where a network change has occurred, the provided script will run, reconfigure the Booster settings if required and then restart the FileWave client. By providing an additional plist file for booster settings per domain, the file may be updated without the requirement to reload the LaunchDaemon to react to the updated settings.

 The verification settings of the LaunchDaemon are set to 'Download if missing'. The script triggered by the daemon will read and write to this file, preventing unnecessary re-runs. Therefore, the file should not be changed to Self-Healing.

 As of FileWave 16, editing the client plist file will automatically cause the client to re-read the plist, without the need to restart the client. Although restarting the client is not necessary, it won't harm. The 'configure\_boosters.sh' script contained within the Fileset restarts the client. This will remain for now, but may be removed in the near future, as more customers update to version 16. Below is the line that could be commented out, if desired.

```
# Restart the client to pick up the change
/usr/local/bin/fwcontrol client restart
```

To comment out the restart, place a hash symbol at the beginning of the line, as per the line above.

 If commenting out this line, the change will only be picked up once the LaunchDaemon is reloaded, after the client receives the newly updated script.

## Directions

- Download and import the Fileset into FileWave Admin
- Edit the 'booster\_list.plist' file to meet requirements

The example provided utilises the domain name from the device and based upon the booster plist file, the relevant dictionary of items is read into the client settings and then the client is restarted. The LaunchDaemon script uses the 'hostname' command to achieve this.

For example, if the command returned the following:

hostname

```
$ hostname
ML1015.filewave.ch
```

The script would look in the booster\_list.plist file for a dictionary called 'filewave.ch' and configure the client to match this dictionary. The example file has been provided with 3 dictionaries for the following domains:

- 'filewave.ch'
- 'local'
- 'home'

The example dictionary from the 'booster\_list.plist' for 'filewave.ch' is:

booster\_list.plist

```
<key>filewave.ch</key>
<dict>
    <key>booster1</key>
    <string>booster1.filewave.ch</string>
    <key>booster1Port</key>
    <integer>20013</integer>
    <key>booster1PublishPort</key>
    <integer>20003</integer>
    <key>booster2</key>
    <string>booster2.filewave.ch</string>
    <key>booster2Port</key>
    <integer>20013</integer>
    <key>booster2PublishPort</key>
    <integer>20003</integer>
    <key>booster3</key>
    <string>booster3.filewave.ch</string>
    <key>booster3Port</key>
    <integer>20013</integer>
    <key>booster3PublishPort</key>
    <integer>20003</integer>
    <key>booster4</key>
    <string>booster4.filewave.ch</string>
    <key>booster4Port</key>
    <integer>20013</integer>
    <key>booster4PublishPort</key>
    <integer>20003</integer>
```

```
<key>booster5</key>
<string>no.booster.set</string>
<key>booster5Port</key>
<integer>0</integer>
<key>booster5PublishPort</key>
<integer>0</integer>
<key>boosterRouting</key>
<false/>

</dict>
```

Edit the name of the domain, booster names and settings for each dictionary to meet requirements. Note, if no booster is required for any of the five boosters, 'use no.booster.set' as in the above example for booster5.

If the device reports a domain that is not in the list, a default domain will be used. Specify which default domain should be used within the 'booster\_list.plist' file. The example provided is set to use 'local' as the default domain and if implemented, the values of the 'local' dictionary keys will be read from the 'booster\_list.plist' file.

Default Domain

```
<key>default</key>
<string>local</string>
```

Once all domains have been set, with relevant criteria for each dictionary, the Fileset may be associated, tested and then implemented.

## Windows

Windows has a similar option to LaunchDaemons, 'Task Scheduler'. Powershell may be used to import scheduled tasks:

<https://docs.microsoft.com/en-us/powershell/module/scheduledtasks/register-scheduledtask?view=win10-ps>

As such, in theory a Scheduled task could be created based upon network event IDs

- Log: Microsoft-Windows-NetworkProfile/Operational
- Source: NetworkProfile
- Event ID 10000 Connection to a network
- Event ID 10001 Disconnection from a network

This task could trigger a script. Once created, it may then be exported, added to a Fileset with an appropriate script to be triggered by the task, designed to update the booster configuration of the client, and installed using an Activation script by the register-scheduledtask Powershell command.

## Conclusion

The above is an example of how boosters could be configured automatically. Similar could be used to specify IP ranges, however it is more likely that IP ranges can overlap between networks which could cause unexpected configurations. It may therefore be necessary to adapt the script to allow for both domain and IP ranges, depending upon network topology.

Each setup is unique and as such it is not possible to provide a Fileset that is a 'one-fit-for-all'. Whichever method is chosen, try to identify something that will distinguish device locations to ensure settings are delivered as intended.

### Incorrect Configuration

- ❗ If devices receive configurations that do not match the location of the device, it may be possible that device management is lost until the mis-configuration is addressed. Thorough testing should be applied before implementation.

🔄Revision #2

★Created 17 July 2023 00:17:27 by Josh Levitsky

✍Updated 9 April 2025 21:49:08 by Sean Holden