

FWAdmin CLI (Command Line Interface)

Using FileWave Admin CLI (Command Line Interface) for OS X and Windows

Admin CLI allowances include:

- Importing
 - Folder
 - Package
 - Image
- Removing
 - Associations
 - Filesets
- Updating model
- Reporting
 - Clients
 - Filesets
 - Associations

Default Location

macOS

```
/Applications/FileWave/FileWave\ Admin.app/Contents/MacOS/FileWave\ Admin
```

Windows (FW v15.4.2 or lower)

```
"C:\Program Files (x86)\FileWave\FileWaveAdmin.exe"
```

Windows (FW v15.5.0 or higher)

```
"C:\Program Files\FileWave\admin\FileWaveAdmin.exe"
```

Just running the above commands with no arguments will launch the UI version of the Admin

Command Options

Running the command with --help will provide the full list of possible options:

macOS

```
/Applications/FileWave/FileWave\ Admin.app/Contents/MacOS/FileWave\ Admin --help
```

Windows (FW v15.4.2 or lower)

```
C:\Program Files (x86)\FileWave\FileWaveAdmin.exe --help
```

Windows (FW v15.5.0 or higher)

```
"C:\Program Files\FileWave\admin\FileWaveAdmin.exe" --help
```

Here is a list of some of the options available:

```
FileWave Command Line Tool
```

```
Options:
```

-h, --help	Displays this help.
-v, --version	Displays version information.
-u <user>	The filewave admin username.
-p <password>	The filewave admin password.
-H <host>	The filewave server hostname.
-P <port>	The filewave server port number

	(defaults to 20016).
-k	Allows connections to filewave server without checking certificate.
--listClients	Lists all the client client/clone/group information.
--listFilesets	Lists all the fileset information.
--createFileset <name>	Creates a new empty fileset with the specified name.
--importFolder <path>	Imports a folder as a fileset (not as a package).
--importPackage <path>	Imports a package (pkg, flat, mpkg or msi) as a fileset.
--importFileset <path>	Imports a previously exported FileWave fileset or template.
--exportFileset <path>	Exports the given fileset name/id to the specified path
--setRevisionAsDefault	the importing revision will be set as default.
--addRequirementsScript <path>	Adds requirements script (only valid for --importFolder).
--addPreflightScript <path>	Adds preflight script (only valid for --importFolder).
--addActivationScript <path>	Adds activation script (only valid for --importFolder).
--addPostflightScript <path>	Adds postflight script (only valid for --importFolder).
--addVerificationScript <path>	Adds verification script (only valid for --importFolder).
--addPreuninstallationScript <path>	Adds preuninstallation script (only valid for --importFolder).
--addPostuninstallationScript <path>	Adds postuninstallation script (only valid for --importFolder).
--importImage <path>	Imports an image as a fileset.
--deleteFileset <id>	Deletes a fileset by ID/Name.
--listAssociations	Lists all the associations held in the system.
--createAssociation	Create an association between a client/clone/group ID/Name and a fileset ID/Name. Use the --clientgroup and --fileset options.
--deleteAssociation <id>	Deletes an association between a client/clone/group ID/Name and a fileset ID/Name. Use the --clientgroup and --fileset options.
--kiosk	Make this a kiosk association.
--software_update	Make this a software update association.
--licenseDistribution <model>	The license distribution model (only for associations to VPP filesets). Can be "user" or "device".
--updateModel	Updates the FileWave model (as long as no other admins have locked objects).
--setProperty	Sets a fileset property value, use the --fileset, --key and --value parameters to determine for which fileset this is done
--delProperty	Removes a fileset property value, use the --fileset and --key parameters
--setCriticalFlag	Sets the critical flag value for a fileset ; use the --fileset and --value (0/1) parameters
--name <name>	The name value which will be applied to any newly created object.
--comment <comment>	The comment value which will be applied to any newly created object.
--filesetgroup <id>	The ID/Name of the target fileset container, if not specified all objects are created in their respective root container. If the Name of the container does not exist then its assumed to be a

```
--fileset <id>
--revision <name>
--clientgroup <id>
```

```
--root <root>
```

```
--key <key>
--value <value>
```

```
--listExitCodes
```

Fileset Container and will be created automatically.

The ID/Name value of a fileset object.

The name of a revision object.

The ID/Name value of a client, clone or group object.

When importing, if you specify the root then all the data that was imported will be moved into this root folder. The root folder will be created if required.

The key used in the --setProperty call.


The value which will be used in the --setProperty call.

Lists all exit codes and their description.

**** You are seeing this because the -h option was used ****

Best Practices

You should use a separate FileWave Administrator account in order to protect other administrator passwords from accidentally being exposed in scripts. Along the same lines, if you run a command with an admin who is already logged in. It will auto-kick them off from wherever they are at, and from whatever they are doing.

 Model update WILL update the model, no conformation

Know what the Exit Codes mean

```
$ FileWave\ Admin --listExitCodes
0: No Error
100: Unknown Error
101: The given fileset does not exist
102: The given client does not exist
103: The given group does not exist
104: The given target is not a group
105: Database internal error
106: Error while uploading fileset
107: Error while updating the model
108: Login Error
109: Error while importing a fileset
110: Package Type not supported for import
111: Command line parse failed
112: Can't create association with an imaging fileset
```

Examples

Import Fileset:

```
$ FileWave\ Admin -u api -p <password> --importFolder /Applications/TextEdit.app --name "My New Application"
```


Import Package:

```
$ FileWave\ Admin -u api -p <password> --importPackage ~/Downloads/MyExamplePackage.pkg
```

Import Revision:

To add the above PKG to an existing Fileset with ID 537136 and define a revision name of Revision2.

```
$ FileWave\ Admin -u api -p <password> --importPackage ~/Downloads/MyExamplePackage.pkg --fileset 537136 --revision Revision2
```

 Since FileWave 13, it is not possible to add into a current Fileset.

Undocumented

FileWave Admin includes more than one language option. If unspecified, the Admin Application should open in a language to match

the users chosen language if supported. Current supported languages are:

- English – en_GB or en_US
- German – de_DE
- Korean – ko_KR
- Japanese – ja_JP
- Chinese (Traditional and Simplified) – zh_TW or zh_CN

FileWave Admin will default to English otherwise.

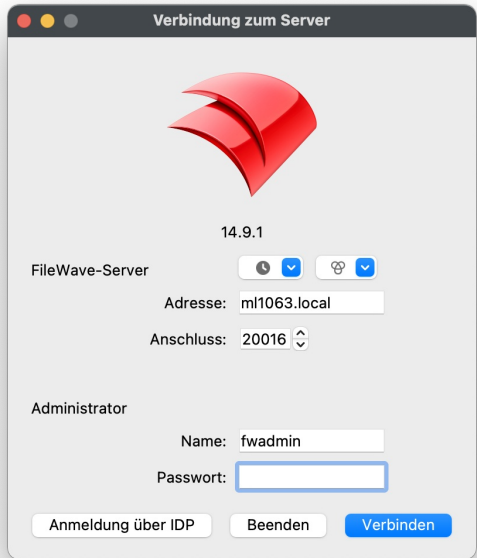
Any of the supported languages may be launched, by use of the language command line option, overriding the current set language:

Windows Korean example

```
& 'C:\Program Files\FileWave\admin\FileWaveAdmin.exe' --lang ko_KR
```

macOS German example

```
/Applications/FileWave/FileWave\ Admin.app/Contents/MacOS/FileWave\ Admin --lang de_DE
```



Related Content

- [FileWave AutoPkg on github](#)