ivanti | File Director

macOS Client Advanced Configuration Guide
Version 2018.3

Copyright Notice

Contents

Mac Client Advanced Configuration

This guide describes how to configure the File Director macOS client by setting keys (property values) in the com.ivanti.filedirector.plist file. This includes basic configuration instructions - such as setting the server and user and whether icon overlays display - and also more advanced settings, such as exclusions.

You can do this via a command line tool or you may find it convenient to use the shell script in the File Director Mac client installer package.

AdminConfig.sh file

The File Director Mac installer package contains a shell script: .AdminConfig.sh.

The script contains instructions to the macOS command line tool (defaults) to update keys in the p-list for the File Director app. For example, this command sets the server for the File Director client to connect to:

```
defaults write com.ivanti.filedirector server "filedirector.mycompany.com"
```

Each command is clearly commented with an explanation of its function and the value of the property, for example:

```
# Set the option to allow the user to uninstall the application to YES
(default) or NO
```

```
# YES = User can uninstall File Director via the GUI, NO = User is blocked
from using the GUI uninstaller to remove File Director
```

```
#defaults write com.ivanti.filedirector allowuninstallapplication -bool
NO/YES
```

The commands, with the exception of default mappings and exclusions, are also commented out.

To use the script:

1. In the File Director Mac installer, locate the Advanced Guide folder and open the AdminConfig.sh file.
2. Uncomment the commands for the settings that you want to apply.
3. Update the commands with the values for the properties as described in the comments.
4. Make the script executable.

    For example, use the change modification command: `chmod +x <scriptname>`

5. Run the script and apply the settings to the users' endpoint.

Basic configuration

Values	Description

Key:
 	
 server
 	
 Value type:
 	
 string
 	
 Value: The name of the server to connect to
	`defaults write com.ivanti.filedirector server "<SERVER NAME>"`
	
	 Sets the server to connect to. Once set, the user cannot change the value.
	
	 Example:
	
	 `defaults write com.ivanti.filedirector server "filedirector.mycompany.com"`

Key:
 	
 user
 	
 Value type:
 	
 string
 	
 Value:
 	
 The username of the user to connect with
	`defaults write com.ivanti.filedirector user "<USER>"`
	
	 Sets the user to connect with the File Director server. Once set, the user cannot change the value.
	
	 Example:
	
	 `defaults write com.ivanti.filedirector user "j.doe@mycompany.com`

Folder settings

Values	Description

Key:
 	
recoveryfolder
 	
Value type:
 	
string
 	
Value: The path for the recovery folder
	`defaults write com.ivanti.filedirector recoveryfolder "~/<FOLDER>"`
	
	Set the folder used to store local data not stored on the server after uninstall or volume deletion. The folder must be under the user home folder. If not set, the value defaults to ~/FileDirector/Salvage.
	
	Example:
	
	`defaults write com.ivanti.filedirector recoveryfolder "~/DNRecovery`

Key:
 	
changerecoveryfolder
 	
Value type:
 	
bool
 	
Value:
 	
YES or **NO**
	`#defaults write com.ivanti.filedirector changerecoveryfolder -bool NO/YES`
	
	Set whether the user is allowed to change the location of the recovery folder from that set by the administrator. Once set, the user cannot change this option.
	
	YES = User can set recovery folder
	
	NO = User cannot change set recovery folder
	
	Example:
	
	`defaults write com.ivanti.filedirector changerecoveryfolder -bool YES`

Key:
 	
user
 	
Value type:
 	
string
 	
Value:
 	
The username of the user to connect with
	`defaults write com.ivanti.filedirector mapping -dict-add "~/<NATIVE FOLDER>" "<MAP POINT>"`
	
	Set the native folder to File Director server map point.
	
	This can be a single map point to local location mapping, for example:
	
	`defaults write com.ivanti.filedirector mapping -dict-add "~/filedirector" "/home"`
	
	Or you can set a range of mappings:
	
	`defaults write com.ivanti.filedirector mapping -dict-add "~/Desktop" "/home/Desktop"`
	
	`defaults write com.ivanti.filedirector mapping -dict-add "~/Documents" "/home/Documents"`
	
	`defaults write com.ivanti.filedirector mapping -dict-add "~/Downloads" "/home/Downloads"`

Values	Description
| | ```
defaults write com.ivanti.filedirector mapping -dict-
add "~/Software" "/home/Software"
```<br><br>```
defaults write com.ivanti.filedirector mapping -dict-
add "~/Movies" "/home/profile/Movies"
```<br><br>ⓘ You can only map to folders or subfolders under the user's home. Do not map more than one map point to the same location. |

UI settings

Values	Description

Key:
 	
dockicon
 	
Value type:
 	
bool
 	
Value:
 	
YES or **NO**
	`defaults write com.ivanti.filedirector dockicon -bool NO/YES`
	
	Set whether the File Director dock icon is displayed. Once you set this, this user cannot alter the setting.
	
	YES = show the dock icon
	
	NO = no dock icon
	
	Example:
	
	`defaults write com.ivanti.filedirector dockicon -bool YES`

Key:
 	
openatlogin
 	
Value type:
 	
bool
 	
Value:
 	
YES or **NO**
	`#defaults write com.ivanti.filedirector changerecoveryfolder -bool NO/YES`
	
	Set whether the user is allowed to change the location of the recovery folder from that set by the administrator. Once set, the user cannot change this option.
	
	YES = User can set recovery folder
	
	NO = User cannot change set recovery folder
	
	Example:
	
	`defaults write com.ivanti.filedirector changerecoveryfolder -bool YES`

Application settings

Values	Description

Key:
 	
allowuninstallapplication
 	
Value type:
 	
bool
 	
Value:
 	
YES or **NO**
	`defaults write com.ivanti.filedirector allowuninstallapplication -bool NO/YES`
	
	Set the option to allow the user to uninstall the application.
	
	YES = The user can uninstall File Director via the GUI (default)
	
	NO = The user is blocked from using the GUI uninstaller to remove File Director
	
	Example:
	
	`defaults write com.ivanti.filedirector allowuninstallapplication -bool YES`

Key:
 	
allowuninstalldata
 	
Value type:
 	
bool
 	
Value:
 	
YES or **NO**
	`defaults write com.ivanti.filedirector allowuninstalldata -bool NO/YES`
	
	Set the option to allow the user to uninstall the data stored locally by File Director.
	
	YES = The user can uninstall File Director data via the GUI (default)
	
	NO = The user is blocked from using the GUI uninstaller to remove File Director data
	
	Example:
	
	`defaults write com.ivanti.filedirector allowuninstalldata -bool YES`

Key:
 	
openatlogin
 	
Value type:
 	
bool
 	
Value:
 	
YES or **NO**
	`defaults write com.ivanti.filedirector allowservercontact -bool NO/YES`
	
	Set the option to allow the user to pause the network. Once set, the user cannot
	
	change this option.
	
	YES = The user can toggle the Contact Server option to pause the network (default)
	
	NO = The user cannot change the contact Server option and pause the network.
	
	Example:
	
	`defaults write com.ivanti.filedirector allowservercontact -bool YES`

Exclusions

File Director keeps all files in sync, regardless of age, type, or size. You may want to tailor what gets synced, saving network bandwidth and storage.

You can define exclusions for macOS File Director clients to exclude files and file types from being uploaded.

The exclusions use the same "language" as the exclusions for the File Director Windows client:

```
<Rule name> = <expression>

<expresssion>= [ext=<extension>]

            [filepath=<regex String>]

rule<Number> = rule<Number> AND rule [Number]
```

For example, the following excludes ISO files:

```
defaults write com.ivanti.filedirector exclusionRule –dict–add
"extensionISO" "ext= .iso"
```

For more information on Windows exclusions, see File Sync in the File Director help.

Default Exclusions

Certain files and folders are excluded by default to save resources, for example:

Exclusion	Description
.DS-Store	Desktop Services Store files
.download	Safari temp download files
.crdownload	Chrome temp download files
.Spotlight-V100	Index data files for Spotlight searches
.fseventstd	File System Events notifications
.DocumentRevisions-V100	Excel temp files
Icon	Icon folders
.part*	Partial downloads
.localized	Localization files
.VolumeIcons.icns	Volume icon folder
.TemporaryItems	Folder for temp items

Exclusion	Description
.Trashes	Folder that stores deleted data
.Trash	Trash folder
~*	Paths that start with a tilde, for example Office temp files