



File Director
powered by AppSense

macOS Client Advanced Configuration Guide

Version 4.4

Copyright Notice

This document contains the confidential information and/or proprietary property of Ivanti, Inc. and its affiliates (referred to collectively as "Ivanti"), and may not be disclosed or copied without prior written consent of Ivanti.

Ivanti retains the right to make changes to this document or related product specifications and descriptions, at any time, without notice. Ivanti makes no warranty for the use of this document and assumes no responsibility for any errors that can appear in the document nor does it make a commitment to update the information contained herein. For the most current product information, please visit www.ivanti.com.

Copyright © 2017, Ivanti. All rights reserved.

Ivanti and its logos are registered trademarks or trademarks of Ivanti, Inc. and its affiliates in the United States and/or other countries. Other brands and names may be claimed as the property of others.

Table of Contents

macOS Client Advanced Configuration Guide	1
Copyright Notice	2
Table of Contents	3
Mac Client Advanced Configuration	4
AdminConfig.sh file	5
Basic configuration	6
Folder settings	7
UI settings	9
Application settings	10
Exclusions	11
Default Exclusions	12

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

Folder settings

Values	Description
<p>Key: recoveryfolder</p> <p>Value type: string</p> <p>Value: The path for the recovery folder</p>	<pre>defaults write com.ivanti.filedirector recoveryfolder "~/<FOLDER>"</pre> <p>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.</p> <p>Example:</p> <pre>defaults write com.ivanti.filedirector recoveryfolder "~/DNRecovery"</pre>
<p>Key: changerecoveryfolder</p> <p>Value type: bool</p> <p>Value: YES or NO</p>	<pre>#defaults write com.ivanti.filedirector changerecoveryfolder -bool NO/YES</pre> <p>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.</p> <p>YES = User can set recovery folder</p> <p>NO = User cannot change set recovery folder</p> <p>Example:</p> <pre>defaults write com.ivanti.filedirector changerecoveryfolder -bool YES</pre>
<p>Key: user</p> <p>Value type: string</p> <p>Value: The username of the user to connect with</p>	<pre>defaults write com.ivanti.filedirector mapping -dict-add "~/<NATIVE FOLDER>" "<MAP POINT>"</pre> <p>Set the native folder to File Director server map point.</p> <p>This can be a single map point to local location mapping, for example:</p> <pre>defaults write com.ivanti.filedirector mapping -dict-add "~/filedirector" "/home"</pre> <p>Or you can set a range of mappings:</p> <pre>defaults write com.ivanti.filedirector mapping -dict-add "~/Desktop" "/home/Desktop"</pre> <pre>defaults write com.ivanti.filedirector mapping -dict-add "~/Documents" "/home/Documents"</pre> <pre>defaults write com.ivanti.filedirector mapping -dict-add "~/Downloads" "/home/Downloads"</pre>

Values	Description
	<pre>defaults write com.ivanti.filedirector mapping -dict-add "~/Software" "/home/Software"</pre> <pre>defaults write com.ivanti.filedirector mapping -dict-add "~/Movies" "/home/profile/Movies"</pre> <hr/> <p> 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.</p> <hr/>

UI settings

Values	Description
<p>Key: dockicon</p> <p>Value type: bool</p> <p>Value: YES or NO</p>	<pre>defaults write com.ivanti.filedirector dockicon -bool NO/YES</pre> <p>Set whether the File Director dock icon is displayed. Once you set this, this user cannot alter the setting.</p> <p>YES = show the dock icon</p> <p>NO = no dock icon</p> <p>Example:</p> <pre>defaults write com.ivanti.filedirector dockicon -bool YES</pre>
<p>Key: openatlogin</p> <p>Value type: bool</p> <p>Value: YES or NO</p>	<pre>#defaults write com.ivanti.filedirector changerecoveryfolder -bool NO/YES</pre> <p>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.</p> <p>YES = User can set recovery folder</p> <p>NO = User cannot change set recovery folder</p> <p>Example:</p> <pre>defaults write com.ivanti.filedirector changerecoveryfolder -bool YES</pre>

Application settings

Values	Description
Key: allowuninstallapplication Value type: bool Value: YES or NO	<pre>defaults write com.ivanti.filedirector allowuninstallapplication -bool NO/YES</pre> <p>Set the option to allow the user to uninstall the application.</p> <p>YES = The user can uninstall File Director via the GUI (default)</p> <p>NO = The user is blocked from using the GUI uninstaller to remove File Director</p> <p>Example:</p> <pre>defaults write com.ivanti.filedirector allowuninstallapplication -bool YES</pre>
Key: allowuninstalldata Value type: bool Value: YES or NO	<pre>defaults write com.ivanti.filedirector allowuninstalldata -bool NO/YES</pre> <p>Set the option to allow the user to uninstall the data stored locally by File Director.</p> <p>YES = The user can uninstall File Director data via the GUI (default)</p> <p>NO = The user is blocked from using the GUI uninstaller to remove File Director data</p> <p>Example:</p> <pre>defaults write com.ivanti.filedirector allowuninstalldata -bool YES</pre>
Key: openatlogin Value type: bool Value: YES or NO	<pre>defaults write com.ivanti.filedirector allowservercontact -bool NO/YES</pre> <p>Set the option to allow the user to pause the network. Once set, the user cannot</p> <p>change this option.</p> <p>YES = The user can toggle the Contact Server option to pause the network (default)</p> <p>NO = The user cannot change the contact Server option and pause the network.</p> <p>Example:</p> <pre>defaults write com.ivanti.filedirector allowservercontact -bool YES</pre>

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>
<expression>= [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
.Volumelcons.icns	Volume icon folder
.TemporaryItems	Folder for temp items
.Trashes	Folder that stores deleted data
.Trash	Trash folder
~*	Paths that start with a tilde, for example Office temp files