



ShockFiler Xtra Help

[Installation](#)

[What ShockFiler Xtra Does](#)

[Getting Started](#)

[Methods at a Glance](#)

[Methods Documentation](#)

[ShockFiler Relative Path Syntax](#)

[Firewall/Proxy](#)

[Error Table](#)

[JavaScript](#)

[Shockwave](#)

[Limitations](#)

[How to Order & Register](#)

[Licensing & Availability](#)

[Technical Support](#)

For up-to-date information please visit our web site:

xtras.tabuleiro.com



SHOCKFILER XTRA HELP: INSTALLATION

The installation procedure is slightly different depending on the version of Director and platform used. Make sure you have administrative rights to create files in the directory where Director is installed on your system.

WINDOWS

MACINTOSH

Director 11

Director 11

Director
MX 2004

Director MX
2004

Director
MX

Director MX

Director 8.5

Director 8.5



SHOCKFILER XTRA HELP: INSTALLATION: WINDOWS - DIRECTOR 11

INSTALLING THE XTRA ON WINDOWS - Director 11

Decompress the installation .zip file. This will unpack the Xtra, documentation and sample files to a folder named "ShockFiler" on your machine. To install the Xtra, just copy the file Windows\ShockFiler.x32 to the Director 11 XTRAS folder. If your copy of Director 11 is installed at the default location, the Windows Xtra file will be located at:

C:\Program Files\Adobe\Adobe Director 11\Configuration\Xtras\ShockFiler.x32

Now you need to install the files necessary for creation of cross-platform projector for Mac OSX. Go back to the "ShockFiler" directory where the Xtra files were unpacked. Open the **Mac Universal** directory. Now copy the file "ShockFiler.cpio" to the "Configuration\Cross Platform Resources\Macintosh\Xtras" directory used by Director 11. In a default installation of Director this file will end up at the following location:

C:\Program Files\Adobe\Adobe Director 11\Configuration\Cross Platform Resources\Macintosh\Xtras\ShockFiler.cpio

Finally, you need to edit the xtrainfo.txt file to include information about ShockFiler. This information is used by the Shockwave and cross-platform publishing features in Director 11, to locate the files needed when assembling the Mac OSX version of your projector. The xtrainfo.txt file is located by default at:

C:\Program Files\Adobe\Adobe Director 11\Configuration\xtrainfo.txt

Double click the file to open it in notepad, or alternatively edit with any other text editor. You need to add the following line to the end of the file:

Online Help

```
[#namePPC:"ShockFiler", #nameW32:"ShockFiler.x32",  
#package:"http://download.tabuleiro.com/packages/ShockFiler/2/ShockFiler"]
```

You may want to customize this line in the future, to instruct Director to download ShockFiler Shockwave packages from the same server that hosts your Shockwave applications. This is covered in more detail at the [Using the Xtra in Shockwave](#) section of the documentation. Restart Director for the changes to take effect. The Xtra should be listed when you issue the command "put the xtralist" in the message window. The Xtra functions will also be listed when you click the message window Scripting Xtras button.



SHOCKFILER XTRA HELP: INSTALLATION: WINDOWS - DIRECTOR MX 2004

INSTALLING THE XTRA ON WINDOWS - Director MX 2004

If you have not done so, we recommend updating to Director MX 2004 version 10.1 before installing the Xtra. This will allow creation of projectors for Mac Classic and OSX (Director MX 2004 without the update can only create cross platform projectors for OSX.)

Decompress the installation .zip file. This will unpack the Xtra, documentation and sample files to a folder named "ShockFiler" on your machine. To install the Xtra, just copy the file Windows\ShockFiler.x32 to the Director MX 2004 XTRAS folder. If your copy of Director MX 2004 is installed at the default location, the Windows Xtra file will be located at:

C:\Program Files\Macromedia\Director MX 2004\Configuration\Xtras\ShockFiler.x32

Now you need to install the files necessary for creation of cross-platform projector for Mac OSX. Go back to the "ShockFiler" directory where the Xtra files were unpacked. Open the **Mac Carbon** directory. Now copy the files "ShockFiler.data" and "ShockFiler.rsrc" files to the "Configuration\Cross Platform Resources\Macintosh\Xtras" directory used by Director MX 2004. In a default installation of Director these files will end up at the following locations:

C:\Program Files\Macromedia\Director MX 2004\Configuration\Cross Platform Resources\Macintosh\Xtras\ShockFiler.data

C:\Program Files\Macromedia\Director MX 2004\Configuration\Cross Platform Resources\Macintosh\Xtras\ShockFiler.rsrc

If you are running Director MX 2004 10.1, you can also install the files necessary for creation of cross-platform projector for Mac Classic. Again, go back to the "ShockFiler" directory where the Xtra files were unpacked. Open the **Mac Classic** directory. Now copy the files "ShockFiler.data" and "ShockFiler.rsrc" files to the "Configuration\Cross Platform Resources\Classic\Xtras" directory used by Director MX 2004. In a default installation of Director these files will end up at the following locations:

Online Help

C:\Program Files\Macromedia\Director MX 2004\Configuration\Cross Platform Resources\Classic\Xtras\ShockFiler.data

C:\Program Files\Macromedia\Director MX 2004\Configuration\Cross Platform Resources\Classic\Xtras\ShockFiler.rsrc

Finally, you need to edit the xtrainfo.txt file to include information about ShockFiler. This information is used by the Shockwave and cross-platform publishing features in Director MX 2004, to locate the files needed when assembling the OSX and Classic versions of your projector. The xtrainfo.txt file is located by default at:

C:\Program Files\Macromedia\Director MX 2004\Configuration\xtrainfo.txt

Double click the file to open it in notepad, or alternatively edit with any other text editor. You need to add the following line to the end of the file:

```
[#namePPC:"ShockFiler", #nameW32:"ShockFiler.x32",  
#package:"http://download.tabuleiro.com/packages/ShockFiler/2/ShockFiler"]
```

You may want to customize this line in the future, to instruct Director to download ShockFiler Shockwave packages from the same server that hosts your Shockwave applications. This is covered in more detail at the [Using the Xtra in Shockwave](#) section of the documentation. Restart Director for the changes to take effect. The Xtra should be listed when you issue the command "put the xtralist" in the message window. The Xtra functions will also be listed when you click the message window Scripting Xtras button.



SHOCKFILER XTRA HELP: INSTALLATION: WINDOWS - DIRECTOR MX AND 8.5

INSTALLING THE XTRA ON WINDOWS - Director MX and Director 8.5

Decompress the installation .zip file. This will unpack the Xtra, documentation and sample files to a folder named "ShockFiler" on your machine. To install the Xtra, just copy the file **Windows\ShockFiler.x32** to the Director 8.5 or Director MX XTRAS folder. If you have previously installed an older copy of the Xtra make sure to remove or replace it.

These are the default locations of the Xtras folder for each application:

Director 8.5- C:\Program Files\Macromedia\Director 8.5\Xtras

Director MX- C:\Program Files\Macromedia\Director MX\Xtras

Finally, you need to edit the xtrainfo.txt file to include information about ShockFiler. This information is used by the Shockwave publishing features in Director. The xtrainfo.txt file is located by default at:

Director 8.5 - C:\Program Files\Macromedia\Director 8.5\xtrainfo.txt

Director MX - C:\Program Files\Macromedia\Director MX\xtrainfo.txt

Double click the file to open it in notepad, or alternatively edit with any other text editor. You need to add the following line to the end of the file:

```
[#namePPC:"ShockFiler", #nameW32:"ShockFiler.x32",  
#package:"http://download.tabuleiro.com/packages/ShockFiler/2/ShockFiler"]
```

You may want to customize this line in the future, to instruct Director to download ShockFiler Shockwave packages from the same server that hosts your Shockwave applications. This is covered in more detail at the [Using the Xtra in Shockwave](#) section of the documentation. Restart Director for the changes to

Online Help

take effect. The Xtra should be listed when you issue the command "put the xtralist" in the message window. The Xtra functions will also be listed when you click the message window Scripting Xtras button (Director MX).



SHOCKFILER XTRA HELP: INSTALLATION: MACINTOSH - DIRECTOR 11

INSTALLING THE XTRA ON MAC OSX - Director 11

Double-click the installation .dmg file. This will mount a disk named "ShockFiler" on your desktop.

The first step is to copy the Universal binary version of the Xtra, which will be used in the authoring environment and also when creating Mac OSX projectors, for both Intel and PPC machines. This file is located in the install disk image, at:

ShockFiler/Mac Universal/ShockFiler.xtra

This file needs to be copied to the Director 11 Xtras folder. The final pathname for the Xtra in a default installation of Director 11 will be:

OSX Volume Name/Applications/Adobe Director 11/Configuration/Xtras/ShockFiler.xtra

Windows projectors can also be created directly on Director 11 running on Mac OSX after installation of the Windows version of the Xtra. It is located on the install disk, at:

ShockFiler/Windows/ShockFiler.x32

Copy this file to the Cross Platform resources directory in Director 11, so that it will be available at:

OSX Volume Name/Applications/Adobe Director 11/Configuration/Cross Platform Resources/Windows/Xtras/ShockFiler.x32

Online Help

Finally, you need to edit the xtrainfo.txt file to include information about ShockFiler. This information is used by the Shockwave and cross-platform publishing features in Director 11, to locate the files needed when assembling the Windows version of your projector. The xtrainfo.txt file is located by default at:

OSX Volume Name/Applications/Adobe Director 11/Configuration/xtrainfo.txt

Double click the file to open it in TextEdit, or alternatively edit with another text editor. Make sure to save the file in plain text format, though. You need to add the following line to the end of the file:

```
[#namePPC:"ShockFiler", #nameW32:"ShockFiler.x32",  
#package:"http://download.tabuleiro.com/packages/ShockFiler/2/ShockFiler"]
```

You may want to customize this line in the future, to instruct Director to download ShockFiler Shockwave packages from the same server that hosts your Shockwave applications. This is covered in more detail at the [Using the Xtra in Shockwave](#) section of the documentation. Restart Director for the changes to take effect. The Xtra should be listed when you issue the command "put the xtralist" in the message window. The Xtra functions will also be listed when you click the message window Scripting Xtras button.



SHOCKFILER XTRA HELP: INSTALLATION: MACINTOSH - DIRECTOR MX 2004

INSTALLING THE XTRA ON MAC OSX - Director MX 2004

Double-click the installation .dmg file. This will mount a disk named "ShockFiler" on your desktop.

The first step is to copy the OSX version of the Xtra, which will be used in the authoring environment and also when creating OSX projectors. This file is located in the install disk image, at:

ShockFiler/Mac Carbon/ShockFiler

This file needs to be copied to the Director MX 2004 Xtras folder. The final pathname for the OSX Xtra in a default installation of Director MX will be:

OSX Volume Name/Applications/Macromedia Director MX 2004/Configuration/Xtras/ShockFiler

Director MX 2004 running on Mac OSX can also be used to create Classic projectors, for Mac OS versions 8 and 9. In order to enable this feature you need to copy the Classic version of ShockFiler to the correct location in your Director MX installation. First locate the Classic version of ShockFiler in the install disk:

ShockFiler/Mac Classic/ShockFiler

This file needs to be copied to the following location in the Director MX 2004 folder, to be used for cross-platform publishing. Copy it to:

OSX Volume Name/Applications/Macromedia Director MX 2004/Configuration/Cross Platform Resources/Classic MacOS/Xtras/ShockFiler

Online Help

Windows projectors can also be created directly on Director MX 2004 running on Mac OSX after installation of the Windows version of the Xtra. It is located on the install disk, at:

ShockFiler/Windows/ShockFiler.x32

Copy this file to the Cross Platform resources directory in Director MX 2004, so that it will be available at:

OSX Volume Name/Applications/Macromedia Director MX
2004/Configuration/Cross Platform Resources/Windows/Xtras/ShockFiler.x32

Finally, you need to edit the xtrainfo.txt file to include information about ShockFiler. This information is used by the Shockwave and cross-platform publishing features in Director MX 2004, to locate the files needed when assembling the Classic MacOS and Windows versions of your projector. The xtrainfo.txt file is located by default at:

OSX Volume Name/Applications/Macromedia Director MX
2004/Configuration/xtrainfo.txt

Double click the file to open it in TextEdit, or alternatively edit with another text editor. Make sure to save the file in plain text format, though. You need to add the following line to the end of the file:

```
[#namePPC:"ShockFiler", #nameW32:"ShockFiler.x32",  
#package:"http://download.tabuleiro.com/packages/ShockFiler/2/ShockFiler"]
```

You may want to customize this line in the future, to instruct Director to download ShockFiler Shockwave packages from the same server that hosts your Shockwave applications. This is covered in more detail at the [Using the Xtra in Shockwave](#) section of the documentation. Restart Director for the changes to

Online Help

take effect. The Xtra should be listed when you issue the command "put the xtralist" in the message window. The Xtra functions will also be listed when you click the message window Scripting Xtras button.



SHOCKFILER XTRA HELP: INSTALLATION: MACINTOSH - DIRECTOR
MX

INSTALLING THE XTRA ON MAC OSX - Director MX

Double-click the installation .dmg file. This will mount a disk named "ShockFiler" on your desktop.

The first step is to copy the OSX version of the Xtra, which will be used in the authoring environment and also when creating OSX projectors. This file is located in the install disk image, at:

ShockFiler/Mac Carbon/ShockFiler

This file needs to be copied to the Director MX Xtras folder. The final pathname for the OSX Xtra in a default installation of Director MX will be:

OSX Volume Name/Applications/Macromedia Director MX/Xtras/ShockFiler

Director MX running on Mac OSX can also be used to create Classic projectors, for Mac OS versions 8 and 9. In order to enable this feature you need to copy the Classic version of ShockFiler to the correct location in your Director MX installation. First locate the Classic version of ShockFiler in the install disk:

ShockFiler Folder/Mac Classic/ShockFiler

This file needs to be copied to the following location in the Director MX folder:

OSX Volume Name/Applications/Macromedia Director MX/Classic
MacOS/Xtras/ShockFiler

Online Help

Finally, you need to edit the `xtrainfo.txt` file to include information about ShockFiler. This information is used by the Shockwave and cross-platform publishing features in Director MX 2004, to locate the files needed when assembling the Classic MacOS version of your projector. The `xtrainfo.txt` file is located by default at:

OSX Volume Name/Applications/Macromedia Director MX/xtrainfo.txt

Double click the file to open it in TextEdit, or alternatively edit with another text editor. Make sure to save the file in plain text format, though. You need to add the following line to the end of the file:

```
[#namePPC:"ShockFiler", #nameW32:"ShockFiler.x32",  
#package:"http://download.tabuleiro.com/packages/ShockFiler/2/ShockFiler"]
```

You may want to customize this line in the future, to instruct Director to download ShockFiler Shockwave packages from the same server that hosts your Shockwave applications. This is covered in more detail at the [Using the Xtra in Shockwave](#) section of the documentation. Restart Director for the changes to take effect. The Xtra should be listed when you issue the command "put the xtralist" in the message window. The Xtra functions will also be listed when you click the message window Scripting Xtras button.



SHOCKFILER XTRA HELP: INSTALLATION: MACINTOSH - DIRECTOR 8.5

INSTALLING THE XTRA ON MAC OS 8 AND 9 - Director 8.5

Running under OSX, double-click the installation .dmg file. This will mount a disk named "ShockFiler" on your desktop. To install the Xtra just copy the file "ShockFiler" from the Mac Classic folder to the Xtras folder of your Director 8.5 installation. The final pathname for the Xtra will be for example:

Macintosh HD:OS9 Applications:Macromedia Director 8.5:Xtras:ShockFiler

Finally, you need to edit the xtrainfo.txt file to include information about ShockFiler. This information is used by the Shockwave publishing features in Director. The xtrainfo.txt file is located in the directory where Director 8.5 was installed, for example at:

Macintosh HD:OS9 Applications:Macromedia Director 8.5:xtrainfo.txt

Double click the file to open it in SimpleText, or another editor capable of saving plain text files. You need to add the following line to the end of the file:

```
[#namePPC:"ShockFiler", #nameW32:"ShockFiler.x32",  
#package:"http://download.tabuleiro.com/packages/ShockFiler/2/ShockFiler"]
```

You may want to customize this line in the future, to instruct Director to download ShockFiler Shockwave packages from the same server that hosts your Shockwave applications. This is covered in more detail at the [Using the Xtra in Shockwave](#) section of the documentation. Restart Director for the changes to take effect. The Xtra should be listed when you issue the command "put the xtralist" in the message window.



SHOCKFILER XTRA HELP: WHAT SHOCKFILER XTRA DOES

ShockFiler Xtra exports text, image, sound members and vList Xtra members to external files saved on an FTP server or to the local hard drive. ShockFiler also sends and receives external files via FTP. With ShockFiler Xtra you can easily:

- Save artwork created by the user
- Save test results
- Append new user records to a file
- Save sounds recorded using Audio Xtra
- Save list data in binary format using vList Xtra
- Send and receive external files via FTP

ShockFiler Xtra was designed specifically for use in Shockwave movies but also works in Director projectors.



SHOCKFILER XTRA HELP: GETTING STARTED

ShockFiler is a Scripting Xtra. Scripting Xtras are used to extend the Lingo language with new functions and datatypes. Unlike Asset Xtras there is no visual representation of a scripting Xtra in the Director interface, and you can not create castmembers or sprites.

The first step is to download and install the ShockFiler Xtra, following the instructions in the [installation](#) page. Now that ShockFiler is installed, let's verify that the installation was successful. If you are using DirectorMX you should see the ShockFiler entry in the Scripting Xtras context menu, appearing at the top of the message window. Selecting the ShockFiler submenu and the "put interface" entry will output a list of all commands understood by ShockFiler in the message window. You can also use the following command

Lingo:

```
put the xtralist
```

JavaScript syntax:

```
trace(_player.xtraList)
```

to verify which Xtras are installed, including the version number for each one.

We will now try a simple scripting session using ShockFiler and the message window. All of ShockFiler functions are available as global Lingo keywords, so there is no need to create an instance of the Xtra in order to use them. For example:

Lingo:

```
mystring = sf_Version()
```

```
put mystring
```

```
--"2.0.0"
```

JavaScript syntax:

```
var mystring = sf_Version()
```

```
trace(myresult)
```

Online Help

That's it. You just confirmed that ShockFiler is installed correctly, and its functions are available as new global keywords in Lingo and Javascript, ready to be used.

REGISTRATION

In addition to displaying an unregistered dialog, a non registered version of ShockFiler always displays a progress dialog for ftp transfers that includes a demo message under the progress bar.

If you have purchased the ShockFiler Xtra you received a registration number. Make the sf_Register registration call in the first movie that uses ShockFiler commands. Make sure you do it before using any other ShockFiler commands. The code can be in any type of script. The startMovie handler is the most convenient place to put it because it will execute before any other code that might try to use ShockFiler functions.

```
on startMovie
```

```
success = sf_Register([111,222,333])
```

```
-- notice that the parameter to sf_Register is a list, not a string!
```

```
if success <> 1 then alert "Registration of ShockFiler Xtra failed"
```

```
end
```



SHOCKFILER XTRA HELP: METHODS AT A GLANCE

Method and arguments	Purpose
<u>sf_Register</u> (serialist)	Prevents the demo dialog from coming up after purchase.
version = <u>sf_Version</u> ()	Returns the version number of the Xtra.
<u>sf_Send</u> (host, directory,username, password,memberName or vListData, castlibName, remoteFileName, fileType, fileProperties, appendFlag, progressDialogFlag, progressDialogTitle)	Exports a text member, sound member,image member, or vList member to a standalone file on an FTP server.
<u>sf_Send</u> ("TODISK", localDirectoryPath,"", "",memberName or vListData, castlibName, fileName, fileType, fileProperties, appendFlag, saveDialogFlag, dialogText)	Exports a text member, sound member,image member, or vList member to a standalone file on the local hard drive.
<u>sf_SendFile</u> (host, directory,username, password,localFilePath,remoteFileName,appendFlag, progressDialogFlag,progressDialogTitle)	Sends a file to an FTP server.
filePath = <u>sf_OpenFileDialog</u> ()	Returns the path to the selected local file. For use with <u>sf_SendFile</u>
<u>sf_GetFile</u> (host, directory,username, password,localFilePath,remoteFileName, progressDialogFlag,progressDialogTitle)	Retrieves a file from an FTP server
filePath = <u>sf_SaveFileDialog</u> (defaultFileName)	Returns the path to the file to save. For use with <u>sf_GetFile</u>
<u>sf_TransferStatus</u> ()	

Online Help

	Returns the status of the current file transfer.
<u>sf_RequestFileList</u> (host, directory,username, password)	Requests a file listing from the remote FTP server.
<u>sf_FileListToLingoList</u> (listType)	Converts a file listing retrieved by <u>sf_RequestFileList</u> into a Lingo list.
<u>sf_CreateDirectory</u> (host, directory, username, password, directoryNameToCreate)	Creates a directory on the remote FTP server.
<u>sf_Disconnect</u> ()	Breaks the connection with the remote FTP server.

Firewall / Proxy Methods

<u>sf_PassiveMode</u> (onOrOffFlag)	Determines whether the next file transfer or list request will be done using active or passive mode.
<u>sf_PortNumber</u> (portNumber)	Determines what port number the next file transfer or list request will use.



SHOCKFILER XTRA HELP: METHODS DOCUMENTATION

The following is a list of functions available after installation of ShockFiler Xtra.

<u>sf_Register</u>	<u>sf_OpenFileDialog</u>	<u>sf_FileListToLingoList</u>
<u>sf_Version</u>	<u>sf_GetFile</u>	<u>sf_CreateDirectory</u>
<u>sf_Send(server)</u>	<u>sf_SaveFileDialog</u>	<u>sf_Disconnect</u>
<u>sf_Send(local)</u>	<u>sf_TransferStatus</u>	
<u>sf_SendFile</u>	<u>sf_RequestFileList</u>	

sf_Register([1111,2222,3333]) -global function, used to register Xtra. It can be called at any time, usually when the Director movie starts. Unregistered versions of the Xtra are fully functional for evaluation purposes, but will display a warning the first time you use a method other than sf_Register.

ShockFiler serial number are strings, and have the generic format SFLXX-1111-2222-3333, where XX is the major Xtra version. In order to protect your serial number from being included as a string in your Director projectors or dcr movies, the sf_Register function requires only the three groups of numbers 1111, 2222 and 3333 inside a Director list. Leading zeroes do not need to be entered.

An example: if your serial number is SFL20-0123-4567-0089 then you should register using the following command, usually on a startmovie handler:

Lingo:

```
sf_Register([123, 4567,89])
```

JavaScript syntax :

```
sf_Register(list(123, 4567,89))
```

version = sf_Version - Returns a string containing version number of the xtra.

Online Help

Example:

```
currentVersion = sf_Version()  
  
put currentVersion  
  
-- "2.0.1"
```

sf_Send(host, directory,username, password,memberName or vListData, castlibName, remoteFileName, fileType, fileProperties, appendFlag, progressDialogFlag, progressDialogTitle) - where:

host is the address of FTP host;

directory is the relative path from login directory to target directory;

username is either the login username or "" for anonymous;

password is either the login password or "" for anonymous, in which case what gets sent is "shockfiler@nodomain.com";

memberName or vListData is the name of member to export or propertyList exported from vList Xtra;

castlibName is the name of castlib containing member to be exported, can be "" which defaults to internal (lib 1). If a vList propertyList is being passed in the memberName param, then this param should be passed as "";

remoteFileName is the file name to save exported member to in remote FTP directory;

fileType is the Lingo symbol, currently supports #text, #jpeg, #wave,#aiff, #bina;

fileProperties is the Lingo list, contains additional specifications for the file to be written. Pass [] for a file type with no properties;

appendFlag is a boolean, 1 to append to existing file, 0 to overwrite existing file and ignored for all file types except text;

progressDialogFlag is a boolean, 1 to display progress bar as file transfers or 0 not to;

progressDialogTitle is the caption for progress dialog title bar.

Online Help

Returns 1 if the transfer can start, or a negative error code if there was a problem. Exports a text member, sound member, image member or vList data to a designated file on an FTP server. sf_Send logs in the the server, transmits the file exported from the specified member, and logs out.

The sf_Send command handles connecting sending and disconnecting automatically. If you are not interested in monitoring the status of the transmit, sf_Send is the only command needed to send a file (other than sf_Register for registered users). Transmission of the file operates concurrently with other Director operations. The movie will continue running and other Lingo will execute after sf_Send, while the file is being sent.

An internet connection must be open before you can call sf_Send successfully. If no connection is open you will get an error. ShockFiler maintains one active connection at a time. If you are sending a file or getting a listing and you attempt a second sf_Send or sf_RequestFileList before the operation in progress finishes, you will get a "connection is in use" error.

SERVER PARAMETERS

The host, directory, username, and password parameters are used to log in to the FTP site where the file will be uploaded. They are the same type of information you usually type in to an FTP client to transmit files. You may need to use additional configuration commands before calling this method if you are connecting through a firewall or proxy server.

Host - The host parameter holds the name or IP address of the FTP server, for example "ftp.macromedia.com" or "192.192.192.8". ShockFiler automatically logs in to the standard FTP port, which is 21.

Directory - The directory parameter holds the path to the target directory relative to the login directory. If the login directory IS the target directory, specify "" for the directory parameter. Specify path separators as "/". Do not use a path separator at the beginning or end of the path. Directory names are case-sensitive. For instance, the string "Exchange/mac" specifies the directory "mac" contained in directory "Exchange" which is a subdirectory of the login directory. If the login directory was:

```
/users/safeway
```


Online Help

then a directory parameter of "Exchange/mac" would produce a target directory of:

```
/users/safeway/Exchange/mac
```

Username - The username parameter specifies the login name of a user allowed to log in to the FTP server. Some servers allow anonymous login. For anonymous login you can specify either "anonymous" or simply "" for the user name.

However, few servers allow users logged in as "anonymous" to upload files. You will probably need the system administrator to give you a username and password for the FTP server to be able to upload files.

Password - The password parameter specifies the password owned by the user allowed to log in to the FTP server. Some servers allow anonymous login. When a username of anonymous is entered, the server expects to see an e-mail address for the password. If you have entered "anonymous" for the username you can specify either your e-mail address or "" for the password. If you use "", ShockFiler sends "shockfiler@nodomain.com" for the e-mail address.

MEMBER TYPES SUPPORTED

sf_Send works with the following internal member types: field, text, bitmap, sound, richtext. Linked cast members are not supported. Compressed members are not supported.

Text, field, richtext members - For field, text, and richtext, the text property of the member is exported. That means that no style information will be exported for richtext, just plain text. However, you can copy any of the following properties of those types of members into a second field member and export the information as a text file through the xtra:

html of member

rtf of member

scriptText of member

Online Help

The text files exported are plain text files with the type of line ending specified in the fileProperties parameter.

#mac - CR (CarriageReturn)

#win - CR, LF (LineFeed)

#unix - LF

Bitmap and picture members - Bitmap and picture members are saved as JPEG files with a 72 dpi resolution and the same pixel dimensions as the original cast member. The quality of the JPEG can be set in the fileProperties parameter.

You can export other graphic member types that have a picture property, by setting the picture property of an existing #bitmap member to the picture property of the other graphic member.

```
member("bitmap member").picture = member("not directly supported graphic type").picture
```

Sound members - Sound members are saved as either WAVE or AIFF files depending on which is specified as the fileType parameter. Sound files are saved with the number of channels, sample rate and sample depth properties of the original sound member. WAVE export only supports standard Win sampling rates (11025, 22050, 44100). Trying to export a member with a different Mac sampling rate, for instance 11027, will return a -4 "member type does not match export file type" error.

ShockFiler can export any sound member recorded at runtime and most sound member types created by importing a sound. The following table shows the various sound member types and any additional information needed to export them using ShockFiler.

Member originally imported from format	Notes
IMA	Not supported on Win for export to AIF, but export to WAVE is OK
AU	OK on both platforms for export to both formats.
System 7	OK on both platforms for export to both formats.

EXPORTED FILES

ShockFiler Xtra uses the data from the specified cast member to create a new file on the FTP server or append data to an existing file (Append applies to text only).

filetype - The filetype parameter currently accepts 5 file types - #text, #jpeg, #wave, #aiff or #bina. Type #bina is for vList data. The file type specified must match the member type or the transfer will error. For instance, specifying #wave to export a text member will not work.

Javascript note: Use the symbol function to convert a string to the Lingo symbol datatype ex: symbol("jpeg")

fileProperties - The file properties list contains additional properties that are particular to the file type. Currently they are:

File type	Property	Values	Default
#text	#lineEndings	#mac, #win, #unix	#mac
#jpeg	#quality	0 - 100	100
#wave	none	--	--
#aiff	none	--	--
#bina	none	--	--

Pass [] for file types that do not have any additional file properties.

JavaScript note: Use the list and propList functions to create a Lingo list to pass in this parameter. You cannot pass a JavaScript Array. EX: propList(symbol("quality"),100)

#lineEndings: The line endings property converts any existing line endings characters in the text to the ending type specified. It does not add line endings to text.

Online Help

#quality: The quality property adjusts the amount of compression applied to the file and therefore the size of the resulting file as well as the time it will take to transmit the data. The lower the quality setting the higher the compression and the smaller the data and the resulting file.

appendFlag - ShockFiler Xtra has the capability to append text to an existing text file on the FTP server, if the server software supports APPEND, if the server has been configured to allow APPEND, and the login account being used has the permission to APPEND.

If the appendFlag parameter is 0 a new file is created on the FTP server if a file of that name does not exist. If the file already exists, and the login user has overwrite permission, the existing file is overwritten. If the file exists and the login user does not have overwrite permission, an error will be returned.

If the appendFlag parameter is 1 and the transfer is for fileType #text, the text will be appended to any existing file of the same name, otherwise a new file will be created. The appendFlag parameter is ignored for any file type other than #text.

You can use the append feature to maintain data files containing data from many users.

PROGRESS DIALOG

Please note the limitations of this feature.

Passing 1 for the progressDialogFlag parameter displays a moveable, nonmodal dialog that shows a progress thermometer and percent done. The progressDialogTitle parameter is the text that will display in the title bar of the progress dialog. Note: The progress dialog always displays in the unregistered version of ShockFiler Xtra.

If the user pressed the cancel button on the dialog during the transfer, the transfer will abort and sf_getTransferStatus will show a status of "Error" and an error string of "Cancelled by server".

INTEGRATION WITH AUDIO XTRA AND VLIST

Online Help

Audio Xtra (version 6 and later) includes a new function, `axConvertToSFData()`, that can be used to pass binary audio data as a vList-compatible property list, ready to be used with ShockFiler. See the Audio Xtra documentation for more information on how to use this function.

vList Xtra saves Lingo lists and other data types to members and files. ShockFiler Xtra includes support for receiving data from vList in a format that will create a standalone vList file when saved by ShockFiler to an FTP server. Getting the data from vList to the FTP server requires two steps:

1. Call a function in vList that returns the packaged vList data into a variable
2. Pass the variable containing the vList data as an argument to ShockFiler's `sf_Send` command

The following vList Xtra function returns data into a variable that you can pass to ShockFiler's `sf_Send` command:

`propList = vList_to_sf (data, compressionFlag)` - where `data` is a Lingo list or other data type and `compressionFlag` is a boolean, 1 for compress the data or 0 for don't compress the data. Returns a Lingo property list in the format:

[#data: binary vList data, #length: size of data in bytes]

Use the following parameters in the `sf_Send` call to send vList data instead of a cast member:

`memberName/propList`: name of property list variable containing vList data

`castlibName`: "". Ignored. No cast member is being sent

`remoteFileName`: name to give exported vList file on FTP server

`fileType`: #bina

`fileProperties`: []. Ignored. No properties can be set for a vList file.

Example:

-- vList function

```
packagedData = vList_to_sf ( [#name: "Andy", #score: 50] , 1)
```

-- the packaged vList data is now in variable packagedData

--

Online Help

-- Variable `packagedData` is passed in the "member" param. It will create a `vList` file on the

-- FTP server named `mydata.lst`.

```
sf_Send ("ftp.macromedia.com", "incoming", "guest",  
"mypassword", packagedData, "", "mydata.lst", #bina, [], 0, 0, "")
```

SF_SEND EXAMPLES

Saving a user's preferences, as stored in a field of the movie

Example:

```
err = sf_Send ("ftp1.domain.com", "langModule/gary", "languageStudent",  
"password", "preference field", "External castlib", "gary.txt", #text, [#lineEndings:  
#win], 0, 1, "Storing preferences on server ...")
```

```
if err <> 1 then
```

```
  alert "Cannot initiate transfer"
```

```
end if
```

Logs on to FTP host "ftp1.domain.com" and into the relative directory of "langModule/gary", with a username of "languageStudent" and a password of "password". Exports the text in field "preference field" of castlib "External castlib" to file `gary.txt` on the FTP server. Any line ending characters in the text will be converted to the Windows convention of CR,LF. The append flag is off, so a new file will be created for the data. The progressDialog flag is on, so a progress dialog will display during the transfer with the caption of "Storing preferences on server ...".

Checks the error value returned. If the error returned is 0 the transfer will start. The movie can continue playing and other Lingo will continue to execute while the file transfer is going on.

Saving a user's art created during the session

Example:

```
member("screenshot member").picture = (the stage).picture
```

```
--
```

Online Help

```
err = sf_Send ("192.243.88.53", "", "george", "dogfight", "screenshot member", "",  
"sketch.jpg", #jpeg, [#quality: 75], 0, 1, "Sending your picture to the server ...")
```

```
--
```

```
if err < 1 then
```

```
  alert "Cannot initiate transfer"
```

```
end if
```

Logs on to FTP host "192.243.88.53" , with a username of "george" and a password of "dogfight". Since no relative directory has been specified, the login directory will receive the file. Exports the internal bitmap in member "screenshot member" of the first internal castlib to file sketch.jpg on the FTP server. Some compression will be applied to the JPEG because a quality setting of 75 was specified. The append flag is off, but it would be ignored for a #jpeg transfer anyway. The progressDailog flag is on, so a progress dialog will display during the transfer with the caption of "Sending your picture to the server ...".

Appending a user's game score to an existing file on the FTP server

Example:

```
put name & TAB & score & RETURN into field "sendScore"
```

```
--
```

```
err = sf_Send ("ftp.games.com", "frogger", "froggerplayer", "ribbet", "sendscore",  
"", "highscores.txt", #text, [#lineEndings: #unix], 1, 0, "")
```

```
--
```

```
if err < 1 then
```

```
  alert "Sorry, could not transmit your game score"
```

```
end if
```

Logs on to FTP host "ftp.games.com" , and into the relative directory of "frogger", with a username of "froggerplayer" and a password of "ribbet". Appends the text in member "sendScore" of the first internal castlib, to existing file highscores.txt on the FTP server. The return in the text will be replaced with the UNIX line ending of a linefeed. The text is appended to the file instead of creating a new file because the append parameter is set to 1. The progressDailog flag is off, so no progress dialog will display during the transfer and a caption for the dialog is therefore not necessary so that parameter has been left blank.

Online Help

Sending sound previously recorded to a member using Audio Xtra

Example:

```
err = sf_Send ("ftp.music.com", "pub/audition", "", "", "recording 1", "Sound  
castlib", "rec1.wav", #wave, [], 0, 1, "Sending your recording to server ...")
```

```
--
```

```
if err < 1 then
```

```
  alert "Sorry, could not transmit your recording"
```

```
end if
```

Logs on to FTP host "ftp.music.com", and into the relative directory of "pub/audition". Since the username and password parameters have both been left blank, the login username will be passed as "anonymous" and the password will be passed as "shockfiler@nodomain.com" Transmits the sound member "recording 1" of castlib "Sound castlib" to create a WAVE file on the FTP server named "rec1.wav". There are no additional properties to set for this media type, so that parameter is passed as an empty list. The append flag is off, but it would be ignored for a #wave transfer anyway. The progressDailog flag is on, so a progress dialog will display during the transfer with the caption of "Sending your recording to server ...".

```
sf_Send ("TODISK", Path, "", "", memberName or vListData, castlibName,  
fileName, fileType, fileProperties, appendFlag, saveDialogFlag, dialogText) -  
where:
```

"TODISK" is a string (pass this string in all caps to sf_Send to indicate that it will be saving the member to disk);

Path is a string, either full path to local directory to save file to or relative path;

notUsed is a string, ignored - pass "";

notUsed is a string, ignored - pass "";

memberName or vListData is a string, the name of member to export or propertyList exported from vList Xtra

Online Help

castlibName is a string, the name of castlib containing member to be exported, can be "" which defaults to internal (lib 1). If a vList propertyList is being passed in the memberName param, then this param should be passed as "";

fileName is a string, the file name to save exported member to if a full directory was passed in "path", or "" if a relative path was passed in "path";

fileType is a Lingo symbol, currently supports #text, #jpeg, #wave,#aiff, #bina;

fileProperties is a Lingo list, contains additional specifications for the file to be written. Pass [] for a file type with no properties;

appendFlag is a boolean, 1 to append to existing file, 0 to overwrite existing file. Ignored for all file types except text;

saveDialogFlag is a boolean, 1 to display saveFile dialog or 0 not to [always displays in unregistered version];

dialogText is a string, one-line message to display on dialog.

Returns 1 if the file was saved successfully, or a negative error code if there was a problem. Exports a text member, sound member, image member or vList data to a designated file on the local hard drive.

This alternate usage of the sf_Send command saves to disk if the first argument that usually contains the FTP server IP address contains the string "TODISK" in all caps instead. Some arguments such as username and password have no use when you are saving locally so those arguments are ignored, but you must still pass an empty string for the unused argument to keep the number of arguments the same.

No progress dialog displays during a save to disk. Control passes to the system during the file copy and the next line of Lingo executes after the file copy, so it is not possible to poll sf_TransferStatus during the copy. Calling sf_TransferStatus after the copy returns the number of bytes copied and any error that happened during the copy.

NOTE: If you use this command to save a text or field member and the member contains no text, the target file will not be created. If the target file already exists, it will be deleted.

PATH: PASSING THE FULL PATH TO THE FILE TO SAVE

If you provide a full path to a directory in param "path", ShockFiler builds the local file path using the path and fileName params. You must pass the directory path without the trailing delimiter which ShockFiler will add. The following strings:

directoryPath: "C:\TEMP" <--- no ending slash

fileName: "animals.jpg"

will save to the path:

"C:\TEMP\animals.jpg"

If you want to use `sf_SaveFileDialog` to prompt the user for a path, leave one or both of the path parameters empty and the dialog will appear automatically. If you want to prompt for the path earlier on, use the following code to break up the full file path returned from the dialog into the two pieces required in the `directoryPath` and `fileName` params:

```
on mouseUp
```

```
  fullpath = sf_saveFileDialog("")
```

```
  directory = splitFilePath(fullpath)[1]
```

```
  filename = splitFilePath(fullpath)[2]
```

```
end
```

```
on splitFilePath path
```

```
  if (the platform).char[1..3] = "Mac" then
```

```
    the itemDelimiter = ":"
```

```
  else
```

```
    the itemDelimiter = "\"
```

```
  end if
```

```
  items = the number of items of path
```

fname = the last item of path

delete the last item of path

return list(path,fname)

end

PATH: PASSING A RELATIVE PATH TO THE FILE TO SAVE

If you pass a relative path in param "Path" for sf_Send, ShockFiler ignores param "fileName" because it has all of the information it needs to build the path from param "Path". If you pass a relative path in param "Path" then pass "" in param "fileName" because it will be ignored. The following examples show two ways to specify a path of "C:\project\graphics\newpict.jpg" in a movie located at "C:\project\movie.dir".

Full path:

```
sf_Send ("TODISK", "C:\project\graphics", "", "", "screenshot member", "",  
"newpict.jpg", #jpeg, [#quality: 75], 0, 1, "" )
```

Relative path:

```
sf_Send ("TODISK", "@graphics\newpict.jpg", "", "", "screenshot member", "", "",  
#jpeg, [#quality: 75], 0, 1, "" )
```

SAVE FILE DIALOG

If you pass 1 in the saveDialogFlag parameter, with both directoryPath and fileName filled, the saveDialog will come up and show the contents of the directory specified in directoryPath and show the file specified in fileName as the default name for the saved file. If the saveDialogFlag parameter is 1, then directoryPath and fileName are only used by the dialog. The actual save path is determined by the user when the dialog appears.

If you pass 0 in the saveDialogFlag parameter, the directoryPath and fileName are used together to form the path where the file will be saved. If either directoryPath or fileName or both are passed as empty strings, even if the saveDialog parameter is 0, the dialog will still display, since there isn't enough information to save the file.

SHOCKWAVE

In Shockwave, the save file dialog always displays before the file is saved and the path the user picks is used. In Shockwave it doesn't matter what is passed in `directoryPath` or `fileName` because those arguments are not used.

Example:

```
on mouseUp

-- capture the stage and store it in a member

member("screenshot member").picture = (the stage).picture

-- save the member to a local jpeg file

err = sf_Send ("TODISK", "C:\TEMP", "", "", "screenshot member", "",
"sketch.jpg", #jpeg, [#quality: 75], 0, 1, "" )

if err < 1 then

alert "Could not save file"

end if

end
```

The example above saves a captured image of the stage to a JPEG file. The initial path is "C:\TEMP\sketch.jpg" but since the `saveDialogFlag` param is 1, a save dialog will display starting at the TEMP directory with sketch.jpg as the default file name. Whatever the user chooses as the path is where the file will be saved. The file will be saved with a quality of 75%.

`sf_SendFile` (`host`, `directory`, `username`, `password`, `localFilePath`, `remoteFileName`, `appendFlag`, `progressDialogFlag`, `progressDialogTitle`) - where host is a string, the address of the FTP host; directory is a string, the relative path from login directory to target directory; username is a string, either the login username or "" for anonymous; password is a string, either the login password or "" for anonymous, in which case what gets sent is "shockfiler@nodomain.com"; `localFilePath` is a string, the full path to the local file to send, or the relative path or "" to put up an open file dialog

Online Help

and use the path the user picks; `remoteFileName` is a string, the file name to save the sent file to on the FTP server; `appendFlag` is a boolean, 1 to append to existing file, 0 to overwrite existing file; `progressDialogFlag` is a boolean, 1 to display progress bar as file transfers or 0 not to; and `progressDialogTitle` is a string, the caption for the progress dialog title bar.

Returns 1 if the transfer can start, or a negative error code if there was a problem. Sends a local file on the user's hard drive to an FTP server. In authoring and projectors any valid file path can be passed for `localFilePath`. In Shockwave, you must use a relative path for the local file path or leave the `localFilePath` parameter empty, to prompt the user for the save file path. Passing a full path in Shockwave for the `localfilePath` will error.

Example:

```
err = sf_SendFile ("ftp.agency.com", "pub/incoming","", "", "C:\temp\rough.jpg",  
"treatment_1.jpg", 1, "Sending your graphic file to server ...")
```

```
--
```

```
if err < 1 then
```

```
  alert "Sorry, could not transmit your file"
```

```
end if
```

`filePath = sf_OpenFileDialog ()` - Returns a string containing the chosen file or "" if the user did not chose a file. In authoring and in projectors the full file path to the local file is returned. In Shockwave, the path to the local file is stored internally by the xtra and only the file name is returned.

Puts up a standard file picker dialog. This function was included to provide a way to prompt the user for a local file to send when using `sf_SendFile`. You first call the function, save its returned file path into a variable, then pass the variable to `sf_SendFile`:

In authoring and projectors the function operates like a normal open file dialog and can be used any time you want to prompt the user for a local file. In Shockwave, the open file dialog does not return a filepath and therefore is useful only with `sf_SendFile`. In Shockwave, the dialog returns only the file name chosen and ShockFiler stores the path to the file internally. When you pass the filename returned from the open file dialog to `sf_SendFile`, ShockFiler determines the full path to the file from the filename and sends the file.

Online Help

In Shockwave the open file dialog function stores the path to the last file chosen by the user. Once the user has chosen a file using the dialog, you can send the same file multiple times by passing the file name returned from the function to `sf_SendFile`. To send another file, you must prompt the user again. The new chosen file path is stored internally and the old path is erased. Passing "" in the `localFilePath` parameter of `sf_SendFile` will also prompt the user for the local file to send, but the path to the file is not saved internally by the dialog after the transfer completes.

Example:

```
usersFileChoice = sf_OpenFileDialog ( )
```

```
sf_SendFile "ftp.soundbank.com", "pub/incoming","", "",userFileChoice,  
"sample.wav", 1, "Sending your sound file to the server)
```

`sf_GetFile` (host, directory,username, password,localFilePath,remoteFileName,progressDialogFlag,progressDialogTitle) - where host is a string, the address of the FTP host; directory is a string, the relative path from login directory to target directory; username is a string, either the login username or "" for anonymous; password is a string, either the login password or "" for anonymous, in which case what gets sent is "shockfiler@nodomain.com"; localFilePath is a string, the full path to the file to save to or the relative path or "" to display a save file dialog; remoteFileName is a string, the file name to get from the FTP server; progressDialogFlag is a boolean, 1 to display progress bar as file transfers or 0 not to; and progressDialogTitle is a string, the caption for the progress dialog title bar.

Returns 1 if the transfer can start, or a negative error code if there was a problem. Downloads a remote file on an FTP server to the local machine. In authoring and projectors any valid save path can be passed for `localFilePath`. In Shockwave, you must use a relative path to save the file to the `dswMedia` folder or leave the `localFilePath` parameter empty, to prompt the user for the save file path. Passing a full path in Shockwave for the `localfilepath` will error.

Example:

```
err = sf_GetFile("ftp.agency.com","updates","ralphieboy",  
"moonbeam","C:\project\newimages.cst", "weds_images.cst", 1, "Retrieving new  
image castlib...")
```

```
--
```

```
if err < 1 then
```

```
alert "Sorry, could not get the file"
```

```
end if
```

`filePath = sf_SaveFileDialog (defaultFileName)` - where `defaultFileName` is a string, the filename to display in the dialog's filename field. Returns a string containing the save file path or "" if the user did not chose a file. In authoring and in projectors the full file path to the local file is returned. In Shockwave, the path to the local file is stored internally by the xtra and only the file name is returned.

Puts up a standard file save dialog. This function was included to provide a way to prompt the user for a place to save the file when using `sf_GetFile`. You first call the function, save its returned file path into a variable, then pass the variable to `sf_GetFile`:

In authoring and projectors the function operates like a normal save file dialog and can be used any time you want to prompt the user for a local path to save a file. In Shockwave, the save file dialog does not return a filepath and therefore is useful only with `sf_GetFile`. In Shockwave, the dialog returns only the file name chosen and ShockFiler stores the path to the file internally. When you pass the filename returned from the save file dialog to `sf_GetFile`, ShockFiler determines the full path to the file from the filename and saves the file from the FTP server to that path.

In Shockwave the save file dialog function stores the path to the last file chosen by the user. Once the user has chosen a file using the dialog, you can save to the same local file path multiple times by passing the same file name returned from the function to `sf_GetFile`. To send another file, you must prompt the user again. The new chosen file path is stored internally and the old path is erased. Passing "" in the `localFilePath` parameter of `sf_GetFile` will also prompt the user for the local file path to save to, but the path to the file is not saved internally by the dialog after the transfer completes.

Example:

```
usersFileChoice = sf_SaveFileDialog ("newimages.cst" )
```

```
err = sf_GetFile("ftp.agency.com","updates","ralphieboy",  
"moonbeam",usersFileChoice, "weds_images.cst", 1, "Retrieving new image  
castlib...)
```

Online Help

`sf_TransferStatus()` - Returns a Lingo property list containing status information. This function call returns the status of the current file transfer or directory listing retrieval. It returns a Lingo property list in the following format:

```
[ #file: "birds.jpg", #state: "InProgress", #bytesSoFar: 1024 , #bytesTotal: 19782, #error: "None"]
```

where `file` is a string filename of the file currently being transferred; `state` is a string, one of the following: "Connecting", "Started", "InProgress", "Complete" or "Error"; `bytesSoFar` is an integer, the number of bytes transferred so far; `bytesTotal` is an integer, the total size of transfer; and `error` is a string, either "None" or string describing the error.

The commands `sf_Send`, `sf_SendFile`, `sf_GetFile` and `sf_RequestFileList` do their work while allowing the Director movie to continue running. If you want to know whether one of these commands was successful or when they have finished, you must call `sf_TransferStatus` periodically. You can also use this function to update a custom file transfer dialog if you don't want to use the one provided. The best place to call it is from an exitframe handler.

CAUTION: Do not call `sf_TransferStatus()` from within a repeat loop. Lingo repeat loops lock out other processing and will interfere with or completely prevent the file transfer.

JavaScript Note: Use bracket syntax to access the list:

```
if (sf_TransferStatus()["state"] == "Complete") {
```

Example:

```
-- Send text button behavior
```

```
property sending
```

```
on mouseUp me
```

```
host = "ftp.macromedia.com"
```

```
directory = "pub"
```

```
username = "george"
```

```
password = "buckley"
```


Online Help

```
sendMember = "newdata"

sendMembersCastlib = ""

remoteFileName = "results.txt"

remoteFileType = #text

remotefileProperties = [#lineEndings:#win]

doappend = 0

showDialog = 1

dialogTitle = "Sending text file ..."

success = sf_Send (host, directory,username, password,sendMember,
sendMembersCastlib, remoteFileName, remoteFileType, remotefileProperties,
doappend, showDialog,dialogTitle)

if success = 1 then

sending = TRUE

else

sending = FALSE

end if

end

on exitframe me

if sending = TRUE then

-- If a file transfer is going on then check the status

statusList = sf_TransferStatus()

-- Take some action if the transfer is done

if statusList.state = "Complete" then

sending = FALSE

go frame "Transfer complete"
```

Online Help

```
-- Take some action if the transfer errored out

else if statusList.state = "Error" then

sending = FALSE

go frame "Transfer errored out"

-- Loop in this frame while the transfer is in progress

else

go the frame

end if

else

go the frames

end if

end
```

`sf_RequestFileList(host, directory,username, password)` - where host is the address of the FTP host; directory is the path from root to directory; username can be "" for anonymous; and password can be "" for anonymous, in which case what gets sent is "webfiler@nodomain.com". Returns 1 if the request can start, a negative error if there is a problem.

Requests a directory listing for the specified directory on the host. While the list is being transferred Director will continue to animate. You may need to use additional configuration commands before calling this method if you are connecting through a firewall or proxy server.

This command does not itself return the file listing. ShockFiler parses and stores the returned file listing internally in a Lingo list variable, which you must retrieve with `sf_FileListToLingoList` after the transfer of list info has finished. The list will not be available until sf_TransferStatus returns a status of "Complete".

Example:

```
err = sf_RequestFileList ("ftp.adobe.com", "pub","", "")

if err <> 1 then
```

Online Help

```
alert "Invalid parameters or another connection is open"
```

```
end if
```

`sf_FileListToLingoList(listType)` - where `listType` is a symbol, either `#full`, `#filename` or `#raw`. Returns a Lingo list containing file information or [] if there was an error, or no files in directory.

`#filename` returns a linear list containing only the file names of the files in the specified directory, the listing does not include directories.

`#full` returns a property list where each file name is a property whose value is a property list of file attributes.

```
[ "loud.wav" : [#size: 40453, #date: "May 1 1999", #type: #file],
```

```
"pictures" : [#size: 0, #date: "Aug 21 2000", #type: #directory] ]
```

`#raw` returns a linear list of the returned text from a file listing, one line of the listing per string, in order

```
["total 48", "-rw-rw-r-- 1 0 ftpadmin 786 Nov 7 1997 README.ftp" ,"d--x--x--x 2  
0 staff 512 Sep 28 1997 bin"]
```

If the FTP server does not return information required for one or all of the fields ShockFiler will still return a list in the above format but a field with no information returned will appear as follows:

```
#size: 0
```

```
#date: ""
```

```
#type: #invalid
```

Use this function after initiating a listing request with `sf_RequestFileList`. Code example 2 shows how to loop in a frame and check the transfer status while waiting for the list to be returned.

Online Help

JavaScript Note: Use bracket syntax to access the list.

```
theList = sf_FileListToLingoList(symbol("full"));

// [ "loud.wav" : [#size: 40453, #date: "May 1 1999", #type: #file], "pictures" :
[#size: 0, #date: "Aug 21 2000", #type: #directory] ]

soundName = theList.getPropAt(1);

// "load.wav"

soundProps = theList.getProp(soundName);

// [#size: 40453, #date: "May 1 1999", #type: #file]

soundSize = soundProps["size"];

// 40453
```

Example 1:

```
fileList = sf_FileListToLingoList(#filename)

put fileList

-- ["6list.html", "7list.html", "404list.html"]

fileList = sf_FileListToLingoList(#full)

put fileList

-- [{"6list.html":[#size: 6400,#date: "Nov 7 1997",#type: #file], "7list.html":[#size:
3604,#date: "May 28 1999",#type: #file], "404list.html":[#size: 1732,#date: "Jun 1
1997",#type: #file] ]

fileList = sf_FileListToLingoList(#raw)

put fileList

-- ["total 3",

"-rw-rw-r-- 1 0 mike 6400 Nov 7 1997 6list.html" ,
```

Online Help

```
"-rw-rw-r-- 2 0 mike 3604 May 28 1999 7list.html",  
"-rw-rw-r-- 2 0 mike 1732 Jun 1 1997 404list.html" ]
```

Example 2:

```
-- Get listing button behavior
```

```
property connecting
```

```
on mouseUp me
```

```
host = "192.192.192.1"
```

```
directory = "junk"
```

```
username = "anonymous"
```

```
password = "gcm@tiac.net"
```

```
success = sf_RequestFileList (host, directory,username, password)
```

```
if success = 1 then
```

```
connecting = TRUE
```

```
-- Initialize the field that will display the listing
```

```
put "" into field "Listing"
```

```
else
```

```
connecting = FALSE
```

```
end if
```

```
end
```

```
on exitframe me
```

```
-- If not connecting then loop in this frame
```

Online Help

```
if connecting = FALSE then
  go the frame
else
  -- If a connection is in progress then check the status
  statusList = sf_TransferStatus()
  -- If the listing was successfully retrieved then
  -- get it in Lingo list format
  if statusList.state = "Complete" then
    connecting = FALSE
    sending = FALSE
    fileList = sf_FileListToLingoList(#raw)
    numLines = fileList.count
    repeat with x = 1 to numLines
      put fileList[x] & RETURN after field "Listing"
    end repeat
    -- Take some action if the transfer errored out
  else if statusList.state = "Error" then
    connecting = FALSE
    alert "Transfer errored out"
    -- Loop in this frame while waiting for the listing
  else
    go the frame
  end if
end if
end
```

Online Help

`sf_CreateDirectory (host, directory, username, password, directoryNameToCreate)`
- where `host` is the address of FTP host; `directory` is the path from root to directory; `username` can be "" for anonymous; `password` can be "" for anonymous in which case what gets sent is "webfiler@nodomain.com"; and `directoryNameToCreate` is a string, the name of the subdirectory to create in the login directory.

Logs in to the specified directory and creates a sub-directory within it specified by the `directoryNameToCreate` parameter. Use `sf_TransferStatus()` to monitor progress and to determine if the directory was successfully created. Some things that will prevent creation of the directory are:

- The user does not have permission to create directories on the FTP server,
- The directory already exists,
- The login information is incorrect,
- The directory name is illegal for the system the FTP server is running on.

Example:

```
property sending
```

```
on beginsprite me
```

```
sending = FALSE
```

```
end
```

```
on mouseUp me
```

```
sf_CreateDirectory("ftp.server.com", "incoming", "rhonda", "secretword",  
"ProjectFiles")
```

```
sending = TRUE
```

```
end
```

```
on exitframe me
```

```
if sending = FALSE then exit

st = sf_TransferStatus().state

case st of

"Error":

sending = FALSE

alert "Could not create directory"

"Complete":

sending = FALSE

alert "Done creating directory"

end case

end
```

The code above logs user "rhonda" into ftp.server.com and changes from her login directory to subdirectory "incoming". Then it creates subdirectory "ProjectFiles" in directory "incoming". It loops in the exitframe handler until sf_TransferStatus() returns a status of "Error" or "Complete", which means that the operation is over.

sf_Disconnect - ShockFiler automatically disconnects after a successful file transfer or listing transfer. In rare circumstances a problem with the FTP server will cause an operation to never return a Complete status. Since only one transfer can happen at a time, use sf_Disconnect to terminate an unsuccessful operation before attempting a new one.



SHOCKFILER XTRA HELP: SHOCKFILER RELATIVE PATH SYNTAX

ShockFiler accepts a relative path syntax to specify a local file to send or save. In authoring and projectors the path is relative to the movie's path. This is true for all projectors, whether or not the projector safemode property is turned on via Lingo. In Shockwave, the path is relative to folder DSWMEDIA in the Shockwave support folder. ShockFiler relative path's start with @ and use either "\" or ":" to separate directory names.

Examples:

If the path to the movie is:

C:\Project\themovie.dir

this relative path:

@image.jpg

Creates a file at these locations:

Projector

C:\Project\image.jpg

Shockwave

C:\Windows\System\Macromed\Shockwave\DSWMEDIA\image.jpg

If the path to the movie is:

Macintosh HD:Project:themovie.dir

this relative path:

@Pictures:Subfolder:image.jpg

Creates a file at these locations:

Projector

Macintosh HD:Project:Pictures:Subfolder:image.jpg

Shockwave

Macintosh

HD:System:Extensions:Macromedia:Shockwave:DSWMEDIA:Pictures:Subfolder:image.jpg



SHOCKFILER XTRA HELP:
FIREWALL/PROXY

The ShockFiler commands in this section will be most helpful for users who must connect through a firewall or proxy server, but they may be useful in other situations as well. Unless the FTP server you are trying to connect to has unusual requirements, if you are not connecting through a firewall or FTP server you can safely ignore this section and these commands. If you DO have to connect via a firewall and/or proxy server but are not really familiar with how firewalls and proxy servers work, the following Question and Answer section should give you a basic understanding.

WHAT IS A PORT?

An IP address like "ftp.someserver.com" or "192.233.255.1" specifies a particular computer on a network. The computer can be running many different software servers at the same time like web servers, FTP servers or chat servers. A port number is a further breakdown that allows IP traffic to be directed to a particular app on the server. A server app such as an web server, on the computer specifies a port that it "listens" to, and a client app, like a web browser, makes a connection using that port. Without ports, all of the server applications on a PC would have to inspect all of the TCP/IP traffic to look for messages they were interested in. That would not be very efficient.

Port numbers range from 0 to 65535. Port numbers within the range 1-1023 are assigned to certain services as the default agreed-upon port for that service. For instance web servers run on port 80 and FTP servers run on port 21. You can configure server software to run on another port, but most client software like a web browser or FTP client will try to

Online Help

connect on the default port.

WHAT IS A FIREWALL?

A firewall is software that filters TCP/IP traffic to protect a machine from intrusion, or from sending sensitive data out. The software can be running on the computer itself or it can be running on a separate computer. If the firewall is running on another computer, the first computer must make all of its internet connections through the firewall computer to benefit from the firewall.

One way hackers gain access to a system is by using "port scanner" programs to find ports they can connect to on the target machine. A firewall is usually set up to reject incoming port connections except by approved IP addresses and/or programs.

WHAT IS A PROXY SERVER?

A proxy server is a type of firewall set up to handle a particular protocol like all HTTP traffic or FTP traffic.

WHAT IS THE DIFFERENCE BETWEEN ACTIVE AND PASSIVE FTP?

FTP sessions operate over 2 ports on the FTP server and over 2 ports on the FTP client. One port, the control port, carries instructions from the client like "send me a file listing" or "I'm going to send a file" and transmits messages back to the client from the server like "no such directory" or "got the file successfully". The other port, the data port,

Online Help

handles the actual data like the text of the file listing or the actual file data being sent or received.

The client decides whether the FTP session will be active or passive. In an active FTP session, which is the default for most clients, the server connects to a port number on the client to be used for the data port. But firewalls are usually set up to reject incoming port connections, except to pre-established port numbers, because that is one way hackers gain access to a system. In that case the FTP client will not be able to work if it uses an active connection. The only way around it would be to set up the firewall to allow inbound connections to any port, which defeats the purpose of the firewall. Due to this problem, a passive connection is required to successfully FTP out of most firewalls.

In a passive connection, the FTP client initiates the connection to a port on the server to be used for the data port. This makes things safer for the computer the client is running on since it does not have to allow arbitrary incoming connections. Ironically, since passive connections pose security risks for the server similar to the risks posed to the client by active connections, some servers do not accept passive connections.

`sf_PassiveMode (onOrOffFlag)`

`onOrOffFlag`: Boolean, TRUE to turn passive mode on. Default is FALSE, active connection.

Returns: No return

Sets the connection mode to use for the next file transfer or file listing request. Has no effect on a transfer currently under way. This command may be necessary when the user is running behind a firewall.

`sf_PortNumber (portNumber)`

`portNumber`: Integer, FTP server control port number. Default is 21.

Returns: No return.

Sets the port number on the FTP server to connect to on the next file transfer or file listing request. Has no effect on a transfer currently under way. Most normal FTP servers listen to port 21. This command may be necessary to connect to an unusually configured FTP server, but it is mostly used for connecting through an FTP proxy server.

CONNECTING TO A PROXY SERVER

You use a proxy server by connecting to it, rather than the FTP server you actually want to access, and telling the proxy server which remote site you actually want to connect to. The proxy server then connects to the intended FTP server and acts as a go-between.

To accomplish this using ShockFiler you must first use `sf_PortNumber` to change the connection port to the one the proxy server is running on, if it is not 21. Then, in the `sf_Send` or `sf_RequestFileListing` command, you must pass the proxy server's IP address in the host parameter, and username and password parameters in the format required by the proxy server. These two parameters must communicate to the proxy server what ACTUAL FTP server you want to connect to, and possibly the usernames and passwords for both the proxy server and the actual remote FTP server.

The requirements for the user and password fields vary among proxy servers, but following example shows the most common

Online Help

format. In this example you want to send files to the following actual FTP server:

host: ftp.realserver.com

username: james

password: topsecret

On your network, the proxy server listens to port 2020, not port 21, for control commands. The proxy server IP address is 192.241.49.49. The proxy server does not require a user name or password to log on to it. To access ftp.realserver.com through the proxy server, you would do the following:

1. Use sf_PortNumber(2020) to set the server connection port to 2020
2. In either the sf_Send or sf_RequestFileList command you would specify the following for host, user and password:

host: 192.241.49.49

username: james@ftp.realserver.com

password: topsecret

The host IP is the IP address of the proxy server. The username parameter contains the address of the actual server you want to connect to after the @ separator and the password for the actual server before the @ separator. The password field contains the password for the actual FTP server.

Some proxy servers require a user name and password to connect to them, which is

Online Help

completely separate from the ones you would use to connect to the actual FTP server. In that case more information would be needed in the username and password fields. Since the format of the information and the characters used to separate the information in the fields varies between proxy servers, the proxy server manual or system admin is the best source of information as to what will work in those parameters.

In some cases the proxy server requires two logins, one to input the proxy server username and password, and the second to input the remote host address, username and password. ShockFiler Xtra will not work with a proxy server that requires a double-login.



SHOCKFILER XTRA HELP: ERROR TABLE

Error Number	Error
1	Operation was successful
0	Registration failed
-1	Another transfer is in progress
-2	Invalid member
-3	Invalid castlib
-4	Member type does not match export file type
-5	Property invalid for file type or invalid file path
-6	User did not grant permission for this transfer
-7	Memory allocation error
-8	Could not create/open a temp file
-9	File not found
-10	File cannot be read
-11	Sound sample rate or depth not supported
-12	Out of memory
-15	Unspecified error. The most common situation to cause this is no internet connection.



SHOCKFILER XTRA HELP: JAVASCRIPT

XTRAS AND JAVASCRIPT

Director MX 2004 added JavaScript as an alternative scripting language, also available in Director 11. The syntax in the methods doc for ShockFiler Xtra works for both Lingo and JavaScript. For more info, see the technote on [JavaScript and Xtras](#).



SHOCKFILER XTRA HELP: SHOCKWAVE

ShockFiler Xtra can be used in Shockwave: the distribution package provides packaged Xtras that are downloaded automatically to the user's machine, and installed on demand. Please consult Adobe's web site for a complete overview of the Xtras automated download mechanism: read the [Shockwave Xtras downloading overview](#) technote. The basic steps required to make ShockFiler Xtra available for download are outlined below.

To create a Shockwave movie that will auto-download the Xtra to the user's hard drive you must do the following, in this order:

1. Upload the packaged Xtra files to your web server
2. Modify the entry for ShockFiler Xtra in file xtrainfo.txt to point to the packaged Xtra files on your server
3. Do Modify -> Movie -> Xtras, select ShockFiler Xtra, and check the "Download if Needed" option

Once you have completed steps 1 and 2, you can create other Shockwave movies by doing only step 3.

PACKAGED FILES

Your ShockFiler Xtra archive contains a subfolder called "Shockwave". There are four files inside it:

ShockFiler.w32 - Win 32 package

ShockFiler.ppc - Mac Classic package

ShockFiler.carb - Mac Carbon package

ShockFiler.xpku - Mac Universal Binary package

Online Help

All packages contain the ShockFiler Xtra for that platform. Depending on the user's platform, a package autdownloaded to the user's hard drive will install the correct ShockFiler Xtra for the user's platform into their Shockwave support folder.

If, for some reason, you choose not to make your Shockwave movies autdownload the package files, you can have the user install the right Xtra for their platform into the Shockwave support folder manually.

Upload all package files to the same directory on your web server. Use a "binary" or "raw", not "text" transfer. If the packages are uploaded to two different directories, autdownloading will not work. Do not rename the package files.

If you are going to distribute ShockFiler Xtra with Shockwave movies, we recommend that you use your own web server to do so. Packages are available at Tabuleiro's download services, but we reserve the right to refuse access, without notice, to any referring URL that generates excessive traffic.

The ShockFiler Xtra packages included with the download have been signed and packaged by Tabuleiro, and will present the following security message to users of your Shockwave movies when they are installed for the first time:



You may choose to repackage ShockFiler Xtra and sign it with your own Verisign certificate. You might want to do this if you want your own company name to appear in the auto-download dialog box the user sees when an auto-download is initiated. [Adobe](#) is the best source of

information on applying for a Verisign certificate and packaging files.

XTRAINFO.TXT

The text file xtrainfo.txt resides in your Director authoring directory. It contains information about Xtras such as file version names for an Xtra on both platforms and the URL for the packages. The information contained in xtrainfo.txt is saved with each movie you create and used by projectors and Shockwave.

You must create an entry for ShockFiler Xtra in your xtrainfo.txt file that specifies the URL on your server for the package files. The last part of the path will always be "ShockFiler". That specifies the filename of the packages within the directory, without the file extension. Do not include a file extension at the end of the path.

```
[#namePPC:"ShockFiler", #nameW32:"ShockFiler.x32",  
#package:"http://www.domain.com/folder/ShockFiler"]
```

Make sure that the line above does not contain any return character after you paste it into your xtrainfo.txt file. Open your text editor wide and make sure the line does not wrap. If the opening and closing brackets are not on the same line, Director will not be able to create a valid list from the entry and the "Download if needed" button will be dimmed for ShockFiler Xtra in Director.

If you edit xtrainfo.txt while Director is open you should quit and restart Director to read in the changed information in xtrainfo.

EDITING THE MOVIE'S XTRAS LIST

Open the Director movie that you want to save as Shockwave. Choose Modify -> Movie -> Xtras and add ShockFiler. Select ShockFiler from the list and check the "Download if needed" option. Director will initiate an internet connection and look for the packages at the URL you specified in xtrainfo.txt.

Online Help

If Director finds the packages, it will transfer information about the package contents for both platforms such as file names and version numbers and embed the information into your Director movie. An informational dialog box will appear that tells you that the packages for both platforms are "downloading". The packages themselves are not downloading, just information about them that the Shockwave movie will need later to compare the version of the Xtra the user possibly already has to the version currently on the server in order to determine if autodownloading is necessary. The Director movie needs information about both platforms because it may find itself running on either platform once it is on the web.

Once "downloading" of the packages has finished, save the movie, then publish as Shockwave. The finished Shockwave movie can reside at any URL. It does not have to be in the same directory or even on the same server as the packaged Xtras.

If a connection cannot be opened, or the packages cannot be found at the specified location, Director will uncheck the "Download as needed" option automatically. You must have a successful connection for the box to remain checked. A Shockwave made out of a Director movie with the "Download as needed" button unchecked will not autodownload ShockFiler Xtra.



SHOCKFILER XTRA HELP: LIMITATIONS

PROGRESS DIALOG

The progress dialog does not work correctly under all circumstances.

- On Mac, in Shockwave, the progress bar works but the numerical percent to the right of it does not update until the user moves the mouse, and the dialog does not disappear when done immediately, unless the user moves the mouse.
- On Win, all environments, the progress dialog does not show the correct progress for files larger than 21MB.

Unfortunately the workaround for both of these problems is to use `sf_TransferStatus` to get the current byte reading and create your own bar using Lingo.

MULTIPLE TRANSFERS ON WIN

If you are connecting to an FTP server running on WinNT or Win2000 it may be running with a low limit of available incoming socket connections. This limit is specified by the Windows license agreement. For instance it might be 16. In that case, if you run 17 or more transfers in rapid sequence, you will get a failed transfer on number 17. The server log will show an error at that point stating something like all the sockets are in use. One workaround is to specify a passive rather than active connection so the incoming sockets are created on the client side. If you can't do that, the other workaround is to put a looping frame delay between the transfers to allow time for ShockFiler to release the previous socket. This issue is for the Win xtra only.

SHOCKFILER DOES NOT INCLUDE RENAME OR DELETE COMMANDS

However, there is a parent script that implements these commands using Macromedia's MultiUser Xtra, which is x-platform and Shockwave-safe.

INTERNET INFORMATION SERVER (IIS) ISSUES

Listings: ShockFiler Xtra expects to see FTP directory listings in the predominant industry-standard UNIX format. Even Windows and Mac FTP servers generally return their listings in this standard format. The Microsoft IIS server can be configured to return file listings in a different DOS format that looks more like Explorer listings. ShockFiler Xtra will not be able to parse such a list format. In that case, use `sf_FileListToLingoList(#raw)` to retrieve a Lingo list containing just the unparsed text of the returned directory listing.

`sf_GetFile`: ShockFiler always returns "File not found" when retrieving a file from a IIS server if the server is configured to return DOS format listings instead of the default UNIX format listing. The workaround is to change the IIS server setting from DOS to UNIX file listing format.

SHOCKFILER XTRA CANNOT TRANSMIT COMPRESSED MEMBERS

If you are planning to transmit existing members in a Shockwave file, be sure to turn off Publish Settings -> Compression -> Shockwave Audio -> Compression Enabled. This setting does not just apply to Shockwave audio (SWA) members. It will compress all sound members.



SHOCKFILER XTRA HELP: HOW TO ORDER & REGISTER

The unregistered version of ShockFiler Xtra is fully-functional and may be used for evaluation, nonprofit and educational purposes only: commercial distribution is strictly prohibited. A registered version of the Xtra can be used in commercial products, and may be purchased online at xtras.tabuleiro.com, using a secure server. At our web site you can also consult our purchase policy, purchase instructions, payment, delivery and security methods.

If you decide to buy the Xtra you don't need to download a new copy of the software. After your order is processed you will receive an e-mail with a serial number to register the software you've already installed on your machine.

To register the Xtra you should use the `sf_Register()` function, usually called at the startup of your movie, or before a ShockFiler Xtra function is used. More information about specific syntax can be found at the [Methods Documentation](#) page. Please keep your serial number archived for future reference.



SHOCKFILER XTRA HELP: LICENSING & AVAILABILITY

ShockFiler Xtra is a commercial product. Current price and updated information can be found at xtras.tabuleiro.com. If your product provides printed documentation and package we ask you to kindly include the following copyright information:

ShockFiler Xtra(tm) (c) Tabuleiro Prod. Ltda 2008

All Rights Reserved

No royalty-fees are required for a distribution of the Xtra with your product.



SHOCKFILER XTRA HELP:
TECHNICAL SUPPORT

Please use the Your Account section available at our web site xtras.tabuleiro.com to submit your questions. The site also contains Technotes and other resources that can help you identify and solve the most common problems quickly.