

☐ [revIgniter User Guide Version 2.1.7](#)

Search User Guide



1. [revIgniter Home](#)
  2. [User Guide Home](#)
  3. FTP Library
  4. [Table of Contents Page](#)
- [revIgniter](#)
  - [User Guide Home](#)
  - [Table of Contents Page](#)
  - [Basic Info](#)
  - [Server Requirements](#)
  - [License Agreement](#)
  - [Change Log](#)
  - [Credits](#)
  - [Installation](#)
  - [Downloading revIgniter](#)
  - [Installation Instructions](#)
  - [Upgrading from a Previous Version](#)
  - [Troubleshooting](#)
  - [Introduction](#)
  - [Getting Started](#)
  - [revIgniter at a Glance](#)
  - [revIgniter Cheatsheets](#)
  - [Supported Features](#)
  - [Application Flow Chart](#)
  - [Model-View-Controller](#)
  - [HMVC - Extending MVC](#)
  - [Architectural Goals](#)
- 
- [General Topics](#)
  - [revIgniter URLs](#)
  - [Controllers](#)
  - [Reserved Names](#)
  - [Views](#)
  - [Models](#)
  - [Helpers](#)
  - [Using revIgniter Libraries](#)
  - [Creating Your Own Libraries](#)
  - [Creating Core Libraries](#)
  - [Using Stacks](#)
  - [Hooks - Extending the Core](#)
  - [Auto-loading Resources](#)
  - [Common Handlers](#)
  - [Scaffolding](#)

- [URI Routing](#)
  - [Modules](#)
  - [Extensions](#)
  - [Error Handling](#)
  - [Caching](#)
  - [Profiling Your Application](#)
  - [Managing Applications](#)
  - [Security](#)
  - [CLI](#)
  - Tutorial
  - [Simple Chat Application](#)
- 
- Library Reference
  - [Authentication Library](#)
  - [Benchmarking Library](#)
  - [Calendaring Library](#)
  - [Captcha Library](#)
  - [Config Library](#)
  - [Database Library](#)
  - [Email Library](#)
  - [Encryption Library](#)
  - [File Uploading Library](#)
  - [Form Validation Library](#)
  - [FTP Library](#)
  - [HTML Table Library](#)
  - [Image Manipulation Library](#)
  - [Input and Security Library](#)
  - [jQuery Library](#)
  - [Language Library](#)
  - [Loader Library](#)
  - [Modules Library](#)
  - [Output Library](#)
  - [Pagination Library](#)
  - [Session Library](#)
  - [Trackback Library](#)
  - [URI Library](#)
  - [User Agent Library](#)
- 
- Helper Reference
  - [Array Helper](#)
  - [Asset Helper](#)
  - [Cookie Helper](#)
  - [Date Helper](#)
  - [Download Helper](#)
  - [Email Helper](#)
  - [File Helper](#)
  - [Form Helper](#)

- [Form-Mail Helper](#)
- [Galleria Helper](#)
- [HTML Helper](#)
- [JWT Helper](#)
- [Language Helper](#)
- [Markdown Helper](#)
- [OTP Helper](#)
- [QueryToJSON Helper](#)
- [QueryValues Helper](#)
- [SiteLinks Helper](#)
- [String Helper](#)
- [URL Helper](#)
- [XML Helper](#)

Table of Contents

Close

# FTP Library

revIgniter's FTP Library permits to:

- get directory listings
- transfer files to a remote server
- download files from a remote server
- move, rename and delete remote files / folders
- create a directory on a remote server
- set file / folder permissions remotely
- recreate an entire local directory remotely using a "mirroring" handler

**Note:** This library requires the tsNet LiveCode external. Asynchronous SFTP and SFTP using public key authentication need LC Business Edition.

**Note:** Asynchronous requests are currently (up to LC server version 9.5.1) not working, presumably this is due to a bug in tsNet, [see bug 18961](#).

## Initializing the Library

The FTP Library is initialized in your controller like most other libraries in revIgniter using the *rigLoaderLoadLibrary* handler:

```
rigLoaderLoadLibrary "Ftp"
```

## Setting FTP Preferences

### Setting Preferences Manually

Preferences are set by passing an array of preference values to the ftp *rigInitializeFTP* handler. Following an example of how you might set ftp preferences:

[Copy](#)

```
put "ftp.example.com/" into tFtpConfA["hostname"]
put "me" into tFtpConfA["username"]
put "myPassword" into tFtpConfA["password"]
put TRUE into tFtpConfA["asyncRequest"]

rigInitializeFTP tFtpConfA
```

**Note:** Many preferences have default values that will be used if you do not set them.

## Setting Preferences in a Config File

If you prefer not to set preferences using the above method, you can instead store them in a config file. Simply create a new file called *ftp.lc* and add the *Config* array in that file. At the end of the file write: *rigRunInitialFtpConfig yourArrayVariableName*. Then save the file at *config/ftp.lc* and it will be used automatically. You will NOT need to use the *rigInitializeFTP* handler if you save your preferences in a config file. Example:

[Copy](#)

```
local sFtpConfA

put "ftp.example.com/" into sFtpConfA["hostname"]
put "me" into sFtpConfA["username"]
put "myPassword" into sFtpConfA["password"]
put TRUE into sFtpConfA["asyncRequest"]

rigRunInitialFtpConfig sFtpConfA
```

## Preferences

The following preferences are available. The default value indicates what will be used if you do not specify that preference.

Preference	Default Value	Options	Description
<b>hostname</b>	No Default	None	FTP hostname, can be a hostname or an IP address (usually something like: ftp.example.com)
<b>username</b>	No Default	None	FTP username
<b>password</b>	No Default	None	FTP password
<b>port</b>	21	Any valid port number	FTP server port number

<b>mode</b>	passive	passive, active	FTP mode to be used
<b>debug</b>	false	false, true	Whether to enable debugging to display status and error messages
<b>asyncRequest</b>	false	false, true	Whether to send asynchronous FTP requests
<b>displayErrors</b>	true	false, true	Whether to display logged error messages, if this is set to true the debug setting is ignored
<b>flavor</b>	ftp	ftp, sftp, ftps	Which protocol to use
<b>use_ssl</b>	false	false, true	Whether to connect using Transport Layer Security (TLS) for FTPS (explicit) connections
<b>verifySSL</b>	false	false, true	Whether to enable SSL peer certificate verification
<b>no_reuse</b>	false	false, true	Specify whether the connection to the server should be disconnected and not left open for any future connections
<b>ssh_passphrase</b>	No Default	None	(LC Business Edition only) Passphrase for any included private key
<b>ssh_host_public_key</b>	No Default	None	(LC Business Edition only) The 128 bit MD5 checksum of the remote host's public key
<b>ssh_private_key</b>	No Default	None	(LC Business Edition only) The private key file (including full system path) to be used in SFTP transfers

## Usage Examples

The following example shows how to upload a local file to the FTP server. The file permissions are set to 755.

Copy

```
rigLoaderLoadLibrary "Ftp"

put "ftp.example.com/" into tFtpConfA["hostname"]
put "me" into tFtpConfA["username"]
put "myPassword" into tFtpConfA["password"]

rigInitializeFTP tFtpConfA

put "/path/to/local/file.pdf" into tLocPath
put "remotefolder/remotefile.pdf" into tRemPath

rigFtpUpload tLocPath, tRemPath, FALSE, TRUE, 0755

put the result into gData["responseData"]
```

This example shows how to retrieve a list of files from the server.

[Copy](#)

```
rigLoaderLoadLibrary "Ftp"

put "ftp.example.com/" into tFtpConfA["hostname"]
put "me" into tFtpConfA["username"]
put "myPassword" into tFtpConfA["password"]

rigInitializeFTP tFtpConfA

rigFtpListFiles "remote/path"

put the result into gData["responseData"]
```

This example shows how to mirror a local directory on the server.

[Copy](#)

```
rigLoaderLoadLibrary "Ftp"

put "ftp.example.com/" into tFtpConfA["hostname"]
put "me" into tFtpConfA["username"]
put "myPassword" into tFtpConfA["password"]

rigInitializeFTP tFtpConfA

put "/local/directory" into tLocPath
put "remote/mirroredDirectory" into tRemPath

rigFtpMirror tLocPath, tRemPath

put the result into gData["responseData"]
```

## Handler Reference

### **rigFtpPrintDebugger( *pWhichMsg* )**

Get FTP debug messages. Don't forget to set the debug preference to TRUE.

#### *Parameters*

**pWhichMsg:** You can optionally choose between "status" and "debug", if you leave this parameter blank you get both types of messages

Example:

```
put rigFtpPrintDebugger() into gData["debugData"]
```

Add the following to your view file:

```
[[gData["debugData"]]]
```

## **rigFtpUpload** *pLocalPath, pRemotePath, pData, pCreateMissingDirs, pPermissions*

Uploads a file to a FTP server. You must at least supply the local path and the remote path.

### *Parameters*

**pLocalPath:** The path to your local file

**pRemotePath:** The path to the file on your server

**pData:** A boolean to specify that you intend to upload data like for example a string instead of a file. This parameter is optional, the default value is FALSE

**pCreateMissingDirs:** An optional boolean to tell the handler that the remote path includes directories which are missing and thus need to be created prior to uploading, the default value is FALSE

**pPermissions:** This optional parameter should be an octal number and is used to set permissions, the default value is 0700

This handler returns as result either FALSE in case of failure or on success "Transfer complete. Transferred xxxxxx bytes."

Example:

```
put TRUE into tData
put "some data to upload as text" into tLocPath
put "remote/path/dat.txt" into tRemPath
```

```
rigFtpUpload tLocPath, tRemPath, tData
```

```
put the result into gData["responseData"]
```

## **rigFtpDownload** *pRemotePath, pLocalPath*

Downloads a file from a remote server to the local server. You must supply the remote path and the local path.

### *Parameters*

**pRemotePath:** The path to the file on your server

**pLocalPath:** Local file path

This handler returns as result either FALSE in case of failure or on success "Transfer complete. Transferred xxxxxx bytes."

Example:

```
put "remote/path/remoteFile.pdf" into tRemPath
put "local/path/local.pdf" into tLocPath
```

```
rigFtpDownload tRemPath, tLocPath
```

```
put the result into gData["responseData"]
```

**Note:** *rigFtpDownload* using SFTP needs LC Business Edition.

## ***rigFtpRenameMove*** *pOldFile, pNewFile, pMissingDirs*

Permits you to rename or move a file. Supply the source file name/path and the new file name/path.

### *Parameters*

**pOldFile:** Old file name/path

**pNewFile:** New file name/path

**pMissingDirs:** An optional boolean to tell the handler that the new path includes directories which are missing and thus need to be created, the default value is FALSE

This handler returns as result either FALSE in case of failure or on success something like "Status: 250 Result: 220 ..."

Example:

```
put "old/path/oldFileName.pdf" into tOldFilePath
put "new/directory/newFileName.pdf" into tNewFilePath

rigFtpRenameMove tOldFilePath, tNewFilePath, TRUE

put the result into gData["responseData"]
```

## ***rigFtpDeleteFile*** *pPath*

Lets you delete a file. Supply the source path with the file name.

### *Parameters*

**pPath:** The file path

This handler returns as result either FALSE in case of failure or on success something like "Status: 250 Result: 220 ..."

Example:

```
put "path/to/file.pdf" into tFilePath

rigFtpDeleteFile tFilePath

put the result into gData["responseData"]
```

## ***rigFtpDeleteFolder*** *pPath, pRecursive, pPurgeOnly*

Lets you delete a folder and / or sub-folders and everything a folder contains. Supply at least the source path to the directory.

### *Parameters*



**pPath:** The folder path

**pRecursive:** This optional boolean specifies if the folder should be deleted recursively (including all sub-folders), the default value is FALSE

**pPurgeOnly:** Use this optional boolean to specify that folders should be purged only and not deleted, the default value is FALSE

This handler returns as result either FALSE in case of failure or TRUE on success.

Example:

```
put "path/to/folder" into tFolder
rigFtpDeleteFolder tFolder, ,TRUE
put the result into gData["responseData"]
```

**Note:** SFTP using public key authentication needs LC Business Edition.

**Note:** Be very careful with this handler, make absolutely sure your folder path is correct. Try to verify that your path is correct using the *rigFtpListFiles* handler.

### **rigFtpListFiles** *pPath, pSimpleList*

Permits you to retrieve a directory listing. You must supply the path to the desired directory.

#### *Parameters*

**pPath:** The folder path

**pSimpleList:** This optional boolean lets you enable "simple" directory listings over FTP and SFTP, the default value is FALSE

This handler returns as result either FALSE in case of failure, otherwise on success the files list.

Example:

```
put "path/to/folder" into tFolderPath
rigFtpListFiles tFolderPath
put the result into gData["responseData"]
```

### **rigFtpMirror** *pLocalPath, pRemotePath*

Recursively reads a local folder including files and sub-folders and creates a mirror on the remote server. You must supply a source path and a destination path.

#### *Parameters*

**pLocalPath:** The local path

**pRemotePath:** The remote path

This handler returns as result either FALSE in case of failure or on success "Transfer complete. Transferred xxxxxx bytes."

Example:

```
put "path/to/localFolder" into tLocalFolder
put "path/to/remoteFolder" into tRemoteFolder

rigFtpMirror tLocalFolder, tRemoteFolder

put the result into gData["responseData"]
```

**Note:** *rigFtpMirror* using SFTP needs LC Business Edition.

### ***rigFtpCreateFolder pPath, pPermissions***

Lets you create a directory on a remote server. Supply the path ending in the folder name you wish to create. Optionally permissions can be set by passing an octal number in the second parameter.

#### *Parameters*

**pPath:** The path to the directory to create

**pPermissions:** The folder permissions (octal), this parameter is optional

This handler returns as result either FALSE in case of failure or TRUE on success.

Example:

```
put "path/to/newFolder" into tNewFolder
put 0755 into tPermissions

rigFtpCreateFolder tNewFolder, tPermissions

put the result into gData["responseData"]
```

### ***rigFtpSetPermissions pPath, pPermissions***

Permits you to set file / folder permissions. Supply the path to the file / folder you wish to alter permissions on.

#### *Parameters*

**pPath:** The path to alter permissions for

**pPermissions:** The file / folder permissions (octal)

This handler returns as result either FALSE in case of failure or TRUE on success.

Example:

```
put "path/to/file.pdf" into tFilePath
put 0755 into tPermissions
```

```
rigFtpSetPermissions tFilePath, tPermissions  
put the result into gData["responseData"]
```

## **rigFtpChangeDir** *pPath*

Changes the current working directory to the specified path, FTP and FTPS only.

### *Parameters*

**pPath:** The directory path

This handler returns as result either FALSE in case of failure or TRUE on success.

Example:

```
put "path/to/folder" into tFolderPath  
rigFtpChangeDir tFolderPath  
put the result into gData["responseData"]
```

## **rigFtpClose**

Lets you remove the current connection from memory. The handler returns as result either FALSE in case of failure or TRUE on success.

Example:

```
rigFtpClose  
put the result into gData["responseData"]
```

Previous Topic: [Form Validation Library](#) :: [Top of Page](#) :: [User Guide Home](#) :: Next Topic: [HTML Table Library](#)

[revIgniter](#) :: Copyright © 2009 - 2020 :: dimensionB Bitter u. Bitter GmbH