SFML

* [Main Page](http://docs.google.com/index.htm)
* [Modules](http://docs.google.com/modules.htm)
* [Classes](http://docs.google.com/annotated.htm)
* [Files](http://docs.google.com/files.htm)
* [Class List](http://docs.google.com/annotated.htm)
* [Class Index](http://docs.google.com/classes.htm)
* [Class Hierarchy](http://docs.google.com/hierarchy.htm)
* [Class Members](http://docs.google.com/functions.htm)
* **sf**
* [Ftp](http://docs.google.com/classsf_1_1Ftp.htm)

[Classes](#_gjdgxs) | [Public Types](#_30j0zll) | [Public Member Functions](#_1fob9te) | [Friends](#_3znysh7) | [List of all members](http://docs.google.com/classsf_1_1Ftp-members.htm)

sf::Ftp Class Reference

[Network module](http://docs.google.com/group__network.htm)

A FTP client. [More...](http://docs.google.com/classsf_1_1Ftp.htm#details)

#include <[Ftp.hpp](http://docs.google.com/Ftp_8hpp_source.htm)>

Inheritance diagram for sf::Ftp:



| Classes | |
| --- | --- |
| class | [DirectoryResponse](http://docs.google.com/classsf_1_1Ftp_1_1DirectoryResponse.htm) |
|  | Specialization of FTP response returning a directory. [More...](http://docs.google.com/classsf_1_1Ftp_1_1DirectoryResponse.htm#details) |
|  | |
| class | [ListingResponse](http://docs.google.com/classsf_1_1Ftp_1_1ListingResponse.htm) |
|  | Specialization of FTP response returning a filename lisiting. [More...](http://docs.google.com/classsf_1_1Ftp_1_1ListingResponse.htm#details) |
|  | |
| class | [Response](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm) |
|  | Define a FTP response. [More...](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#details) |
|  | |

| Public Types | |
| --- | --- |
| enum | [TransferMode](http://docs.google.com/classsf_1_1Ftp.htm#a1cd6b89ad23253f6d97e6d4ca4d558cb) {  [Binary](http://docs.google.com/classsf_1_1Ftp.htm#a1cd6b89ad23253f6d97e6d4ca4d558cba6f253b362639fb5e059dc292762a21ee),  [Ascii](http://docs.google.com/classsf_1_1Ftp.htm#a1cd6b89ad23253f6d97e6d4ca4d558cbac9e544a22dce8ef3177449cb235d15c2),  [Ebcdic](http://docs.google.com/classsf_1_1Ftp.htm#a1cd6b89ad23253f6d97e6d4ca4d558cbabb1e34435231e73c96534c71090be7f4)  } |
|  | Enumeration of transfer modes. [More...](http://docs.google.com/classsf_1_1Ftp.htm#a1cd6b89ad23253f6d97e6d4ca4d558cb) |
|  | |

| Public Member Functions | |
| --- | --- |
|  | [~Ftp](http://docs.google.com/classsf_1_1Ftp.htm#a2edfa8e9009caf27bce74459ae76dc52) () |
|  | Destructor. |
|  | |
| [Response](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm) | [connect](http://docs.google.com/classsf_1_1Ftp.htm#af02fb3de3f450a50a27981961c69c860) (const [IpAddress](http://docs.google.com/classsf_1_1IpAddress.htm) &server, unsigned short port=21, [Time](http://docs.google.com/classsf_1_1Time.htm) timeout=[Time::Zero](http://docs.google.com/classsf_1_1Time.htm#a8db127b632fa8da21550e7282af11fa0)) |
|  | Connect to the specified FTP server. |
|  | |
| [Response](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm) | [disconnect](http://docs.google.com/classsf_1_1Ftp.htm#acf7459926f3391cd06bf84337ed6a0f4) () |
|  | Close the connection with the server. |
|  | |
| [Response](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm) | [login](http://docs.google.com/classsf_1_1Ftp.htm#a686262bc377584cd50e52e1576aa3a9b) () |
|  | Log in using an anonymous account. |
|  | |
| [Response](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm) | [login](http://docs.google.com/classsf_1_1Ftp.htm#a99d8114793c1659e9d51d45cecdcd965) (const std::string &name, const std::string &password) |
|  | Log in using a username and a password. |
|  | |
| [Response](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm) | [keepAlive](http://docs.google.com/classsf_1_1Ftp.htm#aa1127d442b4acb2105aa8060a39d04fc) () |
|  | Send a null command to keep the connection alive. |
|  | |
| [DirectoryResponse](http://docs.google.com/classsf_1_1Ftp_1_1DirectoryResponse.htm) | [getWorkingDirectory](http://docs.google.com/classsf_1_1Ftp.htm#a79c654fcdd0c81e68c4fa29af3b45e0c) () |
|  | Get the current working directory. |
|  | |
| [ListingResponse](http://docs.google.com/classsf_1_1Ftp_1_1ListingResponse.htm) | [getDirectoryListing](http://docs.google.com/classsf_1_1Ftp.htm#a8f37258e461fcb9e2a0655e9df0be4a0) (const std::string &directory="") |
|  | Get the contents of the given directory. |
|  | |
| [Response](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm) | [changeDirectory](http://docs.google.com/classsf_1_1Ftp.htm#a7e93488ea6330dd4dd76e428da9bb6d3) (const std::string &directory) |
|  | Change the current working directory. |
|  | |
| [Response](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm) | [parentDirectory](http://docs.google.com/classsf_1_1Ftp.htm#ad295cf77f30f9ad07b5c401fd9849189) () |
|  | Go to the parent directory of the current one. |
|  | |
| [Response](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm) | [createDirectory](http://docs.google.com/classsf_1_1Ftp.htm#a247b84c4b25da37804218c2b748c4787) (const std::string &name) |
|  | Create a new directory. |
|  | |
| [Response](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm) | [deleteDirectory](http://docs.google.com/classsf_1_1Ftp.htm#a2a8a7ef9144204b5b319c9a4be8806c2) (const std::string &name) |
|  | Remove an existing directory. |
|  | |
| [Response](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm) | [renameFile](http://docs.google.com/classsf_1_1Ftp.htm#a8f99251d7153e1dc26723e4006deb764) (const std::string &file, const std::string &newName) |
|  | Rename an existing file. |
|  | |
| [Response](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm) | [deleteFile](http://docs.google.com/classsf_1_1Ftp.htm#a8aa272b0eb7769a850006e70fcad370f) (const std::string &name) |
|  | Remove an existing file. |
|  | |
| [Response](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm) | [download](http://docs.google.com/classsf_1_1Ftp.htm#a20c1600ec5fd6f5a2ad1429ab8aa5df4) (const std::string &remoteFile, const std::string &localPath, [TransferMode](http://docs.google.com/classsf_1_1Ftp.htm#a1cd6b89ad23253f6d97e6d4ca4d558cb) mode=[Binary](http://docs.google.com/classsf_1_1Ftp.htm#a1cd6b89ad23253f6d97e6d4ca4d558cba6f253b362639fb5e059dc292762a21ee)) |
|  | Download a file from the server. |
|  | |
| [Response](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm) | [upload](http://docs.google.com/classsf_1_1Ftp.htm#a46d6e15cddd719288b5a08b685e11765) (const std::string &localFile, const std::string &remotePath, [TransferMode](http://docs.google.com/classsf_1_1Ftp.htm#a1cd6b89ad23253f6d97e6d4ca4d558cb) mode=[Binary](http://docs.google.com/classsf_1_1Ftp.htm#a1cd6b89ad23253f6d97e6d4ca4d558cba6f253b362639fb5e059dc292762a21ee)) |
|  | Upload a file to the server. |
|  | |

| Friends | |
| --- | --- |
| class | **DataChannel** |
|  | |

## Detailed Description

A FTP client.

[sf::Ftp](http://docs.google.com/classsf_1_1Ftp.htm) is a very simple FTP client that allows you to communicate with a FTP server.

The FTP protocol allows you to manipulate a remote file system (list files, upload, download, create, remove, ...).

Using the FTP client consists of 4 parts:

* Connecting to the FTP server
* Logging in (either as a registered user or anonymously)
* Sending commands to the server
* Disconnecting (this part can be done implicitely by the destructor)

Every command returns a FTP response, which contains the status code as well as a message from the server. Some commands such as getWorkingDirectory and getDirectoryListing return additional data, and use a class derived from [sf::Ftp::Response](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm) to provide this data.

All commands, especially upload and download, may take some time to complete. This is important to know if you don't want to block your application while the server is completing the task.

Usage example:

// Create a new FTP client

[sf::Ftp](http://docs.google.com/classsf_1_1Ftp.htm) ftp;

// Connect to the server

[sf::Ftp::Response](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm) response = ftp.[connect](http://docs.google.com/classsf_1_1Ftp.htm#af02fb3de3f450a50a27981961c69c860)("ftp://ftp.myserver.com");

if (response.[isOk](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#a4dadbe0fe0a3ef2d571a017e1645e675)())

std::cout << "Connected" << std::endl;

// Log in

response = ftp.[login](http://docs.google.com/classsf_1_1Ftp.htm#a686262bc377584cd50e52e1576aa3a9b)("laurent", "dF6Zm89D");

if (response.[isOk](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#a4dadbe0fe0a3ef2d571a017e1645e675)())

std::cout << "Logged in" << std::endl;

// Print the working directory

[sf::Ftp::DirectoryResponse](http://docs.google.com/classsf_1_1Ftp_1_1DirectoryResponse.htm) directory = ftp.[getWorkingDirectory](http://docs.google.com/classsf_1_1Ftp.htm#a79c654fcdd0c81e68c4fa29af3b45e0c)();

if (directory.[isOk](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#a4dadbe0fe0a3ef2d571a017e1645e675)())

std::cout << "Working directory: " << directory.[getDirectory](http://docs.google.com/classsf_1_1Ftp_1_1DirectoryResponse.htm#a500793778ad0ed223aa86ed8fbee28a3)() << std::endl;

// Create a new directory

response = ftp.[createDirectory](http://docs.google.com/classsf_1_1Ftp.htm#a247b84c4b25da37804218c2b748c4787)("files");

if (response.[isOk](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#a4dadbe0fe0a3ef2d571a017e1645e675)())

std::cout << "Created new directory" << std::endl;

// Upload a file to this new directory

response = ftp.[upload](http://docs.google.com/classsf_1_1Ftp.htm#a46d6e15cddd719288b5a08b685e11765)("local-path/file.txt", "files", [sf::Ftp::Ascii](http://docs.google.com/classsf_1_1Ftp.htm#a1cd6b89ad23253f6d97e6d4ca4d558cbac9e544a22dce8ef3177449cb235d15c2));

if (response.[isOk](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#a4dadbe0fe0a3ef2d571a017e1645e675)())

std::cout << "File uploaded" << std::endl;

// Disconnect from the server (optional)

ftp.[disconnect](http://docs.google.com/classsf_1_1Ftp.htm#acf7459926f3391cd06bf84337ed6a0f4)();

Definition at line [47](http://docs.google.com/Ftp_8hpp_source.htm#l00047) of file [Ftp.hpp](http://docs.google.com/Ftp_8hpp_source.htm).

## Member Enumeration Documentation

| enum [sf::Ftp::TransferMode](http://docs.google.com/classsf_1_1Ftp.htm#a1cd6b89ad23253f6d97e6d4ca4d558cb) |
| --- |

Enumeration of transfer modes.

**Enumerator:**

| *Binary* | Binary mode (file is transfered as a sequence of bytes) |
| --- | --- |
| *Ascii* | [Text](http://docs.google.com/classsf_1_1Text.htm) mode using ASCII encoding. |
| *Ebcdic* | [Text](http://docs.google.com/classsf_1_1Text.htm) mode using EBCDIC encoding. |

Definition at line [55](http://docs.google.com/Ftp_8hpp_source.htm#l00055) of file [Ftp.hpp](http://docs.google.com/Ftp_8hpp_source.htm).

## Constructor & Destructor Documentation

| sf::Ftp::~Ftp | ( |  | ) |  |
| --- | --- | --- | --- | --- |

Destructor.

Automatically closes the connection with the server if it is still opened.

## Member Function Documentation

| [Response](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm) sf::Ftp::changeDirectory | ( | const std::string & | *directory* | ) |  |
| --- | --- | --- | --- | --- | --- |

Change the current working directory.

The new directory must be relative to the current one.

Parameters

| directory | New working directory |
| --- | --- |

ReturnsServer response to the request See Also[getWorkingDirectory](http://docs.google.com/classsf_1_1Ftp.htm#a79c654fcdd0c81e68c4fa29af3b45e0c), [getDirectoryListing](http://docs.google.com/classsf_1_1Ftp.htm#a8f37258e461fcb9e2a0655e9df0be4a0), [parentDirectory](http://docs.google.com/classsf_1_1Ftp.htm#ad295cf77f30f9ad07b5c401fd9849189)

| [Response](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm) sf::Ftp::connect | ( | const [IpAddress](http://docs.google.com/classsf_1_1IpAddress.htm) & | *server*, |
| --- | --- | --- | --- |
|  |  | unsigned short | *port* = 21, |
|  |  | [Time](http://docs.google.com/classsf_1_1Time.htm) | *timeout* = [Time::Zero](http://docs.google.com/classsf_1_1Time.htm#a8db127b632fa8da21550e7282af11fa0) |
|  | ) |  |  |

Connect to the specified FTP server.

The port has a default value of 21, which is the standard port used by the FTP protocol. You shouldn't use a different value, unless you really know what you do. This function tries to connect to the server so it may take a while to complete, especially if the server is not reachable. To avoid blocking your application for too long, you can use a timeout. The default value, [Time::Zero](http://docs.google.com/classsf_1_1Time.htm#a8db127b632fa8da21550e7282af11fa0), means that the system timeout will be used (which is usually pretty long).

Parameters

| server | Name or address of the FTP server to connect to |
| --- | --- |
| port | Port used for the connection |
| timeout | Maximum time to wait |

ReturnsServer response to the request See Also[disconnect](http://docs.google.com/classsf_1_1Ftp.htm#acf7459926f3391cd06bf84337ed6a0f4)

| [Response](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm) sf::Ftp::createDirectory | ( | const std::string & | *name* | ) |  |
| --- | --- | --- | --- | --- | --- |

Create a new directory.

The new directory is created as a child of the current working directory.

Parameters

| name | Name of the directory to create |
| --- | --- |

ReturnsServer response to the request See Also[deleteDirectory](http://docs.google.com/classsf_1_1Ftp.htm#a2a8a7ef9144204b5b319c9a4be8806c2)

| [Response](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm) sf::Ftp::deleteDirectory | ( | const std::string & | *name* | ) |  |
| --- | --- | --- | --- | --- | --- |

Remove an existing directory.

The directory to remove must be relative to the current working directory. Use this function with caution, the directory will be removed permanently!

Parameters

| name | Name of the directory to remove |
| --- | --- |

ReturnsServer response to the request See Also[createDirectory](http://docs.google.com/classsf_1_1Ftp.htm#a247b84c4b25da37804218c2b748c4787)

| [Response](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm) sf::Ftp::deleteFile | ( | const std::string & | *name* | ) |  |
| --- | --- | --- | --- | --- | --- |

Remove an existing file.

The file name must be relative to the current working directory. Use this function with caution, the file will be removed permanently!

Parameters

| name | File to remove |
| --- | --- |

ReturnsServer response to the request See Also[renameFile](http://docs.google.com/classsf_1_1Ftp.htm#a8f99251d7153e1dc26723e4006deb764)

| [Response](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm) sf::Ftp::disconnect | ( |  | ) |  |
| --- | --- | --- | --- | --- |

Close the connection with the server.

ReturnsServer response to the request See Also[connect](http://docs.google.com/classsf_1_1Ftp.htm#af02fb3de3f450a50a27981961c69c860)

| [Response](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm) sf::Ftp::download | ( | const std::string & | *remoteFile*, |
| --- | --- | --- | --- |
|  |  | const std::string & | *localPath*, |
|  |  | [TransferMode](http://docs.google.com/classsf_1_1Ftp.htm#a1cd6b89ad23253f6d97e6d4ca4d558cb) | *mode* = [Binary](http://docs.google.com/classsf_1_1Ftp.htm#a1cd6b89ad23253f6d97e6d4ca4d558cba6f253b362639fb5e059dc292762a21ee) |
|  | ) |  |  |

Download a file from the server.

The filename of the distant file is relative to the current working directory of the server, and the local destination path is relative to the current directory of your application.

Parameters

| remoteFile | Filename of the distant file to download |
| --- | --- |
| localPath | Where to put to file on the local computer |
| mode | Transfer mode |

ReturnsServer response to the request See Also[upload](http://docs.google.com/classsf_1_1Ftp.htm#a46d6e15cddd719288b5a08b685e11765)

| [ListingResponse](http://docs.google.com/classsf_1_1Ftp_1_1ListingResponse.htm) sf::Ftp::getDirectoryListing | ( | const std::string & | *directory* = "" | ) |  |
| --- | --- | --- | --- | --- | --- |

Get the contents of the given directory.

This function retrieves the sub-directories and files contained in the given directory. It is not recursive. The *directory* parameter is relative to the current working directory.

Parameters

| directory | Directory to list |
| --- | --- |

ReturnsServer response to the request See Also[getWorkingDirectory](http://docs.google.com/classsf_1_1Ftp.htm#a79c654fcdd0c81e68c4fa29af3b45e0c), [changeDirectory](http://docs.google.com/classsf_1_1Ftp.htm#a7e93488ea6330dd4dd76e428da9bb6d3), [parentDirectory](http://docs.google.com/classsf_1_1Ftp.htm#ad295cf77f30f9ad07b5c401fd9849189)

| [DirectoryResponse](http://docs.google.com/classsf_1_1Ftp_1_1DirectoryResponse.htm) sf::Ftp::getWorkingDirectory | ( |  | ) |  |
| --- | --- | --- | --- | --- |

Get the current working directory.

The working directory is the root path for subsequent operations involving directories and/or filenames.

ReturnsServer response to the request See Also[getDirectoryListing](http://docs.google.com/classsf_1_1Ftp.htm#a8f37258e461fcb9e2a0655e9df0be4a0), [changeDirectory](http://docs.google.com/classsf_1_1Ftp.htm#a7e93488ea6330dd4dd76e428da9bb6d3), [parentDirectory](http://docs.google.com/classsf_1_1Ftp.htm#ad295cf77f30f9ad07b5c401fd9849189)

| [Response](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm) sf::Ftp::keepAlive | ( |  | ) |  |
| --- | --- | --- | --- | --- |

Send a null command to keep the connection alive.

This command is useful because the server may close the connection automatically if no command is sent.

ReturnsServer response to the request

| [Response](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm) sf::Ftp::login | ( |  | ) |  |
| --- | --- | --- | --- | --- |

Log in using an anonymous account.

Logging in is mandatory after connecting to the server. Users that are not logged in cannot perform any operation.

ReturnsServer response to the request

| [Response](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm) sf::Ftp::login | ( | const std::string & | *name*, |
| --- | --- | --- | --- |
|  |  | const std::string & | *password* |
|  | ) |  |  |

Log in using a username and a password.

Logging in is mandatory after connecting to the server. Users that are not logged in cannot perform any operation.

Parameters

| name | User name |
| --- | --- |
| password | Password |

ReturnsServer response to the request

| [Response](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm) sf::Ftp::parentDirectory | ( |  | ) |  |
| --- | --- | --- | --- | --- |

Go to the parent directory of the current one.

ReturnsServer response to the request See Also[getWorkingDirectory](http://docs.google.com/classsf_1_1Ftp.htm#a79c654fcdd0c81e68c4fa29af3b45e0c), [getDirectoryListing](http://docs.google.com/classsf_1_1Ftp.htm#a8f37258e461fcb9e2a0655e9df0be4a0), [changeDirectory](http://docs.google.com/classsf_1_1Ftp.htm#a7e93488ea6330dd4dd76e428da9bb6d3)

| [Response](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm) sf::Ftp::renameFile | ( | const std::string & | *file*, |
| --- | --- | --- | --- |
|  |  | const std::string & | *newName* |
|  | ) |  |  |

Rename an existing file.

The filenames must be relative to the current working directory.

Parameters

| file | File to rename |
| --- | --- |
| newName | New name of the file |

ReturnsServer response to the request See Also[deleteFile](http://docs.google.com/classsf_1_1Ftp.htm#a8aa272b0eb7769a850006e70fcad370f)

| [Response](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm) sf::Ftp::upload | ( | const std::string & | *localFile*, |
| --- | --- | --- | --- |
|  |  | const std::string & | *remotePath*, |
|  |  | [TransferMode](http://docs.google.com/classsf_1_1Ftp.htm#a1cd6b89ad23253f6d97e6d4ca4d558cb) | *mode* = [Binary](http://docs.google.com/classsf_1_1Ftp.htm#a1cd6b89ad23253f6d97e6d4ca4d558cba6f253b362639fb5e059dc292762a21ee) |
|  | ) |  |  |

Upload a file to the server.

The name of the local file is relative to the current working directory of your application, and the remote path is relative to the current directory of the FTP server.

Parameters

| localFile | Path of the local file to upload |
| --- | --- |
| remotePath | Where to put to file on the server |
| mode | Transfer mode |

ReturnsServer response to the request See Also[download](http://docs.google.com/classsf_1_1Ftp.htm#a20c1600ec5fd6f5a2ad1429ab8aa5df4)

The documentation for this class was generated from the following file:

* [Ftp.hpp](http://docs.google.com/Ftp_8hpp_source.htm)

Copyright � Laurent Gomila  ::  Documentation generated by [doxygen](http://www.doxygen.org/)  ::