Ftp Library

Table of Contents

Ftp Library **1**
- Miscellaneous Functions 1
- Server Connection 1
- Directory Functions 1
- File to File Transfer 1
- File to Program Transfer 1
- Utilities 2

FtpDir **3**
- SYNOPSIS 3
- PARAMETERS 3
- DESCRIPTION 3
- RETURN VALUE 3

FtpAccess **4**
- SYNOPSIS 4
- PARAMETERS 4
- DESCRIPTION 4
- RETURN VALUE 4

FtpOptions **5**
- SYNOPSIS 5
- PARAMETERS 5
- DESCRIPTION 5
- RETURN VALUE 6

FtpRead **7**
- SYNOPSIS 7
- PARAMETERS 7
- DESCRIPTION 7
- RETURN VALUE 7

FtpPwd **8**
- SYNOPSIS 8
- PARAMETERS 8
- DESCRIPTION 8
- RETURN VALUE 8

FtpWrite **9**
- SYNOPSIS 9
- PARAMETERS 9
- DESCRIPTION 9
- RETURN VALUE 9

FtpRename **10**
- SYNOPSIS 10
- PARAMETERS 10
- DESCRIPTION 10

Table of Contents

FtpRename
RETURN VALUE....10

FtpRmdir....11
SYNOPSIS....11
PARAMETERS....11
DESCRIPTION....11
RETURN VALUE....11

FtpQuit....12
SYNOPSIS....12
PARAMETERS....12
DESCRIPTION....12
RETURN VALUE....12

FtpChdir....13
SYNOPSIS....13
PARAMETERS....13
DESCRIPTION....13
RETURN VALUE....13

FtpModDate....14
SYNOPSIS....14
PARAMETERS....14
DESCRIPTION....14
RETURN VALUE....14

FtpInit....15
SYNOPSIS....15
PARAMETERS....15
DESCRIPTION....15
RETURN VALUE....15

FtpMkdir....16
SYNOPSIS....16
PARAMETERS....16
DESCRIPTION....16
RETURN VALUE....16

FtpSysType....17
SYNOPSIS....17
PARAMETERS....17
DESCRIPTION....17
RETURN VALUE....17

FtpLogin....18
SYNOPSIS....18
PARAMETERS....18

Table of Contents

FtpLogin
DESCRIPTION....18
RETURN VALUE....18

FtpGet....19
SYNOPSIS....19
PARAMETERS....19
DESCRIPTION....19
RETURN VALUE....19

FtpNlst....20
SYNOPSIS....20
PARAMETERS....20
DESCRIPTION....20
RETURN VALUE....20

FtpConnect....21
SYNOPSIS....21
PARAMETERS....21
DESCRIPTION....21
RETURN VALUE....21

FtpSize....22
SYNOPSIS....22
PARAMETERS....22
DESCRIPTION....22
RETURN VALUE....22

FtpSite....23
SYNOPSIS....23
PARAMETERS....23
DESCRIPTION....23
RETURN VALUE....23

FtpCDUp....24
SYNOPSIS....24
PARAMETERS....24
DESCRIPTION....24
RETURN VALUE....24

FtpLastResponse....25
SYNOPSIS....25
PARAMETERS....25
DESCRIPTION....25
RETURN VALUE....25

Table of Contents

FtpDelete **26**
SYNOPSIS 26
PARAMETERS 26
DESCRIPTION 26
RETURN VALUE 26

FtpClose **27**
SYNOPSIS 27
PARAMETERS 27
DESCRIPTION 27
RETURN VALUE 27

FtpPut **28**
SYNOPSIS 28
PARAMETERS 28
DESCRIPTION 28
RETURN VALUE 28

qftp **29**
Format 29

Ftp Library

Miscellaneous Functions

- FtpInit() - Initialize the library
- FtpSite() - Send a 'SITE' command
- FtpLastResponse() - Retrieve last server response
- FtpSysType() - Determine remote system type <ftplib V3.1>
- FtpSize() - Determine size of remote file <ftplib V3.1>
- FtpModDate() - Determine modification time of file <ftplib V3.1>

Server Connection

- FtpConnect() - Connect to a remote server
- FtpLogin() - Login to remote machine
- FtpQuit() - Disconnect from remote server
- FtpOptions() - Set Connection Options <ftplib V3.1>

Directory Functions

- FtpChdir() - Change working directory
- FtpMkdir() - Create a directory
- FtpRmdir() - Remove a directory
- FtpDir() - List a remote directory
- FtpNlst() - List a remote directory
- FtpCDUp() - Change to parent directory <ftplib V3.1>
- FtpPwd() - Determine current working directory <ftplib V3.1>

File to File Transfer

- FtpGet() - Retreive a remote file
- FtpPut() - Send a local file to remote
- FtpDelete() - Delete a remote file
- FtpRename() - Rename a remote file

File to Program Transfer

These routines were added in V3. They allow programs access to the data streams connected to remote files and directories.

- FtpAccess() - Open a remote file or directory
- FtpRead() - Read from remote file or directory
- FtpWrite() - Write to remote file
- FtpClose() - Close data connection

Utilities

- qftp - Command line ftp utility

Last Updated - June 23, 1998

Thomas Pfau / pfau@cnj.digex.net

FtpDir

Performs a verbose directory.

SYNOPSIS

```
#include <ftplib.h>
int FtpDir(const char *outputfile, const char *path, netbuf *nControl);
```

PARAMETERS

outputfile
: Name of a local file to receive the directory listing.

path
: File specification to pass to remote 'ls'.

nControl
: A handle returned by FtpConnect().

DESCRIPTION

Sends a LIST command to the server with the specified path. The response to this is usually a long format directory listing which will be written to the file named in outputfile. If outputfile is specified as NULL, the list will be written to stdout.

RETURN VALUE

Returns 1 if successful or 0 on error.

FtpAccess

Open a file or directory on the remote system.

SYNOPSIS

```
#include <ftplib.h>
int FtpAccess(const char *path, int typ, int mode, netbuf *nControl,
    netbuf **nData);
```

PARAMETERS

path
: Specifies the name of the remote file or directory to open.

typ
: Specifies the type of transfer to be performed. FTPLIB_DIR performs a terse directory. FTPLIB_DIR_VERBOSE performs a verbose directory. FTPLIB_FILE_READ opens a remote file for reading. FTPLIB_FILE_WRITE creates a remote file and readies it for writing.

mode
: Specifies the transfer mode as FTPLIB_ASCII or FTPLIB_IMAGE.

nControl
: A handle returned by FtpConnect().

nData
: Specifies the address to store a pointer to the created data handle.

DESCRIPTION

FtpAccess() opens a remote file or directory and returns a handle for the calling program to use to transfer data.

RETURN VALUE

Returns 1 if successful or 0 on error.

FtpOptions

Set connection options.

SYNOPSIS

```
#include <ftplib.h>
int FtpOptions(int opt, long val, netbuf *nControl);
```

PARAMETERS

opt
: Specifies the option to change. Valid options are FTPLIB_CONNMODE, FTPLIB_CALLBACK, FTPLIB_IDLETIME, FTPLIB_CALLBACKARG, and FTPLIB_CALLBACKBYTES.

val
: Specifies the new value for the option. The value may need to by cast to a long.

nControl
: A handle returned by FtpConnect() or FtpAccess().

DESCRIPTION

FtpOptions() changes the options for a connection handle. A data connection inherits the options assigned to the control connection it is created from. Callbacks are only called on file data connections.

The following options and values are recognized.

Option	Value
FTPLIB_CONNMODE	Specifies the connection mode. Either FTPLIB_PASSIVE or FTPLIB_PORT.
FTPLIB_CALLBACK	Specifies the address of a user callback routine.
FTPLIB_IDLETIME	Specifies the socket idle time in milliseconds that triggers calling the user's callback routine.
FTPLIB_CALLBACKARG	Specifies an argument to pass to the user's callback routine.
FTPLIB_CALLBACKBYTES	Specifies the number of bytes to transfer between calls to the user's callback routine.

The connection mode tells ftplib if it should use PASV or PORT to establish data connections. The default is specified as a build option.

The user's callback routine is specified as:

```
typedef int (*FtpCallback)(netbuf *nControl, int xfered, void *arg);
```

nControl is the data connection in use. **xfered** specifies how many bytes of data have been transferred on the connection. **arg** is the value specified with option FTPLIB_CALLBACKARG.

The user can request to be called back on either of two events.

If the user wishes to be called when the data socket is idle for some period of time, use FTPLIB_IDLETIME and pass the time in milliseconds.

If the user wishes to be called when a certain amount of data has been transferred, use FTPLIB_CALLBACKBYTES and pass the minimum number of bytes to transfer between callbacks. When using this option, ftplib keeps track of the number of bytes transferred and calls the user once the specified number of bytes or more has been transferred. It then resets the count to 0 and starts again.

If the user wishes to continue the transfer, the callback routine should return true (non-zero). It can abort the transfer by return zero.

RETURN VALUE

Returns 1 if a valid option was specified and the value is legal. Otherwise, returns 0.

FtpRead

Read data from a remote file or directory.

SYNOPSIS

```
#include <ftplib.h>
int FtpRead(void *buf, int max, netbuf *nData);
```

PARAMETERS

buf
: Specifies the address of a buffer where received data will be written.

max
: Specifies the size of the user's buffer.

nData
: A handle returned by FtpAccess().

DESCRIPTION

FtpRead copies up to max bytes of data from the specified data connection and returns it to the user's buffer. If the data connection was opened in ascii mode, no more than one line of data will be returned.

RETURN VALUE

Returns the number of bytes written to the user's buffer or -1 on error or end of file.

FtpPwd

Determine current working directory on server.

SYNOPSIS

```
#include <ftplib.h>
int FtpPwd(char *path, int max, netbuf *nControl);
```

PARAMETERS

path
: A pointer to a buffer where the result should be returned.

max
: Specifies the size of the user's buffer.

nControl
: A handle returned by FtpConnect().

DESCRIPTION

FtpPwd() attempts to determine the current default directory at the server and return it to the user's buffer.

RETURN VALUE

Returns 1 if successful or 0 on error.

FtpWrite

Write data to a remote file.

SYNOPSIS

```
#include <ftplib.h>
int FtpWrite(void *buf, int len, netbuf *nData);
```

PARAMETERS

buf
: A buffer containing the data to be sent to the remote file.

len
: The number of bytes to be sent from 'buf'.

nData
: A handle returned by FtpAccess().

DESCRIPTION

FtpWrite() sends data to a remote file. If the file were accessed in record mode, the necessary conversions are performed.

RETURN VALUE

Returns the number of bytes sent from the user's buffer or -1 on error.

FtpRename

Rename a remote file.

SYNOPSIS

```
#include <ftplib.h>
int FtpRename(const char *src, const char *dst, netbuf *nControl);
```

PARAMETERS

src
: A string containing the current name of the remote file.

dst
: A string containing the desired new name for the remote file.

nControl
: A handle returned by FtpConnect().

DESCRIPTION

FtpRename() sends a rename request to the remote server.

RETURN VALUE

Returns 1 if successful or 0 on error.

FtpRmdir

Remove a directory from the remote file system.

SYNOPSIS

```
#include <ftplib.h>
int FtpRmdir(const char *path, netbuf *nControl);
```

PARAMETERS

path
: A string containing the name of a remote directory.

nControl
: A handle returned by FtpConnect().

DESCRIPTION

FtpRmdir() sends a remove directory request to the remote server.

RETURN VALUE

Returns 1 if successful or 0 on error.

FtpQuit

Close connection to server.

SYNOPSIS

```
#include <ftplib.h>
void FtpQuit(netbuf *nControl);
```

PARAMETERS

nControl
: A handle returned by FtpConnect().

DESCRIPTION

FtpQuit() closes the connection to the remote server and frees any resources associated with the connection.

RETURN VALUE

None.

FtpChdir

Change working directory on server.

SYNOPSIS

```
#include <ftplib.h>
int FtpChdir(const char *path, netbuf *nControl);
```

PARAMETERS

path
: Specifies the desired working directory on the server.

nControl
: A handle returned by FtpConnect().

DESCRIPTION

Sends a change working directory request to the server using the specified path.

RETURN VALUE

Returns 1 if successful or 0 on error.

FtpModDate

Determine last modification time of a remote file.

SYNOPSIS

```
#include <ftplib.h>
int FtpModDate(char *path, char *buf, int max, netbuf *nControl);
```

PARAMETERS

path
: Name of remote file to be checked.

buf
: A pointer to a buffer where the result should be returned.

max
: Specifies the size of the user's buffer.

nControl
: A handle returned by FtpConnect().

DESCRIPTION

FtpModDate() attempts to determine the last access time of a remote file and return it to the user's buffer. The date and time are returned as a string in the format 'YYYYMMDDHHMMSS'.

RETURN VALUE

If a good response is received and the date and time are successfully parsed out of the result, 1 is returned. Otherwise, 0 is returned.

Some servers may not support the MDTM command.

FtpInit

Initialize the library.

SYNOPSIS

```
#include <ftplib.h>
void FtpInit(void);
```

PARAMETERS

None.

DESCRIPTION

Performs any required initialization for the library.

RETURN VALUE

None.

FtpMkdir

Create a directory on the remote system.

SYNOPSIS

```
#include <ftplib.h>
int FtpMkdir(const char *path, netbuf *nControl);
```

PARAMETERS

path
: Specifies the argument to mkdir on the remote system.

nControl
: A handle returned by FtpConnect().

DESCRIPTION

FtpMkdir() sends a make directory request to the remote system.

RETURN VALUE

Returns 1 if successful or 0 on error.

FtpSysType

Determine system type of remote server.

SYNOPSIS

```
#include <ftplib.h>
int FtpSysType(char *buf, int max, netbuf *nControl);
```

PARAMETERS

buf
: A pointer to a buffer where the result will be returned.

max
: Specifies the size of the user buffer.

nControl
: A handle returned by FtpConnect().

DESCRIPTION

FtpSysType() issues a SYST command to the remote system and attempts to parse the system type out of the response and return it to the user's buffer.

RETURN VALUE

Returns 1 if successful or 0 on error.

FtpLogin

Login to remote system.

SYNOPSIS

```
#include <ftplib.h>
int FtpLogin(const char *user, const char *pass, netbuf *nControl);
```

PARAMETERS

user
: Specifies the username.

pass
: Specifies the password.

nControl
: A handle returned by FtpConnect().

DESCRIPTION

FtpLogin() attempts to login to the remote system with the supplied username and password.

RETURN VALUE

Returns 1 if successful or 0 on error.

FtpGet

Retreive a file from the remote system.

SYNOPSIS

```
#include <ftplib.h>
int FtpGet(const char *output, const char *path, char mode, netbuf *nControl);
```

PARAMETERS

output
: Name of a local file to receive the contents of the remote file.

path
: Name of remote file to be retrieved.

mode
: Specifies the transfer mode as FTPLIB_ASCII or FTPLIB_IMAGE.

nControl
: A handle returned by FtpConnect().

DESCRIPTION

FtpGet() copies the contents of a remote file to a local file.

RETURN VALUE

Returns 1 if successful or 0 on error.

FtpNlst

Performs a terse directory.

SYNOPSIS

```
#include <ftplib.h>
int FtpNlst(const char *output, const char *path, netbuf *nControl);
```

PARAMETERS

output
: Specifies the name of a file to receive the directory listing.

path
: Specifies an argument to 'ls' on the remote system.

nControl
: A handle returned by FtpConnect().

DESCRIPTION

Performs a short form directory listing of the specified path on the remote system. The results are written to the specified file.

RETURN VALUE

Returns 1 if successful or 0 on error.

FtpConnect

Connect to an FTP server.

SYNOPSIS

```
#include <ftplib.h>
int FtpConnect(const char *host, netbuf **nControl);
```

PARAMETERS

host
: The name of the host machine to connect to and optionally an alternate port number to use.

nControl
: The address where the pointer to the newly created control handle should be stored.

DESCRIPTION

FtpConnect() establishes a connection to the FTP server on the specified machine and returns a handle which can be used to initiate data transfers. The host name should be specified as or :. may be either a host name or ip address. may be either a service name or a port number.

RETURN VALUE

If the connection to the remote server if successful, FtpConnect() returns 1. Otherwise, 0 is returned.

FtpSize

Determine size of remote file.

SYNOPSIS

```
#include <ftplib.h>
int FtpSize(char *path, int *size, char mode, netbuf *nControl);
```

PARAMETERS

path
: A pointer to a buffer where the result should be returned.

size
: A pointer to an int where the size will be returned.

mode
: Specifies the transfer mode as FTPLIB_ASCII or FTPLIB_IMAGE.

nControl
: A handle returned by FtpConnect().

DESCRIPTION

FtpSize() attempts to determine the size of a remote file.

RETURN VALUE

If a good response is received and the size is successfully parsed out of the result, 1 is returned. Otherwise, 0 is returned.

Some servers may not support the SIZE command. If this request fails, the size may be available in the response to a RETR (FtpOpen() with typ=FTPLIB_FILE_READ).

FtpSite

Send a SITE command to the remote server.

SYNOPSIS

```
#include <ftplib.h>
int FtpSite(const char *cmd, netbuf *nControl);
```

PARAMETERS

cmd
: A string containing a 'SITE' subcommand.

nControl
: A handle returned by FtpConnect().

DESCRIPTION

FtpSite() sends the specified command as an argument to a 'SITE' command.

RETURN VALUE

Returns 1 if successful or 0 on error.

FtpCDUp

Change to parent directory.

SYNOPSIS

```
#include <ftplib.h>
int FtpCDUp(netbuf *nControl);
```

PARAMETERS

nControl
: A handle returned by FtpConnect().

DESCRIPTION

FtpCDUp() sends a CDUP command to the remote server.

RETURN VALUE

Returns 1 if successful or 0 on error.

FtpLastResponse

Retrieve the last response sent by the server.

SYNOPSIS

```
#include <ftplib.h>
char *FtpLastResponse(netbuf *nControl);
```

PARAMETERS

nControl
: A pointer to a control handle returned by FtpConnect().

DESCRIPTION

FtpLastResponse() returns a pointer to the last response string sent by the server. This can be parsed by the user program to determine more information about the last request or can be displayed along with an error message.

RETURN VALUE

If nControl is a valid control handle, FtpLastResponse() returns a pointer to the last server response string. Otherwise, NULL is returned.

FtpDelete

Removes a file from the remote system.

SYNOPSIS

```
#include <ftplib.h>
int FtpDelete(const char *fnm, netbuf *nControl);
```

PARAMETERS

fnm
: The name of the file which is to be removed.

nControl
: A handle returned by FtpConnect().

DESCRIPTION

Requests that the server remove the specified file from the remote file system.

RETURN VALUE

Returns 1 if successful or 0 on error.

FtpClose

Close a data connection.

SYNOPSIS

```
#include <ftplib.h>
int FtpClose(netbuf *nData);
```

PARAMETERS

nData
: A handle returned by FtpAccess().

DESCRIPTION

Closes the data connection specified by 'nData' and frees associated resources.

RETURN VALUE

Returns 1 if successful or 0 on error.

FtpPut

Send a file to the remote system.

SYNOPSIS

```
#include <ftplib.h>
int FtpPut(const char *input, const char *path, char mode, netbuf *nControl);
```

PARAMETERS

input
: Specifies the name of a local file to be transfered to the server.

path
: Specifies the name to be given to the file on the remote system.

mode
: Specifies the transfer mode as FTPLIB_ASCII or FTPLIB_IMAGE.

nControl
: A handle returned by FtpConnect().

DESCRIPTION

FtpPut() transfers a local file to the remote system.

RETURN VALUE

Returns 1 if successful or 0 on error.

qftp

qftp is a utility that performs file transfers using *ftplib* based on instructions presented on the command line.

Format

```
qftp <action> <host> [ -l user [ -p pass ] ] { options/files }...

Actions: send, get, dir, list, rm

Options:
    -v level        Set verbosity
    -r rootpath     Change remote working directory
    -m umask        Set umask for created files
    -a | -i         Set ascii/image transfer mode
```

action
: Specifies what *qftp* is to do. *qftp* can **send** files to a remote system, **get** files from a remote system, remove (**rm**) files from a remote system, or retreive a **dir**ectory of files or a **list** of files on a remote system.

host
: Specifies the name of the remote system. An alternate port may be specified by appending a colon and the port name or number to the host name.

-l user
: Specifies the username to log in with on **host**.

-p pass
: Specifies the password to log in with on **host**.

  If **user** is not specified, *qftp* will use **anonymous**. If **pass** is also not specified, *qftp* attempts to build a password from the translation of the environment variable "USER" and the string returned by gethostname() separated by an "@".

-v level
: Specifies the verbosity level. A level of 1 will cause *qftp* to display messages indicating successful transfers. A level of 2 will cause FTP protocol responses to be displayed. A level of 3 will cause FTP protocol commands to be displayed.

-r rootpath
: Sends a 'change directory' request to the remote server. All subsequent file names on the command line will be parsed relative to this new directory.

-m umask
: Sends a request to change the umask to the remote server. This may not be supported by all servers. It is implemented using the 'SITE' command in the FTP protocol.

-a
: Requests that subsequent files be sent in ascii mode.

-i
: Requests that subsequent files be sent in image mode.

qftp may optionally be invoked through a softlink. *qftp* searches the command which invoked it for 'send', 'get' or 'dir' and, if found, performs the requested function. In this case, leave off the **action** argument on the command line.

For example; I use the following softlinks:

```
ln -s qftp ftpsend
ln -s qftp ftpget
ln -s qftp ftpdir
ln -s qftp ftplist
ln -s qftp ftprm
```

and then invoke transfers with 'ftpsend' instead of 'qftp send', etc.

If no file names are specified on the command line, *qftp* will read file names from stdin. Use your favorite utility to generate a list of files to send/retreive or type them interactively.