NAME

ftp – file transfer program

SYNOPSIS

ftp [**–dgintv**] [*hostname*]

AVAILABILITY

SUNWcsu

DESCRIPTION

The **ftp** command is the user interface to the Internet standard File Transfer Protocol (FTP). **ftp** transfers files to and from a remote network site.

The client host with which **ftp** is to communicate may be specified on the command line. If this is done, **ftp** immediately attempts to establish a connection to an FTP server on that host; otherwise, **ftp** enters its command interpreter and awaits instructions from the user. When **ftp** is awaiting commands from the user, it displays the prompt **ftp>**.

OPTIONS

The following options may be specified at the command line, or to the command interpreter:

–d Enable debugging.

–g Disable filename "globbing."

–i Turn off interactive prompting during multiple file transfers.

–n Do not attempt "auto-login" upon initial connection. If auto-login is not disabled, **ftp** checks the **.netrc** file in the user's home directory for an entry describing an account on the remote machine. If no entry exists, **ftp** will prompt for the login name of the account on the remote machine (the default is the login name on the local machine), and, if necessary, prompts for a password and an account with which to login.

–t Enable packet tracing (unimplemented).

–v Show all responses from the remote server, as well as report on data transfer statistics. This is turned on by default if **ftp** is running interactively with its input coming from the user's terminal.

The following commands can be specified to the command interpreter:

! [*command*]
Run *command* as a shell command on the local machine. If no *command* is given, invoke an interactive shell.

$ *macro-name* [*args*]
Execute the macro *macro-name* that was defined with the **macdef** command. Arguments are passed to the macro unglobbed.

account [*passwd*]
Supply a supplemental password required by a remote system for access to resources once a login has been successfully completed. If no argument is included, the user will be prompted for an account password in a non-echoing input mode.

append *local-file* [*remote-file*]
Append a local file to a file on the remote machine. If *remote-file* is not specified, the local file name is used, subject to alteration by any **ntrans** or **nmap** settings. File transfer uses the current settings for “representation type”, “file structure”, and “transfer mode”.

ascii Set the “representation type” to “network ASCII”. This is the default type.

bell Sound a bell after each file transfer command is completed.

binary Set the “representation type” to “image”.

bye Terminate the FTP session with the remote server and exit **ftp**. An EOF will also terminate the session and exit.

case Toggle remote computer file name case mapping during **mget** commands. When **case** is on (default is off), remote computer file names with all letters in upper case are written in the local directory with the letters mapped to lower case.

cd *remote-directory*
Change the working directory on the remote machine to *remote-directory*.

cdup Change the remote machine working directory to the parent of the current remote machine working directory.

close Terminate the FTP session with the remote server, and return to the command interpreter. Any defined macros are erased.

cr Toggle RETURN stripping during “network ASCII” type file retrieval. Records are denoted by a RETURN/LINEFEED sequence during “network ASCII” type file transfer. When **cr** is on (the default), RETURN characters are stripped from this sequence to conform with the UNIX system single LINEFEED record delimiter. Records on non-UNIX-system remote hosts may contain single LINEFEED characters; when an “network ASCII” type transfer is made, these LINEFEED characters may be distinguished from a record delimiter only when **cr** is off.

delete *remote-file*
Delete the file *remote-file* on the remote machine.

debug
Toggle debugging mode. When debugging is on, **ftp** prints each command sent to the remote machine, preceded by the string **––>**.

dir [*remote-directory*] [*local-file*]
Print a listing of the directory contents in the directory, *remote-directory*, and, optionally, placing the output in *local-file*. If no directory is specified, the current working directory on the remote machine is used. If no local file is specified, or *local-file* is –, output is sent to the terminal.

disconnect
A synonym for **close**.

form [*format-name*]
Set the carriage control format subtype of the "representation type" to *format-name*. The only valid *format-name* is **non-print**, which corresponds to the default "non-print" subtype.

get *remote-file* [*local-file*]
Retrieve the *remote-file* and store it on the local machine. If the local file name is not specified, it is given the same name it has on the remote machine, subject to alteration by the current **case**, **ntrans**, and **nmap** settings. The current settings for "representation type", "file structure", and "transfer mode" are used while transferring the file.

glob Toggle filename expansion, or "globbing", for **mdelete**, **mget** and **mput**. If globbing is turned off, filenames are taken literally.

Globbing for **mput** is done as in **sh**(1). For **mdelete** and **mget**, each remote file name is expanded separately on the remote machine, and the lists are not merged.

Expansion of a directory name is likely to be radically different from expansion of the name of an ordinary file: the exact result depends on the remote operating system and FTP server, and can be previewed by doing **mls** *remote-files* –.

mget and **mput** are not meant to transfer entire directory subtrees of files. You can do this by transferring a **tar**(1) archive of the subtree (using a "representation type" of "image" as set by the **binary** command).

hash Toggle hash-sign (**#**) printing for each data block transferred. The size of a data block is 8192 bytes.

help [*command*]
Print an informative message about the meaning of *command*. If no argument is given, **ftp** prints a list of the known commands.

lcd [*directory*]
Change the working directory on the local machine. If no *directory* is specified, the user's home directory is used.

ls [*remote-directory* | **-al**] [*local-file*]
Print an abbreviated listing of the contents of a directory on the remote machine. If *remote-directory* is left unspecified, the current working directory is used.

The **-a** option lists all entries, including those that begin with a dot (**.**), which are normally not listed. The **-l** option lists files in long format, giving mode, number of links, owner, group, size in bytes, and time of last modification for each file. If the file is a special file, the size field instead contains the major and minor device numbers rather than a size. If the file is a symbolic link, the filename is printed followed by "→" and the pathname of the referenced file.

If no local file is specified, or if *local-file* is –, the output is sent to the terminal.

macdef *macro-name*
Define a macro. Subsequent lines are stored as the macro *macro-name*; a null line (consecutive NEWLINE characters in a file or RETURN characters from the terminal) terminates macro input mode. There is a limit of 16 macros and 4096 total characters in all defined macros. Macros remain defined until a **close** command is executed.

The macro processor interprets **$** and \ as special characters. A **$** followed by a number (or numbers) is replaced by the corresponding argument on the macro invocation command line. A **$** followed by an **i** signals that macro processor that the executing macro is to be looped. On the first pass **$i** is replaced by the first argument on the macro invocation command line, on the second pass it is replaced by the second argument, and so on. A \ followed by any character is replaced by that character. Use the \ to prevent special treatment of the **$**.

mdelete *remote-files*
Delete the *remote-files* on the remote machine.

mdir *remote-files local-file*
Like **dir**, except multiple remote files may be specified. If interactive prompting is on, **ftp** will prompt the user to verify that the last argument is indeed the target local file for receiving **mdir** output.

mget *remote-files*
Expand the *remote-files* on the remote machine and do a **get** for each file name thus produced. See **glob** for details on the filename expansion. Resulting file names will then be processed according to **case**, **ntrans**, and **nmap** settings. Files are transferred into the local working directory, which can be changed with **lcd** *directory*; new local directories can be created with **! mkdir** *directory*.

mkdir *directory-name*
Make a directory on the remote machine.

mls *remote-files local-file*
Like **ls**(1), except multiple remote files may be specified. If interactive prompting is on, **ftp** will prompt the user to verify that the last argument is indeed the target local file for receiving **mls** output.

mode [*mode-name*]
Set the "transfer mode" to *mode-name*. The only valid *mode-name* is **stream**, which corresponds to the default "stream" mode. This implementation only supports **stream**, and requires that it be specified.

mput *local-files*
Expand wild cards in the list of local files given as arguments and do a **put** for each file in the resulting list. See **glob** for details of filename expansion. Resulting file names will then be processed according to **ntrans** and **nmap** settings.

nmap [*inpattern outpattern*]

Set or unset the filename mapping mechanism. If no arguments are specified, the filename mapping mechanism is unset. If arguments are specified, remote filenames are mapped during **mput** commands and **put** commands issued without a specified remote target filename. If arguments are specified, local filenames are mapped during **mget** commands and **get** commands issued without a specified local target filename.

This command is useful when connecting to a non-UNIX-system remote host with different file naming conventions or practices. The mapping follows the pattern set by *inpattern* and *outpattern*. *inpattern* is a template for incoming filenames (which may have already been processed according to the **ntrans** and **case** settings). Variable templating is accomplished by including the sequences **$1**, **$2**, . . . , **$9** in *inpattern*. Use \ to prevent this special treatment of the **$** character. All other characters are treated literally, and are used to determine the **nmap** *inpattern* variable values.

For example, given *inpattern* **$1.$2** and the remote file name **mydata.data**, **$1** would have the value **mydata**, and **$2** would have the value **data**.

The *outpattern* determines the resulting mapped filename. The sequences **$1**, **$2**, . . . , **$9** are replaced by any value resulting from the *inpattern* template. The sequence **$0** is replaced by the original filename. Additionally, the sequence [*seq1* , *seq2*] is replaced by *seq1* if *seq1* is not a null string; otherwise it is replaced by *seq2*.

For example, the command **nmap $1.$2.$3 [$1,$2].[$2,file]** would yield the output filename **myfile.data** for input filenames **myfile.data** and **myfile.data.old**, **myfile.file** for the input filename **myfile**, and **myfile.myfile** for the input filename **.myfile**. SPACE characters may be included in *outpattern*, as in the example **nmap $1 | sed "s/ *$//" > $1**. Use the \ character to prevent special treatment of the **$**, **[**, **]**, and **,**, characters.

ntrans [*inchars* [*outchars*]]

Set or unset the filename character translation mechanism. If no arguments are specified, the filename character translation mechanism is unset. If arguments are specified, characters in remote filenames are translated during **mput** commands and **put** commands issued without a specified remote target filename, and characters in local filenames are translated during **mget** commands and **get** commands issued without a specified local target filename.

This command is useful when connecting to a non-UNIX-system remote host with different file naming conventions or practices. Characters in a filename matching a character in *inchars* are replaced with the corresponding character in *outchars*. If the character's position in *inchars* is longer than the length of *outchars*, the character is deleted from the file name.

Only 16 characters can be translated when using the **ntrans** command under **ftp**. Use **case** (described above) if needing to convert the entire alphabet.

open *host* [*port*]
Establish a connection to the specified *host* FTP server. An optional port number may be supplied, in which case, **ftp** will attempt to contact an FTP server at that port. If the *auto-login* option is on (default setting), **ftp** will also attempt to automatically log the user in to the FTP server.

prompt
Toggle interactive prompting. Interactive prompting occurs during multiple file transfers to allow the user to selectively retrieve or store files. By default, prompting is turned on. If prompting is turned off, any **mget** or **mput** will transfer all files, and any **mdelete** will delete all files.

proxy *ftp-command*
Execute an FTP command on a secondary control connection. This command allows simultaneous connection to two remote FTP servers for transferring files between the two servers. The first **proxy** command should be an **open**, to establish the secondary control connection. Enter the command **proxy ?** to see other FTP commands executable on the secondary connection.

The following commands behave differently when prefaced by **proxy**: **open** will not define new macros during the auto-login process, **close** will not erase existing macro definitions, **get** and **mget** transfer files from the host on the primary control connection to the host on the secondary control connection, and **put**, **mputd**, and **append** transfer files from the host on the secondary control connection to the host on the primary control connection.

Third party file transfers depend upon support of the **PASV** command by the server on the secondary control connection.

put *local-file* [*remote-file*]
Store a local file on the remote machine. If *remote-file* is left unspecified, the local file name is used after processing according to any **ntrans** or **nmap** settings in naming the remote file. File transfer uses the current settings for “representation type”, “file structure”, and “transfer mode”.

pwd Print the name of the current working directory on the remote machine.

quit A synonym for **bye**.

quote *arg1 arg2 . . .*
Send the arguments specified, verbatim, to the remote FTP server. A single FTP reply code is expected in return. (The **remotehelp** command displays a list of valid arguments.)

quote should be used only by experienced users who are familiar with the FTP protocol.

recv *remote-file* [*local-file*]
A synonym for **get**.

remotehelp [*command-name*]
Request help from the remote FTP server. If a *command-name* is specified it is supplied to the server as well.

rename *from to*
Rename the file *from* on the remote machine to have the name *to*.

reset Clear reply queue. This command re-synchronizes command/reply sequencing with the remote FTP server. Resynchronization may be necessary following a violation of the FTP protocol by the remote server.

rmdir *directory-name*
Delete a directory on the remote machine.

runique
Toggle storing of files on the local system with unique filenames. If a file already exists with a name equal to the target local filename for a **get** or **mget** command, a **.1** is appended to the name. If the resulting name matches another existing file, a **.2** is appended to the original name. If this process continues up to **.99**, an error message is printed, and the transfer does not take place. The generated unique filename will be reported. **runique** will not affect local files generated from a shell command. The default value is off.

send *local-file* [*remote-file*]
A synonym for **put**.

sendport
Toggle the use of **PORT** commands. By default, **ftp** will attempt to use a **PORT** command when establishing a connection for each data transfer. The use of **PORT** commands can prevent delays when performing multiple file transfers. If the **PORT** command fails, **ftp** will use the default data port. When the use of **PORT** commands is disabled, no attempt will be made to use **PORT** commands for each data transfer. This is useful when connected to certain FTP implementations that ignore **PORT** commands but incorrectly indicate they have been accepted.

status Show the current status of **ftp**.

struct [*struct-name*]
Set the file structure to *struct-name*. The only valid *struct-name* is **file**, which corresponds to the default “file” structure. The implementation only supports **file**, and requires that it be specified.

sunique
Toggle storing of files on remote machine under unique file names. The remote FTP server must support the **STOU** command for successful completion. The remote server will report the unique name. Default value is off.

tenex Set the “representation type” to that needed to talk to TENEX machines.

trace Toggle packet tracing (unimplemented).

type [*type-name*]
Set the “representation type” to *type-name*. The valid *type-name*s are **ascii** for “network ASCII”, **binary** or **image** for “image”, and **tenex** for “local byte size” with a byte size of 8 (used to talk to TENEX machines). If no type is specified, the current type is printed. The default type is “network ASCII”.

user *user-name* [*password*] [*account*]
Identify yourself to the remote FTP server. If the password is not specified and the server requires it, **ftp** will prompt the user for it (after disabling local echo). If an account field is not specified, and the FTP server requires it, the user will be prompted for it. If an account field is specified, an account command will be relayed to the remote server after the login sequence is completed if the remote server did not require it for logging in. Unless **ftp** is invoked with "auto-login" disabled, this process is done automatically on initial connection to the FTP server.

verbose
Toggle verbose mode. In verbose mode, all responses from the FTP server are displayed to the user. In addition, if verbose mode is on, when a file transfer completes, statistics regarding the efficiency of the transfer are reported. By default, verbose mode is on if **ftp**'s commands are coming from a terminal, and off otherwise.

? [*command*]
A synonym for **help**.

Command arguments which have embedded spaces may be quoted with quote (") marks.

If any command argument which is not indicated as being optional is not specified, **ftp** will prompt for that argument.

ABORTING A FILE TRANSFER

To abort a file transfer, use the terminal interrupt key. Sending transfers will be immediately halted. Receiving transfers will be halted by sending an FTP protocol **ABOR** command to the remote server, and discarding any further data received. The speed at which this is accomplished depends upon the remote server's support for **ABOR** processing. If the remote server does not support the **ABOR** command, an **ftp>** prompt will not appear until the remote server has completed sending the requested file.

The terminal interrupt key sequence will be ignored when **ftp** has completed any local processing and is awaiting a reply from the remote server. A long delay in this mode may result from the **ABOR** processing described above, or from unexpected behavior by the remote server, including violations of the ftp protocol. If the delay results from unexpected remote server behavior, the local **ftp** program must be killed by hand.

FILE NAMING CONVENTIONS

Local files specified as arguments to **ftp** commands are processed according to the following rules.

1) If the file name – is specified, the standard input (for reading) or standard output (for writing) is used.

2) If the first character of the file name is **|**, the remainder of the argument is interpreted as a shell command. **ftp** then forks a shell, using **popen**(3S) with the argument supplied, and reads (writes) from the standard output (standard input) of that shell. If the shell command includes SPACE characters, the argument must be quoted; for example **"| ls –lt"**. A particularly useful example of this mechanism is: **"dir | more"**.

3) Failing the above checks, if globbing is enabled, local file names are expanded according to the rules used in the **sh**(1); see the **glob** command. If the **ftp** command expects a single local file (for example, **put**), only the first filename generated by the globbing operation is used.

4) For **mget** commands and **get** commands with unspecified local file names, the local filename is the remote filename, which may be altered by a **case**, **ntrans**, or **nmap** setting. The resulting filename may then be altered if **runique** is on.

5) For **mput** commands and **put** commands with unspecified remote file names, the remote filename is the local filename, which may be altered by a **ntrans** or **nmap** setting. The resulting filename may then be altered by the remote server if **sunique** is on.

FILE TRANSFER PARAMETERS

The FTP specification specifies many parameters which may affect a file transfer.

The “representation type” may be one of “network ASCII”, “EBCDIC”, “image”, or “local byte size” with a specified byte size (for PDP-10’s and PDP-20’s mostly). The “network ASCII” and “EBCDIC” types have a further subtype which specifies whether vertical format control (NEWLINE characters, form feeds, etc.) are to be passed through (“non-print”), provided in TELNET format (“TELNET format controls”), or provided in ASA (FORTRAN) (“carriage control (ASA)”) format. **ftp** supports the “network ASCII” (subtype “non-print” only) and “image” types, plus “local byte size” with a byte size of 8 for communicating with TENEX machines.

The “file structure” may be one of **file** (no record structure), **record**, or **page**. **ftp** supports only the default value, which is **file**.

The “transfer mode” may be one of **stream**, **block**, or **compressed**. **ftp** supports only the default value, which is **stream**.

FILES

~/.netrc

SEE ALSO

ls(1), **rcp**(1), **sh**(1), **tar**(1), **ftpd**(1M), **popen**(3S), **netrc**(4)

NOTES

Correct execution of many commands depends upon proper behavior by the remote server.

An error in the treatment of carriage returns in the 4.2 BSD code handling transfers with a “representation type” of “network ASCII” has been corrected. This correction may result in incorrect transfers of binary files to and from 4.2 BSD servers using a “representation type” of “network ASCII”. Avoid this problem by using the “image” type.