NAME

lftp − Sophisticated file transfer program

SYNTAX

lftp [**−d**] [**−e** *cmd*] [**−p** *port*] [**−u** *user*[**,***pass*]] [*site*]
lftp −f *script_file*
lftp −c *commands*
lftp −−version
lftp −−help

VERSION

This man page documents lftp version 4.8.1.

DESCRIPTION

lftp is a file transfer program that allows sophisticated FTP, HTTP and other connections to other hosts. If *site* is specified then lftp will connect to that site otherwise a connection has to be established with the open command.

lftp can handle several file access methods - FTP, FTPS, HTTP, HTTPS, HFTP, FISH, SFTP and file (HTTPS and FTPS are only available when lftp is compiled with GNU TLS or OpenSSL library). You can specify the method to use in 'open URL' command, e.g. 'open http://www.us.kernel.org/pub/linux'. HFTP is ftp-over-http-proxy protocol. It can be used automatically instead of FTP if ftp:proxy is set to 'http://proxy[:port]'. Fish is a protocol working over an ssh connection to a unix account. SFtp is a protocol implemented in SSH2 as SFTP subsystem.

Besides FTP-like protocols, lftp has support for BitTorrent protocol as 'torrent' command. Seeding is also supported.

Every operation in **lftp** is reliable, that is any non-fatal error is handled properly and the operation is repeated. So if downloading breaks, it will be restarted from the point automatically. Even if FTP server does not support the REST command, **lftp** will try to retrieve the file from the very beginning until the file is transferred completely.

lftp has shell-like command syntax allowing you to launch several commands in parallel in background (&). It is also possible to group commands within () and execute them in background. All background jobs are executed in the same single process. You can bring a foreground job to background with ^Z (c-z) and back with command 'wait' (or 'fg' which is alias to 'wait'). To list running jobs, use command 'jobs'. Some commands allow redirecting their output (cat, ls, ...) to file or via pipe to external command. Commands can be executed conditionally based on termination status of previous command (&&, ||).

If you exit **lftp** before all jobs are not finished yet, **lftp** will move itself to nohup mode in background. The same thing happens with a real modem hangup or when you close an xterm.

lftp has built-in mirror which can download or update a whole directory tree. There is also reverse mirror (mirror −R) which uploads or updates a directory tree on server. Mirror can also synchronize directories between two remote servers, using FXP if available.

There is command 'at' to launch a job at specified time in current context, command 'queue' to queue commands for sequential execution for current server, and much more.

On startup, **lftp** executes */etc/lftp.conf* and then *~/.lftprc* and *~/.lftp/rc* (or *~/.config/lftp/rc* if *~/.lftp* does not exist). You can place aliases and 'set' commands there. Some people prefer to see full protocol debug, use 'debug' to turn the debug on. Use 'debug 3' to see only greeting messages and error messages.

lftp has a number of settable variables. You can use 'set −a' to see all variables and their values or 'set −d' to see list of defaults. Variable names can be abbreviated and prefix can be omitted unless the rest becomes ambiguous.

If lftp was compiled with OpenSSL (configure −−with−openssl) it includes software developed by the

OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org/)

Commands

! *shell command*

Launch shell or shell command.

```
!ls
```

To do a directory listing of the local host.

alias [*name* [*value*]]

Define or undefine alias *name*. If *value* is omitted, the alias is undefined, else it takes the value *value*. If no argument is given the current aliases are listed.

```
alias dir ls −lF
alias less zmore
```

at *time* [−− *command*]

Wait until the given time and execute given (optional) command. See also **at**(1).

attach [*PID*]

Attach the terminal to specified backgrounded lftp process.

bookmark [*subcommand*]

The bookmark command controls bookmarks.

Site names can be used in the *open* command directly as-is or in any command that accepts input URLs using the *bm:site/path* format.

add <name> [<loc>]	add current place or given location to bookmarks and bind to given name
del <name>	remove bookmark with name
edit	start editor on bookmarks file
import <type>	import foreign bookmarks
list	list bookmarks (default)

cache [*subcommand*]

The cache command controls local memory cache. The following subcommands are recognized:

stat	print cache status (default)
on|off	turn on/off caching
flush	flush cache
size *lim*	set memory limit, -1 means unlimited
expire *Nx*	set cache expiration time to *N* seconds (*x*=s) minutes (*x*=m) hours (*x*=h) or days (*x*=d)

cat *files*

cat outputs the remote file(s) to stdout. (See also **more**, **zcat** and **zmore**)

cd rdir

Change current remote directory. The previous remote directory is stored as ‘−’. You can do ‘cd −’ to change the directory back. The previous directory for each site is also stored on disk, so you can do ‘open site; cd −’ even after lftp restart.

chmod [**OPTS**] *mode files...*

Change permission mask on remote files. The mode can be an octal number or a symbolic mode (see chmod(1)).

-c,	--changes	like verbose but report only when a change is made
-f,	--quiet	suppress most error messages
-v,	--verbose	output a diagnostic for every file processed
-R,	--recursive	change files and directories recursively

close [**−a**]

Close idle connections. By default only with the current server, use −a to close all idle connections.

cls [**OPTS**] *files...*

'cls' tries to retrieve information about specified files or directories and outputs the information according to format options. The difference between 'ls' and 'cls' is that 'ls' requests the server to format file listing, and 'cls' formats it itself, after retrieving all the needed information.

−1		single-column output
−a,	−−all	show dot files
−B,	−−basename	show basename of files only
	−−block−size=SIZ	use SIZ-byte blocks
−d,	−−directory	list directory entries instead of contents
−F,	−−classify	append indicator (one of /@) to entries
−h,	−−human−readable	print sizes in human readable format (e.g., 1K)
	−−si	likewise, but use powers of 1000 not 1024
−k,	−−kilobytes	like −−block−size=1024
−l,	−−long	use a long listing format
−q,	−−quiet	don't show status
−s,	−−size	print size of each file
	−−filesize	if printing size, only print size for files
−i,	−−nocase	case-insensitive pattern matching
−I,	−−sortnocase	sort names case-insensitively
−D,	−−dirsfirst	list directories first
	−−sort=OPT	"name", "size", "date"
−S		sort by file size
	−−user, −−group, −−perms, −−date, −−linkcount, −−links	show individual fields
	−−time−style=STYLE	use specified time format

command *cmd args...*

execute given command ignoring aliases.

debug [**OPTS**] *level*|**off**

Switch debugging to *level* or turn it off. Options:

−T	truncate output file
−o <file>	redirect debug output to the file
−c	show message context
−p	show PID
−t	show timestamps

du [**OPTS**] *path...*

Summarize disk usage. Options:

−a,	−−all	write counts for all files, not just directories
	−−block−size=SIZ	use SIZ-byte blocks
−b,	−−bytes	print size in bytes
−c,	−−total	produce a grand total

−d,	−−max−depth=N	print the total for a directory (or file, with −−all) only if it is N or fewer levels below the command line argument; −−max−depth=0 is the same as −−summarize
−F,	−−files	print number of files instead of sizes
−h,	−−human−readable	print sizes in human readable format (e.g., 1K 234M 2G)
−H,	−−si	likewise, but use powers of 1000 not 1024
−k,	−−kilobytes	like −−block−size=1024
−m,	−−megabytes	like −−block−size=1048576
−S,	−−separate−dirs	do not include size of subdirectories
−s,	−−summarize	display only a total for each argument
	−−exclude=PAT	exclude files that match PAT

echo [**−n**] *string*

Prints (echos) the given string to the display.

edit [**OPTS**] *file*

Retrieve remote file to a temporary location, run a local editor on it and upload the file back if changed. Options:

−k	keep the temporary file
−o <temp>	explicit temporary file location

eval [**-f** *format*] *args...*

without -f it executes given arguments as a command. With -f, arguments are transformed into a new command. The format can contain plain text and placeholders $0...$9 and $@, corresponding to the arguments.

exit [**bg**] [**top**] [**parent**] [**kill**] [*code*]

exit will exit from lftp or move to background if there are active jobs. If no job is active, *code* is passed to operating system as lftp's termination status. If *code* is omitted, the exit code of last command is used.

'exit bg' forces moving to background when cmd:move-background is false. 'exit top' makes top level 'shell' (internal lftp command executor) terminate. 'exit parent' terminates the parent shell when running a nested script. 'exit kill' kills all numbered jobs before exiting. The options can be combined, e.g. 'at 08:00 −− exit top kill &' kills all jobs and makes lftp exit at specified time.

fg

Alias for 'wait'.

find [**OPTS**] *directory...*

List files in the directory (current directory by default) recursively. This can help with servers lacking ls −R support. You can redirect output of this command. Options:

−d *MD*,	−−max−depth=*MD*	specify maximum scan depth
−l,	−−ls	use long listing format

ftpcopy

Obsolete. Use one of the following instead:

```
get ftp://... −o ftp://...
get −O ftp://... file1 file2...
put ftp://...
mput ftp://.../*
mget −O ftp://... ftp://.../*
```

or other combinations to get FXP transfer (directly between two FTP servers). lftp would fallback to plain copy (via client) if FXP transfer cannot be initiated or ftp:use-fxp is false.

get [**−E**] [**−a**] [**−c**] [**−e**] [**−P** *N*] [**−O** *base*] *rfile* [**−o** *lfile*] ...

Retrieve the remote file *rfile* and store it as the local file *lfile*. If −o is omitted, the file is stored to local file named as base name of *rfile*. You can get multiple files by specifying multiple instances of *rfile* (and −o *lfile*). Does not expand wildcards, use **mget** for that.

−c	continue, reget
−E	delete source files after successful transfer
−e	delete target file before the transfer
−a	use ascii mode (binary is the default)
−P *N*	download *N* files in parallel
−O <base>	specifies base directory or URL where files should be placed

Examples:

get README
get README −o debian.README
get README README.mirrors
get README −o debian.README README.mirrors −o debian.mirrors
get README −o ftp://some.host.org/debian.README
get README −o ftp://some.host.org/debian-dir/ **(end slash is important)**

get1 [*OPTS*] *rfile*

Transfer a single file. Options:

−o <lfile>	destination file name (default - basename of rfile)
−c	continue, reget
−E	delete source files after successful transfer
−a	use ascii mode (binary is the default)
−d	create the directory of the target file
−−source−region=<from-to>	transfer specified region of source file
−−target−position=<pos>	position in target file to write data at

glob [*OPTS*] [*command*] *patterns*

Glob given patterns containing metacharacters and pass result to given command or return appropriate exit code.

−f	plain files (default)
−d	directories
−a	all types
−−exist	return zero exit code when the patterns expand to non-empty list
−−not−exist	return zero exit code when the patterns expand to an empty list

Examples:

glob echo *
glob --exist *.csv && echo "There are *.csv files"

help [*cmd*]

Print help for *cmd* or if no *cmd* was specified print a list of available commands.

history [**OPTS**] [*cnt*]

View or manipulate the command history. Optional argument *cnt* specifies the number of history lines to list, or "all" to list all entries. Options:

−w <file>	Write history to file.
−r <file>	Read history from file; appends to current history.
−c	Clear the history.
−l	List the history (default).

jobs [*OPTS*] [*job_no...*]

List running jobs. If *job_no* is specified, only list a job with that number. Options:

−v verbose, several −v increase verbosity
−r list just one specified job without recursion

kill all|*job_no*

Delete specified job with *job_no* or all jobs. (For *job_no* see **jobs**)

lcd *ldir*

Change current local directory *ldir*. The previous local directory is stored as '−'. You can do 'lcd −' to change the directory back.

ln [**−s**] *existing-file new-link*

Make a hard/symbolic link to an existing file. Option −s selects creation of a symbolic link.

local *command*

Run specified command with local directory file:// session instead of remote session. Examples:

local pwd
local ls
local mirror /dir1 /dir2

lpwd

Print current working directory on local machine.

ls *params*

List remote files. You can redirect output of this command to file or via pipe to external command. By default, ls output is cached, to see new listing use **rels** or **cache flush.**

mget [**−c**] [**−d**] [**−a**] [**−E**] [**−e**] [**−P** *N*] [**−O** *base*] *files*

Gets selected files with expanded wildcards.

−c	continue, reget.
−d	create directories the same as file names and get the files into them instead of current directory.
−E	delete source files after successful transfer
−e	delete target file before the transfer
−a	use ascii mode (binary is the default)
−P *N*	download *N* files in parallel
−O <base>	specifies base directory or URL where files should be placed

mirror [*OPTS*] [*source* [*target*]]

Mirror specified source directory to the target directory.

By default the source is remote and the target is a local directory. When using −R, the source directory is local and the target is remote. If the target directory is omitted, base name of the source directory is used. If both directories are omitted, current local and remote directories are used.

The source and/or the target may be URLs pointing to directories.

If the target directory ends with a slash (except the root directory) then base name of the source directory is appended.

−c,	−−continue	continue a mirror job if possible
−e,	−−delete	delete files not present at the source
	−−delete-excluded	delete files excluded at the target

	−−delete−first	delete old files before transferring new ones
	−−depth−first	descend into subdirectories before transferring files
	−−scan−all−first	scan all directories recursively before transferring files
−s,	−−allow−suid	set suid/sgid bits according to the source
	−−allow−chown	try to set owner and group on files
	−−ascii	use ascii mode transfers (implies −−ignore−size)
	−−ignore−time	ignore time when deciding whether to download
	−−ignore−size	ignore size when deciding whether to download
	−−only−missing	download only missing files
	−−only−existing	download only files already existing at target
−n,	−−only−newer	download only newer files (−c won't work)
	−−upload−older	upload even files older than the target ones
	−−transfer−all	transfer all files, even seemingly the same at the target site
	−−no−empty−dirs	don't create empty directories (implies −−depth−first)
−r,	−−no−recursion	don't go to subdirectories
	−−recursion=*MODE*	go to subdirectories on a condition
	−−no−symlinks	don't create symbolic links
−p,	−−no−perms	don't set file permissions
	−−no−umask	don't apply umask to file modes
−R,	−−reverse	reverse mirror (put files)
−L,	−−dereference	download symbolic links as files
	−−overwrite	overwrite plain files without removing them first
	−−no−overwrite	remove and re-create plain files instead of overwriting
−N,	−−newer−than=*SPEC*	download only files newer than specified time
	−−older−than=*SPEC*	download only files older than specified time
	−−size−range=*RANGE*	download only files with size in specified range
−P,	−−parallel[=*N*]	download N files in parallel
	−−use-pget[−n=*N*]	use pget to transfer every single file
	−−on−change=*CMD*	execute the command if anything has been changed
	−−loop	repeat mirror until no changes found
−i *RX*,	−−include=*RX*	include matching files
−x *RX*,	−−exclude=*RX*	exclude matching files
−I *GP*,	−−include−glob=*GP*	include matching files
−X *GP*,	−−exclude−glob=*GP*	exclude matching files
	−−include−rx−from=*FILE*	
	−−exclude−rx−from=*FILE*	
	−−include−glob−from=*FILE*	
	−−exclude−glob−from=*FILE*	load include/exclude patterns from the file, one per line
−f *FILE*,	−−file=*FILE*	mirror a single file or globbed group (e.g. /path/to/*.txt)
−F *DIR*,	−−directory=*DIR*	mirror a single directory or globbed group (e.g. /path/to/dir*)
−O *DIR*,	−−target-directory=*DIR*	target base path or URL
−v,	−−verbose[=level]	verbose operation
	−−log=*FILE*	write lftp commands being executed to FILE
	−−script=*FILE*	write lftp commands to FILE, but don't execute them
	−−just-print, −−dry-run	same as −−script=−
	−−max−errors=*N*	stop after this number of errors
	−−skip−noaccess	don't try to transfer files with no read access.
	−−use-cache	use cached directory listings
	−−Remove−source−files	remove source files after transfer (use with caution)

−−Remove−source−dirs	remove source files and directories after transfer (use with caution). Top level directory is not removed if it's name ends with a slash.
−−Move	same as −−Remove−source−dirs
−a	same as −−allow-chown −−allow-suid −−no-umask

RX is an extended regular expression, just like in **egrep**(1).

GP is a glob pattern, e.g. '*.zip'.

Include and exclude options can be specified multiple times. It means that a file or directory would be mirrored if it matches an include and does not match to excludes after the include, or does not match anything and the first check is exclude. Directories are matched with a slash appended.

Note that symbolic links are not created when uploading to remote server, because FTP protocol cannot do it. To upload files the links refer to, use 'mirror −RL' command (treat symbolic links as files).

For options −−newer−than and −−older−than you can either specify a file or time specification like that used by **at**(1) command, e.g. 'now-7days' or 'week ago'. If you specify a file, then modification time of that file will be used.

Verbosity level can be selected using −−verbose=level option or by several −v options, e.g. −vvv. Levels are:

0 - no output (default)
1 - print actions
2 - +print not deleted file names (when −e is not specified)
3 - +print directory names which are mirrored

−−only−newer turns off file size comparison and uploads/downloads only newer files even if size is different. By default older files are transferred and replace newer ones.

−−upload−older allows replacing newer remote files with older ones (when the target side is remote). Some remote back-ends cannot preserve timestamps so the default is to keep newer files.

Recursion mode can be one of 'always', 'never', 'missing', 'newer'. With the option 'newer' mirror compares timestamps of directories and enters a directory only if it is older or missing on the target side. Be aware that when a file changes the directory timestamp may stay the same, so mirror won't process that directory.

The options −−file and −−directory may be used multiple times and even mixed provided that base directories of the paths are the same.

You can mirror between two servers if you specify URLs instead of directories. FXP is automatically used for transfers between FTP servers, if possible.

Some FTP servers hide dot-files by default (e.g. *.htaccess*), and show them only when LIST command is used with −a option. In such case try to use 'set ftp:list-options −a'.

The recursion modes 'newer' and 'missing' conflict with −−scan−all−first, −−depth−first, −−no−empty−dirs and setting mirror:no−empty−dirs=true.

mkdir [**−p**] [**−f**] *dir(s)*

Make remote directories. If −p is used, make all components of paths. The −f option makes mkdir quiet and suppresses messages.

module *module* [*args*]

Load given module using **dlopen**(3) function. If module name does not contain a slash, it is searched in directories specified by module:path variable. Arguments are passed to module_init function. See README.modules for technical details.

more *files*

Same as ‘cat *files* | more’. if **PAGER** is set, it is used as filter. (See also **cat**, **zcat** and **zmore**)

mput [**−c**] [**−d**] [**−a**] [**−E**] [**−e**] [**−P** *N*] [**−O** *base*] *files*

Upload files with wildcard expansion. By default it uses the base name of local name as remote one. This can be changed by ‘−d’ option.

−c	continue, reput
−d	create directories the same as in file names and put the files into them instead of current directory
−E	delete source files after successful transfer (dangerous)
−e	delete target file before the transfer
−a	use ascii mode (binary is the default)
−P *N*	upload *N* files in parallel
−O <base>	specifies base directory or URL where files should be placed

mrm *file(s)*

Same as ‘glob rm’. Removes specified file(s) with wildcard expansion.

mmv [**−O** *directory*] *file(s) directory*

Move specified files to a target directory. The target directory can be specified after −O option or as the last argument.

−O <dir>	specifies the target directory where files should be placed

mv *file1 file2*

Rename *file1* to *file2*. No wildcard exmapsion is performed. If you give more than two arguments, or the last argument ends with a slash, then **mmv** command is executed instead.

nlist [*args*]

List remote file names

open [*OPTS*] *site*

Select a server by host name, URL or bookmark. When an URL or bookmark is given, automatically change the current working directory to the directory of the URL. Options:

−e *cmd*	execute the command just after selecting the server
−u *user*[,*pass*]	use the user/password for authentication
−p *port*	use the port for connection
−s *slot*	assign the connection to this slot
−d	enable debug
−B	don’t look up bookmarks
−−user *user*	use the user for authentication
−−password *pass*	use the password for authentication
−−env−password	take password from **LFTP_PASSWORD** environment variable
site	host name, URL or bookmark name

pget [*OPTS*] *rfile* [**-o** *lfile*]

Gets the specified file using several connections. This can speed up transfer, but loads the net and server heavily impacting other users. Use only if you really have to transfer the file ASAP. Options:

−c	continue transfer. Requires *lfile.lftp-pget-status* file.
−n *maxconn*	set maximum number of connections (default is taken from **pget:default-n** setting)

put [**−E**] [**−a**] [**−c**] [**−e**] [**−P** *N*] [**−O** *base*] *lfile* [**−o** *rfile*]

Upload *lfile* with remote name *rfile*. If −o omitted, the base name of *lfile* is used as remote name. Does not expand wildcards, use **mput** for that.

−o <rfile>	specifies remote file name (default - basename of lfile)
−c	continue, reput. It requires permission to overwrite remote files
−E	delete source files after successful transfer (dangerous)
−e	delete target file before the transfer
−a	use ascii mode (binary is the default)
−P *N*	upload *N* files in parallel
−O <base>	specifies base directory or URL where files should be placed

pwd [**−p**]

Print current remote URL. Use '−p' option to show password in the URL.

queue [**−n** *num*] *cmd*

Add the given command to queue for sequential execution. Each site has its own queue. '−n' adds the command before the given item in the queue. Don't try to queue 'cd' or 'lcd' commands, it may confuse lftp. Instead do the cd/lcd before 'queue' command, and it will remember the place in which the command is to be done. It is possible to queue up an already running job by 'queue wait <jobno>', but the job will continue execution even if it is not the first in queue.

'queue stop' will stop the queue, it will not execute any new commands, but already running jobs will continue to run. You can use 'queue stop' to create an empty stopped queue. 'queue start' will resume queue execution. When you exit lftp, it will start all stopped queues automatically.

'queue' with no arguments will either create a stopped queue or print queue status.

queue −−delete|-d [*index or wildcard expression*]

Delete one or more items from the queue. If no argument is given, the last entry in the queue is deleted.

queue −−move|-m <*index or wildcard expression*> [*index*]

Move the given items before the given queue index, or to the end if no destination is given.

−q	Be quiet.
−v	Be verbose.
−Q	Output in a format that can be used to re-queue. Useful with −−delete.

Examples:

```
> get file &
[1] get file
> queue wait 1
> queue get another_file
> cd a_directory
> queue get yet_another_file
```

queue −d 3	Delete the third item in the queue.
queue −m 6 4	Move the sixth item in the queue before the fourth.
queue −m "get*zip" 1	Move all commands matching "get*zip" to the beginning of the queue. (The order of the items is preserved.)
queue −d "get*zip"	Delete all commands matching "get*zip".

quote *cmd*

For FTP - send the command uninterpreted. Use with caution - it can lead to unknown remote state and thus will cause reconnect. You cannot be sure that any change of remote state because of quoted command is solid - it can be reset by reconnect at any time.

For HTTP - specific to HTTP action. Syntax: ''quote <command> [<args>]''. Command may be ''set-cookie'' or ''post''.

```
open http://www.site.net
quote set-cookie "variable=value; othervar=othervalue"
set http:post-content-type application/x-www-form-urlencoded
quote post /cgi-bin/script.cgi "var=value&othervar=othervalue" > local_file
```

For FISH - send the command uninterpreted. This can be used to execute arbitrary commands on server. The command must not take input or print ### at new line beginning. If it does, the protocol will become out of sync.

```
open fish://server
quote find -name \*.zip
```

reget *rfile* [**-o** *lfile*]

Same as 'get -c'.

rels [*args*]

Same as 'ls', but ignores the cache.

renlist [*args*]

Same as 'nlist', but ignores the cache.

repeat [*OPTS*] [[**-d**] *delay*] [*command*]

Repeat specified command with a delay between iterations. Default delay is one second, default command is empty.

-c <count>	maximum number of iterations
-d <delay>	delay between iterations
--while-ok	stop when command exits with non-zero code
--until-ok	stop when command exits with zero code
--weak	stop when lftp moves to background.

Examples:

```
repeat at tomorrow -- mirror
repeat 1d mirror
```

reput *lfile* [**-o** *rfile*]

Same as 'put -c'.

rm [**-r**] [**-f**] *files*

Remove remote files. Does not expand wildcards, use **mrm** for that. -r is for recursive directory remove. Be careful, if something goes wrong you can lose files. -f suppress error messages.

rmdir *dir(s)*

Remove remote directories.

scache [*session*]

List cached sessions or switch to specified session.

set [*var* [*val*]]

Set variable to given value. If the value is omitted, unset the variable. Variable name has format ''name/closure'', where closure can specify exact application of the setting. See below for details. If set is called with no variable then only altered settings are listed. It can be changed by options:

-a	list all settings, including default values

−d list only default values, not necessary current ones

site *site_cmd*

Execute site command *site_cmd* and output the result. You can redirect its output.

sleep *interval*

Sleep given time interval and exit. Interval is in seconds by default, but can be suffixed with 'm', 'h', 'd' for minutes, hours and days respectively. See also **at**.

slot [*name*]

Select specified slot or list all slots allocated. A slot is a connection to a server, somewhat like a virtual console. You can create multiple slots connected to different servers and switch between them. You can also use *slot:name* as a pseudo-URL evaluating to that slot location.

Default readline binding allows quick switching between slots named 0-9 using Meta-0 - Meta-9 keys (often you can use Alt instead of Meta).

source *file*
source −e *command*

Execute commands recorded in file *file* or returned by specified external command.

source ~/.lftp/rc
source −e echo help

suspend

Stop lftp process. Note that transfers will be also stopped until you continue the process with shell's fg or bg commands.

torrent [**OPTS**] *torrent-files...*

Start BitTorrent process for the given *torrent-files*, which can be a local file, URL, magnet link or plain *info_hash* written in hex or base32. Local wildcards are expanded. Existing files are first validated unless *−−force−valid* option is given. Missing pieces are downloaded. Files are stored in specified *directory* or current working directory by default. Seeding continues until ratio reaches *torrent:stop-on-ratio* setting or time of *torrent:seed-max-time* runs out.

Options:

−O <directory>	specifies base directory where files should be placed
−−force−valid	skip file validation (if you are sure they are ok).
−−only−new	stop if the metadata is known already or the torrent is complete.
−−only−incomplete	stop if the torrent is already complete.
−−dht−bootstrap=<node>	bootstrap DHT by sending a query to specified *node*. This option should be used just once to fill the local node cache. Port number may be given after colon, default is 6881. Here are some nodes for bootstrapping: dht.transmissionbt.com, router.utorrent.com, router.bittorrent.com.
−−share	share specified file or directory using BitTorrent protocol. Magnet link is printed when it's ready.

user *user* [*pass*]
user *URL* [*pass*]

Use specified info for remote login. If you specify an URL with user name, the entered password will be cached so that future URL references can use it.

version

Print **lftp** version.

wait [*jobno*]
wait all

Wait for specified job to terminate. If jobno is omitted, wait for last backgrounded job.

'wait all' waits for all jobs to terminate.

zcat *files*

Same as cat, but filter each file through zcat. (See also **cat**, **more** and **zmore**)

zmore *files*

Same as more, but filter each file through zcat. (See also **cat**, **zcat** and **more**)

Settings

On startup, lftp executes *~/.lftprc* and *~/.lftp/rc* (or *~/.config/lftp/rc* if *~/.lftp* does not exist). You can place aliases and 'set' commands there. Some people prefer to see full protocol debug, use 'debug' to turn the debug on.

There is also a system-wide startup file in */etc/lftp.conf* . It can be in different directory, see FILES section.

lftp has the following settable variables (you can also use 'set −a' to see all variables and their values):

bmk:save-passwords (boolean)
save plain text passwords in *~/.local/share/lftp/bookmarks* or *~/.lftp/bookmarks* on 'bookmark add' command. Off by default.

cache:cache-empty-listings (boolean)
When false, empty listings are not cached.

cache:enable (boolean)
When false, cache is disabled.

cache:expire (time interval)
Positive cache entries expire in this time interval.

cache:expire-negative (time interval)
Negative cache entries expire in this time interval.

cache:size (number)
Maximum cache size. When exceeded, oldest cache entries will be removed from cache.

cmd:at-exit (string)
the commands in string are executed before lftp exits or moves to background.

cmd:at-exit-bg (string)
the commands in string are executed before backgrounded lftp exits.

cmd:at-exit-fg (string)
the commands in string are executed before foreground lftp exits.

cmd:at-background (string)
the commands in string are executed before lftp moves to background.

cmd:at-terminate (string)
the commands in string are executed before lftp terminates (either backgrounded or foreground).

cmd:at-finish (string)
the commands in string are executed once when all jobs are done.

cmd:at-queue-finish (string)
the commands in string are executed once when all jobs in a queue are done.

cmd:cls-completion-default (string)
default **cls** options for displaying completion choices. For example, to make completion listings show file sizes, set cmd:cls-completion-default to '-s'.

cmd:cls-default (string)
default **cls** command options. They can be overridden by explicitly given options.

cmd:cls-exact-time (boolean)
when true, **cls** would try to get exact file modification time even if it means more requests to the server.

cmd:csh-history (boolean)
enables csh-like history expansion.

cmd:default-protocol (string)
The value is used when 'open' is used with just host name without protocol. Default is 'ftp'.

cmd:fail-exit (boolean)
if true, exit when a command fails and the following command is unconditional (i.e. does not begin with || or &&). lftp exits after the unconditional command is issued without executing it.

cmd:interactive (tri-boolean)
when true, lftp acts interactively, handles terminal signals and outputs some extra messages. Default is auto and depends on stdin being a terminal.

cmd:long-running (seconds)
time of command execution, which is considered as 'long' and a beep is done before next prompt. 0 means off.

cmd:ls-default (string)
default ls argument

cmd:move-background (boolean)
when false, lftp refuses to go to background when exiting. To force it, use 'exit bg'.

cmd:move-background-detach (boolean)
when true (default), lftp detaches itself from the control terminal when moving to background, it is possible to attach back using 'attach' command; when false, lftp tricks the shell to move lftp to background process group and continues to run, then fg shell command brings lftp back to foreground unless it has done all jobs and terminated.

cmd:prompt (string)
The prompt. lftp recognizes the following backslash-escaped special characters that are decoded as follows:

\@	insert @ if the current remote site user is not default
\a	an ASCII bell character (07)
\e	an ASCII escape character (033)
\h	the remote hostname you are connected to
\n	newline
\s	the name of the client (lftp)
\S	current slot name
\u	the username of the remote site user you are logged in as
\U	the URL of the remote site (e.g., ftp://g437.ub.gu.se/home/james/src/lftp)
\v	the version of **lftp** (e.g., 2.0.3)
\w	the current working directory at the remote site
\W	the base name of the current working directory at the remote site
\l	the current working directory at the local site
\L	the base name of the current working directory at the local site

nnn the character corresponding to the octal number *nnn*
\\ a backslash
\? skips next character if previous substitution was empty.
\[begin a sequence of non-printing characters, which could be used to embed a terminal control sequence into the prompt
\] end a sequence of non-printing characters

cmd:parallel (number)
Number of jobs run in parallel in non-interactive mode. For example, this may be useful for scripts with multiple 'get' commands. Note that setting this to a value greater than 1 changes conditional execution behaviour, basically makes it inconsistent.

cmd:queue-parallel (number)
Number of jobs run in parallel in a queue.

cmd:remote-completion (boolean)
a boolean to control whether or not lftp uses remote completion. When true, **Tab** key guesses if the word being completed should be a remote file name. **Meta-Tab** does remote completion always. So you can force remote completion with **Meta-Tab** when **cmd:remote-completion** is false or when the guess is wrong.

cmd:save-cwd-history (boolean)
when true, lftp saves last CWD of each site to *~/.local/share/lftp/cwd_history* or *~/.lftp/cwd_history*, allowing to do ''cd -'' after lftp restart. Default is true.

cmd:save-rl-history (boolean)
when true, lftp saves readline history to *~/.local/share/lftp/rl_history* or *~/.lftp/rl_history* on exit. Default is true.

cmd:show-status (boolean)
when false, lftp does not show status line on terminal. Default is true.

cmd:set-term-status (boolean)
when true, lftp updates terminal status if supported (e.g. xterm). The closure for this setting is the terminal type from TERM environment variable.

cmd:status-interval (time**interval)**
the time interval between status updates.

cmd:stifle-rl-history (number)
the number of lines to keep in readline history.

cmd:term-status (string)
the format string to use to display terminal status. The closure for this setting is the terminal type from TERM environment variable. Default uses ''tsl'' and ''fsl'' termcap values.

The following escapes are supported:

\a bell
\e escape
\n new line
\s "lftp"
\v lftp version
\T the status string

cmd:time-style (string)
This setting is the default value for cls −−time−style option.

cmd:trace (boolean)
when true, lftp prints the commands it executes (like sh −x).

cmd:verify-host (boolean)
if true, lftp resolves host name immediately in ‘open’ command. It is also possible to skip the check for a single ‘open’ command if ‘&’ is given, or if ^Z is pressed during the check.

cmd:verify-path (boolean)
if true, lftp checks the path given in ‘cd’ command. It is also possible to skip the check for a single ‘cd’ command if ‘&’ is given, or if ^Z is pressed during the check. Examples:
set cmd:verify-path/hftp://* false
cd directory &

cmd:verify-path-cached (boolean)
When false, ‘cd’ to a directory known from cache as existent will succeed immediately. Otherwise the verification will depend on cmd:verify-path setting.

color:use-color (tri-boolean)
when true, cls command and completion output colored file listings according to color:dir-colors setting. When set to auto, colors are used when output is a terminal.

color:dir-colors (string)
file listing color description. By default the value of LS_COLORS environment variable is used. See dircolors(1).

dns:SRV-query (boolean)
query for SRV records and use them before gethostbyname. The SRV records are only used if port is not explicitly specified. See RFC2052 for details.

dns:cache-enable (boolean)
enable DNS cache. If it is off, lftp resolves host name each time it reconnects.

dns:cache-expire (time interval)
time to live for DNS cache entries. It has format <number><unit>+, e.g. 1d12h30m5s or just 36h. To disable expiration, set it to ‘inf’ or ‘never’.

dns:cache-size (number)
maximum number of DNS cache entries.

dns:fatal-timeout (time interval)
limit the time for DNS queries. If DNS server is unavailable too long, lftp will fail to resolve a given host name. Set to ‘never’ to disable.

dns:order (list of protocol names)
sets the order of DNS queries. Default is ‘‘inet6 inet’’ which means first look up address in inet6 family, then inet and use them in that order. To disable inet6 (AAAA) lookup, set this variable to ‘‘inet’’.

dns:use-fork (boolean)
if true, lftp will fork before resolving host address. Default is true.

dns:max-retries (number)
If zero, there is no limit on the number of times lftp will try to lookup an address. If > 0, lftp will try only this number of times to look up an address of each address family in dns:order.

dns:name (string)
This setting can be used to substitute a host name alias with another name or IP address. The host name alias is used as the setting closure, the substituted name or IP address is in the value. Multiple names or IP addresses can be separated by comma.

file:charset (string)
local character set. It is set from current locale initially.

file:use-lock (boolean)
when true, lftp uses advisory locking on local files when opening them.

file:use-fallocate (boolean)
when true, lftp uses fallocate(2) or posix_fallocate(3) to pre-allocate storage space and reduce file fragmentation in pget and torrent commands.

fish:auto-confirm (boolean)
when true, lftp answers ‘‘yes’’ to all ssh questions, in particular to the question about a new host key. Otherwise it answers ‘‘no’’.

fish:charset (string)
the character set used by fish server in requests, replies and file listings. Default is empty which means the same as local.

fish:connect-program (string)
the program to use for connecting to remote server. It should support ‘−l’ option for user name, ‘−p’ for port number. Default is ‘ssh −a −x’. You can set it to ‘rsh’, for example. For private key authentication add ‘−i’ option with the key file.

fish:shell (string)
use specified shell on server side. Default is /bin/sh. On some systems, /bin/sh exits when doing cd to a non-existent directory. lftp can handle that but it has to reconnect. Set it to /bin/bash for such systems if bash is installed.

ftp:acct (string)
Send this string in ACCT command after login. The result is ignored. The closure for this setting has format *user@host*.

ftp:anon-pass (string)
sets the password used for anonymous FTP access authentication. Default is "lftp@".

ftp:anon-user (string)
sets the user name used for anonymous FTP access authentication. Default is "anonymous".

ftp:auto-sync-mode (regex)
if first server message matches this regex, turn on sync mode for that host.

ftp:catch-size (boolean)
when there is no support for SIZE command, try to catch file size from the "150 Opening data connection" reply.

ftp:charset (string)
the character set used by FTP server in requests, replies and file listings. Default is empty which means the same as local. This setting is only used when the server does not support UTF8.

ftp:client (string)
the name of FTP client to send with CLNT command, if supported by server. If it is empty, then no CLNT command will be sent.

ftp:compressed-re (regex)
files with matching name will be considered compressed and "MODE Z" will not be used for them.

ftp:bind-data-socket (boolean)
bind data socket to the interface of control connection (in passive mode). Default is true, exception is the loopback interface.

ftp:fix-pasv-address (boolean)
if true, lftp will try to correct address returned by server for PASV command in case when server address is in public network and PASV returns an address from a private network. In this case lftp would substitute server address instead of the one returned by PASV command, port number would not be changed. Default is true.

ftp:fxp-passive-source (boolean)
: if true, lftp will try to set up source FTP server in passive mode first, otherwise destination one. If first attempt fails, lftp tries to set them up the other way. If the other disposition fails too, lftp falls back to plain copy. See also ftp:use-fxp.

ftp:home (string)
: Initial directory. Default is empty string which means auto. Set this to '/' if you don't like the look of %2F in FTP URLs. The closure for this setting has format *user@host*.

ftp:ignore-pasv-address (boolean)
: If true, lftp uses control connection address instead of the one returned in PASV reply for data connection. This can be useful for broken NATs. Default is false.

ftp:list-empty-ok (boolean)
: if set to false, empty lists from LIST command will be treated as incorrect, and another method (NLST) will be used.

ftp:list-options (string)
: sets options which are always appended to LIST command. It can be useful to set this to '−a' if server does not show dot (hidden) files by default. Default is empty.

ftp:mode-z-level (number)
: compression level (0-9) for uploading with MODE Z.

ftp:nop-interval (seconds)
: delay between NOOP commands when downloading tail of a file. This is useful for FTP servers which send "Transfer complete" message before flushing data transfer. In such cases NOOP commands can prevent connection timeout.

ftp:passive-mode (boolean)
: sets passive FTP mode. This can be useful if you are behind a firewall or a dumb masquerading router. In passive mode lftp uses PASV command, not the PORT command which is used in active mode. In passive mode lftp itself makes the data connection to the server; in active mode the server connects to lftp for data transfer. Passive mode is the default.

ftp:port-ipv4 (ipv4 address)
: specifies an IPv4 address to send with PORT command. Default is empty which means to send the address of local end of control connection.

ftp:port-range (from-to)
: allowed port range for the local side of the data connection. Format is min-max, or 'full' or 'any' to indicate any port. Default is 'full'.

ftp:prefer-epsv (boolean)
: use EPSV as preferred passive mode. Default is 'false'.

ftp:proxy (URL)
: specifies FTP proxy to use. To disable proxy set this to empty string. Note that it is a FTP proxy which uses FTP protocol, not FTP over HTTP. Default value is taken from environment variable **ftp_proxy** if it starts with ''ftp://''. If your FTP proxy requires authentication, specify user name and password in the URL. If ftp:proxy starts with http:// then hftp protocol (FTP over HTTP proxy) is used instead of FTP automatically.

ftp:proxy-auth-type (string)
: When set to ''joined'', lftp sends ''user@proxy_user@ftp.example.org'' as user name to proxy, and ''password@proxy_password'' as password.

  When set to ''joined-acct'', lftp sends ''user@ftp.example.org proxy_user'' (with space) as user name to proxy. The site password is sent as usual and the proxy password is expected in the following ACCT command.

  When set to ''open'', lftp first sends proxy user and proxy password and then ''OPEN ftp.example.org'' followed by ''USER user''. The site password is then sent as usual.

When set to ‘‘user’’ (default), lftp first sends proxy user and proxy password and then ‘‘user@ftp.example.org’’ as user name. The site password is then sent as usual.

When set to ‘‘proxy-user@host’’, lftp first sends ‘‘USER proxy_user@ftp.example.org’’, then proxy password. The site user and password are then sent as usual.

ftp:rest-list (boolean)
allow usage of REST command before LIST command. This might be useful for large directories, but some FTP servers silently ignore REST before LIST.

ftp:rest-stor (boolean)
if false, lftp will not try to use REST before STOR. This can be useful for some buggy servers which corrupt (fill with zeros) the file if REST followed by STOR is used.

ftp:retry-530 (regex)
Retry on server reply 530 for PASS command if text matches this regular expression. This setting should be useful to distinguish between overloaded server (temporary condition) and incorrect password (permanent condition).

ftp:retry-530-anonymous (regex)
Additional regular expression for anonymous login, like ftp:retry-530.

ftp:site-group (string)
Send this string in SITE GROUP command after login. The result is ignored. The closure for this setting has format *user@host*.

ftp:skey-allow (boolean)
allow sending skey/opie reply if server appears to support it. On by default.

ftp:skey-force (boolean)
do not send plain text password over the network, use skey/opie instead. If skey/opie is not available, assume failed login. Off by default.

ftp:ssl-allow (boolean)
if true, try to negotiate SSL connection with FTP server for non-anonymous access. Default is true. This and other SSL settings are only available if lftp was compiled with an ssl/tls library.

ftp:ssl-auth (string)
the argument for AUTH command, can be one of SSL, TLS, TLS-P, TLS-C. See RFC4217 for explanations. By default TLS or SSL will be used, depending on FEAT reply.

ftp:ssl-data-use-keys (boolean)
if true, lftp loads ssl:key-file for protected data connection too. When false, it does not, and the server can match data and control connections by session ID. Default is true.

ftp:ssl-force (boolean)
if true, refuse to send password in clear when server does not support SSL. Default is false.

ftp:ssl-protect-data (boolean)
if true, request SSL connection for data transfers. This provides privacy and transmission error correction. Was cpu-intensive on old CPUs. Default is true.

ftp:ssl-protect-fxp (boolean)
if true, request SSL connection for data transfer between two FTP servers in FXP mode. CPSV or SSCN command will be used in that case. If SSL connection fails for some reason, lftp would try unprotected FXP transfer unless ftp:ssl-force is set for any of the two servers. Default is true.

ftp:ssl-protect-list (boolean)
if true, request SSL connection for file list transfers. Default is true.

ftp:ssl-use-ccc (boolean)
if true, lftp would issue CCC command after logon, thus disable ssl protection layer on control connection.

ftp:stat-interval (time interval)
interval between STAT commands. Default is 1 second.

ftp:strict-multiline (boolean)
when true, lftp strictly checks for multiline reply format (expects it to end with the same code as it started with). When false, this check is relaxed.

ftp:sync-mode (boolean)
if true, lftp will send one command at a time and wait for response. This might be useful if you are using a buggy FTP server or router. When it is off, lftp sends a pack of commands and waits for responses - it speeds up operation when round trip time is significant. Unfortunately it does not work with all FTP servers and some routers have troubles with it, so it is on by default.

ftp:timezone (string)
Assume this timezone for time in listings returned by LIST command. This setting can be GMT offset [+|-]HH[:MM[:SS]] or any valid TZ value (e.g. Europe/Moscow or MSK-3MSD,M3.5.0,M10.5.0/3). The default is GMT. Set it to an empty value to assume local timezone specified by environment variable TZ.

ftp:too-many-re (regexp)
Decrease the dynamic connection limit when 421 reply line matches this regular expression.

ftp:trust-feat (string)
When true, assume that FEAT returned data are correct and don't use common protocol extensions like SIZE, MDTM, REST if they are not listed. Default is false.

ftp:use-abor (boolean)
if false, lftp does not send ABOR command but closes data connection immediately.

ftp:use-allo (boolean)
when true, lftp sends ALLO command before uploading a file.

ftp:use-feat (boolean)
when true (default), lftp uses FEAT command to determine extended features of ftp server.

ftp:use-fxp (boolean)
if true, lftp will try to set up direct connection between two ftp servers.

ftp:use-hftp (boolean)
when ftp:proxy points to an http proxy, this setting selects hftp method (GET, HEAD) when true, and CONNECT method when false. Default is true.

ftp:use-ip-tos (boolean)
when true, lftp uses IPTOS_LOWDELAY for control connection and IPTOS_THROUGHPUT for data connections.

ftp:lang (boolean)
the language selected with LANG command, if supported as indicated by FEAT response. Default is empty which means server default.

ftp:use-mdtm (boolean)
when true (default), lftp uses MDTM command to determine file modification time.

ftp:use-mdtm-overloaded (boolean)
when true, lftp uses two argument MDTM command to set file modification time on uploaded files. Default is false.

ftp:use-mlsd (boolean)
when true, lftp will use MLSD command for directory listing if supported by the server.

ftp:use-mode-z (boolean)
when true, lftp will use "MODE Z" if supported by the server to perform compressed transfers.

ftp:use-site-idle (boolean)
when true, lftp sends 'SITE IDLE' command with net:idle argument. Default is false.

ftp:use-site-utime (boolean)
when true, lftp sends 5-argument 'SITE UTIME' command to set file modification time on uploaded files. Default is true.

ftp:use-site-utime2 (boolean)
when true, lftp sends 2-argument 'SITE UTIME' command to set file modification time on uploaded files. Default is true. If 5-argument 'SITE UTIME' is also enabled, 2-argument command is tried first.

ftp:use-size (boolean)
when true (default), lftp uses SIZE command to determine file size.

ftp:use-stat (boolean)
if true, lftp sends STAT command in FXP mode transfer to know how much data has been transferred. See also ftp:stat-interval. Default is true.

ftp:use-stat-for-list (boolean)
when true, lftp uses STAT instead of LIST command. By default '.' is used as STAT argument. Using STAT, lftp avoids creating data connection for directory listing. Some servers require special options for STAT, use ftp:list-options to specify them (e.g. **−la**).

ftp:use-telnet-iac (boolean)
when true (default), lftp uses TELNET IAC command and follows TELNET protocol as specified in RFC959. When false, it does not follow TELNET protocol and thus does not double 255 (0xFF, 0377) character and does not prefix ABOR and STAT commands with TELNET IP+SYNCH signal.

ftp:use-tvfs (tri-boolean)
When set to auto, usage of TVFS feature depends on FEAT server reply. Otherwise this setting tells whether use it or not. In short, if a server supports TVFS feature then it uses unix-like paths.

ftp:use-utf8 (boolean)
if true, lftp sends 'OPTS UTF8 ON' to the server to activate UTF-8 encoding (if supported). Disable it if the file names have a different encoding and the server has a trouble with it.

ftp:use-quit (boolean)
if true, lftp sends QUIT before disconnecting from ftp server. Default is true.

ftp:verify-address (boolean)
verify that data connection comes from the network address of control connection peer. This can possibly prevent data connection spoofing which can lead to data corruption. Unfortunately, this can fail for certain ftp servers with several network interfaces, when they do not set outgoing address on data socket, so it is disabled by default.

ftp:verify-port (boolean)
verify that data connection has port 20 (ftp-data) on its remote end. This can possibly prevent data connection spoofing by users of remote host. Unfortunately, too many windows and even unix ftp servers forget to set proper port on data connection, thus this check is off by default.

ftp:web-mode (boolean)
disconnect after closing data connection. This can be useful for totally broken ftp servers. Default is false.

ftps:initial-prot (string)
specifies initial PROT setting for FTPS connections. Should be one of: C, S, E, P, or empty. Default is empty which means unknown, so that lftp will use PROT command unconditionally. If PROT command turns out to be unsupported, then Clear mode would be assumed.

hftp:cache (boolean)
allow server/proxy side caching for ftp-over-http protocol.

hftp:cache-control (string)
specify corresponding HTTP request header.

hftp:decode (boolean)
when true, lftp automatically decodes the entity in hftp protocol when Content-Encoding header value matches deflate, gzip, compress, x-gzip or x-compress.

hftp:proxy (URL)
specifies HTTP proxy for FTP-over-HTTP protocol (hftp). The protocol hftp cannot work without a HTTP proxy, obviously. Default value is taken from environment variable **ftp_proxy** if it starts with ‘‘http://’’, otherwise from environment variable **http_proxy**. If your FTP proxy requires authentication, specify user name and password in the URL.

hftp:use-allprop (boolean)
if true, lftp will send ‘<allprop/>’ request body in ‘PROPFIND’ requests, otherwise it will send an empty request body.

hftp:use-authorization (boolean)
if set to off, lftp will send password as part of URL to the proxy. This may be required for some proxies (e.g. M-soft). Default is on, and lftp will send password as part of Authorization header.

hftp:use-head (boolean)
if set to off, lftp will try to use ‘GET’ instead of ‘HEAD’ for hftp protocol. While this is slower, it may allow lftp to work with some proxies which don’t understand or mishandle ‘‘HEAD ftp://’’ requests.

hftp:use-mkcol (boolean)
if set to off, lftp will try to use ‘PUT’ instead of ‘MKCOL’ to create directories with hftp protocol. Default is off.

hftp:use-propfind (boolean)
if set to off, lftp will not try to use ‘PROPFIND’ to get directory contents with hftp protocol and use ‘GET’ instead. Default is off. When enabled, lftp will also use PROPPATCH to set file modification time after uploading.

hftp:use-range (boolean)
when true, lftp will use Range header for transfer restart.

hftp:use-type (boolean)
If set to off, lftp won’t try to append ‘;type=’ to URLs passed to proxy. Some broken proxies don’t handle it correctly. Default is on.

http:accept, http:accept-charset, http:accept-encoding, http:accept-language (string)
specify corresponding HTTP request headers.

http:authorization (string)
the authorization to use by default, when no user is specified. The format is ‘‘user:password’’. Default is empty which means no authorization.

http:cache (boolean)
allow server/proxy side caching.

http:cache-control (string)
specify corresponding HTTP request header.

http:cookie (string)
send this cookie to server. A closure is useful here:
set cookie/www.somehost.com "param=value"

http:decode (boolean)
when true, lftp automatically decodes the entity when Content-Encoding header value matches deflate, gzip, compress, x-gzip or x-compress.

http:post-content-type (string)
specifies value of Content-Type HTTP request header for POST method. Default is ‘‘application/x-www-form-urlencoded’’.

http:proxy (URL)
specifies HTTP proxy. It is used when lftp works over HTTP protocol. Default value is taken from environment variable **http_proxy**. If your proxy requires authentication, specify user name and password in the URL.

http:put-method (PUT or POST)
specifies which HTTP method to use on put.

http:put-content-type (string)
specifies value of Content-Type HTTP request header for PUT method.

http:referer (string)
specifies value for Referer HTTP request header. Single dot ‘.’ expands to current directory URL. Default is ‘.’. Set to empty string to disable Referer header.

http:set-cookies (boolean)
if true, lftp modifies http:cookie variables when Set-Cookie header is received.

http:use-allprop (boolean)
if true, lftp will send ‘<allprop/>’ request body in ‘PROPFIND’ requests, otherwise it will send an empty request body.

http:use-mkcol (boolean)
if set to off, lftp will try to use ‘PUT’ instead of ‘MKCOL’ to create directories with HTTP protocol. Default is on.

http:use-propfind (boolean)
if set to off, lftp will not try to use ‘PROPFIND’ to get directory contents with HTTP protocol and use ‘GET’ instead. Default is off. When enabled, lftp will also use PROPPATCH to set ‘Last-Modified’ property after a file upload.

http:use-range (boolean)
when true, lftp will use Range header for transfer restart.

http:user-agent (string)
the string lftp sends in User-Agent header of HTTP request.

https:proxy (string)
specifies https proxy. Default value is taken from environment variable **https_proxy**.

log:enabled (boolean)
when true, the log messages are output. The closure for this and other ‘log:’ variables is either ‘debug’ for debug messages or ‘xfer’ for transfer logging.

log:file (string)
the target output file for logging. When empty, **stderr** is used.

log:level (number)
the log verbosity level. Currently it’s only defined for ‘debug’ closure.

log:max-size (number)
maximum size of the log file. When the size is reached, the file is renamed and started anew.

log:prefix-error (string)

log:prefix-note (string)

log:prefix-recv (string)

log:prefix-send (string)
the prefixes for corresponding types of debug messages.

log:show-ctx (boolean)

log:show-pid (boolean)

log:show-time (boolean)
select additional information in the log messages.

mirror:dereference (boolean)
when true, mirror will dereference symbolic links by default. You can override it by −−no−dereference option. Default if false.

mirror:exclude-regex (regex)
specifies default exclusion pattern. You can override it by −−include option.

mirror:include-regex (regex)
specifies default inclusion pattern. It is used just after mirror:exclude-regex is applied. It is never used if mirror:exclude-regex is empty.

mirror:no-empty-dirs (boolean)
when true, mirror doesn't create empty directories (like −−no−empty−dirs option).

mirror:sort-by (string)
specifies order of file transfers. Valid values are: name, name-desc, size, size-desc, date, date-desc. When the value is name or name-desc, then mirror:order setting also affects the order or transfers.

mirror:order (list of patterns)
specifies order of file transfers when sorting by name. E.g. setting this to "*.sfv *.sum" makes mirror to transfer files matching *.sfv first, then ones matching *.sum and then all other files. To process directories after other files, add "*/" to the end of pattern list.

mirror:overwrite (boolean)
when true, mirror will overwrite plain files instead of removing and re-creating them.

mirror:parallel-directories (boolean)
if true, mirror will start processing of several directories in parallel when it is in parallel mode. Otherwise, it will transfer files from a single directory before moving to other directories.

mirror:parallel-transfer-count (number)
specifies number of parallel transfers mirror is allowed to start. You can override it with −−parallel option. A closure can be matched against source or target host names, the minimum number greater than 0 is used.

mirror:require-source (boolean)
When true, mirror requires a source directory to be specified explicitly, otherwise it is supposed to be the current directory.

mirror:set-permissions (boolean)
When set to off, mirror won't try to copy file and directory permissions. You can override it by −−perms option. Default is on.

mirror:skip-noaccess (boolean)
when true, mirror does not try to download files which are obviously inaccessible by the permission mask. Default is false.

mirror:use-pget-n (number)
specifies −n option for pget command used to transfer every single file under mirror. A closure can be matched against source or target host names, the minimum number greater than 0 is used. When the value is less than 2, pget is not used.

module:path (string)
colon separated list of directories to look for modules. Can be initialized by environment variable LFTP_MODULE_PATH. Default is 'PKGLIBDIR/VERSION:PKGLIBDIR'.

net:connection-limit (number)
maximum number of concurrent connections to the same site. 0 means unlimited.

net:connection-limit-timer (time interval)
increase the dynamic connection limit after this time interval.

net:connection-takeover (boolean)
if true, foreground connections have priority over background ones and can interrupt background transfers to complete a foreground operation.

net:idle (time interval)
disconnect from server after this idle time. Default is 3 minutes.

net:limit-rate (bytes per second)
limit transfer rate on data connection. 0 means unlimited. You can specify two numbers separated by colon to limit download and upload rate separately. Suffixes are supported, e.g. 100K means 102400.

net:limit-max (bytes)
limit accumulating of unused limit-rate. 0 means twice of limit-rate.

net:limit-total-rate (bytes per second)
limit transfer rate of all connections in sum. 0 means unlimited. You can specify two numbers separated by colon to limit download and upload rate separately. Note that sockets have receive buffers on them, this can lead to network link load higher than this rate limit just after transfer beginning. You can try to set net:socket-buffer to relatively small value to avoid this.

If you specify a closure, then rate limitation will be applied to sum of connections to a single matching host.

net:limit-total-max (bytes)
limit accumulating of unused limit-total-rate. 0 means twice of limit-total-rate.

net:max-retries (number)
the maximum number of sequential tries of an operation without success. 0 means unlimited. 1 means no retries.

net:no-proxy (string)
contains comma separated list of domains for which proxy should not be used. Default is taken from environment variable **no_proxy**.

net:persist-retries (number)
ignore this number of hard errors. Useful to login to buggy FTP servers which reply 5xx when there is too many users.

net:reconnect-interval-base (seconds)
sets the base minimal time between reconnects. Actual interval depends on net:reconnect-interval-multiplier and number of attempts to perform an operation.

net:reconnect-interval-max (seconds)
sets maximum reconnect interval. When current interval after multiplication by net:reconnect-interval-multiplier reaches this value (or exceeds it), it is reset back to net:reconnect-interval-base.

net:reconnect-interval-multiplier (real number)
sets multiplier by which base interval is multiplied each time new attempt to perform an operation fails. When the interval reaches maximum, it is reset to base value. See net:reconnect-interval-base and net:reconnect-interval-max.

net:socket-bind-ipv4 (ipv4 address)
bind all IPv4 sockets to specified address. This can be useful to select a specific network interface to use. Default is empty which means not to bind IPv4 sockets, operating system will choose an address automatically using routing table.

net:socket-bind-ipv6 (ipv6 address)
the same for IPv6 sockets.

net:socket-buffer (bytes)
use given size for SO_SNDBUF and SO_RCVBUF socket options. 0 means system default.

net:socket-maxseg (bytes)
use given size for TCP_MAXSEG socket option. Not all operating systems support this option, but Linux does.

net:timeout (time interval)
sets the network protocol timeout.

pget:default-n (number)
default number of chunks to split the file to in pget.

pget:min-chunk-size (number)
minimal chunk size to split the file to.

pget:save-status (time interval)
save pget transfer status this often. Set to ‘never’ to disable saving of the status file. The status is saved to a file with suffix *.lftp-pget-status*.

sftp:auto-confirm (boolean)
when true, lftp answers ‘‘yes’’ to all ssh questions, in particular to the question about a new host key. Otherwise it answers ‘‘no’’.

sftp:charset (string)
the character set used by SFTP server in file names and file listings. Default is empty which means the same as local. This setting is only used for SFTP protocol version prior to 4. Version 4 and later always use UTF-8.

sftp:connect-program (string)
the program to use for connecting to remote server. It should support ‘−l’ option for user name, ‘−p’ for port number. Default is ‘ssh −a −x’. For private key authentication add ‘−i’ option with the key file.

sftp:max-packets-in-flight (number)
The maximum number of unreplied packets in flight. If round trip time is significant, you should increase this and size-read/size-write. Default is 16.

sftp:protocol-version (number)
The protocol number to negotiate. Default is 6. The actual protocol version used depends on the server.

sftp:server-program (string)
The server program implementing SFTP protocol. If it does not contain a slash ‘/’, it is considered a ssh2 subsystem and −s option is used when starting connect-program. Default is ‘sftp’. You can use rsh as transport level protocol like this:
set sftp:connect-program rsh
set sftp:server-program /usr/libexec/openssh/sftp-server
Similarly you can run SFTP over SSH1.

sftp:size-read (number)
Block size for reading. Default is 0x8000.

sftp:size-write (number)
Block size for writing. Default is 0x8000.

ssl:ca-file (path to file)
use specified file as Certificate Authority certificate.

ssl:ca-path (path to directory)
use specified directory as Certificate Authority certificate repository (OpenSSL only).

ssl:check-hostname (boolean)
when true, lftp checks if the host name used to connect to the server corresponds to the host name in its certificate.

ssl:crl-file (path to file)
use specified file as Certificate Revocation List certificate.

ssl:crl-path (path to directory)
use specified directory as Certificate Revocation List certificate repository (OpenSSL only).

ssl:key-file (path to file)
use specified file as your private key. This setting is only used for ftps and https protocols. For sftp and fish protocols use sftp:connect-program and fish:connect-program respectively (add '−i' option to ssh).

ssl:cert-file (path to file)
use specified file as your certificate.

ssl:use-sni (boolean)
when true, use Server Name Indication (SNI) TLS extension.

ssl:verify-certificate (boolean)
if set to yes, then verify server's certificate to be signed by a known Certificate Authority and not be on Certificate Revocation List. You can specify either host name or certificate fingerprint in the closure.

ssl:priority (string)
free form priority string for GnuTLS. If built with OpenSSL the understood values are + or − followed by SSL3.0, TLS1.0, TLS1.1 or TLS1.2, separated by *:*. Example:
set ssl:priority "NORMAL:−SSL3.0:−TLS1.0:−TLS1.1:+TLS1.2"

torrent:ip (ipv4 address)
IP address to send to the tracker. Specify it if you are using an HTTP proxy.

torrent:ipv6 (ipv6 address)
IPv6 address to send to the tracker. By default, first found global unicast address is used.

torrent:max-peers (number)
maximum number of peers for a torrent. Least used peers are removed to maintain this limit.

torrent:port-range (from-to)
port range to accept connections on. A single port is selected when a torrent starts.

torrent:retracker (URL)
explicit retracker URL, e.g. 'http://retracker.local/announce'.

torrent:save-metadata (boolean)
when true, lftp saves metadata of each torrent it works with to *~/.local/share/lftp/torrent/md* or *~/.lftp/torrent/md* directory and loads it from there if necessary.

torrent:seed-max-time (time interval)
maximum seed time. After this period of time a complete torrent shuts down independently of ratio. It can be set to infinity if needed.

torrent:seed-min-peers (number)
minimum number of peers when the torrent is complete. If there are less, new peers are actively searched for.

torrent:stop-min-ppr (real number)
minimum per-piece-ratio to stop seeding. Use it to avoid a situation when a popular piece causes quick raise of the total ratio.

torrent:stop-on-ratio (real number)
torrent stops when it's complete and ratio reached this number.

torrent:timeout (time interval)
maximum time without any progress. When it's reached, the torrent shuts down.

torrent:use-dht (boolean)
when true, DHT is used.

xfer:auto-rename(boolean)
suggested filenames provided by the server are used if user explicitly sets this option to 'on'. As this could be security risk, default is off.

xfer:backup-suffix (string)
a time format string (see strftime(3)) for backup file name when replacing an existing file.

xfer:clobber (boolean)
if this setting is off, get commands will not overwrite existing files and generate an error instead.

xfer:destination-directory (path or URL to directory)
This setting is used as default −O option for get and mget commands. Default is empty, which means current directory (no −O option).

xfer:disk-full-fatal (boolean)
when true, lftp aborts a transfer if it cannot write target file because of full disk or quota; when false, lftp waits for disk space to be freed.

xfer:eta-period (seconds)
the period over which weighted average rate is calculated to produce ETA.

xfer:eta-terse (boolean)
show terse ETA (only high order parts). Default is true.

xfer:keep-backup (boolean)
when true, the backup file created before replacing an existing file is not removed after successful transfer.

xfer:make-backup (boolean)
when true, lftp renames pre-existing file adding xfer:backup-suffix instead of overwriting it.

xfer:max-redirections (number)
maximum number of redirections. This can be useful for downloading over HTTP. 0 prohibits redirections.

xfer:parallel (number)
the default number of parallel transfers in a single get/put/mget/mput command.

xfer:rate-period (seconds)
the period over which weighted average rate is calculated to be shown.

xfer:temp-file-name (string)
temporary file name pattern, first asterisk is replaced by the original file name.

xfer:timeout (time interval)
maximum time without any transfer progress. It can be used to limit maximum time to retry a transfer from a server not supporting transfer restart.

xfer:use-temp-file (boolean)
when true, a file will be transferred to a temporary file in the same directory and then renamed.

xfer:verify (boolean)
when true, verify-command is launched after successful transfer to validate file integrity. Zero exit code of that command should indicate correctness of the file.

xfer:verify-command (string)
the command to validate file integrity. The only argument is the path to the file.

The name of a variable can be abbreviated unless it becomes ambiguous. The prefix before ':' can be omitted too. You can set one variable several times for different closures, and thus you can get a particular settings for particular state. The closure is to be specified after variable name separated with slash '/'.

The closure for 'dns:', 'net:', 'ftp:', 'http:', 'hftp:' domain variables is currently just the host name as you specify it in the 'open' command (with some exceptions where closure is meaningless, e.g. dns:cache-size). For some 'cmd:' domain variables the closure is current URL without path. For 'log:' domain variables the closure is either 'debug' or 'xfer'. For other variables it is not currently used. See examples in the sample *lftp.conf*.

Certain commands and settings take a time interval parameter. It has the format Nx[Nx...], where N is time amount (floating point) and x is time unit: d - days, h - hours, m - minutes, s - seconds. Default unit is second. E.g. 5h30m or 5.5h. Also the interval can be 'infinity', 'inf', 'never', 'forever' - it means infinite interval. E.g. 'sleep forever' or 'set dns:cache-expire never'.

Boolean settings can be one of (true, on, yes, 1, +) for a True value or one of (false, off, no, 0, -) for a False value.

Tri-boolean settings have either a boolean value or 'auto'.

Integer settings can have a suffix: k - kibi, m - mebi, g - gigi, etc. They can also have a prefix: 0 - octal, 0x - hexadecimal.

FTP asynchronous mode (pipelining)

Lftp can speed up FTP operations by sending several commands at once and then checking all the responses. See ftp:sync-mode variable. Sometimes this does not work, thus synchronous mode is the default. You can try to turn synchronous mode off and see if it works for you. It is known that some network software dealing with address translation works incorrectly in the case of several FTP commands in one network packet.

RFC959 says: ''The user-process sending another command before the completion reply would be in violation of protocol; but server-FTP processes should queue any commands that arrive while a preceding command is in progress''. Also, RFC1123 says: ''Implementors MUST NOT assume any correspondence between READ boundaries on the control connection and the Telnet EOL sequences (CR LF).'' and ''a single READ from the control connection may include more than one FTP command''.

So it must be safe to send several commands at once, which speeds up operation a lot and seems to work with all Unix and VMS based ftp servers. Unfortunately, windows based servers often cannot handle several commands in one packet, and so cannot some broken routers.

OPTIONS

−d Switch on debugging mode.

−e *commands*
Execute given commands and don't exit.

−p *port* Use the given port to connect.

−u *user*[**,***pass*]
Use the given username and password to connect. Remember to quote the password properly in the shell. Also note that it is not secure to specify the password on command line, use *~/.netrc* file or **LFTP_PASSWORD** environment variable together with −−env−password option. Alternatively you can use ssh-based protocols with authorized keys, so you don't have to enter a password.

−−norc Don't execute rc files from the home directory.

−−rcfile *file*
Execute commands from the file. May be specified multiple times.

−f *script_file*
Execute commands in the file and exit. This option must be used alone without other arguments (except **−−norc**).

−c *commands*
Execute the given commands and exit. Commands can be separated with a semicolon, '&&' or '||'. Remember to quote the commands argument properly in the shell. This option must be used alone without other arguments (except **−−norc**).

Other **open** options may also be given on the **lftp** command line.

ENVIRONMENT VARIABLES

The following environment variables are processed by **lftp**:

EDITOR
Used as local editor for the **edit** command.

HOME
Used for (local) tilde ('~') expansion.

SHELL
Used by the **!** command to determine the shell to run.

PAGER
This should be the name of the pager to use. It's used by the **more** and **zmore** commands.

http_proxy, **https_proxy**
Used to set initial http:proxy, hftp:proxy and https:proxy variables.

ftp_proxy
Used to set initial ftp:proxy or hftp:proxy variables, depending on URL protocol used in this environment variable.

no_proxy
Used to set initial net:no-proxy variable.

LFTP_MODULE_PATH
Used to set initial module:path variable.

LFTP_HOME
Used to locate the directory that stores user-specific configuration files. If unset, *~/.lftp* will be used. Please note that if this directory does not exist, then XDG directories will be used.

LFTP_PASSWORD
Used for −−env−password **open** option.

LS_COLORS
used to set initial color:dir-colors variable.

XDG_CONFIG_HOME, **XDG_DATA_HOME**, **XDG_CACHE_HOME**
Used to locate the directories for user-specific files when *~/.lftp* (or **$LFTP_HOME** directory) does not exist. Defaults are *~/.config*, *~/.local/share* and *~/.cache* respectively. The suffix */lftp* is

appended to make the full path to the directories.

FILES

/etc/lftp.conf
system-wide startup file. Actual location depends on −−sysconfdir configure option. It is */etc* when prefix is */usr*, */usr/local/etc* by default.

~/.config/lftp/rc or *~/.lftp/rc*, *~/.lftprc*
These files are executed on lftp startup after */etc/lftp.conf*.

~/.local/share/lftp/log or *~/.lftp/log*
The file things are logged to when lftp moves into the background in nohup mode.

~/.local/share/lftp/transfer_log or *~/.lftp/transfer_log*
The file transfers are logged to when log:enabled/xfer setting is set to 'yes'. The location can be changed by log:file/xfer setting.

~/.local/share/lftp/bookmarks or *~/.lftp/bookmarks*
The file is used to store lftp's bookmarks. See the **bookmark** command.

~/.local/share/lftp/cwd_history or *~/.lftp/cwd_history*
The file is used to store last working directories for each site visited.

~/.local/share/lftp/bg/ or *~/.lftp/bg/*
The directory is used to store named sockets for backgrounded lftp processes.

~/.cache/lftp/DHT/ or *~/.lftp/DHT/"*
The directory is used to store DHT id and nodes cache for IPv4 and IPv6. File name suffix is the host name.

~/.cache/lftp/edit/ or *~/.lftp/edit/"*
The directory is used to store temporary files for **edit** command.

~/.local/share/lftp/torrent/md/ or *~/.lftp/torrent/md/"*
The directory is used to store torrent metadata. It is especially useful for magnet links, cached metadata can be loaded from the directory. It can also serve as torrent history, file names are the info_hash of torrents.

~/.netrc The file is consulted to get default login and password to a server when it is specified without a protocol to the 'open' command. Passwords are also searched here if an URL with user name but with no password is used.

SEE ALSO

ftpd(8), **ftp**(1)
RFC854 (telnet), RFC959 (ftp), RFC1123, RFC1945 (http/1.0), RFC2052 (SRV RR), RFC2228 (ftp security extensions), RFC2389 (ftp FEAT), RFC2428 (ftp/ipv6), RFC2518 (WebDAV), RFC2616 (http/1.1), RFC2617 (http/1.1 authentication), RFC2640 (ftp i18n), RFC3659 (ftp extensions), RFC4217 (ftp over ssl), BEP0003 (BitTorrent Protocol), BEP0005 (DHT Protocol), BEP0006 (Fast Extension), BEP0007 (IPv6 Tracker Extension), BEP0009 (Extension for Peers to Send Metadata Files), BEP0010 (Extension Protocol), BEP0012 (Multitracker Metadata Extension), BEP0023 (Tracker Returns Compact Peer Lists), BEP0032 (DHT Extensions for IPv6).
https://tools.ietf.org/html/draft-preston-ftpext-deflate-04 (ftp deflate transmission mode),
https://tools.ietf.org/html/draft-ietf-secsh-filexfer-13 (sftp).
http://wiki.theory.org/BitTorrentSpecification
http://www.bittornado.com/docs/multitracker-spec.txt
http://www.rasterbar.com/products/libtorrent/dht_sec.html (DHT security extension)
http://xbtt.sourceforge.net/udp_tracker_protocol.html (UDP tracker)

AUTHOR

Alexander V. Lukyanov
lav@yars.free.net

ACKNOWLEDGMENTS

This manual page was originally written by Christoph Lameter <clameter@debian.org>, for the Debian GNU/Linux system. The page was improved and updated later by Nicolas Lichtmaier <nick@Feedback.com.ar>, James Troup <J.J.Troup@comp.brad.ac.uk> and Alexander V. Lukyanov <lav@yars.free.net>.