lftp Scriptable file transfer

START HERE

`lftp -u <user> <host>`
Connect and log in

`put <file>`
Upload a file

`ls`
List the remote directory

`mirror <dir>`
Download a whole tree

`cd <dir>`
Change remote directory

`mirror -R <dir>`
Upload a whole tree

`get <file>`
Download a file

`exit`
Close the session

Local dir lcd, lpwd — `put / mirror -R` → **lftp shell** queue, jobs, set — `get / mirror` → **Remote site** ftp, ftps, sftp, http

CONCEPTS

lftp is a shell: connect once, then run ls, cd, get and put inside it. Remote and local commands are separate (cd vs lcd).

Protocols: ftp://, ftps://, sftp://, http(s)://, fish://. A site can also be a bookmark name.

Avoid passwords on the command line: use ~/.netrc, an SSH key, or --env-password with LFTP_PASSWORD.

CONNECT

`lftp <host>`
Open a session

`lftp -u <user>,<pass> <host>`
Connect with credentials

`lftp sftp://<user>@<host>`
Connect over SFTP

`open -u <user> -p <port> <host>`
Connect from inside the shell

`lftp -e "<cmd>" <host>`
Run a command after connecting

`lftp -c "<cmds>"`
Run commands and exit
Separate commands with ;, && or ||.

`lftp -f <script>`
Run a script file and exit

`user <user> [<pass>]`
Set login for later use

`close -a`
Close idle connections

`lftp --norc`
Skip rc files at startup

NAVIGATE AND LIST

`ls`
List remote directory (cached)

`cd <dir>`
Change remote directory
cd - returns to the previous directory.

`lcd <dir>`
Change local directory

`pwd`
Print remote location

`lpwd`
Print local directory

`cls -l -h`
Formatted long listing, sizes human-readable

NAVIGATE AND LIST (cont.)

`cls --sort=size -S`
Sort listing by size

`find <dir>`
Recursive listing

`nlist`
Names only

`cache flush`
Drop the listing cache

`!<cmd>`
Run a local shell command

TRANSFER

`get <file>`
Download a file

`put <file>`
Upload a file

`get <file> -o <local>`
Download under another name

`get -c <file>`
Resume a download

`put -c <file>`
Resume an upload

`mget <glob>`
Download files by wildcard

`mput <glob>`
Upload files by wildcard

`mget -d <glob>`
Download keeping directory names

`get -O <dir> <file>`
Download into a local base directory

`get -P <n> <files>`
Fetch n files in parallel

`pget -n <n> <file>`
Download over n connections
Loads the server heavily; the server may limit connections.

`pget -c <file>`
Resume a pget download

`get -E <file>`
Delete the source after success

`cat <file>`
Print a remote file

`edit <file>`
Edit a remote file in place

MIRROR

`mirror <rdir> [<ldir>]`
Download a directory tree

`mirror -R <ldir> [<rdir>]`
Upload a directory tree

MIRROR (cont.)

`mirror --dry-run <rdir>`
Preview without changing anything

`mirror -c <rdir>`
Continue interrupted transfers

`mirror -n <rdir>`
Only get newer files
Does not combine with -c.

`mirror -e <rdir>`
Delete files missing at the source
Dry-run first: this deletes at the destination.

`mirror -P <n> <rdir>`
Run n transfers in parallel

`mirror -v <rdir>`
Print each file handled

`mirror -x <regex> <rdir>`
Exclude by regex

`mirror -X <glob> <rdir>`
Exclude by glob

`mirror -I <glob> <rdir>`
Include only matching globs

`mirror --only-missing <rdir>`
Only fetch files not present locally

`mirror --use-pget-n=<n> <rdir>`
Split each file over n connections

`mirror -L <rdir>`
Treat symlinks as files

REMOTE FILES

`mkdir -p <dir>`
Create directory with parents

`rm <file>`
Remove a file (no wildcards)

`rm -r <dir>`
Remove a directory recursively

`mrm <glob>`
Remove files by wildcard

`mv <old> <new>`
Rename or move

JOBS AND QUEUE

`<command> &`
Run a command in the background

`jobs -v`
List running jobs

`wait [<job>]`
Wait for a job

`kill <job> / kill all`
Stop jobs

JOBS AND QUEUE (cont.)

`queue <command>`
Add a command to the queue
Do not queue cd or lcd.

`queue stop / queue start`
Pause or resume the queue

`exit bg`
Leave, finish jobs in the background

SETTINGS AND BOOKMARKS

`set -a`
List all settings

`set <name> <value>`
Change a setting

`set <name>/<host> <value>`
Setting for one host only

`set net:limit-rate <down>:<up>`
Limit bandwidth
Bytes per second; suffixes like 100K work.

`set ssl:verify-certificate no`
Skip certificate verification
Only for hosts you trust; lftp verifies by default.

`set ftp:passive-mode on`
Use passive FTP mode

`set sftp:connect-program "ssh -a -x -i <key>"`
Choose SSH key for SFTP

`bookmark add <name>`
Save the current site

`bookmark list`
Show bookmarks

`open <name>`
Connect to a bookmark

`~/.lftp/rc`
Per-user startup commands (set ...)

RECIPES

Mirror a site safely

```
lftp -u <user> sftp://<host>
mirror --dry-run -R ./site /var/www/site
mirror -R -P 4 ./site /var/www/site
```

One-shot script from cron

```
lftp -c "open -u <user>,<pass> <host>; mirror -n -P 2 /pub ./pub; bye"
```

Upload only changed files, skip junk

```
mirror -R -n -x '\.git/' -X '*.tmp' ./out /public
```