File Transfer - Command-line FTP, LFTP, WPUT & cURL

FTP

Commands

to resume a single file upload using the built-in ftp command you will need to know how many bytes of the file you have already sent. This should be accessible by using `ls`. Then you use the following sequence to restart your upload replacing `<#>` with the number of bytes already sent and `<filename>` with the filename you are uploading.

```
restart <#>
put <filename>
```

If the server allows it you should receive a message such as the following...

```
350 Restart position accepted (<#>).
150 Ok to send data.
```

This will resume your upload.

LFTP

Using the LFTP command this allows you to restart a died ftp session

Commands

The following command is the login to the server and go to the folder you are placing the file in

```
lftp user:pass@host/path/to/folder
```

```
lftp user:pass@host/path/to/folder
cd ok, cwd=/path/to/folder
lftp user@host:/path/to/folder> reput file.ext
---> TYPE I
<--- 200 Type set to I
---> SIZE file.ext
<--- 213 11842837120
---> PASV
<--- 227 Entering Passive Mode (10,0,0,15,220,70).
---- Connecting data socket to (10.0.0.15) port 56390
---- Data connection established
---> ALLO 20769244058
<--- 202 No storage allocation necessary
---> REST 11842837120
<--- 350 Restarting at 11842837120. Send STORE or RETRIEVE to initiate transfer
---> STOR file.ext
<--- 150 Opening BINARY mode data connection for file.ext
`file.ext' at 6756302848 (32%) 31.50M/s eta:7m [Sending data]
```

This command is to restart the failed FTP replace the "file.txt" with your file

```
reput file.txt
```

After a quick search, there's a command line program called `lftp` that provides ftp mirroring functionality.

Adapted from a guide here, something like this should do the trick:

```
#!/bin/bash
HOST='address.co.uk'
USER='myuser'
PASS='mypass'
TARGETFOLDER='/public_html/java/desktop/'
SOURCEFOLDER='deploy/'

lftp -f "
open $HOST
user $USER $PASS
lcd $SOURCEFOLDER
mirror --reverse --delete --verbose $SOURCEFOLDER $TARGETFOLDER
bye
"
```

I'd suggest you'd do it without the `--delete` until you're sure you've got the arguments right!

Example of simple bash script ftp client

This script first defines variables such as hostname of the ftp server, username and password and then it creates an ftp session and uploads the specified file into your selected directory:

```
#!/bin/bash


ftp_site=127.0.0.1
username=ftpuser
passwd=YOUR_PASSWORD

PS3='Select a destination directory: '

# bash select
select path in "." "/test" "public_html/myblog/" "backup/images/"
do
ftp -n $ftp_site<<EOF
quote USER $username
quote PASS $passwd
binary
cd $path
put $1
quit
EOF
break
done
```

Be sure to edit the `ftp_site`, `username`, and `passwd` variables above. You should also change the paths to whichever directories you most commonly upload your files to.

Executing the script:

```
$ chmod +x ftp_bash_script.sh
$ ./ftp_bash_script.sh file1
```

Example script output:

```
$ ./ftp_bash_script.sh somerandomfile
1) .
2) /test
3) public_html/myblog/
4) backup/images/
Select a destination directory: 2
Connected to 127.0.0.1.
220 (vsFTPd 3.0.5)
331 Please specify the password.
230 Login successful.
200 Switching to Binary mode.
250 Directory successfully changed.
local: somerandomfile remote: somerandomfile
229 Entering Extended Passive Mode (|||10078|)
150 Ok to send data.
    0        0.00 KiB/s
226 Transfer complete.
221 Goodbye.
```

WPUT

This is to resume a files transfer using wput instead ftp or lftp commands and can be used in Windows or Linux

WPUT Commands

Here is an example how wput continues upload after connection problems:

```
wput -v -u -B upload.zip ftp://login:YOUR_PASSWORD@server.com/dir/upload.zip
--20:14:23-- `upload.zip'
   => ftp://login:YOUR_PASSWORD@111.111.111.111:21/dir/upload.zip
Connecting to 111.111.111.111:21... connected!
Logging in as login ... Logged in!
==> CWD dir
==> TYPE I ... done.
==> SIZE upload.zip ... done (4313 bytes)
==> PASV ... done.
```

```
==> REST 3584 ... done.
==> STOR upload.zip ... done.
Length: 902,153,406 [902,149,822 to go]
 7% [======>                                                                ] 65,658,368       194.0K/s ETA
1:10hError: Error encountered during uploading data (Operation now in progress)
==> ABOR ... Error: recv() timed out. No data received
Receive-Warning: read() timed out. Read '' so far.
failed.
Waiting 10 seconds... Error: recv() timed out. No data received
Receive-Warning: read() timed out. Read '' so far.
Connecting to 111.111.111.111:21... connected!
Logging in as login ... Logged in!
==> CWD dir
==> TYPE I ... done.
==> SIZE upload.zip ... done (65247144 bytes)
==> PASV ... done.
==> REST 65246208 ... done.
==> STOR upload.zip ... done.
Length: 902,153,406 [836,907,198 to go]
 9% [++++++==>
```

CURL

This is the resume a file transfer using curl instead ftp or lftp commands and can be used in Windows or Linux

CURL Commands

```
curl -C - --upload-file source_file ftp://destination.server.com/
```

Revision #2
Created 11 February 2026 13:20:53 by Admin
Updated 24 October 2026 14:12:36 by Admin