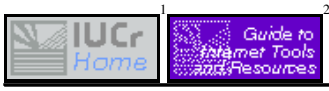

FTP Command line

On many computers the command line is the only means to use the FTP protocol. One starts by typing the `ftp` command. Then a number of subcommands are available. One exits FTP by typing the `quit` subcommand:

```
ftp [machine_address]
.....
....
>quit
```

The address of the remote machine may be given either on the command line or as a subcommand.

ftp example

As an example, we will present the use of an >anonymous ftp[3].

We assume a user who wants to connect to the anonymous ftp server of my laboratory at *ftp.lmcp.jussieu.fr*. Lines starting with > are the answers from the machine as they appear on the screen.

```
ftp ftp.lmcp.jussieu.fr
>Connected to turquoise.lmcp.jussieu.fr.
>220 turquoise FTP server (Version wu-2.4(8) Mon Jun 26 17:20:12 DF 1995) ready.
>Name (ftp:epelboin):
                    anonymous    <---- this is your answer
>Password:
          dupond@rlmc.u-saint_tropez.fr  <<---- this is your e-mail address
                                           It does not appear on the screen
>230-Welcome to the anonymous ftp of the
>230-Laboratoire de Mineralogie-Cristallographie, Paris, France
>230-
>230-                        L.M.C.P.
>230-
>230-
>230-Move to directory:
>230-
>230-     pub             directory where to find information
>230-     incoming        directory where to deposit information.
>230-
>230-
>230-Please read the file README
>230-  it was last modified on Tue Sep 19 08:42:55 1995 - 1 day ago
>230 Guest login ok, access restrictions apply.
>ftp>
```

You may now use the ftp subcommands to list the contents of a directory or a file:

```
>ftp>
          ls      < -----  list the contents of a directory
>200RT command successful.
>150 Opening ASCII mode data connection for file list.
>bin
>lib
>etc
>pub
>usr
>README
>welcome.msg
>incoming
```

Always look at the README when available:

[1] http://www.iucr.ac.uk/iucr-top/welcome.html
[2] int.index.html
[3] int.ftp.gen.html#anonymous

```
        get README -   < ---- the file README will be transfered to you
                                                and written on the screen (do not use - for a
                                                binary!)
>200 PORT command successful.
>150 Opening ASCII mode data connection for README (388 bytes).
>Contents of directory pub
>
>
>mac, pc      software for PCs and MacIntosh   ---> local use only
>unix         same for Unix                    ---> local use only
>
>csiesr       archives for CSIESR distribution list
>eunis        documentation, archives for European Universities
>            Information Systems organization
>            sincris      International Information Server for
>            Crystallography
>
>            226 Transfer complete.
>            399 bytes received in 0.02094 seconds (18.61 Kbytes/s)
```

We now move to the directory pub/sincris:

```
>ftp>
                  cd pub/sincris   < ---- change directory to pub/sincris
>250 CWD command successful.
>ftp>
                  ls      < ---- list its contents
>200 PORT command successful.
>150 Opening ASCII mode data connection for file list.
>software
>iucr
>README
>france
>226 Transfer complete.
>ftp>
```

We now move to software and list its contents:

```
           cd pub/software   < ---- change directory to pub/sincris/software
>250 CWD command successful.
>ftp>
           ls      < ---- list its contents
>200 PORT command successful.
>150 Opening ASCII mode data connection for file list.
>topo
>Fhkl
>XmLmctep
>voxel
>README
>226 Transfer complete.
>ftp>
```

Assume we want to retrieve the Fhkl package. We move to its directory and look at the README. (The answers from the computer have been omitted):

```
         cd Fhkl
> ....
         ls
> .....
>Fhkl-old.tar.Z
>Fhkl1.3.tar.Z
>README
> ....
     get README -
>....
                Fhkl package
>    Software for the computation of data of interest in Dynamical
>               Theory of X-Rays
> ......
>       Run on Unix OS
```

If interested you may retrieve the file Fhkl1.3.tar.Z. Transfer can in general be carried out either in `ASCII` or `binary`. `ASCII` should be used for simple text files and `binary` mode for all files. The ftp setting depends on your computer. You may switch the mode by sending either the command `ascii` or `binary`. For a tar file ASCII is the correct mode. To retrieve the file do

the following:

```
>ftp>
                  get Fhkl1.3.tar.Z   < --- get the file which will be written
                                                          on the current directory of your
                                                          computer
>200 PORT command successful.
>150 Opening ASCII mode data connection for Fhkl1.3.tar.Z (979341 bytes).
>226 Transfer complete.
>983498 bytes received in 0.9732 seconds (986.9 Kbytes/s)
>ftp>
>quit
```

You now have stored the file in your computer. It cannot be used yet since it is written in a compressed format.[4]

You may change the name of the file by providing another name in the `get` command.

ftp commands

Here is a brief summary of ftp subcommands. To see all available subcommands type `help` as a subcommand.

`ascii`
: switchs to ASCII mode

`binary`
: switchs to binary mode

`get file_1 [file_2]`
: get file_1 from remote location and write it as file_2. file_2 may be omitted. If equal to – write file_1 output on the screen.

`put file_1 [file_2]`
: put file_1 in remote file_2 (not valid with an anonymous ftp in most cases unless incoming directory exists).

`mget *.f`
: retrieve all files with extension .f for instance.

`mput *.f`
: multiple send. The user is prompted before each transfer. To disable this feature use subcommand `prompt`.

`ls` or `dir`
: List remote directory

`cd directory_name`
: change to remote directory

`lcd directory_name`
: change to another local directory.
It may happen that the above three subcommands do not work properly. This is quite common with non–Unix systems. On Unix systems ask the supervisor to install a proper version of the ftp server (ftp daemon).

`quit`
: end ftp session

File format

Text files are stored in a computer as a sequence of characters. The representation of each character as a sequence of bits depends on the computer system being used. Many, but by no means all, (a notable example being the IBM mainframes using the EBCIDC code) use the ASCII (7 bit) code or a (8 bit) superset of it. The superset allows the encoding of additional characters, especially accentuated characters used in European languages. The most common ASCII superset in use (but again not the only one) is the ISO–latin–1 (ISO–8859–1) current on UNIX and Windows 3.1/95. Another difference in the way operating systems encode text files is in the treatment of end of lines. In MS–DOS this is done with a CR/LF pair of characters whereas for UNIX only an LF is used. It is thus evident that in transfering text files between computers, care has to be taken to allow for differences operating systems and character representation. With ftp for the transfer of text files one has to choose the (unfortunately named)

[4] #file

`ASCII` mode to achieve this which will undertake the necessary character conversion and end−of−line treatment.

A common error when using FTP protocol is to transfer executable or graphics files in ASCII mode. This may cause changes in the file content which becomes corrupted. Thus non−text files *MUST* be transfered using the binary mode. For instance an image, written either in JPEG or GIF formats must be exchanged in binary mode. On a PC a file compressed with pkzip is a binary file and will be corrupted if transfered in ASCII mode.

A less common error is to transfer text files (ASCII files) as binary. If both sender and receiver computers use a different operating system or character representation, the received file will be misinterpreted. Postscript files are text files and must be transfered in ASCII mode.

Thus the rule is very simple: text files only must be transfered in ASCII mode. All other files must be transfered in binary mode.

Sometimes, however, this is not sufficient. If a PC text file has been stored on a computer in binary mode and then retrieved on a Unix machine as an ASCII file it will be impossible to read it, due to the presence of this aditional character. In such a case the only possibility is to use programs such as *dos2unix* or *unix2dos* to translate the file.

This leads to the conclusion: when in doubt transfer a file in binary mode: it does not modify its contents. If it appears later that the file is a text file it may be repaired using the proper utility.

Unix
: Most unix files are tared and compressed.
  1. To uncompress use `uncompress` or `gunzip`: `uncompress file.tar.Z`, `gunzip file.tar.gz`
  2. To untar: `tar −xvf file.tar`

DOS, Windows
: Most files use the pkzip utility for exchange.

MacOS
: Most files are in binhex format. Use a shareware such as *Stuff'it* to unstuff. Some files are self decompressible.

[Index][5] [ftp−intro][6] [ftp−GUI][7] [ftp−WWW][8] – 19th July 1996 – © Yves Epelboin[9] – **Not to be copied or reproduced without permission**

[5] int.index.html
[6] int.ftp.gen.html
[7] int.ftp.gr.html
[8] int.ftp.www.html
[9] http://www.lmcp.jussieu.fr/~epelboin/