

# Internet File Transfers – FTP

Eric Behr, Mathematics  
Illinois State University  
January 1991  
Revised September 1992

This document describes basics of the Internet standard ‘file transfer protocol’, or FTP. It is not our intention to replace appropriate manuals; we merely want to summarize the most common applications and commands. Note that the bulk of our description will deal with a Unix implementation of FTP. Other systems may be more limited.

## Introduction

FTP is a communications protocol built on top of the standard Internet TCP/IP suite. It provides for fast error-resistant transfers of files in various formats. Aside from allowing users to transmit files between private computer accounts, FTP is also used to disseminate public information within the Internet community. Most publications, references, software, bulletins and archives related to the Internet are available only in this form. Rudimentary ability to use FTP is therefore essential for anyone who wishes to take full advantage of everything that the networks which comprise the Internet have to offer.

It should be pointed out that non-TCP/IP networks, such as Bitnet, do not allow FTP traffic. Some files stored on those networks can be obtained by sending a mail request to the server machine. The file is then sent back to the user by ordinary e-mail. This topic is outside the scope of this memo.

## The software

FTP is similar in principle to other communications protocols like Kermit or X-modem. Since its task is to connect two Internet computers, it is not surprising that both of those machines have to run suitable software. One of the computers uses its FTP program to request file transfer actions from the other one; the first machine then acts as an FTP *client*, and the second one as a *server*.

On most minicomputers and mainframes, an FTP package is included in the operating system, or is readily available from public domain sources. For instance, virtually every implementation of Unix (or its variety) has FTP built-in. Many vendors, such as IBM, offer optional FTP packages for their mainframe operating systems. There are also free and commercial packages compatible with Digital Equipment’s VMS system.

FTP in one form or another is also available for most microcomputers. PC compatibles can use a software package called WiscWare – ISU has purchased a campus site license for it. Macs use one of several public domain programs, such as NCSA Telnet, XferIt, Fetch, or the HyperFTP stack.

## Using FTP

Let us assume that user 'ejbehr' is logged on to a well-connected Internet machine 'rs6000.cmp.ilstu.edu'. We will first trace a typical session in which he wants to connect to his account 'ebehr' on a computer called 'xenon.che.ilstu.edu' (it is also perfectly legal

to use the Internet number, such as '138.87.128.1'). Naturally, details of the computer's responses will vary slightly from one implementation to another. User's commands are in boldface. In this example, the two operating systems are dissimilar (IBM AIX and DEC VMS); this illustrates the unifying power of the FTP protocol.

```
rs6000> ftp xenon.che.ilstu.edu
Connected to xenon.che.ilstu.edu.
220 Service ready for new user.
Name (xenon:ejbehr): ebehr
331 Username Okay, need password.
Password: (that's my secret...)
230 User EBEHR logged in, proceed.
```

The user can now issue commands to the server. We will illustrate a few basic ones below. The first ones will be, in sequence, long and short forms of the remote file directory, and an instruction to transfer a file from the remote account to the client machine:

```
ftp> dir
200 Port 138,87,1,2,4,140 Okay.
125 File status okay; about to open data connection.
$USERS:[EBEHR]DECW$TERMINAL_DEFAULT.DATJNL;1      0/0      4-JAN-1991 16:50
$USERS:[EBEHR]LOGIN.COM;1      2/3      3-OCT-1990 15:15
$USERS:[EBEHR]MAIL.MAI;1      30/30      4-JAN-1991 15:38
226 Closing data connection.
ftp> ls
200 Port 138,87,1,2,4,141 Okay.
125 File status okay; about to open data connection.
DECW$TERMINAL_DEFAULT.DATJNL;1
LOGIN.COM;1
MAIL.MAI;1
226 Closing data connection.
ftp> get mail.mai
200 Port 138,87,1,2,4,142 Okay.
125 File status okay; about to open data connection.
226 Closing data connection.
3219 bytes received in 0.1498 seconds (20.98 Kbytes/s)
```

The following examples illustrate how a file can be transferred and given a different local name, 'log' instead of 'login.com' (this is important when communicating between different systems; for example, it wouldn't be a good idea to transfer a Macintosh file called 'A test of my ftp #3' to an MS-DOS computer...) We then transfer a larger file to the remote machine. At the end of the session, we will erase a file on the remote machine.

```
ftp> get login.com log
200 Port 138,87,1,2,4,143 Okay.
125 File status okay; about to open data connection.
226 Closing data connection.
857 bytes received in 0.004243 seconds (197.2 Kbytes/s)

ftp> put ftplist ftp.txt
200 Port 138,87,1,2,4,144 Okay.
125 File status okay; about to open data connection.
226 Closing data connection.
145592 bytes sent in 3.886 seconds (36.59 Kbytes/s)

ftp> delete ftp.txt
250 Requested file action okay, completed.
ftp> close
221 Service closing control connection.
```



The reader may want to play with two more FTP commands: ‘mget’ and ‘mput’. These are used to transfer multiple files. For example, ‘mget net\*’ will attempt to retrieve all files with names beginning with ‘net’ from the current directory on the server. Transfers of individual files will be preceded with a query (“do you really want to transfer this file?”). For unattended (batch) transfers we can avoid that by issuing a command ‘prompt’, which in most implementations will turn this feature off.

## Anonymous FTP

Many Internet computers contain huge amounts of data which their administrators want to make publicly available. The most common way of allowing such access is to establish a special FTP account belonging to a fictitious user ‘anonymous’. Security measures of the operating system are then usually somewhat relaxed, so that the password for this user is not strictly enforced.

Before we begin, we will note one of the main principles of Internet etiquette. Computers which are accessed using anonymous FTP are seldom dedicated to only that purpose. Most often, they are also used for ordinary tasks by their local users. If excessive FTP activity prevents those users from doing their work, the system manager will most likely abandon the altruistic idea of maintaining an anonymous FTP account. It is therefore expected that larger file transfers will be performed only in off-peak periods. Note that with some sites (Scandinavia, Australia) “off-peak” may mean our morning or early afternoon!

So let’s wait until the evening, and see how it works.

```
ftp> open ucsd.edu
Connected to ucsd.edu.
220 ucsd.edu FTP server (Version 4.24 Sat Dec 22 18:14:29 PST 1990) ready.
Name (ucsd.edu:ejbehr): anonymous
331 Guest login ok, send ident (user@host) as password.
Password:ejbehr
230 Guest login ok, access restrictions apply.
```

Note that some systems *do* require the user to type his full address as password, but in most cases any password whatsoever (including a carriage return) will work fine. Common courtesy, however, requires that the full e-mail address be used; this is how managers of ftp sites track usage and access methods.

```
ftp> ls
200 PORT command successful.
150 Opening data connection for /bin/ls (138.87.1.2,1172) (0 bytes).
graphics
hamradio
images
midi
ncsa.ucsd
pub
sparcaudio
usenet
226 Transfer complete.
```

Suspecting that ‘usenet’ is a subdirectory, we use ‘cd’ to move to it. If we just wanted to list its contents, ‘ls usenet’ would be just as good.

```
ftp> cd usenet  
250 CWD command successful.
```

```
ftp> dir
200 PORT command successful.
150 Opening data connection for /bin/ls (138.87.1.2,1173) (0 bytes).
total 2
drwxr-xr-x  2 0          6          512 Jul 23  1989 bin
drwxr-xr-x  3 0          6          512 Jul 22  1989 comp
-rw-r--r--  1 0          1           0 Jan 28 08:24 index
-rw-r--r--  1 0          1           0 Jan 21 08:24 oldindex
226 Transfer complete.
```

Hmm... nothing interesting here. Let's jump back up ('cdup'; on most systems 'cd ..' would also work), and then check out the directory 'pub'.

```
ftp> cdup
250 CWD command successful.
ftp> cd pub
250 CWD command successful.
ftp> dir
200 PORT command successful.
150 Opening data connection for /bin/ls (138.87.1.2,1174) (0 bytes).
total 6004
drwxr-xr-x  2 0          1          512 Jan 10 20:58 3270
-r--r--r--  1 3          ftp        25322 May 20  1988 68kasm.f.Z
[the entire listing goes on forever; we shortened it]
-r--r--r--  1 102        ftp        25630 Jul 12  1989 traceroute.tar.Z
drwxr-xr-x  2 0          1          512 Sep  5 17:17 ucdmail
-r--r--r--  1 3          ftp        63485 May 19  1988 ucseal.ps.Z
-r--r--r--  1 3          ftp        17270 Feb 25  1988 vmsmodem.shar.Z
-rw-rw-r--  1 ftp        ftp        176293 Jan 25  1990 zoo2.tar.Z
226 Transfer complete.
```

For sentimental reasons, we would like to access the PostScript rendition of the University of California seal. The extension '.Z', however, indicates that it is a binary (and compressed) file. We will switch to that mode, transfer the file (giving it a local name 'seal.Z'), and then change back to text mode. Finally, we show how to ask the server for help, and (after losing track of where we are) how to check the current directory path.

```
ftp> binary
200 Type set to I.
ftp> get ucseal.ps.Z seal.Z
200 PORT command successful.
150 Opening data connection for ucseal.ps.Z (138.87.1.2,1175) (63485 bytes).
226 Transfer complete.
63485 bytes received in 10.62 seconds (5.837 Kbytes/s)
ftp> ascii
200 Type set to A.
ftp> remotehelp
214-The following commands are recognized (* =>'s unimplemented).
  USER  PORT  RETR  MSND*  ALLO  DELE  SITE*  XMKD  CDUP
  PASS  PASV  STOR  MSOM*  REST*  CWD  STAT*  RMD  XCUP
  ACCT*  TYPE  APPE  MSAM*  RNFR  XCWD  HELP  XRMD  STOU
  REIN*  STRU  MLFL*  MRSQ*  RNT0  LIST  NOOP  PWD
  QUIT  MODE  MAIL*  MRCP*  ABOR  NLST  MKD  XPWD
214 Direct comments to ftp-bugs@ucsd.edu.
ftp> pwd
257 "/"pub" is current directory.
ftp> quit
221 Goodbye.
```

It's time to check the fruit of our labors. Here is the local directory listing.





```
rs6000> dir
total 228
-rw----- 1 ejbehr  mat      143599 Feb 01 21:31 ftplist
-rw----- 1 ejbehr  mat         830 Feb 01 21:35 log
-rw----- 1 ejbehr  mat       3203 Feb 01 21:34 mail.mai
-rw----- 1 ejbehr  mat      63485 Feb 01 21:42 seal.Z
```

Note how in the two examples above we connected to two completely different systems (one was a DEC VMS machine, the other probably an Ultrix VAX or a Sun), but we were still able to use the same commands to do what we wanted. Even though FTP is limited when compared with some other sophisticated proprietary protocols such as AppleShare or NFS, its main advantage is reliability and popularity on the Internet.

## Miscellaneous

As we mentioned in the Introduction, there are several varieties of FTP implementations. Some support only server functions, which makes them unuseable in the anonymous FTP scheme. Some, like XferIt, Fetch, or the Mac HyperCard stack HyperFTP, operate in a ‘point and click’ fashion, translating the user’s actions into the arcane FTP commands on the fly.

Macintosh users will want to learn the specifics of Macintosh file transfers. Check out the [sumex.stanford.edu](http://sumex.stanford.edu) archive, directory /info-mac/report.

## Recommended reading

Appropriate manuals.

## Interesting FTP sites

You may want to visit [pilot.njin.net](http://pilot.njin.net) (128.6.7.38), where a master list of major anonymous FTP sites is kept. Among oodles of those, here are some that stand out:

[brownvm.brown.edu](http://brownvm.brown.edu)  
[emx.utexas.edu](http://emx.utexas.edu)  
[ftp.apple.com](http://ftp.apple.com)  
[ncsa.uiuc.edu](http://ncsa.uiuc.edu)  
[sumex.stanford.edu](http://sumex.stanford.edu)  
[trident.arc.nasa.gov](http://trident.arc.nasa.gov)  
[ftp.cso.uiuc.edu](http://ftp.cso.uiuc.edu)  
[ftp.acns.nwu.edu](http://ftp.acns.nwu.edu)

It is highly recommended that adventurers snooping around anonymous FTP sites use *archie* rather than making dozens of hit-or-miss connections. Archie is a system in which several computers gather information about available FTP archives, and make their databases available for searching. Check with your administrator if you have an “archie client” installed on your machine; if not, ask that it be installed. In desperation you can use telnet and login as “archie” to one of the following computers:

[archie.rutgers.edu](http://archie.rutgers.edu)

[archie.sura.net](http://archie.sura.net)

[archie.ans.net](http://archie.ans.net)