

Chapter 5

File Transfer

Chapter Overview

File Transfer Terminology

File Transfer Guidelines

- Full Path Specification: Conversion of Forward Slashes

- Simultaneous Activity

- Rules for Filenames Sent to the PC

- File Transfer Reports

File Transfer Using FTP

- Setting Up FTP

- Using FTP

- Invoking FTP on the Host Computer

- Invoking FTP Using Alternate Methods

- Using FTP Commands

File Transfer Using rcp

- Setting Up rcp

- Using rcp

- Avoiding Use of NCSA Telnet's rcp Server

- Using rcp Between the PC and UNIX

- Using Wildcards

Chapter Overview

This chapter describes procedures for transferring files between a PC and a network host, using NCSA Telnet's two built-in file transfer servers: FTP and rcp. Guidelines and information are presented for file name conventions, path specification, and background file transfer.

File Transfer Terminology

This chapter uses the following terms:

ASCII file, text file

Humans can read an *ascii* or *text* file, and can edit the files using standard editors on the PC or host. When you transfer text files, their end-of-line markers are changed.

Binary, Graphics, or Image File

Humans can't read a *binary*, *graphics*, or *image* file.

When transferred, binary files are not changed in any way (unlike text files).

Client/Server

The *client* is the system that requests services while the *server* is the system that provides or "serves" them. The client is not always your PC, despite appearances. When you use NCSA Telnet to connect to a host, your PC is the telnet client. When you request a file transfer from your PC, the transfer is actually initiated on the host, making the host the FTP client and your PC the FTP server. Thus, the PC is a telnet client and an FTP server at the same time.

File Transfer

In a file transfer, the contents of a file are copied, and the copy is placed on another computer's disk drive.

File Transfer Guidelines

Full Path Specification: Conversion of Forward Slashes

If you do not specify a full pathname for either FTP or rcp, NCSA Telnet will transfer the files to the default directory. If you need to specify a different directory, type the full pathname as documented in the DOS manual. The full pathname may include the disk drive name, with a colon. For example:

```
D:\myfiles\graphics\image.dat
```

or

```
D:/myfiles/graphics/image.dat
```

You may use forward slashes (/) rather than backslashes (\), because NCSA Telnet automatically converts any forward slashes in UNIX pathnames to backslashes for compatibility with PC-DOS. Because backslashes are special characters in the UNIX shell and a special format is required to enter them, forward slashes are more convenient to use.

NOTE: rcp is a Berkeley UNIX feature. Refer to the section, "Using rcp File Transfer," in this chapter for more information.

Simultaneous Activity

With both FTP and rcp, file transfers are processed at the same time as other telnet sessions. Therefore, while a file transfer is in progress, you can perform other NCSA Telnet activities such as interacting with a telnet session, switching sessions, adding new sessions, or changing parameters.

However, do *not* initiate another file transfer while one is already in progress. While an FTP request is being processed, a second FTP request will be ignored. Furthermore, if you try to start an rcp transfer while another transfer is already underway, both transfers may fail.

NOTE: Do not quit NCSA Telnet while a file transfer is in progress, because the file transfer may fail.

Rules for Filenames Sent to the PC

Whether you are using FTP or rcp to transfer files to or from a PC, check these rules for PC filenames:

- A legal PC filename is eight characters or less, followed by an optional period and an up to three-character extension.
- Upper and lower case are considered identical for filenames on PC disks.
- Names of files which are transferred *from* the PC are given in all lower case.

- A filename longer than eight characters is truncated.
- If a period occurs in a filename, the next three characters are read as the extension, and any others are truncated.
- A filename should *not* contain "special characters" or spaces. If it does, the file may be lost or unreadable. Examples of special characters are the asterisk (*), the dollar sign (\$), the pound sign (#), the hyphen (-), and control characters.

File Transfer Reports

The FTP server produces several kinds of information for you to help keep tabs on FTP access to the local hard disk. You can view this information by pressing the ALT-Z keys (show console messages), as documented in Chapter 2, "Introduction to Managing Sessions." The following list describes the information which FTP provides.

- Reports when a command connection is initiated, with the IP address or host name of the remote machine which initiated the connection
- Reports the name of the user who logs in, if you use the USER command
- Reports the file names which were transferred to or from local disk
- Reports when the command connection ends

The rcp server also produces reports for the console screen.

- Reports when an rcp transfer begins
- Reports when the rcp transfer ends

File Transfer Using FTP

The FTP server in NCSA Telnet is the minimum standard FTP server, similar to that in 4.2 BSD UNIX. NCSA's FTP supports the following features:

- Stream transfer in text format (ASCII) or binary format (IMAGE)
- Change, create, or delete a directory
- Print current directory
- List files in current directory (with wildcard specifications)
- Send and receive multiple files with one command, using wildcards

- Delete file

Setting Up FTP

The host machine must support FTP file transfer. If you do not know whether it does, see your system administrator.

You must *not* have disabled the file transfer capability of NCSA Telnet. That is done with the Parameter menu (see Chapter 3, "More About Managing Sessions"), or in the configuration file (see Chapter 7, "Installation and Configuration"). If the file transfer capability has been switched off, when you attempt to start up FTP, you will get a message from the host to the effect that the computer is not responding. If you get such a message, check the file transfer mode on the Parameter menu to make sure that it is *enabled*.

Using FTP

Use FTP rather than rcp in either of the following situations.

1. You are not thoroughly familiar with the host operating system. FTP is easier to use than rcp, and requires less familiarity with the host commands and directory structure.
2. You are transferring PC *text* (ASCII) files to a host. FTP is a better choice than rcp, because FTP automatically converts the end-of-line characters in text files from CRLF, used by PC-DOS, to LF, used by UNIX machines.

Invoking FTP on the Host Computer

FTP is initiated by the host, so the FTP commands vary, depending on the host system. For full documentation of FTP and commands within FTP, you must refer to the manuals for the host computer. With UNIX systems, you can see online documentation by using the `man ftp` command. See Chapter 4, "Utility Programs," for more information.

The FTP Command

On most systems, the FTP command is entered at the prompt, with the name or IP address of the target machine. For example, if your PC is named "mymachine" and your IP address is 192.17.20.22, then you might enter:

```
% ftp mymachine
```

or

```
% ftp 192.17.20.22
```

which generates a response like this:

```
Connected to 192.17.20.22.  
220 PC Resident FTP server, ready  
Name (192.17.20.22:timk):
```

Most FTP clients will now prompt you for your username and password. If NCSA Telnet is configured for passwords (see Chapter 7, "Installation and Configuration"), then these are required. Otherwise, just press RETURN to bypass the prompts. If you are not prompted for username and password, then assume that you are logged in and continue with your FTP commands.

Invoking FTP Using Alternate Methods

The most common procedure for starting FTP uses the shortcut keystroke, ALT-F. When you press ALT-F, NCSA Telnet types the FTP command described above, automatically including your PC's IP address and the RETURN to initiate the command. For convenience, you will probably prefer ALT-F to the other ways of entering the command. You may want to think of this as a keyboard macro that is pre-set with the FTP command and your IP address.

Another available macro is ALT-I which types your IP address for you. When entering any networking command, FTP for example, you can instantly produce your own network IP address with ALT-I. The following sequence is equivalent to pressing ALT-F.

```
ftp  
(SPACEBAR)  
(ALT-I)  
(RETURN)
```

Use whichever method of invoking FTP that works best for you. Your host computer may or may not accept FTP commands as described here, so you may have to try some variations to find the easiest method for your site. Your system administrator may be able to help.

Using FTP Commands

For most FTPs, after FTP has been invoked and passwords have been checked, you are prompted for individual FTP commands. These commands are also documented in the manuals for the host computer. Most of the FTP implementations have similar commands because they are modeled after the Berkeley UNIX version of FTP. Table 5.1 lists FTP commands that are common to most implementations.

Table 5.1 Common FTP Commands

Command	Action
ascii	Sets mode to ASCII transfer mode (default)
binary	Sets mode to binary (image) transfer mode
cd <i>path</i>	Sets a new default directory on PC
dir	Shows filenames in PC's default directory
get <i>filename</i>	Gets a file from PC and sends it to host
help	Shows online list of FTP commands
put <i>filename</i>	Sends a file from host to PC
pwd	Shows current PC directory name
quit	Exits FTP
remotehelp	Shows list of supported commands on FTP server

ASCII, Binary

The default mode for FTP transfers is ASCII format. If you are transferring graphics or binary data files, change to binary mode before you use the put or get commands. Do this by entering the command **binary**. To reset the ASCII format after sending a binary file, enter the command **ascii**. Figure 5.1 shows an FTP transaction with an ASCII file. Figure 5.2 shows an FTP transaction with a binary file.

Default Directory

Unless you include a full pathname with the command, NCSA Telnet transfers the file to the default directory. As Table 5.1 indicates, you can use FTP commands to identify the current directory (**pwd**), or change the current directory (**cd**).

The change directory command, which you type as **cd**, allows you to change the default disk drive in addition to the default directory. Just use PC standard disk notation, with forward slashes instead of backslashes, for example:

```
ftp> cd D:/files
```

The **pwd** command also returns disk information. After the previous example **cd** command, a **pwd** command returns:

```
D:\FILES
```

Transfer to the PC

Even though you seem to be initiating the transfer from the PC, the transaction operates from the host's side. The practical effect of this could make the commands seem intuitively "backwards." To transfer a file from the host to your PC, you use a `put` command. The form this command takes is:

```
put filename.ext
```

Figure 5.1 shows an example of a `put` command used with an actual file, named `temp2`. The boldface type represents user entries.

Figure 5.1 Put to the PC, ASCII File

```
newton_45% ftp -n 192.17.20.124
Connected to 192.17.20.124.
220 PC Resident FTP server, ready
ftp> put temp2
200 This space intentionally left blank < >
150 Opening connection
226 Transfer complete
262145 bytes sent in 32.61 seconds (7.8 Kbytes/s)
ftp> quit
221 Goodbye
newton_46%
```

Transfer to the Host

A request to send a file from the PC to the host is called a `get`.

```
get filename.ext
```

Figure 5.2 shows a `get` operation, using a binary file named `bridge.pic`. Note that the file was in the directory named `ibmg`, so the `cd` command was used. Again, the boldface type represents user entries. If a text file were to be sent after this, the ASCII mode would have to be reset, using the FTP `ascii` command.

Figure 5.2 Get from the PC, Binary File

```
newton_41% ftp -n 192.17.20.124
Connected to 192.17.20.124.
220 PC Resident FTP server, ready
ftp> bin
200 Type set to I, binary transfer mode
ftp> cd /ibmg
250 Chdir okay
ftp> get bridge.pic
200 This space intentionally left blank < >
150 Opening connection
226 Transfer complete
262144 bytes received in 9.22 seconds (28 Kbytes/s)
ftp>
ftp> quit
221 Goodbye
newton_42%
```


FTP Status Indicator

After you have entered a `put` or `get` command, in the lower right hand corner of your screen you will see the filename and an indication of the number of bytes transferred (in the case of a `put`) or left to be transferred (in the case of a `get`). These numbers are updated every couple of seconds to help you keep track of the progress of the file transfer. After the command has completed, the filename goes away and the host usually prints a confirmation message, as seen in Figures 5.1 and 5.2.

mput, mget

On many telnet hosts, you can transfer multiple files sequentially with one command, either `mput` or `mget`, used with wildcard characters. (PC wildcard characters are `?` for a single character and `*` for multiple characters.) For instance, to transfer the multiple files, `teln.2`, and `teln.3`, you could enter `mget` (or `mput`)`*?`

Bug with mget

If you transfer multiple binary files using a UNIX host, note that there is a bug in `mget` as implemented on some systems (especially 4.2 BSD UNIX). When used in binary mode, `mget` adds a carriage return to the filenames as they are transferred. The files themselves are not affected. Use a UNIX utility to remove the carriage return from the filename. In ASCII mode, there is no problem.

File Transfer Using rcp

`rcp` is a Berkeley UNIX feature found in Sun UNIX and other 4.2 BSD systems. The `rcp` program cannot do the end of line translation required for text files, so it is typically used for binary files only. UNIX `rcp` is designed for files copied UNIX-to-UNIX, but NCSA Telnet's `rcp` server supports transfer between a UNIX system and a PC.

NOTE: NCSA Telnet's `rcp` server does not support the recursive option (`-r`).

Setting Up rcp

To use `rcp` efficiently, you should be familiar with the UNIX host, including:

- `rcp` command and options
- directory structures used with `rcp`
- the relationship between the UNIX shell and `rcp` wildcards

Details on these concepts are in the manuals for UNIX.

The internet (IP) number and machine name for each PC must be in the UNIX `/etc/hosts` file for every machine that is transferring files to and from your PC. If `rcp` does not recognize your PC's name, have your system administrator check or add that name in the UNIX machine's hosts file.

Using rcp

Use `rcp` for binary transfers to or from a UNIX system if you are familiar with the UNIX hosts, as described earlier.

There are two advantages of using `rcp` rather than FTP, on binary file transfers to or from a UNIX machine.

- `rcp` handles wildcards more efficiently than FTP does.
- `rcp` handles directories more easily than FTP does.

Avoiding Use of NCSA Telnet's rcp Server

Do not use the `rcp` server to transfer text files. Text files on the PC contain extra RETURN characters that NCSA Telnet's `rcp` server does not remove when transferring to UNIX.

NOTE: Do not initiate an `rcp` while a file transfer is already in process, or both transfers may fail.

Using rcp Between the PC and UNIX

The `rcp` program is fully documented in the manuals for UNIX hosts that support `rcp` transfers. Refer to those manuals to learn about `rcp`. As a reminder, the general syntax of the `rcp` command is given here. The target filename can be a directory name and wildcards are allowed, as noted in the next section.

% <code>rcp mypc:filename unixfile</code>	From PC to UNIX
% <code>rcp unixfile mypc:filename</code>	From UNIX to PC

Example: PC to UNIX

Following is a sample command to transfer a file named `image` from the PC named `mypc` to the current directory on the UNIX host (indicated by a dot or period). Note that the only blank spaces are after the command `rcp`, and between the filename and the target directory name (in this case, the dot for the current directory).

```
% rcp mypc:image .
```

Example: UNIX to PC

Following is a sample command to transfer a file named `mygraf` in the directory `graphs` from the UNIX host and put it in the directory `grafs` on the PC.

```
% rcp /graphs/mygraf mypc:/grafs
```

Using Wildcards

You can use wildcard characters with the `rcp` command to transfer multiple files between a PC client and a UNIX host. Wildcards that refer to UNIX files should be in the UNIX regular expression format. You must set wildcards that refer to PC files apart by using backslashes or quotes. See the following examples.

Example: Backslashes Used With Wildcards

One way of using PC wildcards with the `rcp` server is to use the backslash character (`\`) to keep the UNIX shell from interpreting the special characters. Suppose you want to transfer all files that have `image` at the beginning of the filename. Instead of the usual PC format, `image*.*`, you must precede each asterisk with a backslash. The full `rcp` command would look like this:

```
% rcp mypc:image\*.* .
```

Without the backslashes, the UNIX shell would try to translate the asterisk on the host rather than on the PC.

Example: Quotes Used With Wildcards

The second way of using PC wildcards with the `rcp` server is to insert quotes around the entire argument. The command to transfer the files beginning with `image` would look like this:

```
% rcp "mypc:image*.*" .
```