

For Xerox Internal Use Only -- November 17, 1977

Alto Pup FTP

October 30, 1977

1

Alto Pup File Transfer Program

FTP is a Pup-based File Transfer Program for moving files to and from an Alto file system. The program comes in 3 parts:

- 1) An FTP Server, which listens for file transfer requests from other hosts,
- 2) An FTP User, which initiates file transfers under control of either the keyboard or the command line, and
- 3) A User Telnet for logging into a remote host using the Pup Telnet protocol.

1. Concepts and Terminology

Transferring a file from one machine (or "host") to another over a network requires the active cooperation of programs on both machines. In a typical scenario for file transfer, a human user (or a program acting on his behalf) invokes a program called an "FTP User" and directs it to establish contact with an "FTP Server" program on another machine. Once contact has been established, the FTP User initiates requests and supplies parameters for the actual transfer of files, which the User and Server proceed to carry out cooperatively. The FTP User and FTP Server roles differ in that the FTP User interacts with the human user (usually through some sort of keyboard interpreter) and takes the initiative in user/server interactions, whereas the FTP Server plays a comparatively passive role.

The question of which machine is the FTP User and which is the FTP Server is completely independent of the direction of file transfer. The two basic file transfer operations are called "Retrieve" and "Store"; the Retrieve operation causes a file to move from Server to User, whereas Store causes a file to move from User to Server.

The Alto FTP subsystem contains both an FTP User and an FTP Server, running as independent processes. Therefore, to transfer files between a pair of Altos, one should start up the FTP subsystem on both machines, then issue commands to the FTP User process on one machine directing it to establish contact with the FTP Server process in the other machine. Subsequent file transfers are controlled entirely from the FTP User end, with no human intervention required at the Server machine.

Transferring files to or from a Maxc system or an IFS involves establishing contact with FTP Server processes that run all the time on those machines. Hence, one may simply invoke the Alto FTP subsystem and direct its FTP User process to connect to the machine.

In the descriptions that follow, the terms "local" and "remote" are relative to the machine on which the FTP User program is active. That is, we speak of typing commands to our "local" FTP User program and directing it to establish contact with an FTP Server on some "remote" system.

For Xerox Internal Use Only -- November 17, 1977

Alto Pup FTP

October 30, 1977

2

machine. A Retrieve command then copies a file from the "remote" file system to the "local" file system, whereas a Store command copies a file from the "local" file system to the "remote" file system. Furthermore, we refer to "local" and "remote" filenames. These must conform to the conventions used by the "local" and "remote" host computers, which may be dissimilar (for example, Alto versus Maxc). The Alto FTP knows nothing about Maxc filename conventions or vice versa.

The Alto FTP subsystem also includes a third process, called a "User Telnet", which simulates a terminal in a manner exactly analogous to the Chat subsystem (though lacking some of its finer features). By this means, you may log in to a file system machine to perform operations not directly available via the basic file transfer mechanisms. If you log into Maxc, it is even possible to run "PupFTP", the Maxc FTP User program, and direct it to establish contact with the FTP Server in your own Alto. (Nova users will recognize this as analogous to the old Minx/NewMCA arrangement for file transfer.) You should probably not try this unless you really understand what you are doing, however, since the terms "local" and "remote" are relative to Maxc rather than to your Alto (since the FTP User program is running on Maxc in this case), which can be confusing.

2. Calling the FTP Subsystem

A number of options are available when running FTP. The program decides which parts of itself to enable and where user commands will come from by inspecting the command line. The general form of the command line to invoke FTP looks like:

FTP[/<Global-switches>] [<Host-name> [<Command-list>]]

The square brackets denote portions of the command line that are optional and may be omitted.

Global switches, explained below, select some global program options such as using the Trident disk instead of the Diablo. The first token after the <global-switches>, if present, is assumed to be a <host-name> (a discussion of which appears later in the description of the "Open" command). The User FTP will attempt to connect to the FTP Server on that host. After connecting to the server, if a <command-list> is present, an interpreter is started which feeds these commands to the User FTP. When the command list is exhausted, FTP returns to the Alto Executive. If no command list is present, an interactive keyboard command interpreter is started.

Each global switch has a default value which is used if the switch is not explicitly set. To set a switch to 'false' proceed it with a 'minus' sign (thus FTP/-S means 'no Server'), to set a switch to 'true' just mention the switch.

Switch	Default	Function
--------	---------	----------

/S true [Server] starts the FTP Server. The Server is not started if the User is enabled and is being controlled from the command line.

↑L

For Xerox Internal Use Only -- November 17, 1977

Alto Pup FTP

October 30, 1977

3

- | | | |
|----|-------|---|
| /U | true | [User] starts the FTP User. As explained above, the interactive command interpreter or the command line interpreter will be started depending on the contents of the command line. |
| /C | true | [Chat] starts the Telnet. The Telnet is not started if the User is enabled and is being controlled from the command line, or if the system disk is TPO. |
| /T | false | [Trident] sets the system disk to be a Trident drive. The default is 0, but can be changed by following the /T with a unit number between 0 and 7 (thus FTP/T5 means use Trident unit 5). User and Server commands apply to files on this disk but command line input is still taken from Com.cm on the Diablo drive. |
| /R | true | [Ram] allows Ftp to use some microcode which speeds things up slightly. If your Alto has no ram, this switch is ignored. |
| /B | false | [Boot] creates 'FTP.Boot'. When this file is booted, it will read the disk descriptor, open the system log file, and enable the interactive command interpreter. |
| /D | false | [Debug] starts FTP in debug mode. |
| /A | false | [AllocatorDebug] enables storage allocator checking. |

The rest of the global switches are explained below under 'Server Options'.

3. The FTP Display

The top inch or so of the display contains a title line and an error window. The title line displays the release date of that version of FTP, the current date and time, the machine's internetwork address, and the number of free pages on the disk. The error window displays certain error messages if they arrive from the network (errors are discussed in more detail below). A window is created below the title line for each part of FTP which is enabled during a session (server, user, and telnet).

If the FTP Server is enabled, it opens a window and identifies itself. If a User FTP subsequently connects to this Server, the User's network address will be displayed. The Server will log the commands it carries out on behalf of the remote User in this window. The Server is not enabled when FTP is being controlled from the command line.

The FTP User opens the next window down and identifies itself. The command herald is an asterisk.

The User Telnet opens the bottommost window, identifies itself, and waits for a host name to be entered. The Telnet is not enabled when FTP is being controlled from the command line.

↑L

For Xerox Internal Use Only -- November 17, 1977

Alto Pup FTP

October 30, 1977

4

3.1. Directing Keyboard input to the User and Telnet Windows

The bottom two unmarked keys control which window gets characters from the keyboard. Hitting the unmarked key to the right of 'right-shift' (also known as the 'Swat key') directs keyboard input to the Telnet window. Hitting the unmarked key to the right of the 'return' key (also known as the 'Chat key') directs keyboard input to the FTP User window. The window which currently owns the keyboard will blink a cursor at the next character position if it is waiting for type-in.

4. Keyboard Command Syntax

FTP's interactive command interpreter presents a user interface very similar to that of the Alto Executive. Its command structure is also very similar to that of the Maxc Pup FTP program (PupFTP), and the Maxc ArpaNet FTP program (FTP). The standard editing characters, command recognition features, and help facility (via "?") are available.

4.1. Keyboard Commands

OPEN <host name>

Opens a connection to the FTP Server in the specified host. FTP permits only one user connection at a time. In most cases the word "OPEN" may be omitted: i.e., a well formed <host name> is a legal command and implies a request to "OPEN" a connection. FTP will try for one minute to connect to the specified host. If you made a mistake typing the host name and wish to abort the connection attempt, hit the middle unmarked key (to the right of <return>).

Ordinarily, the host name can be a string, e.g., "Maxc". Most Altos and Novas have names which are registered in Name Lookup Servers. So long as a name lookup server is available, FTP is able to obtain the information necessary to translate a known host name to an inter-network address.

If the host name of the server machine is not known, you may specify an inter-network address in place of the host name. The general form of an inter-network address is:

<network> # <host> # <socket>

where each of the three fields is an octal number. The <network> number designates the network to which the Server host is connected (which may be different from the one to which the User host is connected); this (along with the "#" that follows it) may be omitted if the Server and User are known to be connected to the same network. The <host> number designates the Server host's address on <network>. The <socket> number designates the actual Server process on that host; ordinarily it should be omitted, since the default is the regular FTP server socket. Hence, to connect to the FTP server running in Alto host number 123 on the directly-connected Ethernet, you should say "OPEN 123#" (the trailing "#" is required).

↑L

For Xerox Internal Use Only -- November 17, 1977

Alto Pup FTP

October 30, 1977

5

CLOSE

Closes the currently open User FTP connection.

LOGIN <user name> <password>

Supplies any login parameters required by the remote server before it will permit file transfers. FTP will use the user name and password in the Operating System, if they are there. Logging into FTP will set the user name and password in the OS (in the same manner as the Alto Executive's "Login" command).

When you issue the "Login" command, FTP will first display the existing user name known to the OS. If you now type a space, FTP will prompt you for a password, whereas if you want to provide a different user name, you should first type that name (which will replace the previous one) followed by a space. The command may be terminated by carriage return after entering the user name to omit entering the password.

The parameters are not immediately checked for legality, but rather are sent to the server for checking when the next file transfer command is issued. If a command is refused by the server because the name or password is incorrect, FTP will prompt you as if you had issued the LOGIN command and then retry the command. Hitting delete in this context will abort the command.

A user name and password must be supplied when transferring files to and from a Maxc system or an IFS. The Alto FTP Server requires a user-password to be supplied if the server machine's disk is password-protected and if the password in the server machine's OS does not match the password on the disk. Thus if the OS was booted and FTP invoked because a Request-for-Connection was received (which bypasses password checking), FTP will refuse access to files unless a password is supplied. However if the OS was booted normally, FTP assumes that the disk owner (who knew the password) will control access by using the server option switches. The user-name is ignored.

CONNECT <directory name> <password>

Requests the FTP server to "connect" you (in the Tenex sense) to the specified directory on the remote system. The password may be omitted by typing carriage return after the directory name. As with LOGIN, these parameters are not verified until the next transfer command is issued. At present, the "Connect" command is meaningful only when transferring files to or from a Maxc system or an IFS; the Alto FTP server currently ignores connect requests. If the "multiple directory" feature of the Alto Operating System ever comes into widespread use, this may be changed.

RETRIEVE <remote filename>

Initiates transfer of the specified remote file to the local host. The syntax of <remote filename> must conform to the remote host's file system name conventions.

If the server can find the file, FTP will then print out the complete filename followed by the message "to local file <local filename> [Old|New file]", where the local filename is generally the same as the remote filename without directory or version. At this point you may make one of three choices:

For Xerox Internal Use Only -- November 17, 1977

Alto Pup FTP

October 30, 1977

6

1. Type Carriage Return to cause the data to be transferred to <local filename>.
2. Type Delete to indicate that the file is not to be transferred.
3. Type any desired local filename followed by Return. The previous local filename will disappear, the new filename will replace it, and FTP will tell you whether a file exists with that name. This filename must conform to local conventions. You now have the same three choices.

If the remote filename designates multiple files (the remote host permits "*" or some equivalent in file names), each file will be transferred separately and FTP will ask you to make one of the above three choices for each file. At present, only Maxc and IFS support this capability. That is, you may supply "*"s in the remote filename when retrieving files from Tenex or an IFS, but not when retrieving files from another Alto.

STORE <local filename>

Initiates transfer of the specified local file to the remote host. Alto file name conventions apply to the <local filename>; "*" expansion is not supported. FTP will suggest a remote filename to which you should respond in a manner similar to that described under RETRIEVE except that if you supply a different filename, it must conform to the remote file system's conventions. The default remote filename is one with the same 'name body' (name and extension) as the local file; the remote server defaults other fields as necessary. If the remote host is a Maxc system or an IFS, then the directory is that most recently supplied in LOGIN or CONNECT commands and the version is the next higher.

DUMP <remote filename>

Bundles together a group of files from the local file system into a 'dump-format' file (see the Alto Executive documentation for the dump-file format and more on dump-files in general) and stores the result as <remote filename>. FTP will ask you for the names of local files to include in the dump-file. Terminate the dump by typing just <return> when FTP asks for another filename. By convention, files in dump-format have extension '.dm'.

LOAD <remote filename>

Performs the inverse operation of DUMP, unbundling a dump-format file from the remote file system and storing the constituent files in the local file system. For each file in the dump-file, FTP will suggest a local file name and tell you whether a file by that name exists on your disk. You should respond in the manner described under RETRIEVE.

LIST <remote file designator>

Lists all files in the remote file system which correspond to <remote file designator>. The remote file designator must conform to file naming conventions on the remote host, and may designate multiple files if "*" expansion or some equivalent is supported there. If the <remote filename> is terminated by a comma, FTP prints a prompt of "**" at the left margin and prepares to accept one or more subcommands. These subcommands request printout of additional information about each file. To terminate subcommand

↑L

For Xerox Internal Use Only -- November 17, 1977

Alto Pup FTP

October 30, 1977

7

input, type a <return> in response to the subcommand prompt. The subcommands are:

Type	Print file type and byte size.
Length	Print length of file in bytes.
Creation	Print date of creation.
Write	Print date of last write.
Read	Print date of last read.
Times	Print times as well as dates.
Author	Print author (creator) of file.
Verbose	Same as Type+Write+Read+Author.
Everything	Print all information about the file.

This information is only as reliable as the Server that provided it, and not all Servers provide all of these file properties. Altos derive much of this information from hints, so do not be alarmed if it is sometimes wrong.

DELETE <remote filename>

Deletes <remote filename> from the remote filesystem. The syntax of the remote filename must conform to the remote host's file system name conventions. After determining that the remote file exists, FTP asks you to confirm your intention to delete it. If the remote filename designates multiple files (the remote host permits "*" or some equivalent in file names), FTP asks you to confirm the deletion of each file.

RENAME <old filename> <new filename>

Renames <old filename> in the remote filesystem to be <new filename>. The syntax of the two filenames must conform to the remote host's file system name conventions, and each filename must specify exactly one file.

QUIT

Returns control to the Alto Executive, closing all open connections.

TYPE <data type>

Forces the data to be interpreted according to the specified <data type>, which may be TEXT or BINARY. Initially the type is UNSPECIFIED, meaning that the source process should, if possible, decide on the appropriate type based on local information.

BYTE-SIZE <decimal number>

Applicable only to files of type Binary, BYTE-SIZE specifies the logical byte size of the data to be transferred. The default is 8.

EOL <convention>

Applicable only to files of type Text, EOL specifies the End-of-Line Convention to be used for transferring text files. The values for <convention> are CR, CRLF, and TRANSPARENT. The default is CR.

DEVICE <device>

Causes <device> to be used as the default device in data transfer commands (essentially it causes <device> to be attached to all remote filenames that do not explicitly mention one). The punctuation separating <device> from the other components of a

↑L

For Xerox Internal Use Only -- November 17, 1977

Alto Pup FTP

October 30, 1977

8

remote filename should not be included. For example you might specify "Device DSK" to Tenex, not "Device DSK:"

DIRECTORY <directory name>

Causes <directory name> to be used as the default remote directory in data transfer commands (essentially it causes <directory-name> to be attached to all remote filenames that do not explicitly mention a directory). Specifying a default directory in no way modifies your access privileges, whereas CONNECTing gives you 'owner access' (and usually requires a password). Explicitly mentioning a directory in a file name overrides the default directory, which overrides the connected directory, which overrides the login directory. Punctuation separating <directory name> from other parts of a remote filename should not be included. For example you might type "Directory Alto" not "Directory <Alto>".

USER

Allows you to toggle switches which control operation of the FTP User. There is currently only one: DEBUG, which controls display of protocol interactions. Warning: this printout (and the corresponding one in the SERVER command below) sometimes includes passwords.

SERVER

Allows you to toggle switches which control operation of the FTP Server. The switches are PROTECTED, OVERWRITE, KILL, and DEBUG. The first three are explained below under 'Server Options'.

TELNET

Allows you to toggle switches which control operation of the Telnet. There is currently only one: CLOSE, which closes the Telnet connection if one is open, and clears the Telnet window.

5. Command Line Syntax

The User FTP can also be controlled from the command line. As explained above, the first token after the subsystem name and server switches must be a legal host name; if the User FTP can't connect to the FTP Server on that host it will abort and return control to the Alto Executive. If a command list follows the host name, the command line interpreter is invoked instead of the interactive keyboard interpreter. This permits the full capabilities of the Alto Executive (filename recognition, "*" expansion, command files, etc.) to be used in constructing commands for FTP.

Each command is of the form:

<Keyword>/<SwitchList> <arg> ... <arg>

To get a special character (any one of "*#';") past the Alto Executive, it must be preceded by a single quote. To get a "/" into an FTP argument, the "/" must be proceeded by two single quotes (the second one tells FTP to treat the "/" as an ordinary character in the argument, and the first one gets the second one past the Alto Executive).

↑L

For Xerox Internal Use Only -- November 17, 1977

Alto Pup FTP

October 30, 1977

9

Unambiguous abbreviations of command keywords (which in most cases amount to the first letter) are legal. However, when constructing command files, you should always spell commands in full, since the uniqueness of abbreviations in the present version of FTP is not guaranteed in future versions.

A command is distinguished from arguments to the previous command by having a switch on it, so every command must have at least one switch. The switch "/C" has no special meaning and should be used on commands where no other switches are needed or desired.

When the command line has been completely executed, FTP returns control to the Alto Executive. If the Debug command has been given or any errors have occurred, FTP will pause before finishing. This prevents error messages from disappearing before you have a chance to see them. In this situation, FTP will terminate after one minute has elapsed or any key has been struck on the keyboard.

5.1. Command Line Commands

LOGIN/C <user name> <password>

See description in "Keyboard commands". The <password> may be omitted.

LOGIN/Q <user name>

Causes FTP to prompt the user for the password. This form of LOGIN should be used in command files since including passwords in command files is bad practice.

CONNECT/C <directory name> <password>

See description in "Keyboard commands". The <password> may be omitted.

CONNECT/Q <directory name>

Causes FTP to prompt the user for the password needed to connect to the specified <directory name>. This form of CONNECT should be used in command files since including passwords in command files is bad practice.

RETRIEVE/C <remote filename> ... <remote filename>

Retrieves each <remote filename> and writes it in the local file system, constructing a local file name from the name body of the actual remote file name as received from the Server. FTP will overwrite an existing file unless the /N (No overwrite) switch is appended to the RETRIEVE command keyword. If the remote host allows "*" (or some equivalent) in a filename, a single remote filename may result in the retrieval of several files. (Note that you must quote the "*" to get it past the Alto Executive's command scanner.) As mentioned previously, this capability is implemented only by Maxc and IFS FTP Servers at present.

RETRIEVE/S <remote filename> <local filename>

Retrieves <remote filename> and names it <local filename> in the local file system. This version of RETRIEVE must have exactly two arguments. FTP will overwrite an existing file unless the /N (No overwrite) switch is also appended to the RETRIEVE command keyword. The remote filename should not cause the server to send multiple files.

↑L

For Xerox Internal Use Only -- November 17, 1977

Alto Pup FTP

October 30, 1977

10

RETRIEVE/U <remote filename> ... <remote filename>

Retrieves <remote filename> if its creation date is later than the creation date of the local file. A file will not be retrieved unless a local file with name and extension equal to the name and extension of the remote filename exists, or if the FTP server does not send a CREATION-DATE property. This option can be combined with RETRIEVE/S to rename the file as it is transferred.

RETRIEVE/V

Requests confirmation from the keyboard before writing a local file. This option is useful in combination with the Update option since creation date is not a fool-proof criterion for updating a file.

STORE/C <local filename> ... <local filename>

Stores each <local filename> on the remote host, constructing a remote filename from the name body of the local filename. A local filename may contain "*", since it will be expanded by the Alto Executive into the actual list of filenames before the FTP subsystem is invoked.

STORE/S <local filename> <remote filename>

Stores <local filename> on the remote host as <remote filename>. The remote filename must conform to the file name conventions of the remote host. This version of store must have exactly two arguments.

DUMP/C <remote filename> <local filename>...<local filename>

See the description in "Keyboard Commands".

LOAD/C <remote filename>

See the description in "Keyboard Commands". If the /V switch is appended to the LOAD command keyword, FTP will request confirmation before writing each file. Type <return> to write the file, to skip it. FTP will overwrite an existing file unless the /N (No overwrite) switch is appended to the LOAD command keyword.

DELETE/C <remote filename>

See the description in "Keyboard Commands". If the /V switch is appended to the DELETE command keyword, FTP will request confirmation before deleting each file. Type <return> to delete the file, and (oops!) if you don't want to delete it.

RENAME/C <old filename> <new filename>

See the description in "Keyboard Commands".

TYPE/C <data type>

See the description in "Keyboard Commands".

BYTE-SIZE/C <decimal number>

See the description in "Keyboard Commands".

EOL/C <convention>

See the description in "Keyboard Commands".

DEVICE/C

See the description in "Keyboard Commands".

DIRECTORY/C <default directory>
See description in "Keyboard commands".
↑L

For Xerox Internal Use Only -- November 17, 1977

Alto Pup FTP

October 30, 1977

11

DEBUG/C

See the description of the DEBUG subcommand under the USER command in "Keyboard Commands".

6. Telnet

FTP provides a simple User Telnet as a convenience for logging into a remote host (e.g., Maxc) to poke around without having to leave the FTP subsystem and start Chat. It lacks most of the creature comforts Chat provides, such as automatic attaching to detached jobs, automatic logging in, etc. The Telnet is not enabled when the User FTP is being controlled from the command line. When the Telnet does not have an open connection, it waits for you to type a host name with the syntax explained above for the OPEN command, and then attempts to connect to the specified host. If you wish to abort the connection attempt, hit the bottom unmarked key (opposite right-shift). You can get a larger Telnet window by not starting a server (type FTP/-S to the Executive).

7. File Property Defaulting

Without explicit information from the file system, it is often difficult to determine whether a file is Binary or Text, if Binary, what its byte-size is, and if Text, what End-Of-Line convention is used. The User and Server FTPs use some simple heuristics to determine the correct manner in which to transfer a file. The heuristics generally do the right thing in the face of incomplete information, and can be overridden by explicit commands from a human user who knows better.

The FTP protocol specifies a standard representation for a file while in transit over a network. If the file is of type Binary, each logical byte is packed right-justified in an integral number of 8-bit bytes. The byte-size is sent as a property along with the file. If the file is of type Text, each character is sent right-justified in an 8-bit byte. An EOL\convention may be sent as a file property. The default is that <return> marks the end of a line.

7.1. File Types

FTP determines the type of a local file by reading it and looking for bytes with the high-order bit on. If any byte in the file has a high-order bit on, the file is assumed to be Type Binary, otherwise it is assumed to be Type Text.

FTP will warn you, but allow you to send what it thinks to be a text file as type Binary, since no information is lost. It will refuse to send a binary file as type text.

Don't specify a Type unless you know what you are doing. The heuristic will not lose information.

↑L

For Xerox Internal Use Only -- November 17, 1977

Alto Pup FTP

October 30, 1977

12

7.2. Byte-Size

If a file is type Binary, the byte-size is assumed to be 8 unless otherwise specified. The FTP User and Server will both accept binary files of any byte-size and write them as 8 bit bytes on the disk. No transformation is done on the data as it is written to the disk: it is stored in network default format. Since there is no place in the Alto file system to save the byte-size property, it is lost.

Similarly, requests for Binary files will be honored with any byte size, and whatever is on the disk will be sent to the net without transformation. Since Alto files have no byte size information, the byte-size property will be defaulted to 8 unless otherwise specified (by the BYTE command), in which case whatever was otherwise specified will be sent as the byte size.

Don't specify a Byte-size unless you know what you are doing. Alto-Alto transfers can't go wrong. Alto-Maxc transfers with weird byte-sizes will not work unless the byte-size specified in the Alto to Maxc direction is the same as the byte-size in which the file was stored on the Alto. If it isn't, the Alto will not give any error indication, but the result will be garbage.

7.3. End-of-Line Conventions

FTPs are expected to be able to convert text files between the local file system End-Of-Line (EOL) convention and the network convention. Conveniently enough, the Alto file system's internal representation of a text file is the same as the network standard (a bare <return> marks the end of a line). The Alto FTP does not do any transformations on text files. It will refuse to store a text file coming in from the net whose EOL convention is CRLF.

As an escape to bypass conversion and checking, EOL convention 'transparent' tells both ends NOT to convert to network standard, but rather send a file 'as is'. This is included for Lisp source files which use EOL convention CRLF and contain internal character pointers that are messed up by removing line feed characters.

Don't specify an EOL convention unless you know what you are doing. If your text file is a Lisp source file, specify EOL convention 'Transparent'.

8. Abort and Error messages

Error and Abort packets are displayed in a window above the title line. Abort packets are fatal; Error packets are not necessarily so. The most common Abort message is "Timeout. Good bye", generated when a server process has not received any commands for a long time (typically 5 minutes).

The most common Error message is "Port IQ overflow" indicating a momentary shortage of input buffers at the remote host. Receiving an Error Pup does not imply that the file in transit has been damaged.

↑L

For Xerox Internal Use Only -- November 17, 1977

Alto Pup FTP

October 30, 1977

13

Loss of or damage to a file will be indicated by an explicit message in the User FTP window. The next iteration of the Pup protocols will probably rename 'Error Pups' to be 'Information Pups'.

9. Server Options

Server options are controlled by switches on the subsystem name and subcommands of the SERVER keyboard command. There are currently four options:

switch Default Function

none	If no server option is specified, retrieve requests (disk to net) are allowed. Store requests (net to disk) are allowed unless the store would overwrite an already existing file.
/P	false [Protected] Retrieve requests are allowed. No stores are allowed.
/O	false [Overwrite] Retrieve requests are allowed. Store requests can overwrite files.
/K	false [Kill] FTP will return to the Alto Exec when the server connection is closed. A simple form of remote job entry can be performed if the user FTP stores into Rem.cm (Com.cm on Novas).

10. CLI Examples

Here are some examples of Command lines.

To transfer files Ftp.run and Ftp.syms from the Alto called "Michelson" to the Alto called "Morley", one might start up FTP on Michelson (to act as an FTP Server), then walk over to Morley and type:

FTP Michelson Retrieve/c Ftp.run Ftp.syms

Alternatively, one could start an FTP server on Morley (invoking it by "FTP/O" to permit files to be overwritten on Morley's disk), then issue the following command to Michelson:

FTP Morley Store/c Ftp.run Ftp.syms

The latter approach is recommended for transferring large groups of files such as "*.run" (since expansion of the "*" will be performed by the Alto Executive).

To retrieve User.cm from the FTP server running on Alto serial number 123 (name unknown, but it is on the local Ethernet):

FTP 123'# Retrieve User.cm

For Xerox Internal Use Only -- November 17, 1977

Alto Pup FTP

October 30, 1977

14

Note that the "#" must be preceded by a single quote when included in a command line, since otherwise the Alto Executive does funny things with it. (Quotes are not necessary when typing to FTP's interactive keyboard interpreter).

To start FTP, have the FTP User connect to Maxc, and then accept further commands from the keyboard:

FTP Maxc

To retrieve <System>Pup-Network.txt from Maxc and store it on the Alto as PupDirectory.bravo, and store PupRTP.bcpl, Pup1b.bcpl, and PupBSPStreams.bcpl on <DRB> with their names unchanged:

```
FTP Maxc Connect/c drb mypassword Retrieve/s <System>Pup-
Network.txt PupDirectory.bravo Store/c PupRTP.bcpl Pup1b.bcpl
PupBSPStreams.bcpl
```

To retrieve the latest copy of all .RUN files from the <alto> directory, overwriting copies on the Alto disk (The single quote is necessary to prevent the Alto Executive from expanding the "*"):

FTP Maxc Ret/c <alto>'*.run

To update the Alto disk with new copies of all <alto> files whose names are contained in file UpdateFiles.cm, requesting confirmation before each retrieval:

FTP Maxc Dir/c Alto Ret/u/v @UpdateFiles.cm@

To store all files with extension .BCPL from the local Alto disk to your login directory on Maxc (the Alto Executive will expand "*.bcpl" before invoking FTP):

FTP Maxc St/c *.bcpl

To retrieve <System>Host-name/descriptor-file.txt';43 (two single quotes are necessary to get the "/" past the Alto Executive and the FTP command scanner, and one quote is necessary to get the ";" past the Alto Executive):

FTP Maxc Ret/c <System>Host-name''/descriptor-file.txt';43

To send Prog.f4, Data.f4, and Command.f4 to Fortran-Machine and then cause the FTP server on Fortran-Machine to quit (presumably to execute Prog.f4 on Data.f4 according to the commands in Command.f4):

```
FTP Fortran-Machine Store/c Prog.f4 Data.f4 Store/s
Command.f4 Rem.cm
```

FTP on Fortan-Machine must be started with the /K server option switch, and Command.f4 should re-invoke FTP as its last act so that the results can be retrieved.

To release a new version of FTP, I incant:

@ReleaseAltoFtp.cm@

which the Alto Executive expands into:

↑L

For Xerox Internal Use Only -- November 17, 1977

Alto Pup FTP

October 30, 1977

15

FTP Maxc Connect/q Alto Store/c Ftp.run Ftp.syms Connect/q
AltoSource Dump/c Ftp.dm @ftp.cm@

and then into:

FTP Maxc Connect/q Alto Store/c Ftp.run Ftp.syms Connect/q
AltoSource Dump/c Ftp.dm @FtpSubsys.cm@ @FtpPackage.cm@
Ftp.cm

and finally into:

FTP.run Maxc Connect/q Alto Store/c Ftp.run Ftp.syms
Connect/q AltoSource Dump/c Ftp.dm Ftp.bcpl FtpNv.bcpl
FtpInit.bcpl FtpInit1.bcpl FtpNvInit.bcpl FtpUserInit.bcpl
FtpKbd.decl FtpKbdInit.bcpl FtpKbd.bcpl FtpKbd1.bcpl
FtpKbd2.bcpl FtpCliInit.bcpl FtpCli.bcpl FtpCli1.bcpl
FtpCli2.bcpl FtpCliUtil(bcpl FtpMiscb.bcpl FtpMisca.asm
FtpServerInit.bcpl FtpServer.bcpl FtpTelnetInit.bcpl
FtpTelnet.bcpl FtpKeys.bcpl FtpMc.mu RamTrap.mu GetFrame.mu
PupChecksum.mu CompileFtpMc.cm FtpSubsys.cm
CompileFtpSubsys.cm CompileAltoFtp.cm LoadAltoFtp.cm
LoadHiddenFtp.cm ReleaseAltoFtp.cm CompileNovaFtp.cm
LoadDosFtp.cm LoadRDosFtp.cm FtpProt.decl FtpUserProt.bcpl
FtpUserProtFile.bcpl FtpUserProtMail.bcpl
FtpServProtFile.bcpl FtpServProtMail.bcpl FtpPLInit(bcpl
FtpPLInitProt(bcpl FtpPLInit1(bcpl FtpUtilInit(bcpl
FtpUtil1B.bcpl FtpUtilA.asm FtpUtilXfer(bcpl FtpUtilDmpLd(bcpl
FtpOEPIInit(bcpl CompileFtpPackage.cm DumpFtpPackage.cm
FtpPackage.cm Ftp.cm

To load Ftp.dm from <AltoSource>, expanding it out into its constituent files:

FTP Maxc Load/c <AltoSource>Ftp.dm

To cause Memo.ears to be spooled for printing on Ears by the Maxc printing system:

FTP Maxc Store/s Memo.ears LPT:

This also works for Press files and unformatted text files if you know what you are doing. It does not do the right thing for Bravo-format files.

To use FTP as a stop-gap IFS:

FTP/T-UO

This starts only a server with overwriting of existing files permitted. When using the trident, there isn't enough space to start both a User and a Server.

11. Nova FTP

FTP is also available running under Dos Rev 4 and RDos Rev 3. Since
the Nova versions are nearly identical to the Alto version (the same
↑L

For Xerox Internal Use Only -- November 17, 1977

Alto Pup FTP

October 30, 1977

16

source files except for initialization), only the differences are listed here.

- 1) Ignore all references to display windows. All printout goes to device #11, whatever that is.
- 2) Ignore all references to 'unmarked keys' such as for aborting connection attempts and directing keyboard input to various windows.
- 3) Lack of memory and lack of a windowing display made including a Telnet impractical on the Nova.
- 4) The syntax of the command line is limited to that acceptable to the Nova operating system. Warning: the command line examples given above may not all work on a Nova.
- 5) The Nova OS does not maintain a username or password, so all interactions with a Maxc system or an IFS will require the user to supply them.
- 6) File creation dates are not supported, so there is no Update option to RETRIEVE, and the LIST command does not show dates.

11.1. FTP releases

The Nova FTP subsystem consists of a save-file, FTP.SV, and an overlay-file, FTP.BB. You must get BOTH files when a new version of FTP is released. If you rename FTP.SV you must rename FTP.BB to have the same name (for instance if you rename FTP.SV to be OLDFTP.SV you must also rename FTP.BB to be OLDFTP.BB). New releases of FTP will be distributed as dump files with a consistant pair of save- and overlay-files.

11.2. Device codes

FTP assumes that Nova Ethernet interfaces have device codes 73 and 74, 63 and 64, or 53 and 54. It will use all interfaces with these codes that seem (from reading some status registers) to be Ethernets. The Dos version of FTP assumes that Nova MCA interfaces are device code 6 and 7, or 46 and 47. It will use all interfaces with these codes that seem to be MCAs.

11.3. RDos notes

FTP is big, and will not run under some RDos systems. If you have trouble, generate a smaller system and boot from it when running FTP. FTP disables parts of RDos with patches which may not work for versions other than Rev 3. It will NOT work under an RDos that uses the memory map hardware. The RDos version does not include MCA drivers.

12. Revision History

April 1976

↑L

For Xerox Internal Use Only -- November 17, 1977

Alto Pup FTP

October 30, 1977

17

First release.

May 1976

/Q switch added to CONNECT. Connection requests to the User FTP and Telnet can be aborted. Login prompt changed. 1 minute Timeout added when waiting to finish after a command line error. User FTP automatically recovers from more "No" responses from the remote server.

June 1976

Dos version released. DIRECTORY and LIST, commands added. Update (/U) option added. File creation dates added. 5 minute no-activity timeout added to FTP Server. FTP version, time-of-day, and machine address added in top window. "Ding" now flashes only the affected window instead of the whole display.

August 1976

RDos version released. Same as June release for Dos and Alto.

October 1976

DUMP and LOAD commands added to user FTP. KILL command added. Free disk page count added to the title line. Verify (/V) switch added to the RETRIEVE command.

November 1976

Bug fixes to the October release.

May 1977

This version was only released to friends. KILL command removed and turned into a server option. DEBUG command moved into new USER and SERVER commands. Trident disk option (/T) added. User LIST command improved and Server LIST response implemented. Password checking by the FTP server implemented. Telnet window enlarged at the expense of possibly losing information from the top of the window if the lines are very full. DELETE, RENAME, and DEVICE commands implemented. Much internal reorganization so that the protocol modules could be used in IFS and released as a package.

July 1977

Global switches changed. <Shift-Swat> should work more reliably now. User LIST command further improved. Keyboard command interpreter is much more robust and consistant. Command line STORE and DUMP go much faster since they look up files using MDI. FTP/Tx opens Trident unit 'x'. LOGIN command added to command line interpreter.

October 1977

Special microcode added to speed up execution.

↑L

. For Xerox Internal Use Only -- November 17, 1977

Alto Pup FTP

October 30, 1977

1

Alto Pup File Transfer Program

FTP is a Pup-based File Transfer Program for moving files to and from an Alto file system. The program comes in 3 parts:

- 1) An FTP Server, which listens for file transfer requests from other hosts,
- 2) An FTP User, which initiates file transfers under control of either the keyboard or the command line, and
- 3) A User Telnet for logging into a remote host using the Pup Telnet protocol.

1. Concepts and Terminology

Transferring a file from one machine (or "host") to another over a network requires the active cooperation of programs on both machines. In a typical scenario for file transfer, a human user (or a program acting on his behalf) invokes a program called an "FTP User" and directs it to establish contact with an "FTP Server" program on another machine. Once contact has been established, the FTP User initiates requests and supplies parameters for the actual transfer of files, which the User and Server proceed to carry out cooperatively. The FTP User and FTP Server roles differ in that the FTP User interacts with the human user (usually through some sort of keyboard interpreter) and takes the initiative in user/server interactions, whereas the FTP Server plays a comparatively passive role.

The question of which machine is the FTP User and which is the FTP Server is completely independent of the direction of file transfer. The two basic file transfer operations are called "Retrieve" and "Store"; the Retrieve operation causes a file to move from Server to User, whereas Store causes a file to move from User to Server.

The Alto FTP subsystem contains both an FTP User and an FTP Server, running as independent processes. Therefore, to transfer files between a pair of Altos, one should start up the FTP subsystem on both machines, then issue commands to the FTP User process on one machine directing it to establish contact with the FTP Server process in the other machine. Subsequent file transfers are controlled entirely from the FTP User end, with no human intervention required at the Server machine.

Transferring files to or from a Maxc system or an IFS involves establishing contact with FTP Server processes that run all the time on those machines. Hence, one may simply invoke the Alto FTP subsystem and direct its FTP User process to connect to the machine.

In the descriptions that follow, the terms "local" and "remote" are relative to the machine on which the FTP User program is active. That is, we speak of typing commands to our "local" FTP User program and directing it to establish contact with an FTP Server on some "remote"
↑L

For Xerox Internal Use Only -- November 17, 1977

Alto Pup FTP

October 30, 1977

2

machine. A Retrieve command then copies a file from the "remote" file system to the "local" file system, whereas a Store command copies a file from the "local" file system to the "remote" file system. Furthermore, we refer to "local" and "remote" filenames. These must conform to the conventions used by the "local" and "remote" host computers, which may be dissimilar (for example, Alto versus Maxc). The Alto FTP knows nothing about Maxc filename conventions or vice versa.

The Alto FTP subsystem also includes a third process, called a "User Telnet", which simulates a terminal in a manner exactly analogous to the Chat subsystem (though lacking some of its finer features). By this means, you may log in to a file system machine to perform operations not directly available via the basic file transfer mechanisms. If you log into Maxc, it is even possible to run "PupFTP", the Maxc FTP User program, and direct it to establish contact with the FTP Server in your own Alto. (Nova users will recognize this as analogous to the old Minx/NewMCA arrangement for file transfer.) You should probably not try this unless you really understand what you are doing, however, since the terms "local" and "remote" are relative to Maxc rather than to your Alto (since the FTP User program is running on Maxc in this case), which can be confusing.

2. Calling the FTP Subsystem

A number of options are available when running FTP. The program decides which parts of itself to enable and where user commands will come from by inspecting the command line. The general form of the command line to invoke FTP looks like:

FTP[/<Global-switches>] [<Host-name> [<Command-list>]]

The square brackets denote portions of the command line that are optional and may be omitted.

Global switches, explained below, select some global program options such as using the Trident disk instead of the Diablo. The first token after the <global-switches>, if present, is assumed to be a <host-name> (a discussion of which appears later in the description of the "Open" command). The User FTP will attempt to connect to the FTP Server on that host. After connecting to the server, if a <command-list> is present, an interpreter is started which feeds these commands to the User FTP. When the command list is exhausted, FTP returns to the Alto Executive. If no command list is present, an interactive keyboard command interpreter is started.

Each global switch has a default value which is used if the switch is not explicitly set. To set a switch to 'false' proceed it with a 'minus' sign (thus FTP/-S means 'no Server'), to set a switch to 'true' just mention the switch.

Switch	Default	Function
--------	---------	----------

/S true [Server] starts the FTP Server. The Server is not
 started if the User is enabled and is being
 controlled from the command line.

↑L

For Xerox Internal Use Only -- November 17, 1977

Alto Pup FTP

October 30, 1977

3

- | | | |
|----|-------|---|
| /U | true | [User] starts the FTP User. As explained above, the interactive command interpreter or the command line interpreter will be started depending on the contents of the command line. |
| /C | true | [Chat] starts the Telnet. The Telnet is not started if the User is enabled and is being controlled from the command line, or if the system disk is TPO. |
| /T | false | [Trident] sets the system disk to be a Trident drive. The default is 0, but can be changed by following the /T with a unit number between 0 and 7 (thus FTP/T5 means use Trident unit 5). User and Server commands apply to files on this disk but command line input is still taken from Com.cm on the Diablo drive. |
| /R | true | [Ram] allows Ftp to use some microcode which speeds things up slightly. If your Alto has no ram, this switch is ignored. |
| /B | false | [Boot] creates 'FTP.Boot'. When this file is booted, it will read the disk descriptor, open the system log file, and enable the interactive command interpreter. |
| /D | false | [Debug] starts FTP in debug mode. |
| /A | false | [AllocatorDebug] enables storage allocator checking. |

The rest of the global switches are explained below under 'Server Options'.

3. The FTP Display

The top inch or so of the display contains a title line and an error window. The title line displays the release date of that version of FTP, the current date and time, the machine's internetwork address, and the number of free pages on the disk. The error window displays certain error messages if they arrive from the network (errors are discussed in more detail below). A window is created below the title line for each part of FTP which is enabled during a session (server, user, and telnet).

If the FTP Server is enabled, it opens a window and identifies itself. If a User FTP subsequently connects to this Server, the User's network address will be displayed. The Server will log the commands it carries out on behalf of the remote User in this window. The Server is not enabled when FTP is being controlled from the command line.

The FTP User opens the next window down and identifies itself. The command herald is an asterisk.

The User Telnet opens the bottommost window, identifies itself, and waits for a host name to be entered. The Telnet is not enabled when FTP is being controlled from the command line.

↑L

For Xerox Internal Use Only -- November 17, 1977

Alto Pup FTP

October 30, 1977

4

3.1. Directing Keyboard input to the User and Telnet Windows

The bottom two unmarked keys control which window gets characters from the keyboard. Hitting the unmarked key to the right of 'right-shift' (also known as the 'Swat key') directs keyboard input to the Telnet window. Hitting the unmarked key to the right of the 'return' key (also known as the 'Chat key') directs keyboard input to the FTP User window. The window which currently owns the keyboard will blink a cursor at the next character position if it is waiting for type-in.

4. Keyboard Command Syntax

FTP's interactive command interpreter presents a user interface very similar to that of the Alto Executive. Its command structure is also very similar to that of the Maxc Pup FTP program (PupFTP), and the Maxc ArpaNet FTP program (FTP). The standard editing characters, command recognition features, and help facility (via "?") are available.

4.1. Keyboard Commands

OPEN <host name>

Opens a connection to the FTP Server in the specified host. FTP permits only one user connection at a time. In most cases the word "OPEN" may be omitted: i.e., a well formed <host name> is a legal command and implies a request to "OPEN" a connection. FTP will try for one minute to connect to the specified host. If you made a mistake typing the host name and wish to abort the connection attempt, hit the middle unmarked key (to the right of <return>).

Ordinarily, the host name can be a string, e.g., "Maxc". Most Altos and Novas have names which are registered in Name Lookup Servers. So long as a name lookup server is available, FTP is able to obtain the information necessary to translate a known host name to an inter-network address.

If the host name of the server machine is not known, you may specify an inter-network address in place of the host name. The general form of an inter-network address is:

<network> # <host> # <socket>

where each of the three fields is an octal number. The <network> number designates the network to which the Server host is connected (which may be different from the one to which the User host is connected); this (along with the "#" that follows it) may be omitted if the Server and User are known to be connected to the same network. The <host> number designates the Server host's address on <network>. The <socket> number designates the actual Server process on that host; ordinarily it should be omitted, since the default is the regular FTP server socket. Hence, to connect to the FTP server running in Alto host number 123 on the directly-connected Ethernet, you should say "OPEN 123#" (the trailing "#" is required).

↑L

For Xerox Internal Use Only -- November 17, 1977

Alto Pup FTP

October 30, 1977

5

CLOSE

Closes the currently open User FTP connection.

LOGIN <user name> <password>

Supplies any login parameters required by the remote server before it will permit file transfers. FTP will use the user name and password in the Operating System, if they are there. Logging into FTP will set the user name and password in the OS (in the same manner as the Alto Executive's "Login" command).

When you issue the "Login" command, FTP will first display the existing user name known to the OS. If you now type a space, FTP will prompt you for a password, whereas if you want to provide a different user name, you should first type that name (which will replace the previous one) followed by a space. The command may be terminated by carriage return after entering the user name to omit entering the password.

The parameters are not immediately checked for legality, but rather are sent to the server for checking when the next file transfer command is issued. If a command is refused by the server because the name or password is incorrect, FTP will prompt you as if you had issued the LOGIN command and then retry the command. Hitting delete in this context will abort the command.

A user name and password must be supplied when transferring files to and from a Maxc system or an IFS. The Alto FTP Server requires a user-password to be supplied if the server machine's disk is password-protected and if the password in the server machine's OS does not match the password on the disk. Thus if the OS was booted and FTP invoked because a Request-for-Connection was received (which bypasses password checking), FTP will refuse access to files unless a password is supplied. However if the OS was booted normally, FTP assumes that the disk owner (who knew the password) will control access by using the server option switches. The user-name is ignored.

CONNECT <directory name> <password>

Requests the FTP server to "connect" you (in the Tenex sense) to the specified directory on the remote system. The password may be omitted by typing carriage return after the directory name. As with LOGIN, these parameters are not verified until the next transfer command is issued. At present, the "Connect" command is meaningful only when transferring files to or from a Maxc system or an IFS; the Alto FTP server currently ignores connect requests. If the "multiple directory" feature of the Alto Operating System ever comes into widespread use, this may be changed.

RETRIEVE <remote filename>

Initiates transfer of the specified remote file to the local host. The syntax of <remote filename> must conform to the remote host's file system name conventions.

If the server can find the file, FTP will then print out the complete filename followed by the message "to local file <local filename> [Old|New file]", where the local filename is generally the same as the remote filename without directory or version. At this point you may make one of three choices:

For Xerox Internal Use Only -- November 17, 1977

Alto Pup FTP

October 30, 1977

6

1. Type Carriage Return to cause the data to be transferred to <local filename>.
2. Type Delete to indicate that the file is not to be transferred.
3. Type any desired local filename followed by Return. The previous local filename will disappear, the new filename will replace it, and FTP will tell you whether a file exists with that name. This filename must conform to local conventions. You now have the same three choices.

If the remote filename designates multiple files (the remote host permits "*" or some equivalent in file names), each file will be transferred separately and FTP will ask you to make one of the above three choices for each file. At present, only Maxc and IFS support this capability. That is, you may supply "*"s in the remote filename when retrieving files from Tenex or an IFS, but not when retrieving files from another Alto.

STORE <local filename>

Initiates transfer of the specified local file to the remote host. Alto file name conventions apply to the <local filename>; "*" expansion is not supported. FTP will suggest a remote filename to which you should respond in a manner similar to that described under RETRIEVE except that if you supply a different filename, it must conform to the remote file system's conventions. The default remote filename is one with the same 'name body' (name and extension) as the local file; the remote server defaults other fields as necessary. If the remote host is a Maxc system or an IFS, then the directory is that most recently supplied in LOGIN or CONNECT commands and the version is the next higher.

DUMP <remote filename>

Bundles together a group of files from the local file system into a 'dump-format' file (see the Alto Executive documentation for the dump-file format and more on dump-files in general) and stores the result as <remote filename>. FTP will ask you for the names of local files to include in the dump-file. Terminate the dump by typing just <return> when FTP asks for another filename. By convention, files in dump-format have extension '.dm'.

LOAD <remote filename>

Performs the inverse operation of DUMP, unbundling a dump-format file from the remote file system and storing the constituent files in the local file system. For each file in the dump-file, FTP will suggest a local file name and tell you whether a file by that name exists on your disk. You should respond in the manner described under RETRIEVE.

LIST <remote file designator>

Lists all files in the remote file system which correspond to <remote file designator>. The remote file designator must conform to file naming conventions on the remote host, and may designate multiple files if "*" expansion or some equivalent is supported there. If the <remote filename> is terminated by a comma, FTP prints a prompt of "##" at the left margin and prepares to accept one or more subcommands. These subcommands request printout of additional information about each file. To terminate subcommand

↑L

For Xerox Internal Use Only -- November 17, 1977

Alto Pup FTP

October 30, 1977

7

input, type a <return> in response to the subcommand prompt. The subcommands are:

Type	Print file type and byte size.
Length	Print length of file in bytes.
Creation	Print date of creation.
Write	Print date of last write.
Read	Print date of last read.
Times	Print times as well as dates.
Author	Print author (creator) of file.
Verbose	Same as Type+Write+Read+Author.
Everything	Print all information about the file.

This information is only as reliable as the Server that provided it, and not all Servers provide all of these file properties. Altos derive much of this information from hints, so do not be alarmed if it is sometimes wrong.

DELETE <remote filename>

Deletes <remote filename> from the remote filesystem. The syntax of the remote filename must conform to the remote host's file system name conventions. After determining that the remote file exists, FTP asks you to confirm your intention to delete it. If the remote filename designates multiple files (the remote host permits "*" or some equivalent in file names), FTP asks you to confirm the deletion of each file.

RENAME <old filename> <new filename>

Renames <old filename> in the remote filesystem to be <new filename>. The syntax of the two filenames must conform to the remote host's file system name conventions, and each filename must specify exactly one file.

QUIT

Returns control to the Alto Executive, closing all open connections.

TYPE <data type>

Forces the data to be interpreted according to the specified <data type>, which may be TEXT or BINARY. Initially the type is UNSPECIFIED, meaning that the source process should, if possible, decide on the appropriate type based on local information.

BYTE-SIZE <decimal number>

Applicable only to files of type Binary, BYTE-SIZE specifies the logical byte size of the data to be transferred. The default is 8.

EOL <convention>

Applicable only to files of type Text, EOL specifies the End-of-Line Convention to be used for transferring text files. The values for <convention> are CR, CRLF, and TRANSPARENT. The default is CR.

DEVICE <device>

Causes <device> to be used as the default device in data transfer commands (essentially it causes <device> to be attached to all remote filenames that do not explicitly mention one). The punctuation separating <device> from the other components of a

↑L

For Xerox Internal Use Only -- November 17, 1977

Alto Pup FTP

October 30, 1977

8

remote filename should not be included. For example you might specify "Device DSK" to Tenex, not "Device DSK:"

DIRECTORY <directory name>

Causes <directory name> to be used as the default remote directory in data transfer commands (essentially it causes <directory-name> to be attached to all remote filenames that do not explicitly mention a directory). Specifying a default directory in no way modifies your access privileges, whereas CONNECTing gives you 'owner access' (and usually requires a password). Explicitly mentioning a directory in a file name overrides the default directory, which overrides the connected directory, which overrides the login directory. Punctuation separating <directory name> from other parts of a remote filename should not be included. For example you might type "Directory Alto" not "Directory <Alto>".

USER

Allows you to toggle switches which control operation of the FTP User. There is currently only one: DEBUG, which controls display of protocol interactions. Warning: this printout (and the corresponding one in the SERVER command below) sometimes includes passwords.

SERVER

Allows you to toggle switches which control operation of the FTP Server. The switches are PROTECTED, OVERWRITE, KILL, and DEBUG. The first three are explained below under 'Server Options'.

TELNET

Allows you to toggle switches which control operation of the Telnet. There is currently only one: CLOSE, which closes the Telnet connection if one is open, and clears the Telnet window.

5. Command Line Syntax

The User FTP can also be controlled from the command line. As explained above, the first token after the subsystem name and server switches must be a legal host name; if the User FTP can't connect to the FTP Server on that host it will abort and return control to the Alto Executive. If a command list follows the host name, the command line interpreter is invoked instead of the interactive keyboard interpreter. This permits the full capabilities of the Alto Executive (filename recognition, "*" expansion, command files, etc.) to be used in constructing commands for FTP.

Each command is of the form:

<Keyword>/<SwitchList> <arg> ... <arg>

To get a special character (any one of "*#';") past the Alto Executive, it must be preceded by a single quote. To get a "/" into an FTP argument, the "/" must be proceeded by two single quotes (the second one tells FTP to treat the "/" as an ordinary character in the argument, and the first one gets the second one past the Alto-Executive).

↑L

For Xerox Internal Use Only -- November 17, 1977

Alto Pup FTP

October 30, 1977

9

Unambiguous abbreviations of command keywords (which in most cases amount to the first letter) are legal. However, when constructing command files, you should always spell commands in full, since the uniqueness of abbreviations in the present version of FTP is not guaranteed in future versions.

A command is distinguished from arguments to the previous command by having a switch on it, so every command must have at least one switch. The switch "/C" has no special meaning and should be used on commands where no other switches are needed or desired.

When the command line has been completely executed, FTP returns control to the Alto Executive. If the Debug command has been given or any errors have occurred, FTP will pause before finishing. This prevents error messages from disappearing before you have a chance to see them. In this situation, FTP will terminate after one minute has elapsed or any key has been struck on the keyboard.

5.1. Command Line Commands

LOGIN/C <user name> <password>

See description in "Keyboard commands". The <password> may be omitted.

LOGIN/Q <user name>

Causes FTP to prompt the user for the password. This form of LOGIN should be used in command files since including passwords in command files is bad practice.

CONNECT/C <directory name> <password>

See description in "Keyboard commands". The <password> may be omitted.

CONNECT/Q <directory name>

Causes FTP to prompt the user for the password needed to connect to the specified <directory name>. This form of CONNECT should be used in command files since including passwords in command files is bad practice.

RETRIEVE/C <remote filename> ... <remote filename>

Retrieves each <remote filename> and writes it in the local file system, constructing a local file name from the name body of the actual remote file name as received from the Server. FTP will overwrite an existing file unless the /N (No overwrite) switch is appended to the RETRIEVE command keyword. If the remote host allows "*" (or some equivalent) in a filename, a single remote filename may result in the retrieval of several files. (Note that you must quote the "*" to get it past the Alto Executive's command scanner.) As mentioned previously, this capability is implemented only by Maxc and IFS FTP Servers at present.

RETRIEVE/S <remote filename> <local filename>

Retrieves <remote filename> and names it <local filename> in the local file system. This version of RETRIEVE must have exactly two arguments. FTP will overwrite an existing file unless the /N (No overwrite) switch is also appended to the RETRIEVE command keyword. The remote filename should not cause the server to send multiple files.

↑L

For Xerox Internal Use Only -- November 17, 1977

Alto Pup FTP

October 30, 1977

10

RETRIEVE/U <remote filename> ... <remote filename>

Retrieves <remote filename> if its creation date is later than the creation date of the local file. A file will not be retrieved unless a local file with name and extension equal to the name and extension of the remote filename exists, or if the FTP server does not send a CREATION-DATE property. This option can be combined with RETRIEVE/S to rename the file as it is transferred.

RETRIEVE/V

Requests confirmation from the keyboard before writing a local file. This option is useful in combination with the Update option since creation date is not a fool-proof criterion for updating a file.

STORE/C <local filename> ... <local filename>

Stores each <local filename> on the remote host, constructing a remote filename from the name body of the local filename. A local filename may contain "*", since it will be expanded by the Alto Executive into the actual list of filenames before the FTP subsystem is invoked.

STORE/S <local filename> <remote filename>

Stores <local filename> on the remote host as <remote filename>. The remote filename must conform to the file name conventions of the remote host. This version of store must have exactly two arguments.

DUMP/C <remote filename> <local filename>...<local filename>

See the description in "Keyboard Commands".

LOAD/C <remote filename>

See the description in "Keyboard Commands". If the /V switch is appended to the LOAD command keyword, FTP will request confirmation before writing each file. Type <return> to write the file, to skip it. FTP will overwrite an existing file unless the /N (No overwrite) switch is appended to the LOAD command keyword.

DELETE/C <remote filename>

See the description in "Keyboard Commands". If the /V switch is appended to the DELETE command keyword, FTP will request confirmation before deleting each file. Type <return> to delete the file, and (oops!) if you don't want to delete it.

RENAME/C <old filename> <new filename>

See the description in "Keyboard Commands".

TYPE/C <data type>

See the description in "Keyboard Commands".

BYTE-SIZE/C <decimal number>

See the description in "Keyboard Commands".

EOL/C <convention>

See the description in "Keyboard Commands".

DEVICE/C

See the description in "Keyboard Commands".

DIRECTORY/C <default directory>
See description in "Keyboard commands".
↑L

For Xerox Internal Use Only -- November 17, 1977

Alto Pup FTP

October 30, 1977

11

DEBUG/C

See the description of the DEBUG subcommand under the USER command in "Keyboard Commands".

6. Telnet

FTP provides a simple User Telnet as a convenience for logging into a remote host (e.g., Maxc) to poke around without having to leave the FTP subsystem and start Chat. It lacks most of the creature comforts Chat provides, such as automatic attaching to detached jobs, automatic logging in, etc. The Telnet is not enabled when the User FTP is being controlled from the command line. When the Telnet does not have an open connection, it waits for you to type a host name with the syntax explained above for the OPEN command, and then attempts to connect to the specified host. If you wish to abort the connection attempt, hit the bottom unmarked key (opposite right-shift). You can get a larger Telnet window by not starting a server (type FTP/-S to the Executive).

7. File Property Defaulting

Without explicit information from the file system, it is often difficult to determine whether a file is Binary or Text, if Binary, what its byte-size is, and if Text, what End-Of-Line convention is used. The User and Server FTPs use some simple heuristics to determine the correct manner in which to transfer a file. The heuristics generally do the right thing in the face of incomplete information, and can be overridden by explicit commands from a human user who knows better.

The FTP protocol specifies a standard representation for a file while in transit over a network. If the file is of type Binary, each logical byte is packed right-justified in an integral number of 8-bit bytes. The byte-size is sent as a property along with the file. If the file is of type Text, each character is sent right-justified in an 8-bit byte. An EOL convention may be sent as a file property. The default is that <return> marks the end of a line.

7.1. File Types

FTP determines the type of a local file by reading it and looking for bytes with the high-order bit on. If any byte in the file has a high-order bit on, the file is assumed to be Type Binary, otherwise it is assumed to be Type Text.

FTP will warn you, but allow you to send what it thinks to be a text file as type Binary, since no information is lost. It will refuse to send a binary file as type text.

Don't specify a Type unless you know what you are doing. The heuristic will not lose information.

↑L

For Xerox Internal Use Only -- November 17, 1977

Alto Pup FTP

October 30, 1977

12

7.2. Byte-Size

If a file is type Binary, the byte-size is assumed to be 8 unless otherwise specified. The FTP User and Server will both accept binary files of any byte-size and write them as 8 bit bytes on the disk. No transformation is done on the data as it is written to the disk: it is stored in network default format. Since there is no place in the Alto file system to save the byte-size property, it is lost.

Similarly, requests for Binary files will be honored with any byte size, and whatever is on the disk will be sent to the net without transformation. Since Alto files have no byte size information, the byte-size property will be defaulted to 8 unless otherwise specified (by the BYTE command), in which case whatever was otherwise specified will be sent as the byte size.

Don't specify a Byte-size unless you know what you are doing. Alto-Alto transfers can't go wrong. Alto-Maxc transfers with weird byte-sizes will not work unless the byte-size specified in the Alto to Maxc direction is the same as the byte-size in which the file was stored on the Alto. If it isn't, the Alto will not give any error indication, but the result will be garbage.

7.3. End-of-Line Conventions

FTPs are expected to be able to convert text files between the local file system End-Of-Line (EOL) convention and the network convention. Conveniently enough, the Alto file system's internal representation of a text file is the same as the network standard (a bare <return> marks the end of a line). The Alto FTP does not do any transformations on text files. It will refuse to store a text file coming in from the net whose EOL convention is CRLF.

As an escape to bypass conversion and checking, EOL convention 'transparent' tells both ends NOT to convert to network standard, but rather send a file 'as is'. This is included for Lisp source files which use EOL convention CRLF and contain internal character pointers that are messed up by removing line feed characters.

Don't specify an EOL convention unless you know what you are doing. If your text file is a Lisp source file, specify EOL convention 'Transparent'.

8. Abort and Error messages

Error and Abort packets are displayed in a window above the title line. Abort packets are fatal; Error packets are not necessarily so. The most common Abort message is "Timeout. Good bye", generated when a server process has not received any commands for a long time (typically 5 minutes).

The most common Error message is "Port IQ overflow" indicating a momentary shortage of input buffers at the remote host. Receiving an Error Pup does not imply that the file in transit has been damaged.

↑L

For Xerox Internal Use Only -- November 17, 1977

Alto Pup FTP

October 30, 1977

13

Loss of or damage to a file will be indicated by an explicit message in the User FTP window. The next iteration of the Pup protocols will probably rename 'Error Pups' to be 'Information Pups'.

9. Server Options

Server options are controlled by switches on the subsystem name and subcommands of the SERVER keyboard command. There are currently four options:

switch Default Function

none	If no server option is specified, retrieve requests (disk to net) are allowed. Store requests (net to disk) are allowed unless the store would overwrite an already existing file.
/P	false [Protected] Retrieve requests are allowed. No stores are allowed.
/O	false [Overwrite] Retrieve requests are allowed. Store requests can overwrite files.
/K	false [Kill] FTP will return to the Alto Exec when the server connection is closed. A simple form of remote job entry can be performed if the user FTP stores into Rem.cm (Com.cm on Novas).

10. CLI Examples

Here are some examples of Command lines.

To transfer files Ftp.run and Ftp.syms from the Alto called "Michelson" to the Alto called "Morley", one might start up FTP on Michelson (to act as an FTP Server), then walk over to Morley and type:

FTP Michelson Retrieve/c Ftp.run Ftp.syms

Alternatively, one could start an FTP server on Morley (invoking it by "FTP/O" to permit files to be overwritten on Morley's disk), then issue the following command to Michelson:

FTP Morley Store/c Ftp.run Ftp.syms

The latter approach is recommended for transferring large groups of files such as "*.run" (since expansion of the "*" will be performed by the Alto Executive).

To retrieve User.cm from the FTP server running on Alto serial number 123 (name unknown, but it is on the local Ethernet):

FTP 123'# Retrieve User.cm

For Xerox Internal Use Only -- November 17, 1977

Alto Pup FTP

October 30, 1977

14

Note that the "#" must be preceded by a single quote when included in a command line, since otherwise the Alto Executive does funny things with it. (Quotes are not necessary when typing to FTP's interactive keyboard interpreter).

To start FTP, have the FTP User connect to Maxc, and then accept further commands from the keyboard:

FTP Maxc

To retrieve <System>Pup-Network.txt from Maxc and store it on the Alto as PupDirectory.bravo, and store PupRTP.bcpl, Pup1b.bcpl, and PupBSPStreams.bcpl on <DRB> with their names unchanged:

```
FTP Maxc Connect/c drb mypassword Retrieve/s <System>Pup-
Network.txt PupDirectory.bravo Store/c PupRTP.bcpl Pup1b.bcpl
PupBSPStreams.bcpl
```

To retrieve the latest copy of all .RUN files from the <alto> directory, overwriting copies on the Alto disk (The single quote is necessary to prevent the Alto Executive from expanding the "*"):

```
FTP Maxc Ret/c <alto>*.run
```

To update the Alto disk with new copies of all <alto> files whose names are contained in file UpdateFiles.cm, requesting confirmation before each retrieval:

```
FTP Maxc Dir/c Alto Ret/u/v @UpdateFiles.cm@
```

To store all files with extension .BCPL from the local Alto disk to your login directory on Maxc (the Alto Executive will expand "*.bcpl" before invoking FTP):

```
FTP Maxc St/c *.bcpl
```

To retrieve <System>Host-name/descriptor-file.txt;43 (two single quotes are necessary to get the "/" past the Alto Executive and the FTP command scanner, and one quote is necessary to get the ";" past the Alto Executive):

```
FTP Maxc Ret/c <System>Host-name''/descriptor-file.txt';43
```

To send Prog.f4, Data.f4, and Command.f4 to Fortran-Machine and then cause the FTP server on Fortran-Machine to quit (presumably to execute Prog.f4 on Data.f4 according to the commands in Command.f4):

```
FTP Fortran-Machine Store/c Prog.f4 Data.f4 Store/s
Command.f4 Rem.cm
```

FTP on Fortan-Machine must be started with the /K server option switch, and Command.f4 should re-invoke FTP as its last act so that the results can be retrieved.

To release a new version of FTP, I incant:

```
@ReleaseAltoFtp.cm@
```

which the Alto Executive expands into:

↑L

For Xerox Internal Use Only -- November 17, 1977

Alto Pup FTP

October 30, 1977

15

FTP Maxc Connect/q Alto Store/c Ftp.run Ftp.sysms Connect/q
AltoSource Dump/c Ftp.dm @ftp.cm@

and then into:

FTP Maxc Connect/q Alto Store/c Ftp.run Ftp.sysms Connect/q
AltoSource Dump/c Ftp.dm @FtpSubsys.cm@ @FtpPackage.cm@
Ftp.cm

and finally into:

FTP.run Maxc Connect/q Alto Store/c Ftp.run Ftp.sysms
Connect/q AltoSource Dump/c Ftp.dm Ftp.bcpl FtpNv.bcpl
FtpInit.bcpl FtpInit1.bcpl FtpNvInit.bcpl FtpUserInit.bcpl
FtpKbd.decl FtpKbdInit.bcpl FtpKbd.bcpl FtpKbd1.bcpl
FtpKbd2.bcpl FtpCliInit.bcpl FtpCli.bcpl FtpCli1.bcpl
FtpCli2.bcpl FtpCliUtil.bcpl FtpMiscb.bcpl FtpMisca.asm
FtpServerInit.bcpl FtpServer.bcpl FtpTelnetInit.bcpl
FtpTelnet.bcpl FtpKeys.bcpl FtpMc.mu RamTrap.mu GetFrame.mu
PupChecksum.mu CompileFtpMc.cm FtpSubsys.cm
CompileFtpSubsys.cm CompileAltoFtp.cm LoadAltoFtp.cm
LoadHiddenFtp.cm ReleaseAltoFtp.cm CompileNovaFtp.cm
LoadDosFtp.cm LoadRDosFtp.cm FtpProt.decl FtpUserProt.bcpl
FtpUserProtFile.bcpl FtpUserProtMail.bcpl
FtpServProtFile.bcpl FtpServProtMail.bcpl FtpPLInit(bcpl
FtpPLInitProt(bcpl FtpPLInit1(bcpl FtpUtilInit(bcpl
FtpUtilB.bcpl FtpUtilA.asm FtpUtilXfer(bcpl FtpUtilDmpLd(bcpl
FtpOEPInit(bcpl CompileFtpPackage.cm DumpFtpPackage.cm
FtpPackage.cm Ftp.cm

To load Ftp.dm from <AltoSource>, expanding it out into its constituent files:

FTP Maxc Load/c <AltoSource>Ftp.dm

To cause Memo.ears to be spooled for printing on Ears by the Maxc printing system:

FTP Maxc Store/s Memo.ears LPT:

This also works for Press files and unformatted text files if you know what you are doing. It does not do the right thing for Bravo-format files.

To use FTP as a stop-gap IFS:

FTP/T-U0

This starts only a server with overwriting of existing files permitted. When using the trident, there isn't enough space to start both a User and a Server.

11. Nova FTP

FTP is also available running under Dos Rev 4 and RDos Rev 3. Since
the Nova versions are nearly identical to the Alto version (the same
↑L

For Xerox Internal Use Only -- November 17, 1977

Alto Pup FTP

October 30, 1977

16

source files except for initialization), only the differences are listed here.

- 1) Ignore all references to display windows. All printout goes to device #11, whatever that is.
- 2) Ignore all references to 'unmarked keys' such as for aborting connection attempts and directing keyboard input to various windows.
- 3) Lack of memory and lack of a windowing display made including a Telnet impractical on the Nova.
- 4) The syntax of the command line is limited to that acceptable to the Nova operating system. Warning: the command line examples given above may not all work on a Nova.
- 5) The Nova OS does not maintain a username or password, so all interactions with a Maxc system or an IFS will require the user to supply them.
- 6) File creation dates are not supported, so there is no Update option to RETRIEVE, and the LIST command does not show dates.

11.1. FTP releases

The Nova FTP subsystem consists of a save-file, FTP.SV, and an overlay-file, FTP.BB. You must get BOTH files when a new version of FTP is released. If you rename FTP.SV you must rename FTP.BB to have the same name (for instance if you rename FTP.SV to be OLDFTP.SV you must also rename FTP.BB to be OLDFTP.BB). New releases of FTP will be distributed as dump files with a consistant pair of save- and overlay-files.

11.2. Device codes

FTP assumes that Nova Ethernet interfaces have device codes 73 and 74, 63 and 64, or 53 and 54. It will use all interfaces with these codes that seem (from reading some status registers) to be Ethernets. The Dos version of FTP assumes that Nova MCA interfaces are device code 6 and 7, or 46 and 47. It will use all interfaces with these codes that seem to be MCAs.

11.3. RDos notes

FTP is big, and will not run under some RDos systems. If you have trouble, generate a smaller system and boot from it when running FTP. FTP disables parts of RDos with patches which may not work for versions other than Rev 3. It will NOT work under an RDos that uses the memory map hardware. The RDos version does not include MCA drivers.

12. Revision History

April 1976

↑L

For Xerox Internal Use Only -- November 17, 1977

Alto Pup FTP

October 30, 1977

17

First release.

May 1976

/Q switch added to CONNECT. Connection requests to the User FTP and Telnet can be aborted. Login prompt changed. 1 minute Timeout added when waiting to finish after a command line error. User FTP automatically recovers from more "No" responses from the remote server.

June 1976

Dos version released. DIRECTORY and LIST commands added. Update (/U) option added. File creation dates added. 5 minute no-activity timeout added to FTP Server. FTP version, time-of-day, and machine address added in top window. "Ding" now flashes only the affected window instead of the whole display.

August 1976

RDos version released. Same as June release for Dos and Alto.

October 1976

DUMP and LOAD commands added to user FTP. KILL command added. Free disk page count added to the title line. Verify (/V) switch added to the RETRIEVE command.

November 1976

Bug fixes to the October release.

May 1977

This version was only released to friends. KILL command removed and turned into a server option. DEBUG command moved into new USER and SERVER commands. Trident disk option (/T) added. User LIST command improved and Server LIST response implemented. Password checking by the FTP server implemented. Telnet window enlarged at the expense of possibly losing information from the top of the window if the lines are very full. DELETE, RENAME, and DEVICE commands implemented. Much internal reorganization so that the protocol modules could be used in IFS and released as a package.

July 1977

Global switches changed. <Shift-Swat> should work more reliably now. User LIST command further improved. Keyboard command interpreter is much more robust and consistant. Command line STORE and DUMP go much faster since they look up files using MDI. FTP/Tx opens Trident unit 'x'. LOGIN command added to command line interpreter.

October 1977

Special microcode added to speed up execution.
↑L