

Index

Commands are given in all caps.

ATTENTION.CHARACTER	308
BREAK	310
Camp-on option	298, 303
CASE.SHIFT.PREFIX.FOR	300, 307
CHAR.LOWER.CASE	307
CHAR.UPPER.CASE	307
CHARACTER.MODE	306
CLEAR.OUTPUT.CHARACTER=	309
CODE	310
COMMAND completion	296
Command interpreter	296
Command mode	297
CONCISE	296, 308
Connection names	298
CONNECTION.TO	297, 303
CONTROL.CHARACTER.ECHO.FOR	305
CURRENT.MODES.ARE	300, 306
DDT	310
DISCONNECT	298, 303-304
Disconnecting	298
Disconnection, spontaneous	298
DIVERT.OUTPUT.STREAM.TO.FILE	308
Diverting output	302
ECHO.MODE.IS	304
Echoes	299
Editing buffered lines	299
End of line conventions	299
Escape character	297
Escape character, initial	297
ESCAPE.CHARACTER=	309
Escaping to command mode	297
EXEC	310
FANCY.COMMAND.INTERPRET	296, 308
Finishing a session	301
FLUSH.HOST	312
FULL-DUPLEX	305
HALF-DUPLEX	305
HELP	309
HOST.NAMES	311

Input from a File	302
IO errors	298
Line at a time	299
LINE.BUFFER	306
LINEFEED.FOR.CARRIAGE.RETURN	304
LIST.CONNECTIONS	311
LOAD.SETTINGS	303
LOCAL	304
LOCAL.MODE	305
LOCK.LOWER.CASE	307
LOCK.UPPER.CASE	307
LOGGER	303
LOGOUT	310
LOWER	306
LOWER.CASE	305
Making a connection	297
Multiple connections	301
NAME.FOR.CONNECTION.IS	303
NAME.FOR.CURRENT.CONNECTION	301
Names, connection	298
NET.EXEC	304
NETSTATUS	309
Network virtual terminal	296
NO	303-309
NVT	296
Question mark, use of	296
QUIT	301, 310
QUOTE.PREFIX	300, 307
RAISE	306
REMOTE	304
Remote mode	297
REMOTE.MODE	306
RESET	310
RETRIEVE.CONNECTION.UNDER.NAME	311
RUN	309
SIGNAL.WAITING.OUTPUT	311
SOCKET.MAP	309
Special characters	300
Special sockets	298, 303
Status commands	300
STATUS.OF	304
SYNCH	311
SYNCH.CHARACTER	307
TAKE.INPUT.STREAM.FROM.FILE	308
TERMINAL.TYPE.IS	305
Terminators, command	296
Terminators, line buffer	299
Time-outs	298

January 1975

TRANSPARENT.MODE	306
TypeScript	301
TYPESCRIPT.TO.FILE	309
UNSHIFT.PREFIX	307
VERBOSE	308
WAIT	303
WAIT.FOR.ANY.ACTIVE.CONNECTION	311
WHERE.AM.I	311
WRITE.MODES.FOR.HOST . . .	311

January 1975

TIPCOPY

TIPCOPY DOCUMENTATION NOV. 1, 1974

Tipcopy will copy an ASCII text file to a TIP port.

The program is set up so that the receiving TIP is the TIP of the
controlling terminal. The port number on that TIP is defaulted
to be 2. You may change these settings. (To be explained
later).

To use the program type:

@TIPCOPY <cr> OR @TIPCOPY <space>

The program will ask for a file name. To send the file to the
TIP and port as explained above, just type a carriage-return to
confirm your file selection. To be allowed to select option such
as a specific TIP or port, confirm with a comma (,).
#

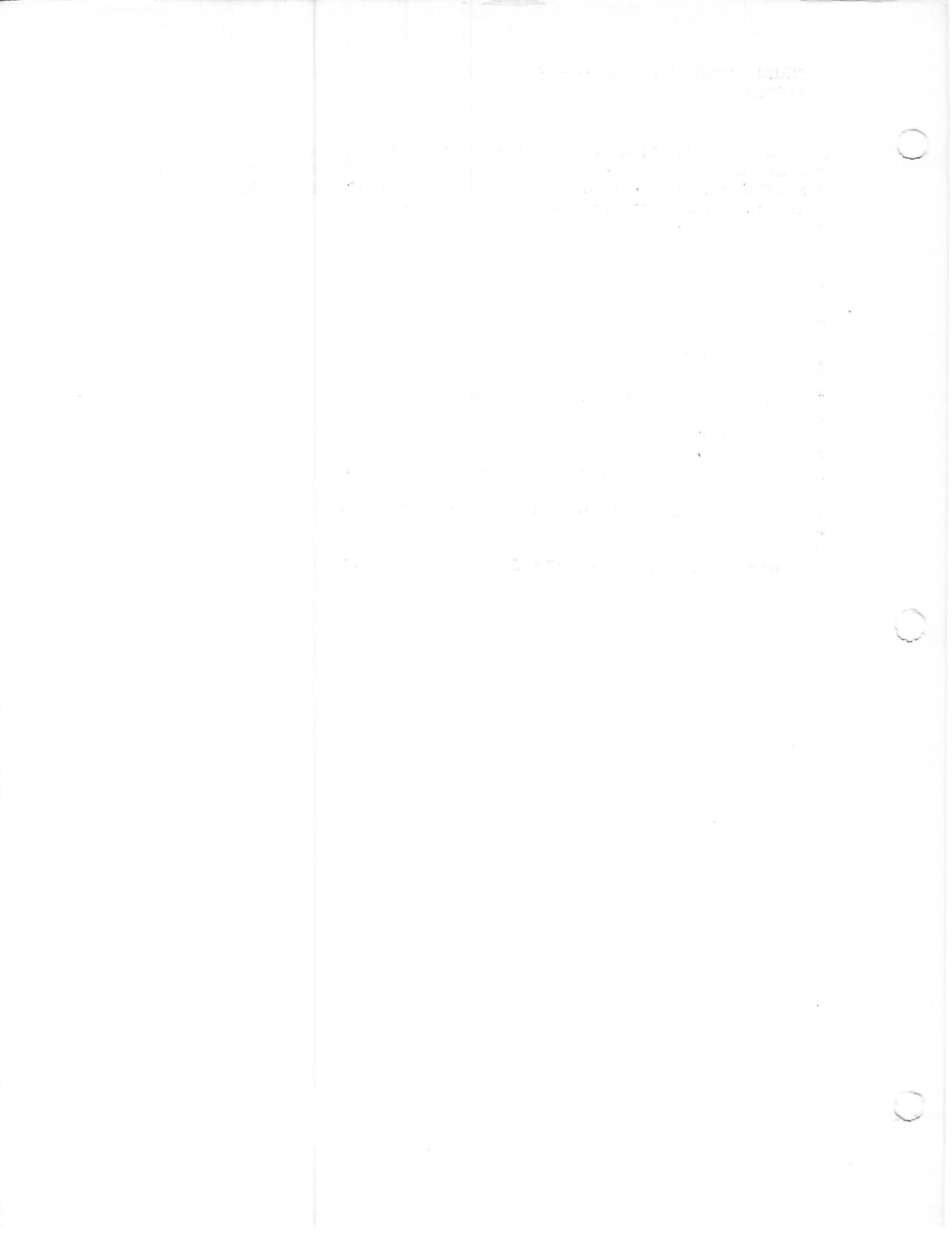
At this point TIPCOPY will allow you to type commands. Typing a
question mark (?) will print a list of the commands currently
available. A list is shown below. You may select any option you
wish, or may undo a defaulted option by typing NO followed by the
command. It is not possible to undo the destination TIP or a
device command. To change these just reuse the command. Type an
escape when you have completed your options selection.

L FOR LIST OF OPTIONS ALREADY SELECTED
T FOR TIP NAME IF OTHER THAN THE CONTROLLING TERMINAL'S TIP
F FOR FORMFEEDS OVER PERFORATION
H FOR HEADINGS
P FOR PAUSES BETWEEN PAGES
SI FOR SIMULATE FORMFEEDS OVER PERFORATIONS
SP FOR SPECIFIC PAGES
D FOR OCTAL DEVICE, IF OTHER THAN PORT 2
C FOR COLUMN 1 FORTRAN INTERPRETATION
M FOR MULTIPLE COPIES OF A FILE
X FOR SELECTABLE LINE SPACING (DEFAULT=1)
Y FOR PRINTING CONTROL CHARACTERS
Z FOR SELECTABLE NUMBER OF FORMFEEDS AT END (DEFAULT=3)
? FOR HELP, PRINTS THIS INFORMATION

TYPE ESCAPE WHEN COMPLETED OPTIONS SELECTION
USE THE "NO" COMMAND TO UNDO A SELECTION, OR TO
REMOVE THE TITLE PAGE WITH THE COMMAND "NO T".
THE DEFAULT SETTINGS INCLUDE FORMFEEDS OVER
PERFORATIONS, HEADINGS AND NOT PRINTING CONTROL CHARACTERS.
CONTROLLING TERMINALS ON THE BBN-TESTIP WILL
AUTOMATICALLY SEND FILES TO THE NCC-TIP.

January 1975

```
# From time to time a new version of TIPCOPY will be written with
# the modifications users requested. This program will be called
# NTIPCOPY. I urge you to try this program and report to MALMAN
# any problems you may have had.
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
# <IMP>PLOT.SAV
#
# This is a Tipcopy like program except:
#
#      2 commands
#
#      T      TIP command if other than TESTIP
#
#      D      DEVICE command if other than port 22[8]
#
#
# 5 bit bytes (CALCOMP STYLES) to a tip port
```



XED - Experimental Editor

? outputs this message.

H HELP

The command letter which follows the H will be explained.
H<CR> generates an editor manual. H<space> describes
the interrupt characters.

The Legal Command Letters are:

^A, ^H, DEL	Delete char	^W	Delete word	^X	Delete line	^R	Re-type line
^Q	Abort command	^V, ^^, ^\	are all escape characters				
A(ppend)	B(ackup)	C(hange)	D(elete)				
E(xit)	F(ind)	G(roup)	H(elp)				
I(nsert)	J(am)	K(ill)	L(ist)				
M(odeset)	N(otemodes)	O(utput)	P(rint)				
Q(uit)	R(ead)	S(earch)	T(ype)				
V(iew)	W(rite)	X(change)					
\$ Go to End	= Current Line	- Move Back	+ Move Forward				
, Context	^ Print Prior	/ Print Current	LF Print Next				
Format	; Comment						
% Call SNDMSG	! Run program	@ Call EXEC	" Mode dialogue				

H HELP - Help accepts a command character following the H command. A description of the command represented by that character is then printed on the terminal. H(help) followed by a carriage return additionally requests a file name. A brief reference manual is created in the file.

; COMMENT - All text input from the terminal following the ";" is ignored until the occurrence of a ^Z or a ^Q.

POSITIONING AND SEARCHING COMMANDS

Most commands in the editor operate on the current line of text within the text buffer. Lines in the text buffer are numbered consecutively starting at 1. The commands in this section describe ways to examine the contents of the text buffer and to change to current line.

\$ DOLLARS - \$ makes the last line in the text buffer the NEW current line. It also prints the number of lines in the text buffer.

NUMBERS - Numbers which are input as commands cause the current line to become the line with the number input. If a digit is

incorrectly input, it may be canceled with a ^A, ^H, or DEL. The entire number may be cancelled with ^Q.

The commands ,(context), +, -, K(ill), T(ype), and M(odeset) all expect to receive numbers following. These numbers may be similarly edited with ^A, ^H, and DEL. ^Q cancels the entire command.

: COLON - K(ill) and T(ype) expect to receive a count of the number of lines on which they should operate. However, if the number is preceded by a :, they operate on the current line through the line number input instead of n lines starting at the current line.

Example: 12T:15. is the same as 12T4.

. DOT - . is a convenient character to terminate numbers, and is a null operator to the editor. Space and carriage return would also work as well.

= EQUALS - = prints the line number of the current line.

^ UP ARROW - ^ moves the current line to the line preceding the current line in the text buffer and prints the NEW current line. This is the same as -/.

LF LINEFEED - LF advances the current line 1 line forward in the text buffer and prints the NEW current line. This is the same as +/.

+ PLUS - Plus accepts a number following it, and advances the current line that number of lines forward. If no number follows, the current line is advanced 1 line forward. If this command is input by accident, typing ^Q will cancel it.

- MINUS - Minus accepts a number following it, and moves the current line that number of lines backwards. If no number follows, the current line is moved 1 line backwards. If this command is input by accident, typing ^Q will cancel it.

F FIND - Find accepts a string from the terminal. It then locates the first occurrence of the string in the text starting with the current line throughout the entire text buffer. The NEW current line is the line found, or the current line if no match was found. If the search string was last found at the current line, the search will start at the NEXT line (so repeated F(ind)s will locate successive occurrences of the search string in the text). If just a <CR> is typed to the prompt for a string, Find uses the string from the last F(ind) or S(earch). If the search string is a single null character (specified by control-V control-@ [^V^@] or control-^ @

[^^@]), then XED will search for a null line (no text, only <CR><LF>). If this command is input by accident, typing ^Q will cancel it. While this command is executing, typing ^Q interrupts it and returns to XED command mode. This command normally looks for the string any place in the line, independent of upper and lower case differences.

^E If ^E is specified, the match will be exact only.

^B If ^B is specified, the match will be only at beginning of lines.

^S If ^S is specified, XED will enter Change on the line(s) containing the string, positioning the Change cursor at the found string.

During the input of text for this command the following control characters are active. Detailed explanations are available from H(elp): ^A, ^H(backspace), ^X, ^W, ^R, ^Q, ^V, ^~, ^\, DEL and \$(escape).

S SEARCH - Search accepts a string from the terminal. It then locates ALL occurrences of the string in the text, starting with the current line. The NEW current line is the last occurrence found, or the starting line, if no occurrences were found. If just a <CR> is typed to the prompt for a string, Search uses the string from the last F(ind) or Search command. If the search string is a single null character (specified by control-V control-@ [^V^@] or control-^ @ [^^@]), then XED will search for a null line (no text, only <CR><LF>). If this command is input by accident, typing ^Q will cancel it. While this command is executing, typing ^Q interrupts it and returns to XED command mode. This command normally looks for the string any place in the line, independent of upper and lower case differences.

^E If ^E is specified, the match will be exact only.

^B If ^B is specified, the match will be only at beginning of lines.

^S If ^S is specified, XED will enter Change on the line(s) containing the string, positioning the Change cursor at the found string.

During the input of text for this command the following control characters are active. Detailed explanations are available from H(elp): ^A, ^H(backspace), ^X, ^W, ^R, ^Q, ^V, ^~, ^\, DEL and \$(escape).

LINE EDITING COMMANDS

I INSERT - The insert command accepts text lines from the terminal. These text lines are inserted BEFORE the current line in the text buffer. This command is normally terminated by a ^Z. If ^Q is typed, any text typed on the current line is ignored and XED returns to command mode. If this command is input by accident, typing ^Q will cancel it. During the input of text for this command the following control

characters are active. Detailed explanations are available from H(elp): ^A, ^H(backspace), ^X, ^W, ^R, ^Q, ^V, ^^, ^\, DEL and \$(escape).

A APPEND - The append command accepts text lines from the terminal. These text lines are appended AFTER the current line in the text buffer. This command is normally terminated by a ^Z. If ^Q is typed, any text typed on the current line is ignored and XED returns to command mode. If this command is input by accident, typing ^Q will cancel it. During the input of text for this command the following control characters are active. Detailed explanations are available from H(elp): ^A, ^H(backspace), ^X, ^W, ^R, ^Q, ^V, ^^, ^\, DEL and \$(escape).

^I (tab) TAB APPEND - acts like APPEND, with a tab as the first character of the first line appended.

D DELETE - The delete command deletes the current line. Once input there is no way to recover the line. For this reason K(ill) is considered a safer command.

K KILL - Kill accepts a number following it, and deletes that number of lines starting at the current line. The NEW current line is the line AFTER the deleted lines. If no number follows, only the current line is deleted. If the number following is preceded by a :, instead of killing n lines, it will kill all lines from the current line up to, and including, line n.

Example: 12k:15. kills lines 12,13,14,15.

All the lines which are deleted are saved in the text dump. Each subsequent kill destroys the previous text dump contents. Commands Print text dump and J(am text dump) operate on the text dump. If this command is input by accident, typing ^Q will cancel it.

P PRINT TEXT DUMP - Each K(ill) operation saves the deleted lines in the text dump. Print text dump prints the current contents of the text dump. See also J(am text dump). While this command is executing, typing ^Q interrupts it and returns to XED command mode.

' SWITCH DUMP - Exchanges the current contents of the text buffer with that of the text dump (see K(ill), J(am), Printdump) commands). This command is useful for performing operations with sections of large files. By K(ill)ing a section of text into the dump, then switching the dump with the buffer, operations may then be performed on the small section. Repeating the command causes repetitive switching (i.e., two ' commands in a row have no effect).

TUG
XED

J JAM - Each K(ill) operation saves the deleted lines in the text dump. J will jam the current contents of the text dump after the current line. The NEW current line will be the last line appended from the text dump. See also Print text dump).

/ SLASH - / prints the current line.

^ UP ARROW - ^ moves the current line to the line preceding the current line in the text buffer and prints the NEW current line. This is the same as -/.

T TYPE - Type accepts a number following it, and prints that number of lines starting at the current line. The NEW current line is the last line printed. If no number follows one line is printed. If the number following is preceded by a :, instead of typing n lines, it will type all lines from the current line up to, and including, line n.

Example: 12t:15. types lines 12,13,14,15.

If a T command follows a T command, then the current line is advanced before the second T command, so the same line is not printed twice. If this command is input by accident, typing ^Q will cancel it. While this command is executing, typing ^Q interrupts it and returns to XED command mode.

V VIEW - starting at the current line, V(iew) prints the next page of text (approx. 20 lines). The NEW current line is the last line printed. While this command is executing, typing ^Q interrupts it and returns to XED command mode.

, CONTEXT - ,(context) accepts a number n following, after which it types the n lines BEFORE the current line, the current line, and n lines AFTER the current line. The effect is to type (2n+1) lines, with the current line in the middle. If n is not given, 5 is assumed. ,(context) does NOT change the current line. If this command is input by accident, typing ^Q will cancel it.

L LIST - List will print the entire text buffer from begining to end on the terminal. If is the same as saying TTY: to W(rite). There is a way to have a document line stating the current file name, time, and person who edited the file in the front of the file each time it is written. See) and (for further details. While this command is executing, typing ^Q interrupts it and returns to XED command mode.

INTRA-LINE EDITING COMMANDS

C CHANGE - The change command is used for editing a single line of text. There are three basic operations in the change command. First characters may be copied from the old copy into the new line (SPACE,E,S). Characters may be deleted from the old copy (D,K,R). Finally characters may be added to the new line from the terminal (I,R). C(hange) operates on the CURRENT line. To get a full explanation of the C(hange) subcommands input a ? while within the change command. If this command is input by accident, typing ^Q will cancel it. During the input of text for this command the following control characters are active. Detailed explanations are available from H(elp) : ^A, ^H(backspace), ^X, ^W, ^R, ^Q, ^V, ^^, ^\, DEL and \$(escape).

Change Subcommands:

? Prints this message.
n SPACE Space n characters forward copying from old to new.
B Break line - Insert current contents of new line before the current line. Change subcommands may continue with the rest of the old line, however Q and ^Q can not affect the inserted text.
n D Delete n characters forward from the old copy.
E Move to end of line copying from old to new.
I Insert mode - all input inserted until next CR,LF, or ^Z.
n K x Delete forward until nth occurrence of 'x'.
n L x Force alphabetics to be Lower case up to the nth occurrence of 'x'.
P Print the rest of old line and current new line.
Q Abort all changes since last B subcommand.
n R Delete next n characters forward and enter insert mode.
n S x Space forward to before the nth occurrence of 'x'.
n U x Force alphabetics to be Upper case up to the nth occurrence of 'x'.
n V x inVert case of all alphabetic characters up to the nth occurrence of 'x'.
CR Copy rest of old line to new, update current line to new
LF Update current line to new, as is
^D Start C(hange) over, forgetting edits so far
ESCAPE Copy rest of old line to new, update current line to new, and re-enter C(hange) on the updated current line.

If the 'x' in K, S or V command is a CR, then it is interpreted as meaning "to the end of the line"; if it is ESCAPE, it is interpreted as "use the same character as last time."

G GROUP - Group combines the current line with the following line to produce a single new text line. The current line is set to the new combined line. If this command is executed accidentally, using the "B" subcommand of the C(hange) command can separate the two lines.

TUG
XED

X XCHANGE - Xchange takes 2 strings, called OLD and NEW. It then replaces occurrences of OLD with NEW, according to the user's directions. Upon finding a line with an occurrence of OLD, it replaces all the instances of OLD with NEW, types the line on the terminal, and asks the user for confirmation. Typing a "?" to the confirmation request displays the possible options. If the user types just a <CR> to the prompt for NEW, then the occurrences of OLD will be deleted. If <CR> is typed for both OLD and NEW, Xchange uses the same strings it used the last time. If the PRINT XCHANGES mode is set, all XCHANGED lines will be printed even if NOCONFIRM (*) is specified. If the search string is a single null character (specified by control-V control-@ [^V@] or control-^ @ [^@]), then XED will search for a null line (no text, only <CR><LF>). If this command is input by accident, typing ^Q will cancel it. While this command is executing, typing ^Q interrupts it and returns to XED command mode. This command normally looks for the string any place in the line, independent of upper and lower case differences.

^E If ^E is specified, the match will be exact only.

^B If ^B is specified, the match will be only at beginning of lines.

^F If ^F is specified, XED will do case-sensitive XCHANGES.

During the input of text for this command the following control characters are active. Detailed explanations are available from H(elp): ^A, ^H(backspace), ^X, ^W, ^R, ^Q, ^V, ^^, ^\, DEL and \$(escape).

X(change) Confirmation Codes

Y,<space>,CR,LF

Accept printed changes and continue through the text buffer.

N,^H(Backspace),^A,DEL

Reject printed changes and continue through the text buffer.

.

Accept printed changes and stop here.

E

Reject printed changes and stop here.

*

Accept printed changes and continue through the text buffer, accepting all changes without requesting confirmation.

If any other character is input, the confirmation request is repeated.

FILE INPUT/OUTPUT COMMANDS

R READ - Read accepts a file name from the terminal and reads the contents of the file into the text buffer by appending the contents AFTER the current line. The old text buffer contents are not changed. Input and output operations to files will be encrypted if the ENCRYPT mode is set. If this command is input by accident, ^Q or DEL will cancel it. While this command is executing, typing ^Q interrupts it and returns to XED command mode.

E EXIT - Exit accepts a file name from the terminal and writes the contents of the text buffer into this file. When the write is complete the editor exits back to the Executive (@). If the termination is not desired, use the W(rite) command. There is a way to have a document line stating the current file name, time, and person who edited the file in the front of the file each time it is written. See) and (for further details. Input and output operations to files will be encrypted if the ENCRYPT mode is set. If this command is input by accident, ^Q or DEL will cancel it. After this command has returned to the Executive(@), inputting CONT<CR> will continue the editor with no changes to the current line, text buffer etc.

B BACKUP - When a file name is entered to R(ead), W(rite), or E(xit), the editor remembers the name. When executed, B(ackup) writes the text buffer into specific versions of that file name. Normally, B(ackup) writes the buffer into version 2 of the file. If version 2 previously existed, it is renamed to version 1 before the write (the previous contents of version 1 being lost). Thus, repeated uses of the B(ackup) command write over the same files, minimizing the proliferation of new versions of the file. If the SPECIAL BACKUP mode is set, B(ackup) uses the two version numbers numerically following the highest version of the file last read or written instead of versions 2 and 1. This has the advantage that the files created by the B(ackup) are the highest versions of the file and will be referenced when the version number is defaulted (which is almost always the case). To recover the backup copy of the file after a system crash simply read as normal. Read always checks for version 2 backup copies with later write times than the requested versions. There is a way to have a document line stating the current file name, time, and person who edited the file in the front of the file each time it is written. See) and (for further details. Input and output operations to files will be encrypted if the ENCRYPT mode is set. While this command is executing, typing ^Q interrupts it and returns to XED command mode.

W WRITE - Write accepts a file name from the terminal and writes the contents of the text buffer into this file. If the user plans to Q(uit) immediately following the W(rite), the E(xit) command might be more convenient. There is a way to have a document line stating the current file name, time, and person who edited the file in the front of the file each time it is written. See) and (for further details. Input and output operations to files will be encrypted if the ENCRYPT mode is set. If this command is input by accident, ^Q or DEL will cancel it. While this command is executing, typing ^Q interrupts it and returns to XED command mode.

() PARENTHESES - Each time a file is written by B(ackup), W(rite), E(xit), or L(ist) it is possible to have an extra line

added to the top of the file stating the current file name, time, and person who edited it. In addition to this information, there is also facility for a prefix string and a suffix string to be put before and after this information. These latter strings are provided to make these lines into comments for the programs that might process the files. This line will automatically be generated ONLY IF a prefix string has been specified.

(PREFIX - The prefix command accepts a string from the terminal which will be used as the prefix to the automatically generated documentation line when the file is written. The documentation line is NOT added to the text buffer, but is obviously included when the file is next read into the editor. The prefix is required to trigger the automatic document line generation. Inputting (<CR> clears the prefix string and turns off the automatic document line generation. If this command is input by accident, typing ^Q will cancel it. During the input of text for this command the following control characters are active. Detailed explanations are available from H(elp): ^A, ^H(backspace), ^X, ^W, ^R, ^Q, ^V, ^^, ^\, DEL and \$(escape).

() PARENTHESES - Each time a file is written by B(ackup), W(rite), E(xit), or L(ist) it is possible to have an extra line added to the top of the file stating the current file name, time, and person who edited it. In addition to this information, there is also facility for a prefix string and a suffix string to be put before and after this information. These latter strings are provided to make these lines into comments for the programs that might process the files. This line will automatically be generated ONLY IF a prefix string has been specified.

) SUFFIX - The suffix command accepts a string from the terminal which will be used as the suffix to the automatically generated documentation line when the file is written. The documentation line is NOT added to the text buffer, but is obviously included when the file is next read into the editor. The PREFIX is required to trigger the automatic document line generation. If this command is input by accident, typing ^Q will cancel it. During the input of text for this command the following control characters are active. Detailed explanations are available from H(elp): ^A, ^H(backspace), ^X, ^W, ^R, ^Q, ^^, ^\, DEL and \$(escape).

O OUTPUT - this command writes a formatted output file suitable for printing on a hardcopy device such as a line printer (it is NOT intended for normal file writing!). It can provide titles, page numbers, and a summary of editing changes. It is affected by several modes which determine what kind of headings will appear in the output, how page

TUG
XED

numbering is to be done, and the size of the output page. These modes are found in the Format/Output mode group of the mode dialogue. When invoked, O(utput) first prompts for a heading (unless the mode affecting the heading specified not to prompt). It then asks for the name of the output file. If a <CR> or <ESCAPE> is specified for the file name, then LPT: is assumed. If this command is input by accident, typing ^Q will cancel it. While this command is executing, typing ^Q interrupts it and returns to XED command mode.

Q QUIT - Quit will exit the editor WITHOUT writing a new copy of the file. After this command has returned to the Executive(@), inputting CONT<CR> will continue the editor with no changes to the current line, text buffer etc.

MODES

" MODE DIALOGUE - Puts the user in an interactive dialogue, allowing the user to set and examine the XED mode settings. It divides the settings into groups of modes, and allows the user to select or skip each group. Within a group, the user then may examine and optionally set any of the modes within the group. At any time during the dialogue, the user can skip the rest of the group selections and individual mode selections, and can also cancel any changes made. It also allows the user to define mode files to contain specified mode settings, and recall them later.

PROGRAM INVOKING COMMANDS

% Call SNDMSG - allows the text buffer to be sent to SNDMSG without going through EXEC or using SNDMSG's ^B option. XED prompts for subject of message, after which user is prompted for To:, cc:, etc., just as in SNDMSG. The message will have the text buffer inserted as the body. It is important not to type ahead while the message body is being inserted, as the input will be merged randomly with the text buffer into the message body. When the message has been sent, XED will return to command mode, with the text buffer untouched. If this command is input by accident, typing ^Q will cancel it. During the input of text for this command the following control characters are active. Detailed explanations are available from H(elp): ^A, ^H(backspace), ^X, ^W, ^R, ^Q, ^V, ^^, ^\, DEL and \$(escape).

! Run program - allows the user to call any TENEX subsystem or other runnable program (including the TENEX EXEC) without disturbing the state of the text buffer, current pointer, etc. XED prompts for the name of a program to run (directory defaults to <SUBSYS>, extension to .SAV) and then runs the

TUG
XED

requested program. When the program is finished running (such as by QUIT to the EXEC), control returns to XED, leaving it in the same state as when the program was called. If this command is input by accident, ^Q or DEL will cancel it.

@ Call EXEC - starts up the TENEX EXEC without disturbing the state of the text buffer, current pointer, etc. When a "QUIT" command is given to the EXEC, control returns to XED.

[Start Fork - starts last program run by a program-running command (!,%,@). If no program has been run, or the last fork was killed by the > command, the user is prompted for the name of a new program as in the ! command.

< Continue Fork - continues last program run by a program-running command (!,%,@). If no program has been run, or the last fork was killed by the > command, the user is prompted for the name of a new program as in the ! command.

> Kill fork - kills the last program run, and its fork. If no program has been run, or it has already been killed, this command has no effect.

FORMATTING

FORMAT - The format command allows the user to do limited text formatting and justification on portions of (or all of) the text buffer. If the command is followed by a <CR> or <SPACE>, then the whole buffer is formatted. It can also be followed by a line range specification, analogous to the T(ype) or K(ill) commands. Thus,

40 means format the next 40 lines in the buffer, starting at the current line

_ :125 means format the lines starting at the current line and extending to line 125

[Note that the formatter is an experimental facility and subject to change as its features are used, critiqued and refined.]

Several modes are relevant to the format command (set and examined by the " command). The command is enabled by the ENABLE FORMAT mode (in the Command Enable/Disable group). If justification (adding of additional spacing to even the right margin) is desired, then the Justification mode (in the Miscellaneous group) must be turned on. The margin used by the formatter is also settable by the " command (in the Miscellaneous group). Its default value is 70.

In order to protect the user from performing a format operation which produces a result the user does not expect, the lines which are being formatted are first copied to the text dump. If the user has formatted the whole file, he can retrieve it with the '(Switch dump) command; otherwise, he can use the J(am dump) command.

For normal operations, the formatter requires no explicit commands. Paragraphs are separated by blank lines. Each paragraph may contain "flags" which direct the formatter to perform indentation on the entire paragraph. The blank lines are retained after formatting, so that subsequent formats of the same text will have the same paragraph boundaries.

The formatter recognizes two special text lines as command directives. The first is a line with only a single, isolated tab. It directs the formatter to turn off formatting until the occurrence of another isolated tab line. This is useful to allow certain pre-formatted information (such as tables) to be skipped over by the formatter. [Note that unintended results can occur if the isolated tab lines are not properly paired!]. The second command is a line with an isolated space. This acts as a paragraph break and directs the formatter to indent the following paragraph the same as the previous one. To help the user recognize the occurrence of these special command lines, XED highlights them (on terminals so equipped) or prints special identifying lines (e.g., << isolated tab >>).

XED PARAGRAPH FORMATTING

Note: [This description has been formatted by the XED format command. Thus it both describes the process and demonstrates the available features.]

I. Normal Paragraph Processing

The first line of a formatted block is treated as a new paragraph. A new paragraph is also created after a blank line (2 consecutive carriage-returns). The paragraph is terminated either:

- the end of the block of lines being formatted
- a blank line
- a line with a single isolated tab

In the latter case, formatting is also turned off until the next line containing an isolated tab is encountered.

A. There are two items in the XED mode file which affect the formatting process. They are the justification flag and the line width.

1. The justification flag specifies whether the text should be justified to the right margin. If the flag is on, lines will be filled with as many words as can fit, then spaces will be added to even the right margin. This example is justified.
2. The line width determines how wide the formatted lines will be. (In the case of no justification, this is the maximum length.) This example has the line width set to 70.

B. The normal paragraph processing puts the proper number of words on each line. If justification is requested, each line is also justified. The basic features of this formatting process (besides line-filling and justification) are:

1. Paragraph indentation is not changed. If the first word is flush to the left margin, the formatted paragraph will also start flush; if the first word is indented five spaces, the formatted paragraph will also have the first word indented five spaces. However, all other spacing characters (space, tab, carriage return, line feed) within the paragraph are simply used to separate words.

All indentation (whether just the first line of a paragraph or for an entire paragraph) is done with spaces. Tabs are converted into the proper number of spaces.

2. Hyphenated words are recognized and possibly broken. This has the side effect of possibly adding spaces (after the hyphen) in the middle of hyphenated words in later versions.
3. All periods, colons, question marks, and exclamation points appearing at the end of words are followed by two spaces (minimum), instead of the one normally used between words.

II. Flagged Paragraph Processing

In addition to normal paragraph processing, the facility also provides for paragraphs which are entirely indented. These are called "flagged" paragraphs.

- A. The general form of a flagged (indented) paragraph
 - 1. A flagged paragraph starts with at least one spacing character (tab or space).
 - 2. This is followed by a flag. The flags are described in section III below.
 - 3. Finally, the flag must be followed by at least one spacing character.
- * To reiterate, in order for a flag to be recognized, it must be the first word in a paragraph with spacing both before and after. This is done to prevent normal paragraphs from being mistaken for flagged paragraphs. Following are the differences between the processing of flagged paragraphs and normal paragraphs.

B. Flag Formatting:

The flag will be indented in the formatted result the same amount it was indented in the original text. This is similiar to the initial paragraph indenting provided for normal paragraphs. The flag will be followed by two spaces which will not be available for expansion during justification.

C. Succeeding Lines in Flagged Paragraphs:

All lines of the flagged paragraph will line up with the first line of the paragraph and will be indented from both margins.

INDENTED PARAGRAPHS WITHOUT FLAGS

When you want more than one paragraph to line up under a flag, insert a line with a single isolated space between the two paragraphs. The next paragraph will line up automatically with the previous one. This section makes extensive use of this facility.

III. Description of Flags

Flags are used to indent entire paragraphs. There are four kinds:

SINGLES: A single is one of the following: * (star), + (plus), - (hyphen), % (percent), and o (lower-case letter O).

SECTIONS: A section flag can be a single letter, a letter and a number (0-9 only), any number 0-99, or a number (0-9) and a letter. It cannot be two letters since a paragraph could start with Mr. Smith or Dr. Jones.

TUG
XED

Examples: A,1,A1,22,1a. A section cannot be more than 2 characters.

ROMANS: Any roman numeral between 1 (I) and 399 (CCCXCIX). The roman numerals can be upper- or lower-case.

NAMES: Any string containing only alpha-numerics, . (periods) or - (dashes).

Four classes of flags are recognized by the following punctuation:

- A. Any SINGLE by itself (spacing before and after) is considered to be a flag.
- B. Any SINGLE, SECTION, or ROMAN followed by a . (period) or a) (close parenthesis) is considered to be a flag.
- C. Any SINGLE or SECTION enclosed in balanced brackets is considered to be a flag. Recognized brackets are: <>, {}, [], and () .
- D. Any NAME followed by a : (colon) is also considered to be a flag.

EXAMPLE FLAGS

Note: [This section is not formatted. The following line contains a single isolated tab which turns off the formatting process.]

Example flags with appropriate punctuation:

SINGLE SECTION ROMAN NAME

Selected from:

Flag group A:
SINGLE ALONE

*

+

-

o

%

Flag group B:
BRACKETED

(*)

[A]

<o>

(D3)

{+}

<7>

Flag group C:
SECTION DESIGNATOR

o)

23.

IV)

*.

2a)

xxxii.

+

Q.

CX)

Flag group D:
KEYWORD

Flagword:
Case-176:
STATUS.LAST:

CONTROL CHARACTERS

Control characters are input by depressing the CONTROL key (which is a shift key and by itself does not transmit any code to the computer) followed by the appropriate character. Control characters are represented as an ^ preceding the character. E. G.: ^A represents Control A.

INTERRUPT CONTROL CHARACTERS

These characters may be input at any time. ^C and ^T are available anywhere in TENEX, while the others are only supported by XED.

^C This stops the editor and goes immediately to the Executive(@). The editor may be continued by inputting CONT<CR>.

^T This will print the current load average of the system.

^N This will print the current line number. This is useful for tracking the progress of R(ead), W(rite), B(ackup), E(xit), F(ind), and S(earch).

TEXT INPUT CONTROL CHARACTERS

These characters are available whenever the user is inputting a text string from the terminal.

^A **^A**, **^H**(backspace), **DEL**(rubout), will all delete the last character input. **DEL**(rubout) additionally aborts commands which accept file names. In this file name situation **DEL** is active during R(ead), W(rite), E(xit), H<CR> [Manual Request], and when B(ackup) requests a file name because not file name has yet been specified.

This control character is active during A(ppend), I(nsert), C(hange), F(ind), S(earch), X(change), ((set prefix),) (set suffix), and % (SNDMSG).

^W **^W** deletes the last word input. Words are delimited by tabs and spaces. **^W** also deletes and spaces or tabs which might have followed the last word input.

This control character is active during A(ppend), I(nsert), C(hange), F(ind), S(earch), X(change), ((set prefix),) (set suffix), and % (SNDMSG).

^X **X** deletes the current line being input from the terminal.

This control character is active during A(ppend), I(nsert), C(hange), F(ind), S(earch), X(change), ((set prefix),) (set suffix), and % (SNDMSG).

^R ^R re-types the current line being input.

This control character is active during A(ppend), I(nsert), C(hange), F(ind), S(earch), X(change), ((set prefix),) (set suffix), and % (SNDMSG).

^Q This will stop an executing command. It can stop B(ackup), F(ind), L(ist), O(utput), P(rint text dump), R(ead), S(earch), T(ype), V(iew), W(rite) and X(change).

^Q is also active whenever a command or text is being input. If typed while entering the number parameter to a command like K(ill) or T(ype), it will abort the command. If typed while Inserting or Appending, the current text line will be discarded and XED will go back to command mode.

^V ^V "quotes" the next character input. It is used to input characters which would normally cause control or editing functions (e.g., ^Z, ^X, ^Q, ^B). For example, to put "ABC^Z" in a text line, one would enter ABC^V^Z. ^V would be entered into text by ^V^V.

This control character is active during A(ppend), I(nsert), C(hange), F(ind), S(earch), X(change), ((set prefix),) (set suffix), and % (SNDMSG).

^^ ^"control shifts" the next character input. (e.g. ^C inputs ^C, while ^^c inputs).

This control character is active during A(ppend), I(nsert), C(hange), F(ind), S(earch), X(change), ((set prefix),) (set suffix), and % (SNDMSG).

^ ^\ "case shifts" then next character input. (e.g., ^\C inputs c, ^\c inputs C, while ^\ inputs ^C).

This control character is active during A(ppend), I(nsert), C(hange), F(ind), S(earch), X(change), ((set prefix),) (set suffix), and % (SNDMSG).

SEARCH MODIFICATION CONTROL CHARACTERS

These characters cause special actions to be performed during the execution of the certain commands. To enter them as text for these commands, they must be "quoted" by ^V or ^^.

^B ^B specifies that Find, Search, and Xchange must match only at the Beginning of a text line. (The default is to match anywhere in the text line). The ^B may be typed either before the command it affects (F,S or X), or during the typing of the

string affected. If this command is entered more than once, it reverses the effect of the previous one (or the prevailing default setting), i.e., it acts as a toggle.

^E **^E** specifies that Find, Search, and Xchange must Exactly match the text string, including case. (The default is to match exclusive of case). The **^E** may be typed either before the command it affects (F,S or X), or during the typing of the string affected. If this command is entered more than once, it reverses the effect of the previous one (or the prevailing default setting), i.e., it acts as a toggle.

^F This command affects the X(change) command. When specified, it causes XED to do case-sensitive XCHANGEing. In this mode, XED categorizes the found string into one of four classes (UPPER CASE, lower case, Capitalized, or aRbitrAry). When it substitutes the replacement string, it maps it into the same class as the found string (if the found string is arbitrary, XED makes no change). The **^F** may be typed either before the X(change) command, or during the typing of the OLD string. If this command is entered more than once, it reverses the effect of the previous one (or the prevailing default setting), i.e., it acts as a toggle.

^S This command affects the searching commands (F,S). When specified, it causes XED to enter Change for the line within which the string was found, positioning the Change cursor at the found string. The **^S** may be typed either before the command it affects (F,S), or during the typing of the string affected. A mode in the Miscellaneous mode group allows the user to specify that the C(hange) cursor be positioned at the end of the found string. Normally, it is positioned at the beginning. If this command is entered more than once, it reverses the effect of the previous one (or the prevailing default setting), i.e., it acts as a toggle.

\$ ESCAPE TO CHANGE - During the insertion of text lines, the escape character passes the line currently being input to C(hange). When C(hange) is terminated in the standard way, the line input process continues. If escape is typed when the text is empty, the text line will first be initialized to the LAST string entered in the current context, e.g., if the string is being entered for a FIND command, the string used in the last FIND/SEARCH command will be used; if the string is being entered into the text buffer by an APPEND, the last string entered during the last APPEND/INSERT command is used. This control character is active during A(ppend), I(nsert), C(hange), F(ind), S(earch), X(change), (.. (set prefix),) (set suffix), and % (SNDMSG).

TUG
ZTYPE

ZTYPE

ZTYPE is a program for listing text files (RUNOFF output files, for example) on a 300 baud terminal such as Diablo, Anderson Jacobson, Bedford, etc.

Example of its use:

@ZTYPE<space> (File) afilename (Starting at page) a page At this point ZTYPE will output a formfeed and ring the bell. It then waits for the user to type a character (signifying the paper is ready in the terminal). The user typed character will not be echoed on the paper. If you wish to start at page 1 of the file, you can end the file name with a carriage return. Typing carriage return to the page number request will also assume page 1.

Whenever ZTYPE rings the bell at the end of a page, you may type one of the following "special" characters:

"X" - This stops ZTYPE and returns you to the EXEC

"P" - Position to a page number. Type "P" followed by the desired page number

"B" - Starts over at the beginning of the file.

Any other character simply tells ZTYPE the paper is ready and to type out the next page.

None of the typed in characters will be echoed on the terminal.

Insert feature: It is possible to type in text on-line while ZTYPE is running. Whenever a control-F is found in the text file, ZTYPE will stop and echo any characters you type in. When a control-F is typed, ZTYPE will continue listing the file.

