

```
*****
***                                     ***
***           QED Screen Editor Manual   ***
***                                     ***
*****
```

Manual Revision 1.0

May 18, 1983

Written by

Carl Galletti

Copyright 1983 by Computer Design Labs



## INTRODUCTION

---

QED is one of the best screen editors available. This is not due to having a lot of features, but rather because of its degree of simplicity in performing powerful operations. QED is a simple editor to master. Yet, it retains a sufficient amount of power to perform almost any task. Although QED has its limitations, you will discover that it gets its speed of operation from the fact that it cuts out some "fancy" features which were not needed in the first place.

For instance, QED can only edit files as big as available memory. This may seem a limitation. But as you use QED, you will discover that the extra speed you gain is worth it. Besides, well organized programs and documents should be segmented into discrete pieces. And since most computers now have 64K of memory, there is more than enough space for any application.

QED does not have word wrap. The strategy here is to use a separate "text processor" such as our "TOP" or "TOP II" to justify the text. As a result, you use QED like a typewriter; type until you reach the end of the line and then press return. The resulting files are viewable with the operating system's "TYPE" command.

With QED you can read other files into the one you are working on and write part or all of the current file to a new file. This gives you an unlimited glossary function. A glossary function makes it possible to save standard blocks of text that get inserted into new material.

QED does not require you to CRUNCH up your fingers on the keyboard to move the cursor around. All of the cursor commands are logical and mnemonic (i.e., the key matches the function); plus, if you have a numeric key pad but no cursor pad on your terminal, the numeric pad can be used as a cursor pad. If you have a cursor pad with some unique character codes, you can modify the existing commands to match your cursor pad.

Unlike other "popular" screen editors which require you to move your hands out of the standard typing position to perform most functions, QED has a unique command mode which makes it easy to perform all functions from the standard typing position. This unique command mode also eliminates "timid keyboard syndrome". That's where you're afraid to hit the keyboard for fear of overwriting some text or destroying large amounts of work. Somewhere along the line someone thought that the overwrite mode is the most often used mode of operation. That's the mode where what you type replaces what is on the screen. We have discovered this to be the LEAST used mode. You can use this mode in QED. The difference is that QED doesn't make it your "home base" mode as other editors do.

Some people have asked us why we don't have an "UNDO" function. That's simple. You make less errors with QED. That's partly because we don't force you to think in the overwrite mode, partly because it's simple to use, and partly because it's easy to recover from mistakes.

Before operating QED, you must make sure that it is set up for your CRT screen. Although the procedure is not difficult, it does require familiarity with assembly language, operating a debugger (such as DDT or ZDDT), and how to save programs. An easy to operate INSTALL program is forthcoming. This program will automatically configure QED if your terminal is on its list of

devices. Otherwise, you will be able to interactively build a driver for one that is not. Until this INSTALL program is ready, you will have to follow the instructions in the CONFIGURATION chapter which follows.

If you are not familiar with assembly language, DDT, and/or how to save programs; you have several choices. First, you may find someone you who is familiar with these things to help you. Or, you may follow the directions carefully, studying the appropriate manuals/literature as you go along. This is actually the best way. It is the way most top programmers got to be top programmers. The learning experience is invaluable. However, lets suppose that you refuse to learn how to use programming tools no matter what. Until we have an INSTALL program ready, we do the installation for you. You need to supply us with the proper information (Clear screen, addressable cursor, and erase to end of line information as well as the number of lines per screen) and we will do the installation. Also, you must send a blank formatted diskette, return postage, and any other information that might be peculiar to your terminal.

After you install the terminal driver into either "QED24.COM" or "QED25.COM", rename the final copy "QED.COM". The major portion of this manual assumes that you are using this name.

```
*****  
*** NOTE: Before you do anything with the diskette ***  
*** you received from us, make a copy of it and make ***  
*** changes ONLY to the copy. ***  
*****
```

## CONFIGURATION

---

You receive two copies of QED, a 24 line and a 25 line version. The 24 line version is set up for a Hazeltine 1500 terminal or one that uses the same screen commands. The 25 line version is set up for a terminal that uses the Lear Siegler ADM-3A screen commands (the ADM3A has 24 lines but many of the 25 line terminals use its screen commands; to actually run an ADM3A, you need to install the ADM3A terminal driver on the 24 line QED). The 24 line version is named "QED24.COM" and the 25 line version is named "QED25.COM". You must select the version which has the same number of lines per screen regardless of which type of terminal it is set up for.

If the terminal driver installed is not correct for your terminal, you must change the proper version of QED to conform to your terminal. In most cases, the terminal driver is a simple modification of either the Hazeltine 1500 or Lear Siegler ADM-3A. There are two additional SETS of files included on this diskette. One set is HAZ15.ASM, HAZ15.PRN, and HAZ15.COM. This set is for the Hazeltine 1500. The other set, ADM3A.ASM, ADM3A.PRN, and ADM3A.COM, is for the Lear Siegler ADM-3A. The "ASM" files are the source of the drivers. The "PRN" files are the listing of the source. And, the "COM" files are the object code files of the terminal drivers. The "COM" files may be patched into the appropriate version of QED. The "ASM" files may be modified at the source level, reassembled with a Computer Design Labs MACRO I, II, or III assembler, and the resultant object code patched to QED. If you do not have any of these assemblers, you may make patches to the terminal driver's "COM" file using DDT, ZDDT or an equivalent debugger and patch the result into QED. To do this, the "PRN" file is helpful in locating actual locations of codes which must be modified.

To modify one of the terminal driver files, you need to know what the computer sends to the terminal to perform the following functions:

```
>>>   Clear screen
>>>   Addressable cursor
>>>   Clear/erase to the end of line
```

Then, examine the "PRN" files. If only difference between your terminal and one of our terminal drivers is in which codes are used to perform the above functions, you should change the codes by patching them into QED. If the modification is more extensive and you have one of our Macro assemblers, you can use one of the "ASM" files as a starting point and write your own driver. This requires knowledge of assembly language. If you are not confident in doing this, send a blank diskette and a copy of your terminal's manual (or just the sections which describe the above three functions), and return postage to Computer Design Labs and we will return a driver for your terminal.

As an example, the following procedure describes how to modify the ADM-3A driver and install it on the 24 line version of QED.

1. Print a copy of ADM3A.PRN to use as a reference.
2. Get the terminal information for the clear screen, address cursor and clear to end of line functions.
3. DDT the "ADM3A.COM" file into memory. Note the "NEXT" address.
4. Substitute your terminal's codes for those used in the ADM-3A driver.
5. Exit from DDT.
6. SAVE the new program using the "NEXT" address above to figure the

- size and assign a name to the file. We'll assume you named it "TERM.COM". (e.g., "SAVE 1 TERM.COM")
7. DDT QED24.COM into memory.
  8. Read the address at location 131. Remember it is ordered low byte then high byte. (e.g., "1CF8")
  9. Read "TERM.COM" into memory at the address obtained in step 8. (e.g., "R1CF8" for "DDT" this needs to be preceded by "ITERM.COM")
  10. Exit from DDT (^C).
  11. Save the program. (e.g., "SAVE 29 QED.COM").
  12. THE END

The following information may also be useful when preparing terminal drivers.

There is a jump table at the beginning of QED which is used by QED in locating the terminal driver routines for performing the screen control functions. If you write your own drivers, the addresses of your driver routines which correspond to this table are likely to change. They will have to change to point to your routines. To find the address to assemble your driver, use the address at location 131. You must start your driver with the name of the terminal. Use the same length string as in our sample drivers. If the cursor position routine is placed immediately after, you won't have to change its address in the jump table.

Here is a sample of the jump table using the Hazeltine driver. Keep in mind the fact that the CDL Macro assemblers print the jump address the correct way in the listing rather than reverse order as it appears in memory. The ".WORD"s are still in reverse order (i.e., "TERM" and "TABLE").

```

;*****
;THIS IS THE START OF THE JUMP TABLE:
0122 C3 0938  PUTCHR: JP PUT1          ;SEND A CHAR TO CONSOLE (0938)
0125 C3 1D1B  POSCRS: JP UPOCRS         ;USER POSITION CURSOR ROUTINE (1D1B)
0128 C3 1D34  CLRSCR: JP UCLRSCR        ;USER CLEAR SCREEN ROUTINE (1D34)
012B C3XXXX  ;***** This is no longer needed *****
012E C3 1D40  ERASL:  JP UERASL         ;USER CLEAR TO END OF LINE (1D40)
0131 F81C          .WORD TERM          ;address of start of terminal name
                                ;message, 32 character spaces are
                                ;available, they must be followed by
                                ;a carriage return/linefeed and a 0.
                                ;(1CF8)
0133 3906          .WORD TABLE       ;address of the command table which may
                                ;be used to customize some commands.
                                ;(0639)
;*****

```

The last entry, TABLE, shows the location of one of the command tables. It is a series of bytes which can be changed to alter the cursor keys which are used to perform cursor movement in the command mode. It is organized as follows:

```
.BYTE 15      ;LENGTH OF TABLE
.BYTE 'X'     ;EXECUTE LAST COMMAND AGAIN
.BYTE '1'     ;MOVE CURSOR TO BOTTOM LEFT OF SCREEN
.BYTE ' '     ;MOVE CURSOR TO THE LAST TAB POSITION
.BYTE '^'     ;MOVE CURSOR UP ONE LINE
.BYTE LF      ;LINEFEED(OAH), MOVE CURSOR DOWN ONE LINE
.BYTE ' '     ;(SPACE-20H), MOVE CURSOR ONE SPACE RIGHT
.BYTE 8H      ;BACKSPACE, MOVE CURSOR ONE SPACE LEFT
.BYTE '6'     ;MOVE CURSOR ONE SPACE RIGHT (DUPLICATE)
.BYTE '4'     ;MOVE CURSOR ONE SPACE LEFT (DUPLICATE)
.BYTE '2'     ;MOVE CURSOR ONE LINE DOWN (DUPLICATE)
.BYTE '8'     ;MOVE CURSOR UP ONE LINE (DUPLICATE)
.BYTE '7'     ;MOVE CURSOR TO UPPER LEFT OF SCREEN (HOME)
.BYTE 9H      ;MOVE CURSOR TO NEXT TAB POSITION
.BYTE 0DH     ;CARRIAGE RETURN
.BYTE 12H     ;ALTERNATE CARRIAGE RETURN (^R) USED IN SEARCHES
```

## HOW TO START THE EDITOR

---

This section assumes that you have configured QED to your terminal and have renamed the proper file from "QED24.COM" (or "QED25.COM") to "QED.COM".

Note the following:

"filename".....signifies a legal CP/M(r) or TPM(tm) filename, 8 characters long.  
 "typ".....signifies a legal file type of 1 to 3 characters long.  
 R.....this is the only command which is not very mnemonic, it means: cReate a new file.  
 N.....this means do Not make a backup file.

To edit a NEW file:

A>QED filename.typ R

This command cReates a new file with the name given.

To edit a previously created file with NO backup file:

A>QED filename.typ N

This command edits the file given and does NOT create a backup.

To edit a previously created file WITH BACKUP FILE:

A>QED filename.typ

This command generates a file called "filename.OLD" which is a duplicate of "filename.typ" before any editing is done on it. If "filename.OLD" already exists from a previous edit, it will be replaced by the new "filename.OLD".

If you leave off the file name:

A>QED

The editor prompts for the file name:

NAME?

Use the same formats as above:

NAME?  
filename.typ R

This cReates a new file with the name and type specified, and

NAME?  
filename.typ N



edits an existing file with NO backup, and

NAME?  
filename.typ

edits an existing file while creating/updating a backup file, "filename.OLD".

POSSIBLE ERRORS

The following errors may occur:

- |                          |   |
|--------------------------|---|
| ERROR C7                 | File not found  |
| ERROR D4                 | File already exists (this means you are trying to cReate a file that already exists). |
| ERROR O2                 | No space left on disk.  |
| ERROR O4                 | Illegal file name or type.  |
| ERROR 1E<br>FILE TOO BIG | This means there is not enough available memory to fit the file.                      |

## HOW TO EXIT FROM QED

---

There are two ways to exit from QED:

QU

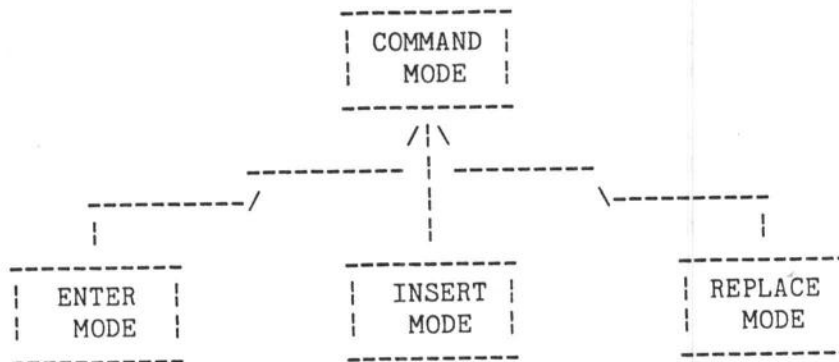
Exits from QED saving the changes that have been made during the edit session, and:

QU N

Exits from QED but NOT saving the changes. That is, the filename entered is left unchanged. If you enter QED with the "N" option (No backup) and use the "QU N" command to exit, no changes are made to any file. If you enter QED without the "N" option, a backup file is created and exiting with "QU N" does NOT affect the backup file. Therefore, if you enter without the "N" option and exit with the "N" option, you have two files, "filename.typ" and "filename.OLD" which are identical. This can be handy in creating a copy of a text file. However, you must rename "filename.OLD" so that it does not get overwritten the next time QED is entered without the "N" option.

THERE ARE 4 MODES  
-----

QED is entered in the COMMAND mode. This is also the primary mode of operation. There are three other modes which can be entered from the command mode. They are ENTER, INSERT, and REPLACE and are referred to as the secondary modes. The secondary modes may only be entered from the COMMAND mode.



COMMAND MODE: While in this mode, the following operations may be performed:

- Cursor movement.
- String searching and changing.
- Deleting lines.
- Setting, moving, copying, and erasing blocks.
- Combining files into the one being edited.
- Re-executing last command.
- Joining two lines into one.
- Entering other modes.
- Exiting the editor.

ENTER MODE: This is the primary text entry mode. All characters typed in this mode is entered into the text with the following exceptions:

- Backspace erases the character to the left of the cursor.
- Escape exits to the COMMAND MODE.

INSERT MODE: This mode is used for inserting characters into the an existing line of text. All characters typed in this mode are entered into the text at the cursor position, moving existing text to the right. The following characters are exceptions:

--- Backspace erases the character to the left of the cursor.

--- DEL (delete-7FH) erases the character to the right of the cursor.

--- Escape OR Return exits to the COMMAND MODE.

REPLACE MODE: All characters entered overwrite the existing characters at the cursor position. The following are exceptions:

--- Backspace moves the to the left WITHOUT erasing any characters.

--- DEL (delete-7FH) erases the character to the right of the cursor.

--- Escape OR Return exits to the COMMAND MODE.

## COMMAND MODE, ELEMENTARY COMMANDS

---

When QED signs on, it is in the command mode. The top line of the screen is the command line and the rest of the screen is for text display. When the other modes are entered, the name of the mode is displayed on the right side of the command (top) line. For example, in the ENTER mode the word ENTER is displayed. The command mode is in effect whenever no mode is displayed.

There are two types of commands in the command mode:

- Immediate commands (single key)
- Top line commands (multi-key)

IMMEDIATE COMMANDS are those which are executed with a single key press. These consist of the basic cursor movement commands:

KEY	FUNCTION
---	-----
^	Moves cursor UP one line.
LF (0AH)	Moves cursor DOWN one line (linefeed= 0A hex).
SPACE	Moves cursor one space to the RIGHT.
Backspace(08H)	Moves cursor one space to the LEFT.
TAB (09H)	Moves cursor to the next tab stop (there is a tab stop every 8 spaces).
DEL (07FH)	Moves cursor to the previous tab stop.
8	Moves cursor UP one line.
2	Moves cursor DOWN one line (linefeed= 0A hex).
6	Moves cursor one space to the RIGHT.
4	Moves cursor one space to the LEFT.
7	Moves cursor to UPPER LEFT of screen (HOME position).
1	Moves cursor to LOWER LEFT of screen.

Note that if your terminal has a numeric key pad, the last six key commands form a cursor key pad where the position of the key on the pad is analogous to the direction of movement.

The four cursor direction keys are duplicated on the regular keyboard (first four keys in the above list) so that touch typists do not have to move their hands out of position to move the cursor.

In addition to the cursor movement commands, the commands used to enter the other modes are also immediate:

KEY	FUNCTION
---	-----
E	Enters ENTER mode.
I	Enters INSERT mode.
R	Enters Replace mode.

Once in these modes, you may re-enter the command mode by pressing the ESC (escape-01B hex) key.

TOP LINE COMMANDS are those commands which require more than one key press. When you type a key that is not an immediate command but is the first key in a top line command, that key is displayed on the left side of the command line. You may then type other keys but the whole command line is not evaluated until you press RETURN. Then, the command is executed if possible. If it is not possible to execute the command, three question marks are displayed on the right side of the command line. QED tries to execute as much of the command line as possible; so, evaluating from left to right, it performs the command sequence until it hits an invalid part and disregards the rest.

T	Moves cursor to TOP of text file.
BO	Moves cursor to BOTTOM of text file.
N	Moves cursor to the Next screen.
Nx	Moves cursor x screens forward, where x is a number.
P	Moves cursor to the Previous screen.
Px	Moves cursor x screens back, where x is a number.
X	EXecutes the last top line command (displayed on the left side of the command line.
DE	Deletes the line at the cursor.
DEx	Deletes the next x lines starting at the cursor.
QU	QUits the editor, saving the changes.
QU N	QUits the editor. Does NOT change the edit file.

ENTER MODE  
-----

Upon entering this mode, if the cursor is at the beginning or middle of a line of text, a carriage return/linefeed (newline) sequence is inserted. This means that all text at and to the right of the cursor is moved to a new line which is inserted below the line the cursor is on. If this command is entered accidentally, the line may be restored by pressing escape to get back to the command mode and then pressing the J (join) command to reconnect the two lines. Also, if a carriage return is pressed, it is removed with a backspace (while still in the ENTER mode). The word ENTER is displayed on the right side of the command line while in this mode.

If, when entering this mode, the cursor is at or to the right of the end of the line, no new line is created until a RETURN is pressed. Of course, text may be entered before the RETURN is pressed. Also, if the cursor was to the right of the end of line, it is moved to the real end of line first. While in this mode all characters typed are entered into the text as-is with the following exceptions:

KEY	FUNCTION
---	-----
Backspace	Erases the character to the left of the cursor, including returns.
ESC	Exits to the command mode.
^Z	DO NOT ENTER THIS CHARACTER. If you do, it enters into the text and is taken as the end of file. If accidentally hit, it is erased with the backspace, which automatically restores the line to its previous length.
^K, ^L	These are not entered as-is. Do not use them.

INSERT MODE

If the cursor is to the right of the end of a line when entering this mode, the cursor is moved to the end of the line. Otherwise the only visible effect is that the word "INSERT" appears on the right side of the command line. After entry, all characters typed are entered as-is and existing text shifted right. Exceptions are noted in the table below.

KEY	FUNCTION
---	-----
Backspace	Erases the character to the left of the cursor.
DEL	Erases the character at the cursor.
RETURN	Exits to the command mode.
ESC	Exits to the command mode.
^Z	DO NOT ENTER THIS CHARACTER. If you do, it enters into the text and is taken as the end of file. If accidentally hit, it is erased with the backspace, which automatically restores the line to its previous length.
^K, ^L	These are not entered as-is. Do not use them.



REPLACE MODE  
-----

Upon entry, this mode looks similar to the INSERT mode with the exception that the REPLACE appears on the right side of the command line. However, when you type, all characters replace the ones which are under the cursor. The exceptions are listed below. Note that a backspace only moves the cursor back through the text without erasing characters. Characters are erased by either overwriting or deleting (DEL) them.

KEY	FUNCTION
---	-----
Backspace	Moves the cursor one space to the left WITHOUT erasing any characters.
DEL	Erases the character at the cursor.
RETURN	Exits to the command mode.
ESC	Exits to the command mode.
^Z	DO NOT ENTER THIS CHARACTER. If you do, it enters into the text and is taken as the end of file. If accidentally hit, it is erased with the backspace, which automatically restores the line to its previous length.
^K, ^L	These are not entered as-is. Do not use them.

COMMAND MODE, ADVANCED TOP LINE COMMANDS

---

J Joins the line at the cursor with the next line.

L/string Locates the string. "/" may be any character not in the string (same for following two commands).  
Uses Control-Rs to represent carriage returns in the string (only in this command).

F/string Finds the string but only if it occurs at the beginning of the line. This is handy for finding labels in programs.

C/str1/str2/nm Changes string 1 (str1) to string 2 (str2), starting search at the cursor position. If n='N', changes next occurrence of string 1 and ignores m. If n is a number, it does the changes on the next n lines, starting at the current cursor position. If n='\*', it does all lines. If n is not given, it does only the current line. If m='\*', changes all occurrences of string 1 on each line. A null string 1 (for example, "C//str2/") places string 2 at the beginning of the line.

M Mid. Moves the TEXT at the cursor position so it is in the Middle of the screen. DOES NOT MOVE the cursor on the text.

Fx Moves the text Forward x number of lines. Keeps the cursor at the same text location unless that location is off screen. If x is not given, places the text at the cursor position at the top of the screen.

Bx Moves the text Backward x number of lines. This is the reverse of the F command. If x is not given, the text is placed at the bottom of the screen.

T1 Tags the START of a block of text with the line the cursor is at. If there is no T2 tag, then this line is the entire block of text.

T2 Tags the END of a block of text with the line ABOVE the one the cursor is at.

O Takes the tags OFF (T1 and T2).

O2 Takes tag 2 OFF (T2 only).

CO Copies the block of text (from T1 to T2) to the current cursor position.

CO filename Copies the block of text to a file.

MO Moves the block of text from where it is to the current cursor position.

MO filename Moves the block of text to a file, removing it from the file being edited.

DET Deletes the block of text between the Tags.

GE filename Gets the file and place it at the cursor.

S Shows the RETURNS for all the lines on the screen.

ST Shows the TABS for all the text on the screen.

;  
Separates commands so that more than one may be placed on a line.

W Waits for any key press from the user before continuing. This is used as a pause for multiple commands per line.

## COMMAND SUMMARY

---

^	Moves cursor UP one line.
LF (0AH)	Moves cursor DOWN one line (linefeed= 0A hex).
SPACE	Moves cursor one space to the RIGHT.
Backspace(08H)	Moves cursor one space to the LEFT.
TAB (09H)	Moves cursor to the next tab stop (there is a tab stop every 8 spaces).
DEL (07FH)	Moves cursor to the previous tab stop.
8	Moves cursor UP one line.
2	Moves cursor DOWN one line (linefeed= 0A hex).
6	Moves cursor one space to the RIGHT.
4	Moves cursor one space to the LEFT.
7	Moves cursor to UPPER LEFT of screen (HOME position).
1	Moves cursor to LOWER LEFT of screen.
T	Moves cursor to TOP of text file.
BO	Moves cursor to BOTTOM of text file.
N	Moves cursor to the Next screen.
Nx	Moves cursor x screens forward, where x is a number.
P	Moves cursor to the Previous screen.
Px	Moves cursor x screens back, where x is a number.
X	EXecutes the last top line command (displayed on the left side of the command line).
DE	Deletes the line at the cursor.
DEx	Deletes the next x lines starting at the cursor.
QU	QUits the editor, saving the changes.
QU N	QUits the editor. Does NOT change the edit file.
E	Enters the ENTER mode.
I	Enters the INSERT mode.
R	Enters the REPLACE mode.
ESC	Exits from the above modes to the command mode or, if entering a top line command (before the RETURN), aborts back to the text.
J	Joins the line at the cursor with the next line.
L/string	Locates the string. "/" may be any character not in the string (same for following two commands). Uses Control-Rs to represent carriage returns in the string (only in this command).
F/string	Finds the string but only if it occurs at the beginning of the line. This is handy for finding labels in programs.
C/str1/str2/nm	Changes string 1 (str1) to string 2 (str2), starting search at the cursor position. If n='N', changes next occurrence of string 1 and ignores m. If n is a number, it does the changes on the next n lines, starting at the current cursor position. If n='*', it does all lines. If n is not given, it does only the current line. If m='*', changes all occurrences of

string 1 on each line. A null string 1 (for example, "C//str2/") places string 2 at the beginning of the line.

M Mid. Moves the TEXT at the cursor position so it is in the Middle of the screen. DOES NOT MOVE the cursor on the text.

Fx Moves the text Forward x number of lines. Keeps the cursor at the same text location unless that location is off screen. If x is not given, places the text at the cursor position at the top of the screen.

Bx Moves the text Backward x number of lines. This is the reverse of the F command. If x is not given, the text is placed at the bottom of the screen.

T1 Tags the START of a block of text with the line the cursor is at. If there is no T2 tag, then this line is the entire block of text.

T2 Tags the END of a block of text with the line ABOVE the one the cursor is at.

O Takes the tags OFF (T1 and T2).

O2 Takes tag 2 OFF (T2 only).

CO Copies the block of text (from T1 to T2) to the current cursor position.

CO filename Copies the block of text to a file.

MO Moves the block of text from where it is to the current cursor position.

MO filename Moves the block of text to a file, removing it from the file being edited.

DET Deletes the block of text between the Tags.

GE filename Gets the file and place it at the cursor.

S Shows the RETURNS for all the lines on the screen.

ST Shows the TABS for all the text on the screen.

; Separates commands so that more than one may be placed on a line.

W Waits for any key press from the user before continuing. This is used as a pause for multiple commands per line.