

TECO Summary¹

Teco is a character oriented editor. This differs from *ed* which is line oriented, and *vi* which is screen oriented. Teco deals with one character at a time and makes only little distinction between one line and another. Therefore, at any given time, the current location teco sees is a character position and not a line.

Interestingly, the original emacs editor was written as a series of teco macros.

In what follows the \$ character is the Escape key. \$ is used to separate one command from another. \$\$ is used to cause teco to execute the entire command. Therefore, teco doesn't actually do anything until \$\$ is hit.

Even though the command are shown in capitol letters, one would normally use lower case letters. They are shown in upper case to avoid confusion (l vs. L, etc.).

\$	Escape key - command delimiter
\$\$	Two escape keys - execute string
n	Some number (implied default is usually 1)
file	The name of a file
text / find / replace	Arbitrary text
q	A named storage area called a q-register. Use any character a-z and 0-9 to name the register.

Exiting

EX\$\$	save & exit
-1EX\$\$	Forced exit without saving

File commands

EC	Save, close, and clear current buffer
EKHK	Abort edit
EBfile\$	Open file
Y	Read file

Displaying text (not for Video Teco)

T	Type current character to EOL
HT	Type entire file
nT	Type n lines starting at current character
V	View entire line
nV	View n lines around current line

¹Blake McBride (blake@mcbride.name), rev 6/6/2015

Cursor movement commands

Macro	Command	Description
	nC	Move n characters forward
	nR	Move n characters reverse
	nL	Move n lines forward (to the beginning of that line)
nBW		Go to beginning of n'th next (previous if negative) word
nEW		Go to end of n'th next (previous if negative) word
BL	0L	Move to beginning of line
EL	L-2C	Move to end of line
nJMP		Go to line n
	J	Beginning of buffer
	ZJ	End of buffer
	nStext\$	Search forward for the n'th occurrence of text (leaves point at end of text)
	S\$	Search again
	-nStext\$	Search backwards for text
	nES	Scroll screen n lines

Entering text

Itext\$	Insert text
nFSfind\$replace\$	Find and replace n'th occurrence of text

Deleting text

Macro	Command	Description
	nD	Delete n characters
nDW		Delete n following (previous if negative) words
DLR		Delete line remainder
	K	Delete from current position to beginning of next line
	0K	Delete from current position to beginning of the line
	0KK	Delete entire line
	0KnK	Delete n lines from current line
	HK	Delete entire buffer
	FDtext\$	Find & delete text
	FKtext\$	Delete from point to beginning of text
	FRtext\$	Replace last found text with text

Variables

B	Zero
Z	Number of characters in file
H	B,Z
.	Current position
x=	Display value of variable x

Cut & Paste

nXq	Copy text from current position and the next n lines into q-register
n.mXq	Copy text from character positions n to m into q-register q
.,+5Xq	Copy following 5 characters into q-register q
Gq	Insert contents of q-register into current position (paste)
E%qFILE\$	Save q-register to file FILE
EQqFILE\$	Read file contents into q-register (good for insert file)

Iteration

n<CMD>	Execute CMD n times
--------	---------------------

Macros

One can put a series of teco commands in a file and execute that command file while editing a file. This is one way teco macros are created. Within one of these command files the \$ character acts like an escape key. Once command files are created they can be executed as follows:

Efile\$	Execute command file when the command line is complete
Efile\$\$	Execute command file immediately

Q-Registers

Commands, text, and contents of files can also be put into and executed from q-registers. These are in-memory registers named with a single character a-z or 0-9. The following operations are supported:

^UqCMD\$	Put CMD into q-register q (note ^U are two characters, not ctl-U)
EQqFILE\$	Put contents of file FILE into q-register q
Mq	Execute q-register q
E%qFILE\$	Save contents of q-register q to file FILE
:Gq	Display contents of q-register q