

TECO

pocket guide

digital

NOTATION

TECO-6	refers to DEC TECO V6
TECO-10	refers to TOPS-10 TECO V24
TECO-11	refers to TECO-11 V28
RSX-11	refers to RSX-11-M, RSX-11-M, IAS and VAX
\$	represents an ESCAPE/ASCII code 30, which is functionally referred to as ALT/ODF
$\textcircled{*}$	represents a single control character (CONTROL)
x	represents either $\textcircled{}$ or the text character sequence CARET x ($\textcircled{*}$ x)
n	represents a numeric value specified by a digit string or TECO command that returns a value

Color Coding

black	represents features of all DEC TECOs [exceptions noted in brackets]
red	represents features of TECO-10 only
blue	represents features of TECO-11 only

Order No. AV-D533A-TK

Copyright © 1978 by Digital Equipment Corporation

INVOKING TECO

Function Initial Commands

USE

R TECO

RUN dev:TECO

MAKE file:src

TECO file:src

TECO file:src1 file:src2

TECO

MUNG file:src

MUNG file:src;text

EW file:src

ER file:src\$Y

ER file:src:\$EW file:src1\$Y

[note 1]

ER file:src\$YHX YHKMY

ER file:src\$YHX YHKMY

RT 11

R TECO

RUN dev:TECO

EQ T/TECO file:src

LU file:src\$Y [note 2]

Switches on CD/T Command

AC/DATE

INSPECT

AC/OUTPUT file:src2

EXECUTE

TEXT/ file:src

EW file:src\$Y

ER file:src\$Y

ER file:src:\$EW file:src2\$Y

ER file:src\$YHX YHKMY

ER file:src\$YHX YHKMY

RSTS/E

TECO

[note 1]

RUN \$TECO

RUN dev:TECO (p.p.)

TECO file:src

TECO file:src1 file:src2

TECO/INSPECT file:src

MAKE file:src

MUNG file:src

MUNG file:src;text

ER file:src\$Y

ER file:src2\$Y file:src1\$Y

ER file:src\$Y

EW file:src\$Y

ER file:src\$Y

ER file:src\$Y

RSX 11

TECO

TECO file:src

TECO file:src1 file:src2

MAKE file:src

MUNG file:src

MUNG file:src;text

LU file:src\$Y

LU file:src2\$Y file:src1\$Y

LU file:src\$Y

LU file:src\$Y

LU file:src\$Y

TOPS-10

R TECO

RUN dev:TECO (p.p.)

MAKE file:src

TECO file:src

TECO file:src1 file:src2

MAKE file:src1 file:src2

TECO

MAKE

[note 3]

EW file:src\$Y

ER file:src\$Y

ER file:src1\$Y

ER file:src2\$Y file:src1\$Y

[note 4]

[note 4]

IMMEDIATE MODE EDITING COMMANDS

\$\$	Executes command string
Ctrl+u	Deletes last character in command string
(C)	Exits from TCCO [note 6]
(H)	Deletes next 1 line in command string
(G) G	Deletes the entire command string
(S) <space>	Types current line of command string
(S)	Types entire command string [not in TCCO-10]

IMMEDIATE MODE AIDS

You must type these immediate mode commands as the first command after TCCO's prompt.

- ?** Types command string and a command that caused an error [note 5]
- /** Types detailed explanation of error
- q** Stores previous command string in Register 0 [note 7] [not in TCCO-6]

EXECUTION-TIME COMMANDS

- (C)** Stops command execution [note 8]
- (C)** Turns off type out; nullifies execution **(O)**
- (S)** Freezes terminal output; nullifies execution **(O)** [note 9]

FILE SPECIFICATION COMMANDS

- ERf filepec\$** Opens file for input
- EWfilepec\$** Opens file for output [note 12]
- EBfilepec\$** Opens file for input and output with backup protection
- ERfilepec\$** Zeros (in on-disk) directory and opens file for output

nEM	Primitive/Action: EM Rewinds 1EM Rewinds 3EM Writes EOF 6EM Skips one record 7EM Backspaces one record 8EM Skips to EOF 9EM Rewinds and unbars 11EM Writes 3 inches blank tape 14EM Advances one file 16EM Backspaces one file
ENFile:pat\$	Search wildcard file specification (not in RT-11)
LN\$	Gets next wildcard match into file specification (not in RT-11)
ERFile:pat\$	Same as ERFile:pat\$ but returns value: -1 if file found, 0 if file not found
EBFile:pat\$	Same as EBFile:pat\$ but returns value: 1 if file found, 0 if file not found
FN\$	Same as LN\$ but returns value: 1 if next file found, 0 if no next file match

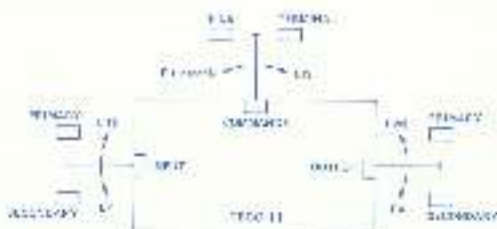
FILE TERMINATION COMMANDS AND EXIT

EF	Closes the output file
EC	Moves the remainder of the input file to the output file, then closes the output file (not in TECO-10)
EX	Same as LC but exits from TECO
EK	Kills the output file (not in TECO-10)
EGS	Same as EK but re-issues the last compile class command (not in 11)
EOcommand\$	Same as EX, but executes the operating system command specified (not in TECO-10)
IC	Exits from TECO (note 12) (not in TECO-10)
IZ	Closes output file and exits from TECO

AUXILIARY FILE SUPPORT (not in RT-11)

ISet - used in linked

IV	Switches to secondary input stream
IPV	Switches to primary input stream
OV	Switches to secondary output stream
OPV	Switches to primary output stream
EFile:pat\$	Opens file for command input; effective at end of current command string
LS	Switches command input to terminal and discards remainder of command for the current command string



FILE SPECIFICATIONS (Table 3)

System	Input	Output
CPM	new filename	destination
TCPS-10	new filename [ppp]	destination, [ppp], [ppp] or [ppp]
TT-10	new filename	destination, and on
RSRT	new [n,pp] filename	dest [p,pp] name, dest, pp, pp
RSX-11	new [n,pp] filename, [pp]	dest [p,pp] name, dest, [pp]

FILE SPECIFICATION SWITCHES

TCPS-10	/SENLIN	Generates line sequence numbers
	/SUPLIN	Suppresses line sequence numbers
	/NOFFin	Discontinues open mode
	/CLUSTER SIZE: n	Specifies output file cluster size
	/B+ /B2 /n	Edits a BASIC-PLUS file Edits a BASIC-PLUS 2 file (same as /B2) Handles BASIC-PLUS continuation character (&) in columns
RSX-11	/NO	Reverts before opening file (master only)
	/CR	Implied carriage control
	/-CR	No internal carriage control
	/F	FORTRAN carriage control

INVOCATION SWITCHES

HSTSA	/SIZE:n	starts with nK word editing area
	/SIZE:fn	starts with extra nK words of editing area
RSX-11	/NO: n	starts with extra nK words of editing area

PAGE MANIPULATION COMMANDS

A	Appends new page of input file to text buffer
LA	Appends next line of input file to text buffer
Y	Same as HKA
PF	Outputs current page and appends a term feed
nPF	Same as n-1 PF
m,nPF	Outputs the characters between text buffer term feed positions m and n (without a term feed)
F	Outputs the text buffer, clears the buffer, and reads the next page of the input file into the buffer; a term feed is appended to the output file if the last page was terminated with a term feed
nF	Same as n-1 PF
m,nF	Same as m,nPF

BUFFER POINTER POSITIONING COMMANDS

n	Moves pointer to immediately following the nth character in the text buffer
O	Moves pointer to beginning of buffer
.	Same as O
nC	Advances pointer n positions
C	Same as 1C
-C	Same as -1C
nB	Backs up pointer n positions (same as -nC)

R	Same as TR
-R	Same as -TR
nL	Moves pointer to beginning of nth line from current pointer position
L	Same as 1L
-L	Same as -1L

INSERTION COMMANDS

The pointer's position relative to the inserted text.

Insert\$	Inserts the text specified
nl\$	Inserts character whose ASCII code is n
character\$	Inserts specified text preceded by a tab
hex\$	Inserts specified text preceded by a tab
n	Inserts radix representation of n
	into buffer (signed if current radix is decimal, unsigned if not). (note 10)
FRhex\$	Same as 1\$Dhex\$

ARITHMETIC OPERATORS

Operators are evaluated from left to right.

n	Same as n
-n	Negation of n
n+n	Addition (note 14)
m-n	Subtraction
m*n	Multiplication
n/n	Division, remainder discarded
n&n	Bitwise Inclusive AND
n&n	Bitwise Inclusive OR
!n	Performs one's complement operation
~n	Two's complement of n

ARITHMETIC PRECISION

TECO-9	$1/2^9 \leq \text{error} < 1/2^8$
TECO-11	$1/2^{11} \leq \text{error} < 1/2^{10}$
TECO-16	$1/2^{16} \leq \text{error} < 1/2^{15}$

TYPE-OUT COMMANDS

nT	Types out all the text from the current pointer position to the beginning of the nth line from the current pointer position
T	Same as 1T
-T	Same as -1T
mT	Types the characters between pointer and line m and n
nV	Same as 1V and 1nT
V	Same as 0TT
1Atext	Types the text specified
n=	Types value of n in decimal (signed) followed by <CR> <LF>
n--	Types value of n in octal (note 15) followed by <CR> <LF>
m=	Types value of m in decimal (signed) [not in TECO-10]
m--	Types value of m in octal (unsigned) [not in TECO-10]
nT	Types out the character whose ASCII code is n [not in TECO-10]
-L	Types a line of text [not in TECO-10]

DELETION COMMANDS

nD	Delete n characters following the current pointer position
-nD	Delete n characters preceding the current pointer position
D	Same as 1D
-D	Same as -1D
nDk	Same as nDk
nK	Delete all characters in the text buffer from the current pointer position to the beginning of the nth line from the current pointer position.
K	Same as 1K
-K	Same as -1K
m.nK	Delete all characters in the text buffer between pointer position m and n; the pointer moves to position n

BUFFER POSITIONS

B	First valid position, same as 0
Z	Last pointer position; equal to number of characters in text buffer
.	Current pointer position, equal to number of characters to the left of the pointer
nH	Just left of the nth character in the text buffer
H	Same as B, Z
1H	Same as -1H,

SEARCH COMMANDS

Successful search leaves the pointer at position 0.
Successful search leaves pointer where text found

nStext\$	Searches for the nth occurrence of the specified text following the pointer, but does not go beyond the end of the current text buffer
Stext\$	Same as 1Stext\$
-Stext\$	Same as nStext\$ but the search proceeds backwards
nNtext\$	Same as nStext\$, except that if the text is not found on the current page, pages are input and output until it is found
Ntext\$	Same as 1Ntext\$
n_text\$	Same as nNtext\$ except that it does input only, no output
_text\$	Same as 1_text\$
nFStext1\$text2\$	Executes the nStext1\$ command, then deletes text1 and replaces it with text2
FStext1\$text2\$	Same as 1FStext1\$text2\$
nIStext1\$text2\$	Executes the nNtext1\$ command, then deletes text1 and replaces it with text2
IStext1\$text2\$	Same as 1IStext1\$text2\$
nF_text1\$text2\$	Executes the n_text1\$ command, then deletes text1 and replaces it with text2
F_text1\$text2\$	Same as 1F_text1\$text2\$

BOUNDED SEARCH

Bounded searches that fail leave the pointer unchanged

- m_StartS Same as $StartS$, but the search does not occur in a range more than $m-1$ characters.
- J_StartS Same as $StartS$, but if the pointer position remains unchanged on search string failure
- m_StartS Same as m_StartS
- m_StartS Same as m_StartS
- $StartS$ Same as $StartS$ (unchanged search)

COLON MODIFIED SEARCHES

Any search command preceded by a colon (:) modifier returns a value as follows:

- 1 If the search succeeds
- 0 If the search fails

NUMERIC QUANTITIES

- Cn ASCII code for character following pointer
- nA ASCII code for the n th character in the buffer (note 16) [see in TPOO-10]
- n Equal to the value of the digit string (if any) that follows the pointer in the text buffer (interpreted in the current radix "note 13", and optionally signed); pointer is positioned after the digit string. Fails on 0 if no digit string is present
- $1B$ Operating system dependent encoding of current date
 $RT-11$: $(\{month\} * 32 + \{day\}) * 32 + \{year\} - 1972$
 $RSTS-E$: $(\{year - 1970\} * 1000 + \{day\} + \{hour\} * 60 + \{min\} * 60 + \{sec\})$
 $RX-11$: $(\{year - 1900\} * 15 + \{month\} * 32 + \{day\})$
- $1F$ Value of console switch register
- $1H$ Operating system dependent encoding of time of day
 $COS-0$: 0
 $RT-11$: 0
 $RSTS-E$: minutes until midnight
 $RX-11$: seconds since midnight
 $TOS-10$: 90ths of a second since midnight
- nG Number of characters stored in G register
- nO Number of characters between the buffer pointer and the n th line separator from the current pointer position
- $1O$ Same as $1O$
- $1S$ Negative of length of last insert, string found, or string inserted with G command, or character assumed hex
- $1T$ ASCII code for the next character typed in the terminal
- $1X$ ASCII code for the character (X) specified

ITERATION AND FLOW CONTROL

n-Commands	Performs the subcommand string n times; skips the commands if n=0
x-commands	Performs the subcommand string x number of times
n;	If n=0, jumps out of the current iteration field
:	Jumps out of the current iteration field if the last search executed failed [note 1/]
tag!	Defines a position in the command string with the name, tag
Jump\$	Jumps to the position defined by tag; must be in same macro-level [note 13]

Ⓢ MODIFIER

- Ⓢ Modifies the next command that takes a text argument to use the delimiter form; text is surrounded by a delimiter (shown here as /) that cannot appear inside the text

The format of the commands becomes:

```

Ⓢn$ /text/
ⓈnR /text/
Ⓢn_ /text/
ⓈnFR /text1/text2/
ⓈnFN /text1/text2/
Ⓢ /text/
Ⓢn /
ⓈLH /filepec/ [not in TECO-10]
ⓈFW /filepec/ [not in TECO-10]
ⓈFB /filepec/ [not in ILCO-10]
ⓈLJ /text/ [not in IFCO-10]
ⓈEG /text/ [not in TECO-10]
ⓈLUG/ [not in TECO-10]
ⓈEN /filepec/
ⓈEL /filepec/
ⓈFR /text/
ⓈIA /text/
Ⓢ /tag/
ⓈC/tag/
ⓈnUq/
  
```

CONDITIONAL COMMANDS

n" /commands	Executes the commands if n is the ASCII code for an alphabetic character (A-Z or a-z) [not in TECO-8]
n' /commands	Executes the commands if n is the ASCII code for a decimal digit (0-9) TECO-8: A-Z, a-z, 0-9 TECO-11: A-Z, a-z, 0-9, ., -, /, % TECO-10: A-Z, a-z, 0-9, ., -, /, %
n# /commands	Executes the commands if n is the ASCII code for a digit (0-9) [not in TECO-8]
n" /commands	Executes the commands if n=0
n' /commands	Executes the commands if n represents FALSE; same as n" /commands
n# /commands	Executes the commands if n>0
n' /commands	Executes the commands if n<0
n# /commands	Executes the commands if n=0
n' /commands	Executes the commands if n is the ASCII code for an alphanumeric (A-Z, a-z, 0-9) [not in TECO-10]

<code>n"Excommands"</code>	Executes the excommands if <code>n</code> represents SUCCESS; same as <code>n"Excommands"</code>
<code>n"Excommands"</code>	Executes the commands if <code>n</code> represents TRUE; same as <code>n"Excommands"</code>
<code>n"Uexcommands"</code>	Executes the excommands if <code>n</code> represents UNSUCCESSFUL; same as <code>n"Excommands"</code>
<code>n"Vexcommands"</code>	Executes the commands if <code>n</code> is the ASCII code for a lower case alphabetic (a-z) [not in TECO-8]
<code>n"Xexcommands"</code>	Executes the excommands if <code>n</code> is the ASCII code for an upper case alphabetic (A-Z) [not in TECO-8]
<code>n"Excommands"</code>	Same as <code>n"Excommands"</code> [not in TECO-10]
<code>n"Excommands"</code>	Same as <code>n"Excommands"</code> [not in TECO-10]

Q-REGISTER LOADING [page 15]

<code>nUc</code>	Stores integer <code>n</code> in Q-register <code>c</code>
<code>Uq</code>	Adds 1 to the number in Q-register <code>q</code> and returns this value
<code>nUq</code>	Adds <code>n</code> to the number in Q-register <code>q</code> and returns this value [not in TECO-10]
<code>nXc</code>	Stores, in Q-register <code>c</code> , all characters from the current pointer position to the beginning of the <code>n</code> th line from the current pointer position
<code>m,nXq</code>	Stores, in Q-register <code>c</code> , all characters between text buffer pointer positions <code>m</code> and <code>n</code>
<code>Xc</code>	Same as <code>1Xq</code>
<code>Xq</code>	Same as <code>1Xq</code>
<code>nXq</code>	Same as <code>1Xq</code> , but appends to Q-register
<code>m,nXq</code>	Same as <code>m,nXq</code> , but appends to Q-register
<code>1Xq</code>	Same as <code>1Xq</code>
<code>1Xq</code>	Same as <code>1Xq</code>
<code>Uqtext\$</code>	Stores the specified text in Q-register <code>q</code> [not in TECO-10]
<code>nUqtext\$</code>	Same as <code>Uqtext\$</code> , but appends to Q-register
<code>nUq\$</code>	Stores character whose ASCII code is <code>n</code> in Q-register <code>q</code>
<code>mUq\$</code>	Appends character whose ASCII code is <code>m</code> to contents of Q-register <code>q</code>
<code>1q</code>	Pushes from the Q-register push down list into Q-register <code>q</code> [note 21] [not in TECO-8]
<code>1c</code>	Same as <code>1q</code> , but returns value <code>-1</code> if unsuccessful, <code>0</code> if push down stack was empty

Q-REGISTER RETRIEVAL [page 18]

<code>Qq</code>	Returns the numeric value stored in Q-register <code>q</code>
<code>Uq</code>	Places the text in Q-register <code>q</code> in to the text buffer before the current pointer position
<code>cU</code>	Types the contents of the text stored in Q-register <code>q</code> on the terminal [not in TECO-10]
<code>Wq</code>	Evaluates the text in Q-register <code>q</code> as a command string
<code>nWq</code>	Evaluates the text in Q-register <code>q</code> as a command string and uses <code>n</code> as a numeric argument for the first command executed

mult%	Executes the text in G-register as a command string and does not save any of number components for the final command executed
q	Pushes the contents of G-register onto the U-register push-down list (line + 2)
q⁺	Pastes the text from the T-register buffer into the text buffer before the current pointer position
q⁺	Types the contents of the T-register buffer on the terminal
q₊	Pastes the text from the search string buffer into the text buffer before the current pointer position
q₊	Types the contents of the search string buffer on the terminal

RETRIEVING NODE CONTROL FLAGS

Flag	Name	Initial Value
FE	Event feed flag	0
FN	End of file flag	0
FX	Search Mode flag	0
LC	edit level flag	0 [PT-11, 0 [FSTS-E] 1 [RJK-11]
EC	edit level flag	2 [TECO-10, 0 [other systems]
EC	Version number flag	2 [TECO-10] 3 [TECO-8] 20 [TECO-11]
ES	Search verification flag	0
ET	Terminal mode flag	

TECO-10:		0
TECO-8:		0 if SET TTY VDSOPE 2 if SET TTY SCOPE
TECO-11:		
Bit value		Initial Value
1	0	
2	1	if terminal is a scope
	0	if terminal is not a scope
4	1	if terminal has lowest code (null)
	0	if terminal does not have lowest code input
8	0	
16	0	
32	0	
64	0	
128	1	before TECO's prompt
	0	after ILU's prompt
256	0	
512	1	if terminal is VT52 and VT52 support is available
	0	if VT52 support is not available
1024	1	if VT11 is present and VT11 support is available
	0	if VT11 support is not available

LU	Upper/lower case flag	
	-1 if terminal supports lower case	
	0 if terminal does not support lower case [note 22]	
LV	Edit verify flag	0

SETTING MODE CONTROL FLAGS

Command	Bit Value	Meaning
rFD	1	0 Find search arguments means next character is CTRL
		1 Find search arguments means ?
	2	0 disables Y or ____ if data exists in list
		1 no Yank protection
rEE	1,2	0 error no 2
		1 types 3 and 4 error code only
		2 types error code and short message
		3 types error code and long message
	4	1 Simulates 2 after error
rED		Allows ED version of features to work

m *255+ ES

If m *255+ ES is typed out of a command line=1, an effective mV command is executed when a search is completed. If it is between 1 and 31, ASCII % represents the pointer; if it is between 32 and 128, it is ASCII code for character to represent the pointer; [TFC0-13] (parameter) - If m < 0, m is treated as 1.

mt	Set terminal character sets [note 23]
----	---------------------------------------

Bit Value	Meaning (T)
1	type-out is in image mode
2	terminal is a serial
4	needs lower case
8	Traces with no wait
16	traces (IC) on type-out
32	Traces with no wait
	TT returns 1 if terminal
64	detached flag
128	shorts on error
256	traces output to terminal with
512	terminal is VT32 and VT32 support is available
1024	VT11 is chosen and VT11 support is available
32768	Traces (IC) % (Base) if (IC) typed
rLU	0 flag error case characters
	>0 flag case and characters
	<0 no case flagging on output [note 22]

NEV		Same as NE5 but lines to be viewed appear before TECO's prompt
NTX	0	With no case matches in searches
	1	Exact case matches required [not in TECO-8]

RADIX CONTROL (note 13)

TD	Switches to octal radix [note 24]
TU	Switches to decimal radix [not in TECO-10]

TRACE CONTROL

T	Enters trace mode; nullified by a 2nd T
---	---

VT52 SCOPE CONTROL

-TW	Updates window into text buffer
OW	Resets cursor line to default position
nW	If n>0, sets cursor to line n If n< -1, "jumps" cursor to -1 line of screen

VT12, VT11, or VS80 SCOPE CONTROL

[not in TECO-10]

W	Updates window into text buffer
nW	Sets window display to n lines on either side of pointer

CASE CONTROL

TV	Enters translate to lower case mode
UW	Enters translate to upper case mode

MATCH CONTROL CONSTRUCTS

[Match control constructs are used in character strings]

(X)	Accepts any character
(S)	Accepts any non-alphanumeric separator [note 26]
(N)	Accepts any character except a character matched by the following character for match control constructs
(F)A	Accepts any alphabetic character [A-Z or a-z]
(F)G	Accepts any symbol character and TECO-11: A-Z, a-z, 0-9, _ \$
(F)D	Accepts any digit (0-9) [not in TECO-8]
(F)Gq	Accepts any character in G-register q
(F)L	Accepts any line terminator (LF, CR, CRLF) [not in TECO-8]
(F)H	Accepts any alphanumeric (A-Z, 0-9)
(F)S	Accepts any non-null string of spaces and/or tabs [not in TECO-8]
(F)V	Accepts any lower case alphabetic (a-z) [not in TECO-8]

(E) W	Accepts any upper case syllabic (A-Z; not in TECO-8)
(E) X	Same as (X)
(F) <nn>	Accepts the character whose ASCII code (in fact) is nn
(E) [abc...]	Accepts exclusive OR of the characters or match control constructs specified

SEARCH STRING BUILDING CHARACTERS

The following characters have special meaning when used inside a search string.

(C)	Use the next character literally (note 26)
(V)	Converts next character, if alphabetic, to lower case (not in TECO-8)
(W)	Converts next character, if alphabetic, to upper case (not in TECO-8)
(T)	Converts next character (if it is A, B, C, F, or G) to lower case
(R)	Uses the next character literally
(V) (V)	Converts all following alphabetic characters to lower case
(W) (W)	Converts all following alphabetic characters to upper case
(T)	Treats all succeeding control characters except (R) and (T) as text, nullified by a second (T)
(C)	Accepts either case for all succeeding alpha- betic characters; nullified by a second (C)
I	If EDBI = 0, converts the following alphabetic character to the corresponding control char- acter
(E) C_i	Uses the string stored in C _i -register at this position
(L) L_i	Uses the string stored in the filestore buffer at this position
(E) C_j	Uses the string stored in the search string buffer at this position

INSERT STRING BUILDING CHARACTERS

The following characters have special meaning when used inside insert strings on TECO-13. They have the same meaning that they have when used inside search strings.

(V)	(T)
(W)	(R)
(V) (V)	(T)
(W) (W)	

ERROR MESSAGES

Code	Explanation
?ARG	Improper arguments
?BAK	Cannot delete old backup file
?BNH	? not in an iteration
?CUI	Same as ?MFV
?CUI	OCI not found or EG argument too big
?COS	Can not store output switches
?CPD	Cannot load Q register
?DEV	Invalid device
?DIU	Directory I/O error
?DIB	Delete too big
?EBD	Bad EB device
?EBF	Bad EB filename
?EBO	Same as ?UFU
?EBP	EO illegal because of protection
?ELL	Unable to read error message file
?EMA	EM with illegal argument
?EMD	EM with no input device open
?ENT	Error error
?EUA	Joe argument to ?D
?EUD	ISDSF monitor not active
?ERR	File error
?ENF	File not found
?FUL	Output command would have overwritten device in ELCO-10, same as ?RPO
?IAD	Invalid reading of C, D, or E
?ICT	Illegal CTRL/E command in search argument
?ICT	Illegal control character in text argument
?IDV	Bad input device
?ILC	Illegal character after I
?ILM	Limit is line after FM
?IFC	Illegal character after F
?IFN	Bad filename
?IA	Illegal insert argument
?ILL	Illegal command
?IN	Same as ?OCT
?IB	Cannot backup input file to archive
?INF	Input error
?IOS	Illegal character in IO switch
?IQC	Illegal character after "
?IDR	Illegal Q-register name
?IRB	Cannot read input filename
?IRB	Failure in remote access
?ISA	Illegal search argument
?ISS	Illegal search string
?VAP	Missing "
?VCF	Macro ending with F
?VEF	Macro ending with F
?VEM	Storage capacity exceeded
?VED	Macro ending with D
?VLU	Macro ending with "
?VEH	Macro ending with H
?VFN	Missing filename
?VQ	Missing Q-register name
?VLA	Same as ?UBI
?VLP	Missing I
?VRA	Missing R
?VRP	Missing P
?VLU	Macro ending with H
?VAB	No argument before
?VAC	TECC-1: No argument before TECC-8: Negative argument in
?VAE	No argument before
?VAF	No alternate entered

ERROR MESSAGES (Cont.)

Code	Explanation
ZDAP	EDD-11: Same as ZM-1
ZDAP	EDD-8: Negative or 0 argument to ?
ZDAD	No argument to "end"
ZDAS	TECO-11: No argument before
ZDAS	TECO-8: Negative or 0 argument to search
ZDAU	No argument before U
ZDAY	Numeric argument with Y
ZDGS	No command string given to input
ZDFI	No file for input
ZDFD	No file for output
ZDGO	No number in G register
ZDIO	No room for output
ZDLO	No text in Q register
ZKEY	Command not yet implemented
ZOCT	Bit 0 - must digit string
ZODV	Output device not available
ZOFO	Output file already open
ZOFI	Cannot link in output file to rename it
ZOHF	Output error
ZPAR	Confirmed use of argument
ZPDO	Push down list overflow
ZPOP	Attempt to move pointer off page
ZPPH	Back PPH
ZOMO	Register memory overflow
ZONG	Cannot rename output file
ZGAL	Second argument less than first
ZGVA	Initial search with no argument
ZGVI	and in an iteration
ZSRH	Search failure
ZSTC	Search string has an empty character
ZSTL	Search string too long
ZTAG	Missing tag
ZTAL	Two arguments with L
ZTFL	Tag too long
ZTTY	Legal TTY device
ZUCA	Unterminated U A command
ZUEA	Unterminated EA
ZUEL	Unterminated EL
ZULN	Unterminated LN
ZULP	Unterminated LP
ZULS	Macro ending with unterminated file select
ZULN	Unterminated L user command
ZULS	Undefined FO switch
ZUSI	Unterminated search command
ZULI	Unterminated command
ZUTG	Unterminated tag
ZUTV	Unterminated value
ZUUC	Illegal UUC
ZULC	System device is not connected
ZKAB	Execution aborted by ^C
ZYCA	Y command executed
ZYFI	RSK-11 monitor call failed

OCTAL AND DECIMAL ASCII CODES

chr	OCT	DEC	chr	OCT	DEC
nul	000	000	@	100	084
ts	001	001	A	101	085
th	002	002	B	102	086
tc	003	003	C	103	087
td	004	004	D	104	088
te	005	005	E	105	089
tf	006	006	F	106	090
tg	007	007	G	107	091
th	010	008	H	110	092
ti	011	009	I	111	093
tj	012	010	J	112	094
tk	013	011	K	113	095
tl	014	012	L	114	096
tm	015	013	M	115	097
tn	016	014	N	116	098
to	017	015	O	117	099
tp	020	016	P	120	100
tq	021	017	Q	121	101
tr	022	018	R	122	102
ts	023	019	S	123	103
tt	024	020	T	124	104
tu	025	021	U	125	105
tv	026	022	V	126	106
tw	027	023	W	127	107
tx	030	024	X	130	108
ty	031	025	Y	131	109
tz	032	026	Z	132	110
0	033	027	.	135	101
1	034	028	.	134	102
2	035	029	.	135	103
3	036	030	.	136	104
4	037	031	.	137	105
5	040	032	.	140	106
6	041	033	.	141	107
7	042	034	.	142	108
8	043	035	.	143	109
9	044	036	.	144	100
:	045	037	.	145	101
;	046	038	.	146	102
<	047	039	.	147	103
=	050	040	.	150	104
>	051	041	.	151	105
?	052	042	.	152	106
@	053	043	.	153	107
A	054	044	.	154	108
B	055	045	.	155	109
C	056	046	.	156	110
D	057	047	.	157	111
E	060	048	.	160	112
F	061	049	.	161	113
G	062	050	.	162	114
H	063	051	.	163	115
I	064	052	.	164	116
J	065	053	.	165	117
K	066	054	.	166	118
L	067	055	.	167	119
M	070	056	.	170	120
N	071	057	.	171	121
O	072	058	.	172	122
P	073	059	.	173	123
Q	074	060	.	174	124
R	075	061	.	175	125
S	076	062	.	176	126
T	077	063	.	177	127

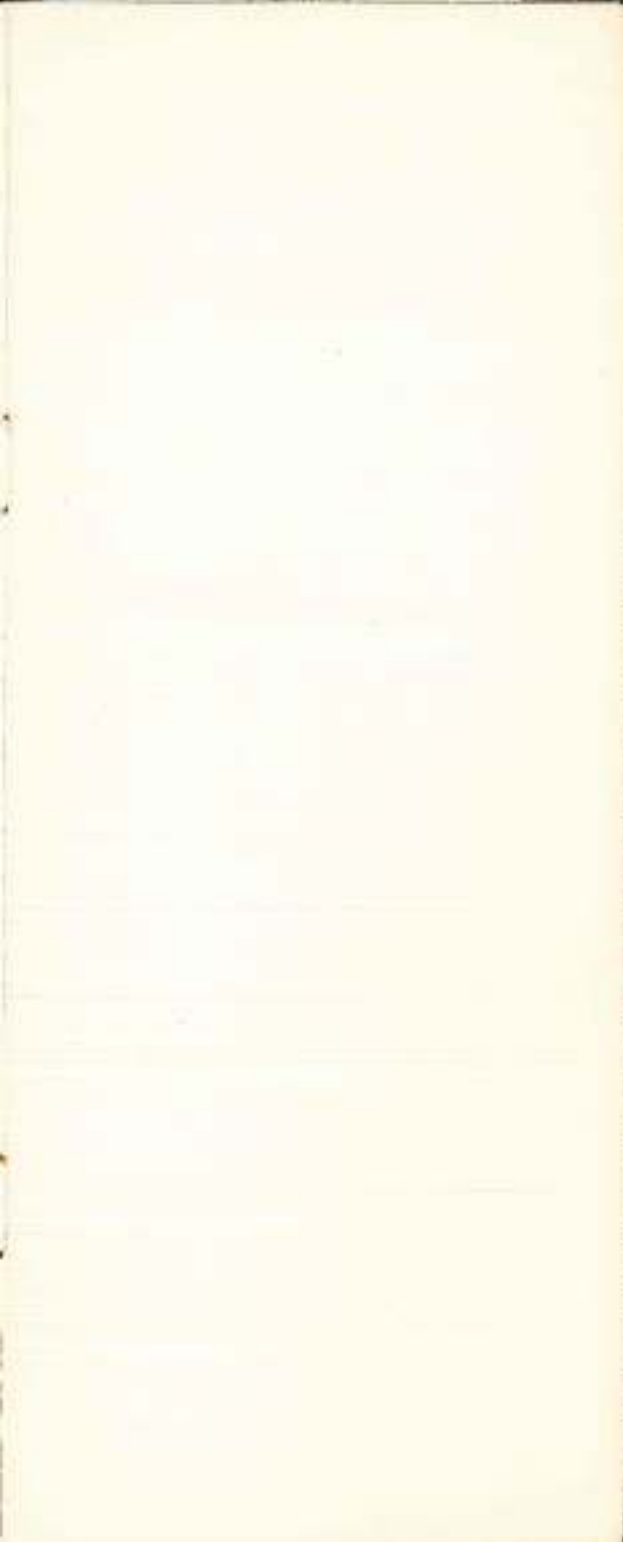
NOTES

- [1] If a word after HLOC command ignores Blank2, it prints on previous command
- [2] -TECO switch not necessary if SET EDITOR TECO command has been previously specified
- [3] SFD allowed in TOPS 10
- [4] If a word after MACI command
- [5] TECO under FTS/6 and OS/8 requires two  or  is not an immediate mode command in RT-11 and RSX-11
- [6] TECO 10 copies the last 10 characters of the command string before the error occurs
- [7] Not an immediate mode command in TECO-10 (execution is deferred until next value is then asked. In HLOC-11, an immediate mode command  is the same as T/C
- [8] Two  s may be necessary on systems that support type ahead
- [9] Only if word is entered in the operating system
- [10] TECO 10 contains previous fix
- [11] Not in RSTS TECO-11 does not require S
- [12] In HLOC-11  is a macro requires control to HLOC-11 program
- [13] TECO-10 has no general hold radix macro
- [14] In TECO 10, a space between two numbers also acts as an addition
- [15] TECO-10 prints sign on line in HLOC-11 and TECO 11 print unsigned res. in
- [16] In HLOC-10, nA, and nA to CA
- [17] Must immediately follow search in TECO 11
- [18] Cannot jump to before start of current search in TECO 8 or TECO 11
- [19] HLOC-10 stores either hex or numbers; this fix, not both simultaneously, in a single area within the G register
- [20] Same as T/C; in TECO-10
- [21] Arguments "pass through" and command in TECO-11
- [22] On TECO 10, no case flagging occurs if the terminal supports two cases, regardless of state of EU flag
- [23] TECO-10 uses only bit value 1, TECO-8 uses only bit values 1, 2, and 8
- [24] In TECO-10, 1/2 must be followed by a digit string and the octal radix mode applies only to this string
- [25] HLOC-11 has two kinds of radix, 1/2 and 4/8 match
- [26] Cannot quote \$ in TECO-10

NOTES

NOTES

NOTES



NOVICE SUBSET OF TECO

These commands allow you to do most common editing jobs on a version of TECO.

W <i>file</i>	Creates a new file (number 1-11 use, CRLF/L command)
E <i>file</i>	Edits an existing file (number 1-11, use ED T command)
.	Moves pointer to beginning of buffer
Z	Moves pointer to end of buffer
L	Advances pointer to beginning of next line
n	Advances pointer <i>n</i> lines
-n	Moves pointer back <i>n</i> lines
OL	Moves pointer to beginning of current line
C	Advances pointer across one character
nC	Advances pointer across <i>n</i> characters
-nC	Moves pointer backwards across <i>n</i> characters
T	Types text from current pointer position to end of current line
nT	Types <i>n</i> lines on terminal
-T	Types entire content of text buffer
OTT	Types current line
text\$	Inserts text before current pointer position
K	Deletes text from current pointer position to end of line
nK	Deletes <i>n</i> lines of text
-K	Deletes entire content of text buffer
D	Deletes the character following the pointer
nD	Deletes <i>n</i> characters (excludes the pointer)
-nD	Deletes <i>n</i> characters before the pointer
P	Outputs the current page and reads in the next page
PF	Executes the <i>Program</i> macro lines
A	Appends the current input page to the end of the buffer
FX	Moves the remainder of the input file to the output file, closes the output file, and exits from TECO
\$text\$	Beginning at the current pointer position, searches for the specified text on the current page. The buffer pointer is positioned after the specified text if it is found. If the text is not found, an error message is given and the buffer pointer is left at the beginning of the next page.
!text \$	Same as <i>\$text\$</i> except that the search continues across page boundaries, if necessary, until the string or the end of the input file is reached.
!\$text1\$ \$text2\$	Searches for text1 and, if found, replaces it with text2.