the MOBY MUNGER

The official organ of the TECO Special Interest Group

Issue #2 March 1979

Contributions and correspondence should be sent to the editor:

Stanley Rabinowitz
6 Country Club Lane
Merrimack, NH 03054

or to

TECO SIG
c/o DECUS, MR2-3/E55
One Iron Way
Marlboro, MA 01752

Camera ready copy is preferred but scribblings on the side of a napkin will be accepted. Ready-to-use material should be prepared on 8 1/2 x 11 inch white bond paper. A one inch margin should be maintained on both sides, on the top and on the bottom. Material should be reasonably clean, legible, sufficiently dark copy for printing. Material prepared on electrostatic printers (e.g. Versatec printers, Xerox machines, etc.) are often unsuitable for photographic reproduction.

Readers are urged to submit articles, techniques, notes, hacks, ideas, comments, and bug reports to the Moby Munger for publication. This publication will publish on a more regular basis if more readers will submit material.
The newsletter editor prefers camera ready copy, but he will retype material submitted that is not in the appropriate form if it is worthy enough. Machine-readable copy is also acceptable if the margins are the right size. TECO programs considerably over 1ØØ characters in length should be sent in machine-readable form (if not camera-ready) in order to preserve accuracy during the publication cycle.

©1979, DECUS
It is assumed that all articles submitted to the editor of this newsletter are with the authors' permission to publish in any DECUS publication. The articles are the responsibility of the authors and, therefore, DECUS, Digital Equipment Corporation, and the editor assume no responsibility or liability for articles or information appearing in the document.

TABLE OF CONTENTS

SIG NEWS

The TECO SIG (Special Interest Group) has been quite active during the last year. Perhaps our best accomplishment was the creation of a TECO reference card. This pocket guide is in three colors and covers all TECO's available from DEC. The reference card is now available from the SDC (Software Distribution Center). Its order number is AV-D53ØA-TK.

We have held sessions at each MINI/MIDI DECUS symposium since the SIG's creation. Our TECO Tutorials (held at each symposium) have always greeted large audiences (standing room only).

At the Spring '79 Symposium, we hope to bring down several new versions of TECO for distribution. (Media will most likely be MAGtape and DECtape.) It is hoped that we will have TECO-11 version 34 available for RSX, VAX, and RSTS. Two new TECO's, one for OS/8 and one for TOPS-1Ø will be available (hopefully). These versions have been written at the prompting of the SIG and should be compatible with TECO-11. More news about these versions of TECO will be given in the next issue of the Moby Munger.

There will be at least one TECO paper given at the Spring Symposium.
There is also a Birds of a Feather Session scheduled, so users are urged to prepare small talks to be given there.

If you are not already a member of the TECO SIG, send your request for membership (it's free) to

DECUS
MR2-3/E55
One Iron Way
Marlboro, MA 01752

The Moby Munger is published sporadically by the TECO SIG. Articles are needed. We especially need macros of interest.

Also, anyone interested in helping out with SIG administrative work should write to the TECO SIG, c/o DECUS at the above address. We need someone to help catalog and solicit and review TECO submissions to the DECUS program library and we need someone to help organize activities at and solicit papers for DECUS symposia.

The SIG would also appreciate hearing information about other versions of TECO available. Also, we coordinate bug reporting and bug fixing for various versions of TECO that are not officially supported by DEC. Send all bug reports and other comments or suggestions to the TECO SIG, c/o DECUS or to the editor.

NOTES FROM RT-11

RT-11 V3B included TECO-11 (version 28) as an unsupported product. Users of that version of TECO should be warned that later versions of TECO-11 have made some changes that may affect existing macros. The CTRL/W window command was changed to W and the Qq command no longer throws away any numeric argument. Users should plan ahead to the time that a later version of TECO-11 is available for RT-11. (Some users may have received version 29 at a DECUS symposium.)
Current plans are that version 34 of TECO-11 will be released with version 4 of RT-11. That release is currently scheduled for early in 198Ø. The delay is because the TECO maintainers are too busy with the RT-11 release to find time to upgrade TECO.

If any reader who is a user of RT-11 would like to volunteer to upgrade TECO-11 for RT-11, send your name and qualifications in to the TECO SIG (c/o DECUS). The lucky winner will receive the source of the RT-11 TECO I/O module along with the new I/O interface spec and an OBJ of the main TECO kernel.

NOTE ON VT52 SUPPORT

Users trying to use the VT52 support macro under CTS-3ØØ or the FB version of RT-11 must be aware that they must first issue the command

```
              SET  TT:  NOCRLF
```

to the operating system.

NOTES FROM VAX

Version 29 of TECO-11 was included with the first VAX-11/780 release as an unsupported product. The manual is available from DECUS (DECUS-11-35Ø). We have been using version 33 in-house at DEC for quite a while now and it seems very solid. We hope to be able to submit this version to the DECUS library very shortly.

PROBLEM WITH CTRL/C

The CTRL/C immediate action command (in VAX TECO v29) does not properly get you out of TECO. Temporarily, you can use the CTRL/Y command to abort TECO, but this alternative has the disadvantage that you may be left with an unwanted Ø-length file if you had a file open when you aborted. The way you can avoid this is to type EK$$ before doing the CTRL/Y, or by using the ↑Z↑Z↑Z command. CTRL/C has been fixed in v33.

PROBLEM WITH TECO.TEC

TECO.TEC will not work properly if you are using sub-directories under VAX. This problem has been fixed and a new TECO.TEC has been created and tested and hopefully will shortly be available from the DECUS program library.

NOTES FROM OS/8

TECO is a supported product running under the OS/8 operating system. Version 5 was released with OS/8 V3D (as part of the extension kit). This version will also run under OS/78, although it will not run properly while a symbiont is running. The PDP-8 group at DEC has no plans to re-release TECO in the near future.

INDEX TO PUBLISHED PATCHES:

Seq. Number	DSN issue	Title/Description
31.2Ø.Ø1	Mar 78	CHANGING THE DEFAULT EU VALUE Gives an optional patch to change the initial value of the EU flag.
31.2Ø.Ø2	Mar 78	CHANGING THE DEFAULT EH VALUE Gives an optional patch to change the initial value of the EH flag.
31.2Ø.Ø3	Mar 78	REMOVING YANK PROTECTION Gives an optional patch for users who don't like Yank protection. This causes the Y and _ commands to always work.
31.2Ø.Ø4	Mar 78	SCOPE SUPPORT FOR VTØ5 USERS Gives an optional patch to allow command line scope editing to work from VTØ5 terminals.
31.2Ø.Ø5	Mar 78	PROBLEM WITH AY COMMAND Fixes bug: Y command sometimes aborts with ?NYA error (such as after an A command) even though no numeric argument is specified.
31.2Ø.Ø6	Mar 78	CONDITIONALS INSIDE ITERATIONS Fixes bug having to do with nesting of conditionals and iterations.
31.2Ø.Ø7	Mar 78	ECHOING OF WARNING BELLS Fixes TECO so that the warning you get when about to run out of command line space consists only of a bell ring and no accompanying "^G" (Caret-G) typeout.
31.2Ø.Ø8	May 78	CTRL/U SOMETIMES FAILS AFTER * Fixes many bugs having to do with scope editing and typing rubouts and CTRL/U's after a * has been entered into command string.
31.2Ø.1Ø	May 78	MULTIPLYING BY Ø IN TECO Fixes bug: n*Ø didn't always return Ø.
31.2Ø.11	May 78	Q-REGISTERS DON'T WORK IN 8K Fixes bug. Enables Q-Register commands to work properly on 8K machines.

Seq. Number	DSN issue	Title/Description
31.2Ø.12	May 78	CAN'T SKIP OVER A "W" Fixes bug: While scanning for a ', TECO would blow up if it encounters a W command or a PW command.
31.2Ø.13	Jul 78	UNSPECIFIED ITERATIONS AFTER INSERTS Fixes bug. An iteration with an unspecified iteration count would sometimes act as if the iteration count was Ø (thereby skipping the iteration). For example, IA$<...> exhibits this behavior.
31.2Ø.14	Aug/Sep 78	NEW FEATURES IN TECO V5 Describes the complete set of differences between TECO V4 and V5.
21.Ø3.Ø1	May 78	DEFAULT EXTENSIONS TO TECO Gives an optional patch to V3D CCL to cause the TECO and MAKE commands to use .MAC as the default estension rather than .PAL .

KNOWN RESTRICTIONS:

The following bugs are known to exist in OS/8 TECO and should be considered to be permanent restrictions. There are no plans now or in the future to fix these bugs.

1. No error message is given if the nI command is not followed by an ALTMODE. Use of such a construct inside a conditional will cause unpredictable results.
   Way around: always follow nI command by an altmode.
2. If an ALTMODE is typed into TECO in response to a ↑T command, TECO will return the ASCII code for a dollar sign (decimal 36) rather than the correct decimal 27.
   Way around: Turn echo off first. This bug does not manifest itself when echo is off.
3. A large T or X command will abort with the ?FUL error if the output device is nearly full.
   Way around: issue T or X command in smaller pieces or exit TECO, free up space on output device and then resume editing.
4. The CTRL/S special command (freezes type-out) turns off the window display on a PDP-12.
   Way around: After typing CTRL/Q, the window display resumes.
5. Typing CTRL/C while I/O is going on to a non-system device may cause TECO to return to dot rather than to prompt.
   Way around: Avoid typing CTRL/C while such I/O is in progress.
6. TECO will die if while running under BATCH in 12K or 16K machines, TECO attempts to read from BAT: .
   Way around: Avoid use of BAT: handler on such configurations.

CORRECTIONS TO OS/8 HANDBOOK UPDATE

The OS/8 Handbook Update (DEC-S8-OSHBA-A-DN4) contains much new information about TECO. New features of TECO version 5 are reported. There are a number of errors in table 2-45 (on page 24) which describes TECO's error messages. Specifically, the one-line error message for some errors is incorrect. The error code and corrected message appear below:

Error	Message
?BNI	> not in iteration
?FUL	Output command would have overflowed output device
?NFO	No File for Output
?SNI	; Not in an Iteration
?XAB	Execution Aborted by CTRL/C

Also, it should be noted that each occurrence of X* in the error message listing should really be "x" . This represents the fact that TECO will substitute the character in error for "x". For example, the command FA will produce the error message: ?IFC Illegal Character "A" after F .

WILD.TEC

The program WILD.TEC was distributed with the OS/8 TECO material that was handed out at the last DECUS meeting. For those of you who did not get it, it is reprinted on the following page. It is useful only with the new CCL that is distributed with the latest version of OS/78. (That is the CCL that allows multiple-character switches.) That CCL will run under OS/8 as well as OS/78 and is highly recommended for use by all OS/8 users. It has the special feature that if a COMPILE, PAL, MACRO, or EXECUTE command is issued with a filespec that contains wildcards, then SYS:TECO.TEC is invoked passing it the user's command string in the text buffer. WILD.TEC parses the command and creates the appropriate BATCH stream to properly execute the command. Thus, you can type

```
.MAC *.MAC
```

and OS/8 will assemble all files with a .MA extension on DSK:. Similarly, the command

```
.COMPILE ABC???.FT
```

will assemble all FORTRAN programs whose names begin with the letters ABC. If you don't have the new CCL, the above command can be issue by typing MUNG WILD,COMPILE ABC???.FT .

```
SYSTEM MACRO TO HANDLE WILD CARDS!
ØAU9 Q9-^^""N                  !Get first character in buffer!
-^^^T"E J D HT                 !If it is ^T, type message following!
                               !followed by a CR/LF!
^C'                            !and exit!
                               !If first character is not ^T or ",!
                               !then assume a command line!
                               !is in the buffer.!
S^S$ R ØXA ØK                  !Store the verb in Q-reg A!

JI
                               !Surround the filespecs by CR/LFs!
<JLS-$;.-1U1S^S$ $ RQ1,.X1 Q1,.K ØL2RG1>   !Move all - switches to front!
<JLS/$;.-1U1S^S$ $ RQ1,.X1 Q1,.K ØL2RG1>   !Move all / switches to front!
L 2R ØXB ØLK                   !Get all switches into Q-reg B!
$JOB
IR WILD.DI<$                   !Create BATCH preface!
 2R I/F=1
UNG WILD,"$ GA I"$ GB
$END
                           !Create BATCH epilog!
WILD.TMP$ PW EF HK         !Put created BATCH job in WILD.TMP!
SUBMIT WILD.TM/T/H$ '      !Execute the BATCH job!
-^^""E                     !If the first character is a " !
) FS"$ $                   !then delete it & change matching " to space!
A ØK HXB HK                !place pre-text in Q-reg A and post-text in B!
WILD.DI$Y                  !Read in directory listing of files that matched!
 ZJ -3K                    !delete garbage at beginning and end!
E^ANo such files
'                          !Give error message if no files remaining!
FS $$;>                    !Removes spaces from filespecs!

 2R ØXC GB ØL I.$ GA       !Create COMPILE FILE command!
.I.MUNG WILD,^T$ GC I
.                          !Create TYPE command!
                           !Repeat, for each filespec!
JOB
NG WILD,^T$GAI:
I,R FOTP
LD.DI,WILD.TM,WILD.BI/D
D
                           !Put BATCH template around commands!
ILD.BI$ PW EF HK           !Create second BATCH job!
UBMIT WILD/T/H$            !Execute it!
```

NEW TECO FOR OS/8

A new TECO for OS/8 will shortly be submitted to the DECUS program library. It is called TECO-8 and is based on DEC's OS/8 TECO V5. It was written by Stan Rabinowitz and is upward compatible with DEC's TECO. It is written in MACREL and contains many new extensions to DEC's TECO, mostly compatible with TECO-11. Copies will be available at the next symposium.
Of most interest is the fact that it contains VT support (written by Jim Roth). This support is compatible with TECO-11's VT support and allows maintaining a window into the text buffer on the screen of a VTØ5, VT52, or VT1ØØ.

Also of interest is TECO.INI and TECO.TEC support, read with no wait, CTRL/C trapping, War and Peace Error Message Mode (3EH), type-ahead, V, "A and "D, and much much more. More details will appear in the next Moby Munger. An advance article appears on pages 7-1Ø of the January 1979 issue of The 12-Bit SIG Newsletter (issue #32).

PREVIEW OF VTEDIT

With the above-planned release of TECO-8 will be a macro called VTEDIT.TEC . It is compatible with TECO-11's VTEDIT that is a superset of the VT52.TEC macro. This macro turns TECO into a true scope editor. The buffer is permanently displayed on the scope of the CRT. Typing a character immediately enters it into the buffer and updates the screen. Typing special keys and keys on the VT52 keypad cause special actions to occur. A summary of the keypad layout follows: (See page 54 for corresponding layout for the -11 version)

BLUE Save Text*	RED TECO command*	GREY Paste text	^ Up in column*
7 Open Line*	8 Page*	9 Quote*	v Down in column*
4 Up Line*	5 Delete char*	6 Delete text	➡ Cursor right*
1 Top of page*	2 Bottom of page	3 Start of line	⬅ Cursor left*
Ø Down line*		. Search again*	ENTER Search arg*

ABC, MHB, HJ, JR fecerunt PK, SR composuit

^C	Return to TECO
^D	Kill rest of line*
^K	Kill line *
^U	Kill start of line
^Z	Return to TECO
BK SP . .	Go to end of line*
DELETE .	Delete previous character*
ESC ESC.	Repeat RED-key command*

All *commands take an optional argument as:
ESC *number*

NOTES FROM TOPS-1Ø

TECO is a supported product running under the TOPS-1Ø operating system. The latest release is version 24(2Ø2) which was released in March of 1977.

This version of TECO is pretty solid, however it does not contain any scope support, nor has this TECO kept pace with the other TECO's around DEC. For these reasons, many DECSYSTEM-1Ø users are running versions of TECO written at places other than DEC.

The TECO SIG hopes to remedy this situation shortly by submitting to DECUS a new version of TECO for TOPS-1Ø that will be compatible with TECO-11 and will have scope support. This version is being worked on by Andy Nourse and we hope to have it available from the DECUS program library very shortly now. Copies should be available by the next -1Ø DECUS symposium.

Bug reports for DEC's TECO should be submitted on SPRs in the usual manner. Fixes are published periodically in the Software Dispatch. Following is a complete summary of all published corrections to TECO since the release of version 24 (edit 2Ø2):

<u>Index to TECO Revisions</u>

Edit	SPR-no./ date	PCO-no.	Description of bug fixed
2Ø3	1Ø-23364 19-Aug-77	-Ø51	TECO does not generate correct line sequence numbers when the /GENLSN switch is given as part of an EB command
2Ø4	1Ø-235Ø8 14-Sep-77	-Ø52	TECO generates erroneous errors such as ?FNF if the trace feature (? command) is turned on.
2Ø5	1Ø-21631 15-Sep-77	-Ø53	A ?NNQ error is given when trying to access a Q-register with a Q command after an illegal argument to a U command has been given.
21Ø	1Ø-21632 14-Oct-77	-Ø54	Successful searches within iterations do not return a value of -1. *Editorial note: TOPS-10 users are in for a big shock when they install this fix. TECO has been working 'incorrectly' for the 7 years that I have been using it and all TECO programmers know that searches do not return values unless they are colon modified. The problem is that the documentation has been wrong all these years. TECO-11 made the above change several versions ago and immediately got lots of flack from all their users, so they changed back. The algorithm found most useful is that a search in an iteration acts as if it were colon modified if it is immediately followed by a semicolon.*

Index to TECO Revisions (cont.)

Edit	SPR-no./ date	PCO-no.	Description of bug fixed
2ø6	none 17-Oct-77	-ø55	MACRO version 53 gives an error when TECO is compiled.
2ø7	none 17-Oct-77	-ø56	TECO gets an illegal memory reference if it is loaded with DDT and a command line of 9ø characters is entered.
211	1ø-22148 28-Oct-77	-ø57	TECO gets an illegal memory reference when doing an nXq command if . > 262114.
212	1ø-21417 ø8-Nov-77	-ø58	The *q every once in a while puts garbage in Q-register q.
213	1ø-21561 29-Nov-77	-ø59	The ↑EL match control construct fails to recognize end-of-buffer as a match.
214	1ø-22762 15-Dec-77	-ø6ø	TECO is not able to find the file that is to be edited when FTSFD is turned off.
215	1ø-24918 17-Feb-78	-ø61	Successive searches forget exact case specifications. For example, the command Sa↑VB$ causes the search to proceed using exact case mode since case control constructs are present. A subsequent S$ command uses the preceding search argument, but forgets to use exact case mode. This patch causes the exact mode internal flag to be reset only when an explicit search string is given.
216	1ø-25563 19-Apr-78	-ø62	If the 81st character of a search pattern specification is a CTRL/Q or CTRL/R, the first word of the Q-Register stack gets wiped.
22ø	1ø-25665 ø1-May-78	-ø63	Standard page marks are not given in line sequence mode.
221	1ø-25395 ø2-Jun-78	-ø64	TECO gets confused by garbage after a filespec when processing CCL.
222	1ø-25659 17-Jul-78	-ø65	No error is given if the match control construct CTRL/↑ occurs at the end of a search string. This patch creates the new error message ?MCO (Missing Character Operand).
223	1ø-25648 18-Jul-78	-ø66	S↑E[A,↑ES] will match A and any tabs or spaces following A.
224	none ø5-Sep-78	-ø67	This edit improves the comments in the source of TECO.

Index to TECO Revisions (cont.)

Edit	SPR-no./ date	PCO-no.	Description of bug fixed
225	1∅-26761 12-Oct-78	-∅68	The commands MAKE FILE.EXT and TECO FILE.EXT lose the third character of the extension. (Edit 221 broke the command scanner.)
226	1∅-26914 2∅-Oct-78	-∅69	TECO high segment is too big.
227	1∅-27219 14-Dec-78	-∅7∅	? sometimes doesn't work after an erroneous *q command. Rubout doesn't work after a *q and *q goes into affect after a single ALTmode instead of after two ALTMODEs. *Editorial comment: I believe that *q is an immediate action command and should go into affect without any altmodes being necessary.*

Note: I have not been able to determine the content or purpose of edit 217 to TECO.

<u>Miscellaneous TECO bugs:</u>

The following bugs were reported to me by a person who wishes to remain anonymous. He reported them to DEC in November of 1977 but DEC refused to answer his SPRs since he was a DEC employee and not a customer. These bugs apply to v24 of DEC's TOPS-1∅ TECO:

1. A Fails if buffer is full.
   If a file is open for input and you fill up the text buffer to capacity and then issue an A command, TECO properly issues the message ?COR - Storage Capacity Exceeded, and then proceeds to wipe out the text buffer (sets Z=∅).

2. Excessive Pops bomb TECO.
   If a large number of successive pop commands is given, for example 5∅∅∅<]q> , TECO aborts with an illegal memory reference. TECO ought to give a message such as ?CPQ Can't Pop Q-register if the Q-register push down stack is empty and a]q command is issued.

3. Push Down List overflow is fatal to TECO.
   The command !A!<OA$$ aborts TECO with a ?PDL OV error message. TECO ought to be smart enough to catch this error and give an error message such as ?PDO.

4. Literal-typeout-mode too General
   If the 1ET command is issued to put TECO into literal-typeout-mode, then many TECO features act strangely. For example, while editing a command string, if you try to rubout a CTRL/A, TECO retypes the deleted character by sending a true CTRL/A (rather than an uparrow-A) to the terminal. Also, the Bell-space immediate action command retypes the line literally, and error messages such as ?IFC report erroneous characters literally. This is wrong. Only TECO typeout should be affected by literal mode (see section 3.6.4 of the TECO manual).

5. TECO fails to recognize ↑a .
   Erroneous results can occur if the CTRL/A command is used in its uparrow mode within an unsatisfied conditional and the "a" following the uparrow is in lower case. For example, the command

   ```
   Ø"N<Ø"N↑a text ↑A ' > ' $$
   ```

   produces the ?MAP error message incorrectly.
   The above command works if ↑a is replaced by ↑A .

6. Null FS replacements abort command.
   The command IABCD$JFSB$↑V$T$$ doesn't execute the final T command since TECO thinks it has seen a double ALTMODE and consequently aborts the execution of the command string after replacing B by the null string.

7. @-modifier to tab confuses TECO.
   The TECO documentation fails to specify that the tab and ↑I commands may be @-sign modified. Such commands appear to work. For example, both @→/text/ and @↑I/text/ insert →text into the buffer. (→ denotes a tab character.) However, if an @-sign modified tab command appears within an unsatisfied conditional, TECO gets confused. For example, the command 2"E@↑I/L/'$$ works while the command 2"E@→/L/'$$ fails giving a ?MAP error.
   Suggestion: TECO ought to treat @→ as an error.

NOTES FROM RSX

Version 28 of TECO-11 was released (as an unsupported product) with versions 3 and 3.1 of RSX-11/M. Version 28 is also in the DECUS program library. The order number for the RSX-11/M version is DECUS-11-333 and the order number for the RSX-11/D version is DECUS-11-334. It is expected that TECO-11 version 34 will shortly be submitted to the DECUS library. The DECUS order number for the manual is DECUS-11-35Ø.

Fixing RUNOFF files

Output from RSX RUNOFF is in a very strange ASCII format, so that RUNOFF output can not be conveniently input to SRCCOM. These problems can be fixed up by passing the funny file throguh TECO twice as follows:

```
TECO file/-CR
*EX$$
TECO file
*EX$$
```

Bug with P and full buffer

If the text buffer is full and does not end with a line-terminator, and if the form feed flag is not set, and a P command is issued, TECO will insert an erroneous carriage-return / line-feed at the end of the buffer. This bug is due to inherent RMS problems and there is no plan to fix this bug in future TECO-11 releases.

NOTES FROM RSTS/E

TECO was released with RSTS V6C as an unsupported product. It has also been distributed at recent DECUS symposia. A new version will shortly be put into the DECUS program library.

NEW FEATURES FOR V29:

The following new features are part of the version 29 release and apply only to RSTS:

1. A bug which prevented reading of certain RMS files has been fixed.
2. The EG command has been changed. The new format is

   ```
   EGcommand$     or     @EG/command/
   ```

   This command is the same as EC, however control returns to RSTS and the CCL command specified after the EG command is executed by RSTS.
3. Several new file switches are permitted on file specifications in the ER, EW, and EB commands.

   /B+ is the same as the / (with no switch) and means that you are editing a BASIC PLUS file.

   /B2 is a new switch and means that you are editing a BASIC PLUS 2 file. It is the same as the /72 command (see below).

   /n where n is a decimal number. On input, this causes the trailing &'s (along with all preceding spaces and tabs) to be discarded. On output, if a line is terminated by a carriage-return / line-feed sequence and the next line does not begin with a digit, then enough tabs and/or spaces are appended to the line followed by an & to cause that & to go into column n of the line. If this is not possible, then a single space followed by an & is appended.
4. The new CCL command

   ```
   TECO file2=file1
   ```

   has been implemented. It reads file1 and creates file2. It is equivalent to the TECO command

   ```
   ERfile1$EWfile2$Y
   ```
5. The TECO start-up file, TECO.TEC, has been modified. Users who have their own private copies of TECO.TEC should be aware that it will be invoked with a -1 in Q-register 1.

See elsewhere in this issue for general features of TECO v29 that apply to all TECO-11 implementations.

PROBLEM OF THE MONTH

This column will present interesting and/or unusual problems for the reader to solve.

Problem 1: Write a TECO macro, which, when executed, types out the upper case letters A to Z (in order) on the terminal.

Restrictions:

(a) There must be no other type-out by this macro.

(b) The macro must contain only distinct characters. That is, no character may be repeated. (An upper case character is to be considered different from its lower case counterpart.)

(c) The macro must be as short as possible.

For example, the macro

```
iABCDEFGHIJKLMNOPQRSTUVWXYZ$ht
```

is a 3Ø character solution to the problem. It is far from minimal. Can you cut it down to 15 characters or less?

Solutions should be sent to the editor and must be postmarked by June 15, 1979.

There are two classes of competition:

CLASS A: Macro is written in machine-independent (DEC) TECO† and will run correctly under OS/8, RT-11, RSX-11, RSTS/E, or TOPS-1Ø.

CLASS B: Macro is machine dependent. Specify on which TECO it will run.

Macros will be tested by effectively putting them in a file called MACRO.TEC, invoking TECO, and executing the command string `ERMACRO.TEC$YHXMHKMM$$` .

Winners names will be printed in the first issue of the Moby Munger following the conclusion of this contest.

Solutions and problem proposals for future columns should be sent to the editor: Stanley Rabinowitz, 6 Country Club Lane, Merrimack, NH 03054.

† Consult TECO reference card for system compatible commands.

COMMENTS ON *Z

by Eric Osman

The *Z command in TECO-11 is a real crock because

1) Typing a non-* and then DELETE and then *Z loses.

2) Sometimes you do a long command string, then a few short command strings later (things like HT, etc.) you realize the long command string wasn't quite right and you want it back. *Z is useless at this point.

My suggestion for definition of *Z command, where "Z" is any legal Q-register name is:

> The *Z command takes the last command string that was longer than 15 characters and stores it in Q-register Z.

Notes:

a) The number 15 is arbitrary, but has proven convenient in other TECO's.

b) This definition solves both problems listed above.

c) No conflict with multiplication because * seen with no preceeding argument means "save last long command string".

BEGINNER'S COLUMN

by Stan Rabinowitz

<u>Selectively making changes to a file</u>

A common editing problem is to change all occurrences of one string to another in a file. Although the command <FNstring1$string2$;> is usually good enough to do this, this command has the problem that there might be an unusual occurrence of the string1 that shouldn't be changed. The following technique is a typical way of avoiding this problem. This command string searches for the given string1 and types out the line containing this string on the terminal. It then waits for the user to type a character on the terminal, if he types "Y", then that means that the string1 should be replaced by string2. Typing any other character means that the string1 should not be replaced in this case.

```
<Nstring1$;                 !Search for string, exit if not found!
↑A
↑A                          !Return the carriage!
ØTT                         !Type the current line!
↑A
 Change? ↑A                 !Prompt for decision!
↑T-↑↑Y"E-7DIstring2$'       !If yes, replace string!
>                           !and continue!
```

DIFFERENCES BETWEEN TECO-11 V29 AND V28:

1. The n↑Uq$ command was implemented. It inserts the character whose ASCII code is n into Q-register q. The command may be @-sign modified and colon modified (colon means character appends to end of Q-register text).

2. F_ was put back in.

3. Several commands were changed to prevent conflicting with DEC's TOPS-1Ø TECO. ↑W was changed to W (Window command). ↑V (version number) was changed to EO.

4. The ↑R (Replace) command was eliminated. It is identical to FS.
   (The above two items were necessitated by the fact that the DECsystem-10 group upgraded their TECO to version 22 without consulting what the PDP-8 and PDP-11 groups were doing.)

5. Vertical tab is now an error (it used to be ignored).

6. The ↑V and ↑W string build constructs and the ↑EV and ↑EW match constructs were implemented to be compatible with DEC's TOPS-1Ø TECO version 23. Also, a bug involving the "V and "W conditionals was fixed.

7. The ↑↑ string build construct was eliminated. Its functionality can be accomplished by the new ↑V construct.

8. Some error codes were changed for compatibility reasons. ?ICC became ?IQC and ?IQR became ?IQN.

9. The :A (append single line to text buffer) command was implemented.

1Ø. An error message is generated if a Y command has a numeric argument.
   (This is a good protection feature and is compatible with OS/8 TECO.)

11. The bits in the EH flag were shuffled so as not to conflict with TOPS-1Ø TECO. The old 3EH is now 4EH.

12. The bits in the ED flag were shuffled around so that now the flag is bit-encoded. -1ED is now 2ED.

13. The <bell><space> and <bell>* commands were modified so that the first line of the command string will be preceded by an asterisk (representing TECO's prompt). Now, if you are on a scope terminal and rub out a multi-line command string back to the first line, the * prompt that was on the screen does not disappear.

Version 29 of TECO-11 is the version that is documented in the TECO Reference card available from the SDC. The PDP-11 implementors must be congratulated for their part in ironing out incompatibilites between the various TECOs at DEC. The only item above that may be of some concern to users is the change in the window command. Anyone who wrote scope editing macros under v28 should change all CTRL/W's to W's in their macros.

DIFFERENCES BETWEEN TECO-11 V29 and V3Ø:

TECO-11 has the following new features that were not in V29:

1. The Qq command may now take an argument. nQq returns the value of the (n+1)th character in Q-register q. The argument n must be between Ø and the Q-register's size - 1 . If n is out of range, then a value of -1 is returned. (ØQq returns the first character, etc.) *(Warning - this new command causes many existing macros to break, because many macros were assuming that the Q command would discard any argument. It would be a wise programming practice to always discard an unneeded value by following it with an ALTMODE, a command which is guaranteed to discard any argument.)*

2. The O command can now build the tag to go to using the same constructs as the Search and E commands. Specifically, the ↑EQq construct can be used.

3. The EJ command has been implemented. ØEJ returns your job number and 1EJ returns your console keyboard number. (Ø is returned on single-user systems.) *(Additional EJ commands have been implemented in later versions of TECO-11; these will be described in the next issue of the Moby Munger.)*

4. Conditionals now allow an IF-THEN-ELSE construct. In addition to the usual form of a conditional, you now have the new form:

   ```
   n"X <then-commands> | <else-commands>  '
   ```

   If the condition (denoted above by X) is satisfied, then the then-commands are executed. If the condition is not satisfied, then the else-commands are executed. *(A real advance in structured programming in TECO! This command is amazingly useful. Additional commands that will make TECO programming more structured are planned for the near future. User comments and suggestions are always welcome.)*

5. The :W command can be used to return scope characteristics. Ø:W returns your scope type. Ø means VT52, 2 means VT1ØØ, 4 means VTØ5 with fill, and 6 means VTØ5 without fill. 1:W returns the scope's horizontal size. 2:W returns the scope's vertical size. n,m:W is used to set the m:W information to n. *(This last feature seems to be a little inconsistent with the rest of the language.)*

This version of TECO-11 has been in use in-house at DEC for quite a while, so it has gotten extensive testing. However, it has not yet been made publicly available. Version 34 of TECO-11 will shortly be submitted to DECUS. Watch for more information. The next issue of the Moby M will contain details about version 32, 33, and 34. You can look forward to such obscure commands as F', F<, F>, and hexadecimal radix mode!

DIFFERENCES BETWEEN TOPS-1Ø TECO V24 and V23B:

DEC's TOPS-1Ø TECO version 24 (edit 2Ø2) has the following new features that were not in version 23B (edit 162):

1. TECO now allows editing of files protected via the File Daemon. The File Daemon bit (4ØØ) of the file protection is preserved in the new files created.

2. TECO will now allow ERSATZ devices and SFD's in arguments to the MAKE, TECO, ER, EW, and EB commands, as well as the "[-]" and "[,]" constructions.

3. On an EB or TECO command, TECO will now always put the new source file where the command string specified. (In version 23B, the new file always went into the user's area.)

4. Any time that TECO finds a file on an area other than the one explicitly specified in the command string (because of scanning up directory paths), the warning message %File found in [path] is printed.

5. The command MAKE A=B is now permitted. It is equivalent to invoking TECO and giving it the initial command ERB$EWA$Y$$. An equal sign in a TECO command will be ignored. Thus TECO A=B is equivalent to TECO A . Thus, if a MAKE A=B command is followed by a TECO <crlf> , the right thing will happen (the system will remember the previous argument, and file A will be edited).
(Once again the -10 world went and implemented a feature without seeing how other DEC operating systems handle the feature. OS/8 solved this problem several years prior to the -10's implementation by making the TECO A=B command edit B into A and only remember the argument up to and not including the "=" sign.)

The following deficiencies in V23B were <u>not</u> corrected in V24:

(a) CTRL/C REENTER or EX$$ REENTER sometimes results in lost editing or address checks.

(b) Changing the size of the push down list causes a garbage collect error (?GCE).

(c) The value output in the [nK CORE] message is not always correct for KI processors.

Presumably, the following bugs in V23B were fixed:

(i) TECO couldn't handle magtape with non-standard block size.

(ii) ?FUL was incorrectly generated if a file on a DECtape was open for both input and output.

(iii) Output to device NUL: resulted in a superseding warning.

(iv) ?COS error was not detected for EB commands.

TECO DOCUMENTATION AVAILABLE FROM DEC

Since the last issue of the Moby Munger, the following new TECO documents have become available from DEC:

Order Number	Description
DEC-11-UTECA-A-DN1	UPDATE NOTICE No. 1 to TECO-11 User's Guide This is an update to the July 1977 copy of the TECO-11 User's Guide (DEC-11-UTECA-A-D). It upgrades appendix F (The index to TECO commands and special characters). The original index had the wrong page numbers. This document is also a useful summary of TECO-11 commands (version 28).
AV-D53ØA-TK	TECO POCKET GUIDE This is a pocket reference card that covers OS/8 TECO version 5, TECO-11 version 29, and TOPS-1Ø TECO version 24. It is extremely detailed. Commands peculiar to specific implementations are easy distinguished by a color coding.

ERRORS IN TECO REFERENCE CARD

A few errors have been noticed in the TECO Pocket Guide (AV-D53ØA-TK). These are described below:

1. Several commands are in black and imply that they exist in all 3 flavors of TECO; however, these commands are not implemented in TECO-8. They are:
   (a) (page 1Ø) [q is not in TECO-8
   (b) (page 11) nES is not in TECO-8
   (c) (page 12) ↑EA is not in TECO-8
2. Page 16, the graphic for the character whose octal ASCII code is 174 is vertical bar (|) not close square bracket.
3. Page 17, note [11] should say "Not in RSTS or RSX. ... " .
4. Page 17, note [25] should say "TECO-1Ø allows begin of buffer to match and does not allow ., $, or % to match" .

Readers discovering other errors in the card are urged to write in and inform us so that these errors can be fixed on the next printing.

EDITORIAL - TECO vs. TV

by the Editor

As the reader of this newsletter may notice - TECO runs on most DEC operating systems: OS/8, RT-11, RSX-11/D, RSX-11/M, IAS, VAX, TOPS-1∅, and RSTS/E. A glance at the large number of black entries in the TECO Reference Card shows that these TECO's are largely compatible too.

Of notable absence is the DECSYSTEM-2∅. Why doesn't TECO run on this machine too? Well, in fact it does. Since TECO runs under TOPS-1∅, that version of TECO will run on the -2∅ too in -1∅ mode. However, running TOPS-1∅ TECO v24 on a -2∅ is not very convenient because TOPS-1∅ TECO does not support scope terminals and TOPS-2∅ does. How to get around this problem? Since most of the TOPS-2∅ developers were heavy TECO users from -1∅ land anyhow, it seemed obvious to upgrade TECO to run on the DEC-2∅ and to support scopes. This in fact they did - sort of. Instead of starting with DEC's TECO for TOPS-1∅, they started with BB&N's TENEX TECO (see page 29). After considerable modification and improvement, they wound up with a program called TV. TV supports many CRTs and looks very similar to TECO. It is currently being used quite heavily within DEC by the DEC-2∅ implementors. Unfortunately, since it was not based on DEC's TECO, it has a few features that are incompatible with TOPS-1∅ TECO and TECO-11.

As of the last information that I have (which is about a year old), TV contains 61 commands that are exactly the same as TECO commands; it contains 19 commands that are extensions to TECO; and it contains 1∅ commands that are either slightly different than their TECO counterparts or have different names. Clearly, only a little bit of effort could have made TV completely compatible with TECO. Right now, TV is a very fine editor. However, there are a few commands that it has that would 'wipe out' experienced TECO users. For example, the R command in TV is what the FS command was in TECO. I agree that R consists of fewer keystrokes than FS, but for the sake of only one extra keystroke, I would rather see a compatible TECO on the DEC-2∅.

Unless a lot more incompatiblities have been added since my last information, the following changes, if made to TV, would make it compatible with TECO:

Change names of ;R , ;W , ;X , and ;C commands to ER, EW, EX, and EC respectively. Change ;-space to ; command. Change ;F to _, change F to N, and change R to FS. Change the X command so that it does not delete the characters extracted and change the ↑A command so that the syntax is ↑Atext↑A rather than ↑Atext$. Rename E command to something else.

It seems to me that ER is just as easy to type as ;R, so I don't understand why this command was changed. Just a small amount of work could turn TV into a beautiful editor compatible with TECO, yet still keep its extensions. Why doesn't someone do this?

Replies to this editorial are actively solicited.

HISTORICAL DEPARTMENT

DIFFERENCES BETWEEN TOPS-1Ø TECO V23B and V23A:

1. Lower case FS and FN commands now accepted.
2. ↑S can now match first buffer character.
3. TECO now recognizes SOS page marks.
4. A warning message is given if a file contains line sequence numbers but no /GENLSN or /SUPLSN switch was specified.
5. TECO now uses the monitor commands TTY ALTMODE and TTY NO ALTMODE to determine if altmode conversion should take place.
6. ?NNQ and ?TTL error messages added.
7. ALTMODE terminating a macro now returns to the next macro level rather than reinitializing.
8. Numeric arguments are now preserved across carriage return and line feed.

DIFFERENCES BETWEEN TOPS-1Ø TECO V23A and V23:

The only change was that the DATE-75 revision was added in V23A.

DIFFERENCES BETWEEN TECO-11 V27 and V24:

The following features were in TECO-11 version 27 which were not in version 24: (consult TECO-11 manual for description of what these commands do)

1. scope mode command string editing
2. secondary input and output streams (EP, ER$, EA,EW$)
3. wildcards (EN) and indirect input (EI)
4. filespecs can use string build constructs such as ↑EQq
5. bounded search (m,nStext$, Ø,nStext$, m,Stext$)
6. string build constructs: ↑ , ↑EQ* , and ↑EQ_
7. :]q , :Gq , :G* , :G_ , and :Qq
8. EGtext$
9. m,nD
1Ø. n"V and n"W conditionals
11. EV and nEV
12. ET bits 6, 7, 8, and 15 (detach flag, abort on error, truncate to terminal width, and CTRL/C trap)
13. n↑Q TECO command

RELATIVE ADVANTAGES AND DISADVANTAGES OF KED AND TECO

by Tom McIntyre

KED is a new video text editor for RT-11 which implements a proper subset of the proposed DEC Editor Standard. EDT is the only other DEC editor related to the standard. The readers of this newsletter need no introduction to TECO. The following table gives a brief view of what I consider the chief advantages and disadvantages of both KED and TECO.

TECO

ADVANTAGES	DISADVANTAGES
Programmable	Obscure mnemonics
Powerful editing commands	Limited error recovery
Expression evaluation	Inconvenient file handling
Generic search types	Limited human engineering
Multiple variables	

KED

ADVANTAGES	DISADVANTAGES
Easy to learn	No macro capability
Good error recovery	Limited variable facility
Convenient file handling	No hard copy editing
Good human engineering	

The primary difference I see between KED and TECO is philosophical in its basis. KED is designed from the ground up as an interactive video text editor. The basic goal of the user interface is that one can guess from the function names what any effect will be. In those cases where the guess is incorrect, one can easily recover in context by using the inverse function. TECO on the other hand is basically a text programming language and incidentally an interactive editor. One might properly view TECO as text language interpreter which provides an "immediate" mode of operation. TECO has had a very long period of development and consequently has a very rich set of operations and functions. In my view this development has left TECO with the disadvantage of a very obscure set of keywords.

Although there is no direct relatonship between the editing semantics and the file handling capabilities, KED also has a more natural file handling system. KED scrolls in the text file with no reference on the user's part to edit pages or extents. The user is free to move forward or backward to the limits of the file. In addition KED can maintain an auxiliary input file and an auxiliary output file. In order to rearrange a file with KED and put portions of it in an output file, one need only locate to the various elements in the order they are wanted and write them out. One never needs to close and reopen the input file. The most common compound edit operation, the global replace, is implemented in KED. It is called the substitute function and can be used either for an automatic global replace or a verify interactively and replace. In the global flavor it is equivalent to the TECO command:

```
        <@FN/STRING/STRING/>
```

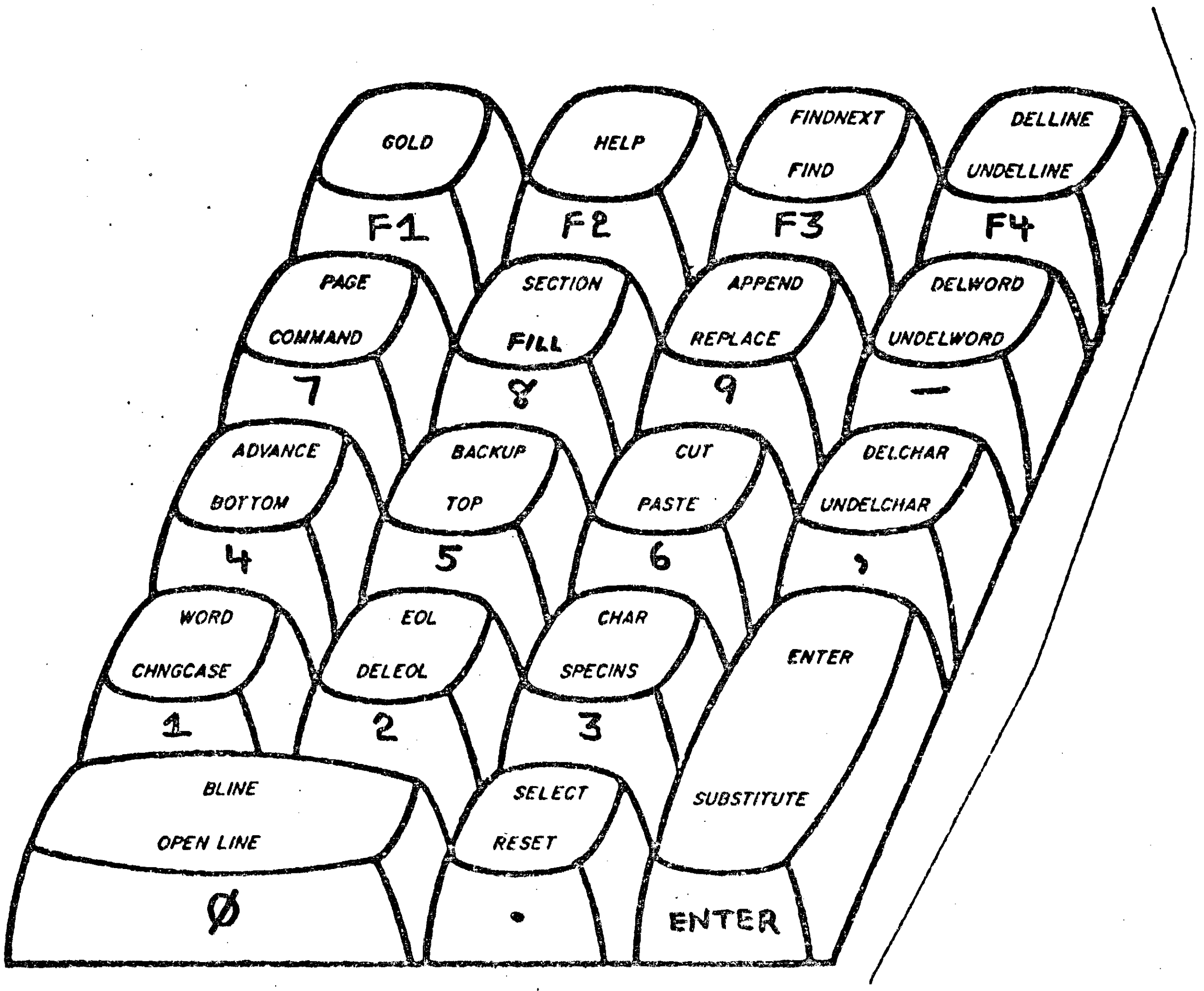
Since TECO is a general text processing language, one can implement almost all of KED in TECO. Indeed, the display management code in KED is based on Herb Jacob's TECO display manager. Some of KED's commands are messy to implement in TECO and the file handling would require changes to the interpreter. KED on the other hand is more efficient since it is a direct implementation.

A KED Emulating TECO macro has been written for TECO-11 by D. J. Duffy. It is hoped that this will shortly become available from the DECUS program library.

KED has the additional "advantage" of lack of flexibility. KED is a fully supported DEC product and will be maintained as a standard on any systems on which it is implemented. TECO standards on the other hand depend on the continued interest of a community of interested users.

In summary, TECO is far and away the best text processing language available over the whole range of DEC computers (and many others as well). The only way a threat could develop for TECO in this area would be for DEC to develop a good SNOBOL for all the systems (an unlikely occurence). KED on the other hand is unsurpassed for ease of learning and effectiveness in standard text editing situations.

Keypad Layout for VT100
Lower Function is GOLD



DELETE	RUB CHAR
LINEFEED	RUB WORD
CTRL/U	RUB LINE
CTRL/W	SCREEN UPDATE
CTRL/Z	CANCEL (PROMPTS)
CTRL/C	CANCEL (MOVEMENT)
<GOLD>nnn	REPEAT ANY FUNCTION

[OPEN] INPUT file		Open Auxiliary Input file
[OPEN] OUTPUT file		Open auxiliary output file
CLOSE		Close Auxiliary output file
PURGE		Purge Auxiliary output file
INCLUDE options		Read from input file
SKIP options		Ignore from input file
WRITE options		Write to output file
options	nnn PAGES	Pages as defined
	nnn LINES	Lines
	REST	Remainder of file
CLEAR PASTE		Clear paste buffer
EXIT		Close files and restart editor
QUIT		Purge files and restart editor
SET [ENTITY] PAGE defn		Define a page
SET [ENTITY] SECTION defn		Define a section
defn	' or " string of chars ' or "	
	nnn LINES	
SET [SEARCH] GENERAL		General searches
SET [SEARCH] EXACT		Exact searches
SET [SEARCH] BEGIN		Cursor at beginning of target
SET [SEARCH] END		Cursor at end of target
SET [SEARCH] BOUNDED		Bound search on PAGE definition
SET [SEARCH] UNBOUNDED		Searches are not bounded
SET [SCREEN] 80		Set screen width
SET [SCREEN] 132		
SET [SCREEN] LIGHT		Screen to reverse background
SET [SCREEN] DARK		Screen to normal background
SET QUIET		Errors do not sound terminal bell
SET NO QUIET		Errors sound terminal bell

HARVARD TECO

This article describes Harvard TECO version 22.5 which runs on a DECsystem-1Ø. Harvard TECO is based upon DEC's TECO for the -1Ø, version 22. (TOPS-1Ø TECO version 21 was the 'traditional' TECO that OS/8 TECO and TECO-11 are based on; whereas the current TOPS-1Ø TECO by DEC is version 24.) Information for this article comes from a memo dated May 1972 and is probably considerably out-of-date. Recent information about Harvard innovations to TECO would be appreciated. Comments in italics are the editor's personal opinions.

1. A new output file may not be opened until an existing output file is closed explicitly.
   (A good safety precaution. TECO-11 works this way too.)

2. Y and _ commands have been replaced by EY and E_ commands respectively.
   (Again, a good safety feature. However, I like TECO-11's and OS/8's Yank protection algorithm better.)

3. Control characters are permitted in searches and inserts just as they were in DEC's TECO version 21.
   (A wise fix. I feel DEC made a mistake when they prohibited control characters from insert strings. I think that the only restricted characters should be CTRL/E, CTRL/Q, CTRL/R, CTRL/S, and CTRL/X and CTRL/N. Future enhancements can all involve CTRL/E constructs so as to avoid incompatibilities with existing macros.)

4. Space is ignored as a TECO command.
   (Very wise. Using space to mean plus upsets all my ideas of readability and good programming practice.)

5. FB bounded search command has been implemented. It takes an argument that is the same as the argument allowed on line commands such as K, T, and X. In other words,

nFBtext$	searches the next n lines
m,nFBtext$	searches between pointer positions m and n.

   The original pointer position does not change if the bounded search fails.
   (I like this scheme better than the scheme used by version 29 of TECO-11. It is easier to remember and use.)

6. The nFSstring1$string2$ command makes n replacements.
   (This is incompatible with all the TECOs at DEC. Its functionality can easily be achieved with the n<FSstring1$string2$> command.)

7. The FC (Find and Change) command is a combination of the FS and FB commands. Thus m,nFCa$b$ finds the first occurrence of "a" between buffer pointer positions m and n and replaces it with the string "b". *(Neat!)*

HARVARD TECO (cont.)

8. The n↑E command can be used to set the form feed flag.

9. The :X command works like an X command but deletes the extracted text from the text buffer. In this case, the X is mnemonic for Xfer rather than eXtract.
   (Many people believe that this is the more useful option. However, I believe it would be too devastating to make such a change in existing TECOs. Perhaps the thing to do is to make it a user option by putting this facility under the control of some ED bit.)

1Ø. Harvard TECO contains a form of the EU command for upper/lower case conversions. However, it has a very strange and incompatible syntax, so I won't describe it here.

11. Position Recording Arguments (VB, VI, and VS).
    VB is set to the buffer pointer position at the start of any search command. Thus, if the search failed, VBJ gets you back to where you were. *(useful)*
    VI returns the value of the buffer pointer position immediately before the most recent insert command. Thus VI,.K kills an incorrect insert.
    (The TECO-11 command FR$ does the same thing.)
    VS returns the value of the text buffer position directly in front of the string matched by the most recent successful search.

12. End-of-Line Counting Arguments (VL and VZ).
    VZ is the number of end-of-line characters in the text buffer. VL is the number of end-of-line characters preceding the current text buffer pointer position.

13. Line Formatting Aids (VC and VH).
    VC returns the number of ASCII characters between the beginning of the current line and the current pointer position. VH is similar to VC but gives special consideration to characters such as tab, carriage-return, and backspace. Thus VH approximates the horizontal position of the current pointer on a printed page.
    (The equivalent of VH would be very convenient for the VTEDIT macro. TECO-11 might want to implement this via something like a 0:↑Q command.)

14. n↑O\ inserts the octal representation of n into the text buffer.
    (General octal radix mode is a better solution to this problem.)

15. n:; exits an iteration if n is negative.
    (This seems very useful to help writing structured code.)

16. :T command is similar to T command but non-printing characters have special echo. For example, carriage-return types out as <CR> .

17. : modifier only affects the very next command and would not extend across commands until it found a command for which it were meaningful.

BB&N TECO

This version of TECO was written by BB&N (Bolt Beranek and Newman, Inc.) and runs under the TENEX operating system. Some of this information comes from Bill Plummer at BB&N. Also, I wish to extend my deepest thanks to James Neeland of the Hughes Research Laboratories who sent me a manual. This information reflects BB&N TENEX TECO as of October 1973.

I am a bit reluctant to call this editor a TECO variant because it contains several radical departures from traditional TECO. These changes are extremely inconsistent with all the TECOs at DEC. An experienced DEC TECO user would have a lot of trouble trying to use this 'TECO'.

The main incompatibilities are:

(a) The R command does a replace (it does what the FS command does in DEC's TECOs).
(b) A single ALTMODE is the immediate action command that means start execution of this command string. In order to enter a single ALTMODE into a command string without beginning execution, you must use the immediate action command CTRL/D which inserts an ALTMODE into the command string.
(c) For no reason that I have been able to discern, the E commands have all been changed to ; commands. Thus, to open a file for input, you type ;Rfilespec$.
EF became ;C , EW became ;W , etc.
The original ; command is now ; followed by a space (*UGH!*)

However, there are enough similarities between this TECO and DEC's TECOs, that some of their extensions are worth mentioning. The following lists some of its extensions to DEC's TOPS-1Ø TECO version 24. My comments are in italics. Reader comments are solicited.

1. Negative arguments to a search command cause the search to proceed backwards.
   (A useful feature; already in TECO-11.)

2. A backward search initiated when the text buffer pointer is currently at the start of the buffer will automatically bring the pointer to the end of the buffer and then the search will commence.

3. A Ø repetition count to a search or replace command is a no-op.

4. EAfilespec$ opens the file for append.

5. Backspace, if typed as the very first character in a command string is an immediate action command that performs an effective -LT$$ command. Linefeed, if typed as the very first character in a command string, is an immediate action command that performs an effective LT$$ command.
   (This feature is going into TECO-11.)

BB&N TECO (cont.)

6. n↑H tells TECO what kind of terminal you are running on.

7. A new Q-register, by the name of @ is defined. It has a special property. If TECO is halted, and the resulting core-image is saved, the result is a runnable program. When that program is run, it starts up by executing the contents of Q-register @, i.e. an effective M@ command occurs. This feature is useful for creating runnable TECO programs.

8. The command m,nG gets a copy of the (n-m) characters beginning with the character after pointer position m and inserts it at the current pointer position.
nG makes a copy of the next n lines leaving the buffer pointer between the copies. Gq works as it always did.
(These commands wreck havoc on TECO's algorithm for skipping commands. A possible use for m,nGq might be to get the mth through nth characters from Q-register q into the text buffer.)

9. The ;P (Pick-up) command performs a ØA command combined with a C command, returning the value accessed.

1Ø. m,nV types the m-l lines before the current line, types the current line, then types the n-l lines after the current line. nV is the same as n,nV . V means lV .
(A nice compatible extension to TECO-11's V command.)

CCA TECO

This version of TECO is a modification of BB&N TECO. It was implemented at the Computer Corporation of America in Cambridge by someone whose initials are MRC. The following information was obtained from Donald Eastlake III and appears to be current as of July 1977. The following is a partial list of differences between CCA TECO and BB&N TECO:

1. ALTMODE was restored to its original status. To quote from MRC: " *The misfortune of using ↑D for a single altmode and altmode for double altmode has been flushed. ↑D is now an ordinary character, and TECO behaves like every other TECO in the world.* "

2. Space, if typed as the very first character in a command string, is an immediate action command that performs an effective ØTT$$ command.

3. Carriage-return and line-feed throw away any numeric argument preceding them.

4. n↑T types out the character whose ASCII code is n.
(OS/8 TECO and TECO-11 already do this.)

Documentation also indicates that phase-of-the-moon commands will shortly be implemented (I gather to be compatible with MIT's TECO.)

WPI TECO

WPI TECO (also known as WACCC TECO) was written at the Worcester Polytechnic Institute Computer Center and is based on DEC's TECO (version 23) for TOPS-1Ø. The following information (current as of April 1977) is based on WPI TECO version 23T. Comments in script type are the editor's personal comments.

1. EP (Page Control) command added. I don't understand the 1EP command.
   2EP returns the current page number.
   4EP sets 'truth-in-paging mode'. In this mode, Y and P commands must proceed to form feed characters, expanding core as necessary. If this mode is not set, Y and P commands may stop when the current buffer is nearly full, setting the form-feed flag (↑E) to Ø to indicate that a partial page has been read in.
   8EP causes the terminal bell to ring (*to wake you up*) upon the completion of a search or P command.
   16EP causes the EB command to create a .OLD backup file instead of a .BAK backup file. This allows for an extra level of backup. *(This mode is not necessary on operating systems that use file version numbers.)*

2. EQq types the text stored in Q-register q.
   (OS/8 TECO and TECO-11 use the :Gq *command to do this.)*

3. EXQ and EGQ are quick versions of the EX and EG commands respectively. They are much faster than EX and EG because they don't mung over each character. Instead, they pad the current buffer with nulls, and then block for block copy the rest of the input file to the output file without looking at the characters. Nulls encountered later in the input file will not be purged. Presumably, these nulls will be purged the next time the file is edited.

4. n,mStext$ searches the text buffer between pointer positions n and m for the specified text.
   (This seems a bit more logical than the TECO-11 bounded search command.)

5. EDfilespec$ inserts into the text buffer (at the current pointer position) a directory listing of the specified directory. Wildcards are permitted. If only a single filename is specified, then this command does not do any insertion into the text buffer, but instead returns a value of -1 or Ø specifying whether or not the file exists.

6. EKfilespec$ deletes the specified file.
   (This capability is missing from TECO-11.)

7. EC core compression command. Causes TECO to relinquish any extra core that may have been obtained by expansion of its low segment.
   (A more general command would be nEC which would tell TECO to expand

WPI TECO (cont.)

8. EAprogram$ causes TECO to abort and run the specified program.
(This is not as general as the OS/8 TECO EGcommand$ *which allows an arbitrary monitor command to be executed. Naturally,* EGr program$ *is a valid TECO command.)*

9. EJ job information command. EJ returns the current job number. 1EJ returns the user's project,programmer number in the text buffer. n,mEJ does a GETTAB of n,m and returns the result.

10. nP has been changed to look for actual form feed characters. In DEC's TECO, the nP command is identical to n<PWY> which is not quite the same thing, since the Y command might stop on buffer full rather than form feed encountered.

11. The EZ and EO commands were removed.

12. TECO may be initialized by a TECO.INI file. If a file named TECO.INI exists in the user's area, it will be read and the data in it (which must be in a particular rigid format) is used to load up TECO Q-registers. If Q-register Ø is loaded by this scheme, TECO will perform an MØ command.
(Because of the special format of the INI file, this method of initialization is not as general as the scheme used by Stevens TECO which effectively treats the text in the .INI file as a macro to be executed. This method is also used by TECO-11 and will soon be used by OS/8 TECO.)

TECA

TECA is a display version of WPI TECO. The meager information I have on it was received in January of 1977. It was written by Andy Nourse at DEC and is probably not in use at very many places outside of DEC. Some of its features are:

1. It maintains a window into the text buffer and updates it after each command string execution.

2. nW command sets size of display window.

3. V command takes same arguments as T command but moves the window to the specified text area.

4. n↑← sets the terminal type. It can also be used to set terminal characteristics.

5. All ASCII characters are legal Q-register names. Each Q-register has two parts and can contain both a text string and a numeric value.
(Having upper case and lower case Q-registers be different is a real lose.)

STEVENS TECO

This version of TECO runs on a DECsystem-]Ø and is based on DEC's TECO (version 23) for TOPS-1Ø. It was written at Stevens Institute of Technology by John Ptochnak, Gary Brown, and Robert McQueen. The following information describes the differences between Stevens TECO version 123 and DEC's TECO version 23. It is accurate as of January 1976.

1. The Y command has been replaced by EY. (Y still works within macros.)

2. The ER and EW commands default unspecified fields from corresponding previous commands.

3. A log-file capability has been added. ELfilespec$ closes any previous log file and opens a new one. A copy of all TECO type-in and typeout goes into the log file. EX automatically closes the log file. Switches such as /NOIN and /NOOUT are permitted to specify that only typeout (or only type-in) is to be logged. /APPEND appends to the log file. TECO macros can dynamically change log modes via an nEL$ command.

4. ENfilespec$ renames the current input file to the name specified. EN$ deletes the current input file.

5. EDfilespec$ instructs TECO to run the specified program whenever TECO is exited. ED$ removes the pending run of such a program.

6. :m,n↑T allows TTCALLs to be executed.

7. The EEfilespec$ (Edit Execute) command saves the state of TECO in the file specified. Subsequent running of that file, causes this image to start up again. It continues executing commands that followed the EE command.
   (Several people have told me that this is an extremely useful command. It is somewhat more convenient to use than the MUNG *command.)*

8. Commands such as ER, C, R, J, etc. can be colon modified and return a value of -1 for success and Ø for failure.

9. The EX command has been changed to act like the EXQ command of WPI TECO.

1Ø. The second CTRL/A in a CTRL/A command now forces out the text. (Under DEC's TECO, terminal characters were buffered up until a line feed was output or the program terminated.)

11. Negative searches are permitted and cause the search to proceed backwards.
    (Most new TECOs are allowing this.)

12. m,nSstring$ searches for the specified string which must begin between pointer positions m and n. If m>n, then the search proceeds backwards. Bounded search also works for the FS command.

STEVENS TECO (cont.)

13. EAfilespec$ (Edit Append) opens the specified file for update (appends at the end).

14. Wq is similar to Mq but jumps to the Q-register. It stores nothing on TECO's internal push-down stack and does not expect you to ever return to the macro level in which the Wq command is issued. It is useful for large TECO programs that are nested so deeply that they might otherwise get PDL overflows.

15. * is a valid Q-register name.

16. EPfilespec$ reads the specified file into Q-register *.

17. EIfilespec$ is the same as EPfilespec$M* executing the contents of the specified file.
(This is similar but not identical to TECO-11's EI command. The difference is that TECO-11 thinks the characters are coming from the terminal, so certain characters will have their immediate effect as immediate action commands rather than as TECO commands. For example, under TECO-11, a CTRL/U in an indirect file would erase the current command line. I do not think that is good and I like the Stevens TECO algorithm better. TECO-11 could be modified by setting a switch on an EI command to disable checking of immediate action commands (except $$) until input returns to a physical terminal.)

18. If a file by the name of TECO.INI resides on the user's area, TECO will EI this file upon startup.

19. The FDstring$ command searches for the specified string and deletes everything from the original pointer position to the new pointer position if the string is found. Bounded Find and Delete is also legal.
(I have been told that this command may change to FK and that FD would mean delete the string that was found.)

2Ø. Search failures now preserve the text buffer pointer.
(This may be very convenient but will kill existing TECO macros.)

21. The EC command will cause TECO to garbage collect and shrink to its original size.

22. n↑P pages out to page n. n↑Y yanks in text until page n has been read in.

23. n,m= is the same as m= but then types out the character whose ASCII code is n.
(The OS/8 and TECO-11 n↑T and n:= seem more useful.)

24. n,m↑G performs a GETTAB UUO.

25. n↑F returns the terminal number being used by job n.

26. Qq= types the text stored in Q-register q.
(This would not work on TECO implementations that have split Q-registers.)

STEVENS/TEXAS TECO

is version of TECO (for the DECsystem-1Ø) is based on Stevens ECO version 123. It was written at Texas University by Clive Dawson. The differences between version 124 of this TECO and the original v123 Stevens version are described below:

1. The CTRL/G<space>immediate action command was fixed so that it never types literally even if 1ET is set.
2. n↑G does a PEEK.
3. i,j↑G does a GETTAB.
4. ↑G returns your job number.
5. n↑F returns the TTY number of job n, -1↑F is your TTY no.
6. ↑G. (bell-dot) immediate action command retypes the entire command string.
7. The EB command always writes into your area and allows a /INPLACE switch to mean don't create a backup file.
8. n\\ and \\ are the same as n\ and \ but work in octal radix. *(This does not appear to be as general as TECO-8's and TECO-11's general octal radix mode.)*
9. The linefeed, backspace, and semicolon immediate action commands were implemented. If typed as the very first keystroke after TECO's prompt, they simulate the action of the LT, -LT, and ØLT commands respectively.
10. The EK command cancels any outstanding EW, EB, or EA commands.
11. CTRL/Gq inside a string substitutes the contents of Q-Register q into that string.
    (This proliferation of new special characters in search strings seems to me to be bad. It just lessens the number of characters that can be easily typed directly. I would prefer to see all new constructs be implemented as CTRL/E constructs and allow all other control characters to be legal inside string arguments.)
12. The /APPEND switch is allowed on the filespec in an EW command and converts the EW command to an EA command.
13. The TECO monitor command can now have a /READONLY switch which causes TECO to perform an ER rather than an EB.
14. CRT support. This version knows over a dozen types of scope terminals and handles command line scope editing correctly (rubout, backspace, CTRL/U, etc.)
    (Users of TECO-11 and TECO-8 already know what a great boon this is. When you rubout a character, it really disappears from your screen. This feature works even when rubbing out obscure characters like tab and linefeed.)

This version of TECO is widely in use in the field. There are probably between 5Ø and 1ØØ installations using this version. It is available from Clive Dawson, Computation Center, University of Texas, Austin, Texas, 78712 for a nominal postage and handling fee. Send Clive a MAGtape (he'll write 9-track 8ØØ bpi).

QUESTIONS AND ANSWERS

This column will attempt to answer questions submitted by readers on any TECO-related topics. Our first set of questions were submitted by David Yost of Hollywood California and apply specifically to RT-11 TECO.

Q1. by David Yost.

How do I reclaim the right brace, }, as a printing character? TECO seems to be translating it into an ESCAPE.

A1. by Roy A. Auerbach (Texas Tech University, Lubbock, Texas)

Both the right brace and tilde (octal 175 and 176) are interpreted by RT-11 TECO as an ESCape. The file TECO.SAV may be patched to eliminate this. Address 124 from the bottom of TECO.SAV must be patched. The value of 227ØØ at this address must be changed to 2Ø7. Running version 28 TECO with RT-11 V-Ø3, my bottom address is 3756. A sample run to patch with my system would be:

```
R PATCH
TECO.SAV
3756; ØR
Ø, 124/ 227ØØ  2Ø7
E
```

Q2. by David Yost

Is there a way to get the text buffer pointer to stay where it is when a search on the current page fails?

A2. by Stan Rabinowitz

Yes. In TECO-11, V28, the command Ø,nStext$ is exactly the same as the command nStext$ except that it has the desired property that the pointer position remains unchanged after search string failure. This command can of course be colon modified so that you can detect the failure.

Q3. by David Yost

The manual I have refers to version 27 of TECO-11, but I have version 28. What do I do?

A3. by the editor

Consult page 7 of issue number 1 of the Moby Munger for a complete list of differences between V27 and V28. The main enhancement was that Xq and ↑Uq permit a colon modifier to mean <u>append</u> to Q-register instead of load.

Queries should be submitted to the editor, Stan Rabinowitz, 6 Country Club Lane, Merrimack, NH 03054.

A NOTE ON THE EN COMMAND

by Jim Burrows

The EN comand, which allows wild-card file lookups is extremely useful, but at least on RSTS, has one drawback. on many disk structures the act of editting can change the order of the files in the directory such that the EN command will return some file names twice and others not at all. A simple coding technique can be used to avoid this.

Basically, the public structure on a RSTS system can span multiple disks, thus editting a file can cause it to migrate from one disk to another changing the order of the directory. Furthermore, unless the NFF bit is set on a disk, new files are added to the end of the directory. Thus if you edit the first file in a directory, it becomes the last file in the directory and EN probably will find it later.

For those not familiar with the EN command, I will briefly review its use. It takes a filespec arguement like the EB or ER commands. Rather than open a file, the filespec is used to perform wildcard directory look-ups. Subsequent invocations of EN without a filespec cause the name of the next file matching the spec to be loaded into the '*' pseudo-Qreg. Thus, if we execute the command:

```
@EN"foobar.*"
```

TECO will perform the necessary setup to allow it to find all files with the filename FOOBAR. Qreg * does not yet contain the name of a file at this point. To get the name of the first file, we must execute the command:

```
@EN""
```

So far there is no problem. The name of the first file matching our wild-card spec will be in Qreg *. However, if we edit the file, say with the command:

```
@EB"^EQ*"
```

we may change the order of the files in the directory. a subsequent call will return us the name of the second file which CURRENTLY matches the wild card spec. If we have indeed changed the order of the files, the file that used to be the second may now be first, and we may get the file that was originally third.

This problem can be avoided if we first gather the filenames and then edit the files. The follwing macro is an example of this technique. In it we will change all occurances of "DNDA" to "TMP:DNDA" in all files that match the wildcard spec "DND???.BAS". The list of filenames is kept on the stack. The :] command is used to detect the end of the stack, and thus the macro will only work on versions of TECO which have implemented this feature.

A note on the format of the macro. The author and several other programmers at DEC who do a lot of work in TECO, have adopted the use of a program to strip the comments and unnecessary whitespace from TECO macros to increase efficiency of the macro when executed. Thus the following conventions are used:

1) All comments start with "!*" and end with "*!" so that they may be distinguished from labels.

2) No literal strings contain CR or LF so that leading and trailing whitespace can be deleted on a per-line basis.

```
                   !***************!
                   !* set up code *!
                   !***************!

<:]1;>                     !* Clean off the stack *!
@EN"DND???.BAS"            !* Initialize the wildcard look-up *!

                   !****************************!
                   !* First get the file names *!
                   !****************************!

<@:EN"";                   !* Get first file name into Qreg * *!
  G* ^YXl ^YK [l           !* Get it in memory, push it, delete it. *!
>                          !* And do it again, till the :EN fails *!

                   !************************!
                   !* Now go do the edits. *!
                   !************************!

<:]l;                      !* Pop the first name off the stack. *!
  @EB"^EQl"                !* Edit (indirect) the file *!

  <@FN"DNDA"TMP:DNDA";V>!* Do the edit throughout the file *!

  EC                       !* Close the file *!
  @^A"  Done with " :Gl !* Tell 'em what you did *!
  13^T 10^T                !* Return the carriage *!
>                          !* Go back and get the next file *!
$$
```

TECO MACRO OF THE MONTH

This macro creates a distinctive IN USE message on a user's display terminal. It prints a user message on the top of the screen, it prints IN USE in block letters in the middle, and it displays a canned message at the bottom of the screen in ticker-tape fashion.

```
ET#1ET ^A$H$J                    ^A
Z"N^A   ----->     ^A ' HT Z"N^A     <-----^A ' HK ^A


    IIIIII    NN      NN              UU      UU    SSSSSSSS  EEEEEEEEEE
    IIIIII    NN      NN              UU      UU    SSSSSSSS  EEEEEEEEEE
      II      NNNN    NN              UU      UU  SS          EE
      II      NNNN    NN              UU      UU  SS          EE
      II      NNNN    NN              UU      UU  SS          EE
      II      NNNN    NN              UU      UU  SS          EE
      II      NN  NN  NN              UU      UU    SSSSSS    EEEEEEEE
      II      NN  NN  NN              UU      UU    SSSSSS    EEEEEEEE
      II      NN    NNNN              UU      UU          SS  EE
      II      NN    NNNN              UU      UU          SS  EE
      II      NN    NNNN              UU      UU          SS  EE
      II      NN    NNNN              UU      UU          SS  EE
    IIIIII    NN      NN                UUUUUU    SSSSSSSS    EEEEEEEEEE
    IIIIII    NN      NN                UUUUUU    SSSSSSSS    EEEEEEEEEE


In use by TECO fanatic John Doe


^A

IThis terminal is presently in use.  $
IUnfortunately, I am forced to leave my post for a short period of time.  $
II will be back soon.  $
IWoe unto him who steals my terminal during my absence.  $
IThe security of the whole free world depends upon my accomplishing my $
Iduties at this terminal.  $
IUnder no circumstances is this terminal to be used by anyone $
Iwithout my permission.    $

0U1 0,80XA ZJ GA J
<Q1,Q1+80T %1-Z+79"E1U1' ^A␍^A>
^C^C
$$
```

where ␍ denotes a CR (with no LF)

This macro runs under all DEC operating systems. On some systems it is first necessary to set up the terminal with a SET TTY NO CRLF command or its equivalent. It is run from RSX, RSTS, IAS, OS/8, VAX, and RT-11 via the command MUNG INUSE,message where "message" is any user message. It is run from TOPS-1Ø via the commands: R TECO
*ERINUSE.TEC$YHXZHKImessage$MZ$$.

COMPATIBILITY CORNER

by Stanley Rabinowitz

The following TECO commands are identical on TECO-11 v29, OS/8 TECO v5, and TOPS-1Ø TECO v24, or have only slight incompatibilities:

ER	EW	EB	EF	EX	A	Y	P	PW	J	C
R	L	I	nI$	↑I	n\	\	+	-	*	/
&	#	(n)	T	↑A	=	==	↑T	D	K	B
Z	.	,	H	S	N	_	FS	FN	:	@
ØA	↑F	↑H	↑↑x	<n>	!	O	"E	"F	"G	"L
"N	"S	"T	"U	U	%	X	Q	G	M	↑E
↑N	EH	EU								

In addition to the above, the are a large number of other features that are compatible, such as the immediate action commands, the execution time commands, and the match control constructs.

Also, there are a large number of commands on each operating system that are extensions to TECO, but are not incompatible with each other.

In the last issue of the Moby M., we listed a set of compatibility issues that are yet to be resolved. No one sent in any comments. These issues and their results are described below. Note that many of these issues have already been resolved; and many cannot be resolved until the -1Ø group is ready to re-release TOPS-1Ø TECO. Notation: $ denotes an ALTMODE and (^X) represents the single character CTRL/X.

1. Implementation Dependent Date

Problem: The (^B) command returns an implementation-dependent value.

Discussion: It is unlikely that the user really wants the date in a strange encoded form. He will probable decode it first thing. Proposal (a) below incurs a possible race condition around midnight of New Year's Eve.

Proposals:

(a) Replace (^B) with n(^B) command. Arguments as follows:
1(^B) returns year (0-1999)
2(^B) returns month (1-12)
3(^B) returns day (1-31)

(b) Have (^B) insert the date into the text buffer as a character string.

(c) Leave (^B) as system dependent.

Decision: issue still pending

2. Implementation dependent time

Problem:	The (^H) command returns an implementation-dependent time of day.
Discussion:	internal word size of -11 cannot store number of seconds in a day. Proposal (a) below incurs a race condition. If at 4:59:59 a program accesses the hour and finds it to be 4 and then the program accesses the minute and finds it to be 00 (since it is now 5:00), the program will incorrectly think the time is 4:00 when it is actually 5:00.

Proposals:

(a) Replace (^H) by n(^H) command with arguments as follows:

1(^H)	returns number of hours since midnight (0-23)
2(^H)	returns number of minutes since last whole hour (0-59).
3(^H)	returns number of seconds since last whole minute (0-59)
4(^H)	returns implementation-dependent number of ticks since last whole second.

(b) Have (^H) insert the time as a character string in the text buffer.

(c) Leave (^H) to be system dependent.

Decision: issued still pending

3. Incompatible form feed command

Problem:	The (^L) command is ignored on -8 but prints a form feed on -10 and -11.
Discussion:	-10 use of (^L) can be performed by the equivalent command (^A)(^L)(^A)

Proposals:

✓ (a) change -8 to conform with -10 and -11.

(b) Change -10 and -11 to conform with -8.

Decision: It was decided that the -8 should conform. OS/8 TECO was changed to conform as of v5.

4. Radix control problem

Problem: -8 and -11 allow TECO to work in two modes (octal radix and decimal radix mode). The current mode of TECO affects many commands. The -10 does everything in decimal, except for a few commands, and treats (^O) as a modifier for numeric strings.

Discussion: Octal mode seems more important on small machines.

Proposals:

(a) change -10 to conform with -8 and -11.

(b) Change -8 and -11's (^O) and (^D) to (^O)$ and (^D)$.

(c) Change -8 and -11's (^O) and (^D) to (^O)(^O) and (^D)(^D) .

(d) Define a bit in the ED flag, such that if this bit is set, (^O) and (^D) will only apply to the next digit string.

(e) Make the O(^R) command mean that future occurrences of (^O) and (^D) only apply to the next digit string.

Decision: Issue still pending.

5. Obsolete use of (^R)

Problem: On -8 and -11, (^R) is a redundant command, having the same meaning as FS.

Discussion: (^R) was implemented before FS was implemented on -10. Then FS was put in for compatibility with -10. (^R) was kept for compatibility with old -8 versions of TECO.
(^R) is easier to type than FS (?).

Proposals:

[✓ (a)] Remove (^R) command from -8 and -11 to free up character for future use.

(b) Leave duplicate command in

Decision: It was felt that (^R) was an unecessary command. (^R) is no longer in TECO-11 or OS/8 TECO.

6. Incompatible (^V) command

Problem: On the -10, (^V) is used to go into 'translate to lower case' mode. On the -8 and -11, (^V) returns TECO's version number.

Discussion: letters V and W are firmly locked into lower and upper case mnemonics on -10.
(^V) on -8 and -11 is not a critical command, unlikely to change running of old macros.

Proposals:

✓ (a) Change (^V) command on -8 and -11 to EO command. For next release on -10, upgrade EO value to -10's TECO version number (e.g. 23).

(b) Change (^V) on -8 and -11 to EV .

Decision: (^V) and (^W) are heavily used on the -10 and should not change. (^V) on the -8 and -11 is not a crucial command and EO would be just as useful.
This change was made to OS/8 TECO as of V5.
This change was made to TECO-11 as of v29.

7. Incompatible (^W) command

Problem: On the -10, (^W) is used to go into 'translate to upper case' mode. On the -8 and -11, (^W) is used to force a display update on scope displays.

Discussion: Both uses are very common on their respective machines.
-11 will eventually want to support upper and lower case, however, they're not sure they like the -10's syntax.

Proposals:

✓ (a) Change (^W) on -8 and -11 to W .

Decision: (^W) is needed by the -10. On the -8 and -11 it is only used by scope TECOs and could more easily change. PW would be considered a single 2-character command and therefore should not be confused with the two commands, P followed by W.
(^W) was changed to W in OS/8 TECO v5 and in TECO-11 v29.

8. Incompatible (^Z) command

Problem:	On the -10, (^Z) causes return to operating system after finishing I/O and closing output file. On -8 and -11, (^Z) returns total number of characters in Q-register storage.
Discussion:	-8 and -11 usage of (^Z) is not very critical. -10 changed (^G) command to (^Z) for EO level 2. -8 still uses (^G). The (^Z), alias (^G) command is very dangerous and can be accomplished by the equivalent commands: EFEX. Use of (^Z) as an immediate-mode command is a different issue.
Proposals:	(a) change -8 and -11 usage of (^Z) to some other command.
	(b) switch (^Z) back to (^G) on -10.
	(c) Remove -10 use of (^Z) from TECO and -8 use of (^G).
Decision:	OS/8 TECO removed the (^Z) command as of v5. However, issue is still pending.

9. Space problem

Problem:	On the -10, spaces are normally ignored, however, between digits, space is treated as plus. On the -8 and -11, spaces are ignored everywhere.
Discussion:	-10 use of space is traditional (but ugly). Ignoring spaces everywhere is more consistent.
Proposals:	(a) change -10 to ignore spaces everywhere
	(b) change -8 and -11 to treat space as + in certain constructs
	(c) change all systems to give error message if space appears within digit string.
Decision:	It was decided that proposal (a) seems best, however issue is still pending.

10. Incompatible "A command

Problem: On -10 and -11, "A is an 'execute if alphabetic' command. On -8, n-m"A is an 'execute if n>=m' command.

Discussion: -8 usage was implemented before -10 usage came along. Normal compare conditionals such as "N, "L, etc. do not always work on the -8 because the -8's maximum number (unsigned) is 4095, yet the -8 allows almost 4000 characters (maximum) in buffer. When comparing buffer pointers, for example, anomalous results can occur with "L command since it requires a signed argument whereas the . command produces an unsigned value. This was gotten around by creating the after and before commands ("A and "B) which check the link instead of the sign bit. This is not a problem on the -10 and -11 because of their increased word sizes.

Proposals: (a) Change "A and "B commands on -8 to "< and ">

Decision: OS/8 TECO removed "A and "B commands as of v5. The problem was gotten around by increasing OS/8 TECO's arithmetic precision to 13 bits.

11. Definition of alphanumeric character

Problem: The "C command needs to know what an alphanumeric character is. This concept differs between implementations. Are the character ., $, %, etc. alphanumeric?

Proposals: ✓(a) Allow this command to be machine dependent.

Decision: It was decided that this command needed to be machine dependent to allow users to search assembly sources for symbols.

12. GeneralizAtion of % command

Problem: The -8 and -11 generalized the % command so that the argument is added to the specified Q-register. On the -10, n%q always means 1%q

Discussion: Code 'fell out' on -8 and -11 implementations. Felt to be a very useful extension.

Proposals: (a) Change -10 to conform with -8 and -11.

Issue is still pending

13. Unspecified repeat counts for iterations

Problem: If no repeat count is specified for an iteration, each implementation uses a different default value.

Discussion: Largest number in machine was a likely default.

Proposals: (a) keep this default to be implementation-dependent.

[✓ (b)] Change all implementations to make the default truly infinite.

Decision: It is beleived that proposal (b) is easy to implement and will give users what they really want.
All DEC TECOs now treat an unspecified iteration count as being a truly infinite count.

14. = problem

Problem: = and == do different things on various implementations. On -11, they reset the radix, on -8 they do not. On -8 they print unsigned values, on -10 and -11 = prints signed value.

Discussion: There is no great need to be able to print a value in the current radix (as the -8 does) without knowing what the radix is. However, I can see cases where a macro may want to do something in a given radix, and then return to the user in the original radix.

Proposals: [✓ (a)] Have standard say that == prints the value in octal and does not change the current radix. = prints the value in decimal and does not change the current radix.

Decision: It was felt that most of the time (except occasionally on the -8), the user wants to see signed numbers. Thus = should print a signed value. The fact that the -11 == reset the radix was considered a bug and has been fixed long ago.

15. Generalized nA command

Problem:	On -10, nA is always the same as 0A on the -8 and -11.	
Discussion:	-8 and -11 generalization is fairly obvious. -10 users may be locked into current use, especially since user's might use 1A instead of 0A.	
Proposals:	(a)	change -10 to conform to -8 and -11 during next E0 level.
	(b)	change -8 and -11 to conform to -10.
Decision:	Problem is still pending.	

16. Location of @

Problem:	-10 manual claims @nS/text/ is legal but n@S/text/ is not. -8 and -11 manuals claim the reverse.	
Discussion:	In point of fact, both are legal on all implementations. Furthermore, @ is merely a modifier and need not be immediately adjacent to the search command. It may appear anywhere and merely affects the outcome of the next search executed.	
Proposals:	(a)	modify all documentation to describe actual behavior properly.
	(b)	pick a syntax and modify all implementations to use it and give error messages for all other uses.
	(c)	Require @ to be first character of a 2 or 3 character command. Thus @ may not occur alone, and must immediately be followed by S, N, _, FS, FN, or F_. Furthermore, : may also not occur alone, it must immediately be followed by @ or one of the above forms.
	[✓ (d)]	Ignore this issue.
Decision:	Problem seems too trivial to bother with.	

Filespecs treated as text

Problem: -11 treats filespecs as text and allows it to contain special characters (such as (^E)Qq) and allows it to be modified by @ modifier. -8 and -10 do not permit this.

Discussion: -11 usage is a nice generalization and causes all text-like occurrences to be treated uniformly.

Proposals: (a) Change -10 and -8 to conform to -11

Decision: OS/8 TECO now (v5) allows filespecs to be @-sign modified. Issue is still pending.

EH value is implementation dependent

Problem: The meaning of the value of the EH flag is implementation dependent.

Proposal: [✓ (a)] Change TECO-11's 3EH to 4EH to prevent conflicting with -10's 3EH. Reserve 3EH on -11 for future use.

Decision: TECO-11 (v27?) changed EH so that 1, 2, and 3 EH are compatible with the -10 (although 3EH is not implemented). The 4's bit in the EH flag is an extension to TECO on the -11.
OS/8 TECO implemented the EH flag compatibly with both TECO-11 and TOPS-10 TECO.

ET value is implementation dependent

Problem: The value of the ET flag is implementation dependent.

Decision: Issue is still pending.

Closing of files with EW

Problem: On the -10, EW closes the current output file. On the -8 and -11, EW creates a new output file and the old one is not replaced by the partially created temporary file.

Discussion: Both algorithms seem pretty poor.

Proposals: (a) Change all systems so that EW gives an error message if an output file is already open.

Decision: TECO-11 quite a while ago implemented proposal (a). OS/8 TECO and TECO-11 also implemented the EK command to 'kill' or take back the effect of an unwanted EW. Issue is still pending.

21. Required altmode on nI

Problem: the -8 does not require the altmode at the end of the nI command. It is required on the -10 and -11.

Discussion: $ was optional on -10 during EO level 1. The $ is necessary in order that this command can be correctly parsed and skipped over when encoutered in an unsatisfied conditional.

Proposals: [✓ (a)] make $ required on -8

(b) Make $ optional on -10 and -11.

Decision: Proposal (a) was picked, however, due to space and time limitations, OS/8 TECO may not implement the necessary error checking in the near future. Users should always put ALTMODEs after an nI command.

22. Passing values through Macros

Problem: the -8 and -11 allow arguments to be passed to macro calls. They allow macros to return values. Both are specifically prohibited on the -10.

Discussion: Macro arguments and values are as useful in TECO as functions are in FORTRAN, BASIC, and ALGOL.

Proposals: [✓ (a)] Change -10 to conform with -8 and -11.

Decision: It was found that TOPS-10 TECO v24 already conforms to this decision (proposal a), however their documentation didn't make this fact clear.

23. Negative arguments to searches

Problem: The -11 allows the argument to certain search commands to be negative. This causes the search to proceed backwards on the same page.

Discussion: This is a most useful and consistent extension.

Proposals: (a) Change -10 and -8 to conform to -11

(b) Leave status quo, call this a compatible extension and not an incompatibility. Change error message on -8 and -10 to read "feature not implemementedⁿ .

Decision: Issue still pending.

24. Strange W command

Problem: The W command is illegal on the -10 and -11. It is ignored on the -8.

Discussion: This is probably a bug in -8 implementation.

Proposals: (a) Change -8 to treat W command as an illegal command.

Decision: Problem went away after issue 7 was resolved. W is now a valid command on -11 and -8.

25. Alpha and Numeric Q-registers

Problem: -8 and -11 have two storage compartments in each Q-register; one to hold strings and one to hold numeric values. Both can retain their values simultaneously. On the -10, if a string is stored in a Q-register which previously contained a number, this number is lost.

Proposals: (a) Document around this problem. Have the 'standard' say that if a number is stored in a Q-register, and then the Q-register is read by a command requiring an alphabetic string, the result is unpredictable.

(b) Modify -10 to have 'split' Q-Registers.

(c) Treat -10 as subset of 'standard' TECO.

Decision: Issue is still pending.

27. Concatenated commands which return values

Problem: If two commands which return values are placed next to each other, the result is not the same on all implementations.

Discussion: For example, if there are 5 characters in the buffer, then the command Z3= produces 3 on the -10 and 53 on the -8. This is probably a bug in the -8 implementation. The -10 impementation ain't that swift either.

Proposals: (a) Pick a scheme and have everyone conform.

(b) Treat this as an error.

(c) Declare action to be undefined.

Decision: Issue is still pending.

28. Checking for erroneous number of arguments

Problem: The -8 and -11 do not give errors for commands of the form 3,5,6D or 2,5L .

Discussion: The -8 and -11 use the last values typed.

Proposals: (a) Change -8 and -11 to give meaningful error messages in the following 2 cases:
(a) more than 2 arguments are typed
(b) two arguments are typed to a command which allows 1 but not 2 arguments

(b) Declare action to be undefined.

Decision: OS/8 TECO now gives an error (?ARG) if three numeric arguments are specified with any command. Issue is still pending.

29. End-of-line character

Problem: The -10 treats any vertical paper motion character as an end-of-line terminator. The -8 and -11 treat <LF> only as end-of-line.

Discussion: -10 changed from -8 and -11 interpretation during EO level 2.

Proposals: ✓ (a) Change -8 and -11 to conform to -10.

Decision: Both OS/8 TECO and TECO-11 now treat line-feed, vertical tab, and form feed as end-of-line terminators, in accordance with the -10 implementation.

30. Echoing of Bell

Problem: On the -10 and -11, the character (^G) echoes as ^G(^G). On the -8, it merely echoes as ^G.

Proposals: ✓ (a) Change -8 to conform with -10 and -11.

Decision: OS/8 TECO now rings the bell as well as types "^G" as of v5.

FUTURE COMPATIBILITY ISSUES

Readers are urged to send in comments, criticisms, and suggestions concerning the preceding compatibility issues. In future issues of the Moby Munger, we will go into detail about other compatibility problems. In order to give readers a chance to comment on such issues in advance, we print below the titles of some future issues:

31. *Z is CTRL/S on PDP-8
32. Vertical tab command
33. Partial error reporting with ?
34. Assumed : modifier for searches in iterations
35. F modifier on PDP-8
36. Label tracing in conditionals
37. Carriage return input to CTRL/T
38. echoing of NULL
39. Matching of begin of buffer on CTRL/S
40. $$ command
41. Jumps to left out of iterations
42. == signed on PDP-1Ø
43. Memory expansion algorithm
44. Forced exact mode searches
45. CTRL/C command
46. Passing arguments through [and]
47. Case flagging on lower case terminals

Readers discovering other compatibility problems amongst DEC TECOs are strongly urged to report them so that they may be discussed in future columns.

TECO TECHNIQUES - How to Write Printable TECO Macros

Many PDP-11 operating systems have the problem that their line printer drivers (and sometimes even their terminal drivers) do not print visible symbols to represent the entire ASCII character set, or even a large percentage of the ASCII set.

In particular, when making listings of TECO programs, it is necessary to see all the control characters and ESCAPEs in the source listing. To get around this problem, TECO-11 has generalized the TECO language so that all TECO programs can now be written using only printable characters. In the table below, we show how to write printable equivalents of typical commands:

Traditional Command	Printable Equivalent
CTRL/X	↑X
Sabc$	@S/abc/
nI$	@nI//
$	↑[
ERfile$	@ER/file/
<TAB>xyz$	@↑I/xyz/
@I/$/	@27I//
Otag$	@O/tag/
CTRL/X in a string	↑X in the string (with ØED)

TECO-RELATED DECUS SUBMISSIONS

The following items of interest to TECO users are available from the DECUS program library:

Number	Description
DECUS-8-863	TECO Overlay for VC8/E. Submitted by M. Boudinot. This is an overlay to OS/8 TECO (version 3) to allow it to display a window into the text buffer on a VC8/E display. It is written in PAL8 and makes the VC8/E look like a VR12. The write-up of the algorithm is quite detailed.
DECUS-L-125	PATCH TO OS/8 TECO FOR LINC-8 DISPLAY. Submitted by Mark Lewis. This is an overlay to OS/8 TECO (version 2) to allow it to display a window into the text buffer on a LINC-8 display. It is written in PAL8.
DECUS-1Ø-264	XTEC Submitted by Jack Krupansky and Mark Crispin of Stevens Institute of Technology. It was written from scratch in MACRO-1Ø. XTEC is a TECO compiler for the DECsystem-1Ø. It is much faster than DEC's TOPS-1Ø TECO since it actually compiles command strings and macros into hard code. In general, it is upward compatible with TOPS-1Ø TECO version 23.
DECUS-1Ø-22Ø	TOPSTEACH This was written by Jacob Palme and is an automated course which introduces you to the TOPS-1Ø operating system. It is of interest to TECO fans because there are several segments which are TECO tutorials and this course makes a fine introduction to TECO for a novice. Unfortunately, it will only run on a -1Ø since it is written in the GNOSIS language (a language designed for computer-aided-instruction), however I have looked at the source and it would not seem to be too hard to write a GNOSIS compiler in TECO if anyone were interested.

TECOs AVAILABLE FROM DECUS

Operating System	DECUS Order Number	TECO version[†]
RT-11	DECUS-11-288	TECO-11 v28
RSX-11/M	DECUS-11-333	TECO-11 v28
RSX-11/D, IAS	DECUS-11-334	TECO-11 v28
DOS-11	DECUS-11-265	TECO-11 v15
RSTS/E	DECUS-RSTS-11-1Ø5	TECO-11 v24
TOPS-1Ø	DECUS-1Ø-264	XTEC %Ø(427)

[†]These submissions are updated from time to time.

BUGS IN TECO-11

Only one bug in TECO-11 has been reported since the last issue. Either TECO-11 is extremely solid, or no one is reporting the bugs. Bugs should be reported to the TECO SIG, care of DECUS.

$ bug in v32

The $ (ESCAPE) command is supposed to discard any numeric arguments. However, the command in all versions of TECO-11 up to v32, is forgetting to reset the comma flag, that indicates that a comma has been seen. Thus, a command such as H$T acts incorrectly. H$T should be exactly identical to T, however, it isn't. It does unpredictible things, like Ø,.T or something. This bug will be fixed in v34. In the meantime, avoid using two arguments to a command that doesn't expect two arguments.

Eat it, EDT® VTEDIT MACRO for TECO-11

Type any key to resume editing.

All * commands take an optional argument as: **ESC** ***expression key(s)***

All † commands operate from Dot to Mark if Mark is set

All ‡ commands are continuous when entered as **ESC** *key*. (Type any key to stop.)

BLUE Cut Text*†	RED TECO command*	GREY Paste text	↑ Up in column*
7 Open Line*	8 Page*	9 Mark/ quote*	↓ Down in column*
4 Up Line*‡	5 Delete char*	6 Delete/ restore	→ Cursor right*‡
1 Top of page*†	2 Bottom of page	3 Start of line	← Cursor left*‡
0 Down line*‡		. Search again*	ENTER Search arg*

^C [or] ^Z	Return to TECO
ESC ^Z	Exit from TECO
ESC - ^Z	Kill output, exit

[ESC] ^A	Append [No FF] text *
[ESC] ^B	Delete [next] word*
^D	Kill rest of line*
[ESC] ^E*q*	Execute [iterate] Q-reg *q**
^F	Forward over word*‡
^G*q*	Get text from Q-reg *q*
^K	Kill line*†
[ESC] ^N	Non-stop [destructive] search*
^R	Back over word*‡
^T[:]*q*	Cut text to Q-reg *q**†
^U	Kill start of line
^V	Toggle display mode
^W	Set line*/re-paint
^X[:]*q*	Save text in Q-reg *q**†
^Y	Yank next page*
^\x	Insert ASCII "*x*"
BK SP	Go to end of line*
DELETE	Delete previous character*
ESC ESC	Repeat RED-key command*
ESC F	Find front of file
ESC H	Display help frame
ESC M	Set left margin*
ESC N	Get next word*
ESC T	Select word delimiters*

DECUS

DIGITAL EQUIPMENT COMPUTER USERS SOCIETY
ONE IRON WAY, MR2-3/E55
MARLBORO, MASSACHUSETTS 01752

MOVING OR REPLACING A DELEGATE?

Please notify us immediately to guarantee continuing receipt of DECUS literature. Allow up to six weeks for change to take effect.

( ) Change of Address
( ) Delegate Replacement

DECUS Membership No.: ____________________

Name: ____________________

Company: ____________________

Address: ____________________

____

State/Country: ____________________

Zip/Postal Code: ____________________

Mail to: DECUS - ATT: Membership
One Iron Way, MR2-3
Marlboro, Massachusetts 01752 USA

Affix mailing label here. If label is not available, print old address here. Include name of installation, company, university, etc.