MASSACHUSETTS INSTITUTE OF TECHNOLOGY

PROJECT MAC

REVISED A.I. Memo. 118.
System Program.

January 1967
Revised June 1967 by
Donald E. Eastlake III

PDP-6 SOFTWARE UPDATE

Table of Contents

Conventions of this Memo 1
Organization of PDP-6 Software 2
MACDMP 3
Changes to TECO 5
Changes to MIDAS 8
Changes to DDT 10
Appendix 12

Conventions of this Memo

Most numbers written in arabic numerals are octal while all those written out in english are decimal.

Underlining a character and immediately preceding it with a vertical bar indicates the character produced by holding down the control key while striking that character except in the case of |$ which represents an ALT MODE. Characters not indicatable with the character set used in this memo or control of such a character are described between angle brackets. The string from the open to the close angle bracket should be considered as one character which may be controlled by underlining and preceding with a verticle bar.

Lower case letters in a command string usually indicate a possibly optional variable while capital letters or special characters are constant.

Note the special conventions involving ¢ in the MACDMP section.

Organization of PDP-6 Software

MACDMP is normally used to load system and user machine language programs. If when one approaches the PDP-6 it is not in MACDMP (which is usually displaying a file directory) one should first try starting at location 177400 which is MACDMP's starting address. If this fails be sure a system tape is mounted of drive number one and try reading in at location 0 (see appendix). If that loses try locations 1 and 2. If still unsuccessful try placing a paper tape of MACDMP in the paper tape reader, turning it on, and starting at location 20 (see appendix). If all else fails you can conclude that most of memory is clobbered and load a paper tape of MACDMP according to the instructions on the inside of the left door of the first bay of the PDP-6 to the left of the console.

When finally in MACDMP one may load and start various systems programs such as LISP or TECO or one may load a fairly arbitrary machine language program with or without DDT. Most systems programs provide a way to get back to MACDMP and are unlikely to die randomly. However, a machine language program being debugged should normally be loaded with DDT so that locations in the program can be examined and changed symbolicly and so that more of the machine conditions (such as AC's) may be saved on a dump by MACDMP if DDT is first started (at location 174000) and control transfered to MACDMP with the ↑U command.

In normal use a system tape is mounted on drive one with such programs as DDT, TECO, MIDAS, STINK, LISP, etc. on it. The other three drives are free for users' tapes or such tapes as the library and console scratch.

MACDMP (replacing memo MAC-M-248 or AI-82)

MACDMP is a program normally resident at the top of that portion of the PDP-6 memory being used (usually 65K) and whose main purpose is to load programs from and dump programs on MAC format (see memo MAC-M-249) DEC tapes. In general MACDMP does not dump locations containing zero or locations 0 to 37 or any part of itself. It does not load into locations 0 to 37 or itself nor does it set these locations to zero when it "clears core". All loading commands set location 40 to zero.

When quiescent it also displays the file directory of its current tape on the 340 display.

A bell will be typed out for any error MACDMP detects such as the following: checksum error on reading, file not in directory for read, not enough room on tape or in file directory for dump, tape write locked for dump, more than one unit of the selected number, disagreement on verify, no symbols in file for T|$.

In the list of commands below ¢ represents a carriage return but an |$ will also be accepted if the immediately proceeding file name is null or begins with two letters. Before its terminal character a command may be cancelled by typing a <RUB OUT>. If only one file name is given it will be assumed to be the second with @ as the first. If no file names are given then @ will be used as both the first and second file name. Aiming the 340 display's light pen at a file is equivalent to typing its file names followed by a carriage return.

n|$ Sets the current tape to tape n if n is in the interval 0 to 7. Tape zero is a pseudo-DEC tape which is both small and rapid and is in reality the bottom 16K of the 262K Fabritek memory which is normally masked by the 16K of fast memory. If n is greater than 7 the current starting address is set to n.

G|$ Transfer control to the current starting address.

fn1 fn2¢ Clear core, load program fn1 fn2, and go to its starting address.

L|$fn1 fn2¢ Clear core, load, set starting address, but stay in MACDMP.

M|$fn1 fn2¢ Load (Merge) and set starting address staying in MACDMP.

I|$fn1 fn2¢ Verify core against program fn1 fn2.

T|$fn1 fn2¢ This command assumes that DDT has been loaded and that fn1 fn2 is a file with symbols. It clears core below DDT, loads fn1 fn2, adds she symbols for fn1 fn2 to DDT's symbol table and goes to DDT.

N|$fn1 fn2¢ The same as T|$ but does not clear core below DDT.

D|$fn1 fn2¢ Dumps the contents of core on the current tape as a file called fn1 fn2 with the current starting address and write out the updated file directory of the current tape. A previous file of that name will be deleted.

K|$fn1 fn2¢ Eliminate the file fn1 fn2 from the file directory. of the current tape. Does not write out its file directory.

D|$n|$ Dump out the current file directory on tape n.

I|$n|$ Verify the current file directory against that for tape n.

Changes to TECO (supplementing memo MAC-M-250 or AI-81)

The following new commands have been introduced:

|R Turns on line printer output and when first used causes one form feed to be sent to the line printer.

|E Turns off line printer output.

|F Has the value of the setting of the data switches.

|Q Generates, on whatever output devices are currently selected, a listing of the file last ER'd or EF'd. Two passes are made over the file during the first of which a symbol table is made up with, for each symbol encountered, the page and line number the symbol is defined, declared a variable, or defined a macro on, if any, and whether the symbol is declared global. On the second pass two title pages are printed with the file names in large letters and the tape label in small letters (see the ES command below). These are followed by the file with page numbering and three columns of numbers to the left. The first of these is the line number while the second and third are the page and line number of the definition of the last symbol appearing on that line ouside of parentheses. Finally a symbol table is printed with all the encountered symbols in alphabetic order followed by the page and line number of their definition. The page and line number are normally separated by a space which is replaced by an asterisk if no reference to the symbol is noticed. If no definition is noticed for a global symbol the word "undef" will appear in place of page and line numbers. A single letter may also appear between the symbol and its defining location with the following meaning: M signifying a macro, V signifying a variable, or G signifying a global symbol.

n|Q Causes |Q to make n listings repeating only pass two if n is greater that zero. If n is less than or equal to zero then the "file name" following the |Q and terminated by an |$ is big printed and no other operations preformed by the command.

EP Takes a file name argument, ER's that file and then big prints its name.

|<up arrow> Has the value of the ASCII code for the immediately following character.

<open bracket> Push the value stored in the Q-register specified by the next character onto the Q-register pushdown list.

<close bracket> Pop the most recently stored entry on the Q-register pushdown list into the Q-register specified by the following character.

<backslash> Without an argument has the value of the number which is assumed to be immediately to the right of the pointer in the buffer and causes the pointer to be moved to the right of this number. With one argument will insert in the buffer at the pointer the decimal print of its argument. With two arguments will insert in the buffer at the pointer the decimal print of its second argument possibly padded on the left with spaces so as to insert at least as many characters as the value of its first argument.

|X, |Y Inside a macro |X and |Y have the value of the two arguments given when the macro was called. If only one argument was given, |X is zero and |Y has the value of the argument given. If no arguments were given both |X and |Y are zero.

nES Assign a three character label to tape n (optional argument, if none given the current tape is assumed). The three character name follows the ES command and is terminated by a |$. This label is inserted if the tapes file directory but the directory is not written out.

nEM Causes a listing of the file directory of tape n (optional argument, if none given the current tape is assumed) to be inserted into the buffer.

|G Takes an optional string argument which, if given, is passed on to MACDMP with a carriage return added to the end of the string. If dollar signs appear in the string they will be replaced by |$'s before being passed to MACDMP.

|<left arrow>, |<close bracket> Used for input-output via PLT to III's PDP-7. Consult a systems programmer for more information.

Special characters in search strings:

|B Matches any break character.

|N Takes the immediately following character as an argument and will match anything but that character. |N|B will match any non-break character.

|O Separates strings in a search command and causes independent searches, one for each string, to occur. The string which occurs earliest in the buffer is the one which is found. If the search is performed with the colon modifier, its value will be zero if it fails or minus n if the nth string is the one found.

|Q Quotes the next character causing any special significance it might normally have to be ignored.

|X Matches any character.

Other changes.

On input from a teletype the characters with the values 33 and 176 are converted to 175.

The Y command may now be used to read part of a "page" longer than the buffer, also nY will read at most n characters.

Numbers that are followed by a period will be considered octal.

TECO's error comment of a question mark has now been supplemented by a variety of three letter codes some of which follow:

2<1 Second argument less than first.
IQN Illegal Q register name.
NIB Not in buffer.
SFL Search failed.
UEC Unexpected end of command.
UMC Unmatched close.
WNA Wrong number of arguments.

Changes to MIDAS (supplementing memo MAC-M-279 or AI-90)

New pseudo-instructions:

ASCIZ The same as ASCII except that a word of zeros if forced after the string if the number of characters is a multiple of five. Thus the character string always end in a zero character.

.ASCII The same as ASCIZ except that if an ! is encountered in the string it will be assumed to be followed by a field whose value will be octal printed into the string at that point.

.TYPE Has a value determined by the status of the symbol that follows it according to this table:

1	pseudo-op or macro
2	defined local symbol
3	local undefined symbol
4	local defined variable
5	local undefined variable
6	global defined variable
7	global undefined variable
10	global entry (defined in program)
11	global exit (not defined in program)
17	not in symbol table

.GLOBAL Takes a list of symbols as arguments and is equivalent to placing a " on an occurance of each.

.LIBRQ Takes a list of symbols as arguments and makes each "seen" and "requested" as far as the loader is. concerned.

.GSSET Set the generated symbol counter to the value of the following expression. The next generated symbol will be of the form Gxxxxx where the number xxxxx is the value of the counter plus one.

.NSTGW Print an error if any storage words are generated.

.YSTGW Permit storage word generation.

.LNKOT Punch out loader words to unlink core in preparation for a load-time transfer to your program.

.ISTOP Stop the processing of the current IRP-class pseudo operation.

.STOP Stop the current iteration of the IRP-class pseudo operation or repeat and take the next element of the IRP list.

.LOP Loader .OP which is the same as .OP except that it occurs at load time and has no value. Instead the values left in AC and AC+1 are placed into the global symbols .LVAL1" and .LVAL2" respectively.

.TAG, .GO Used to control the flow of character handling inside macros and IRP's. .TAG is a NOP when encountered except that it causes the symbol following it to be ignored. .GO takes the symbol following and begins from the start of the macro or IRP body a search for the character string ".TAG". Upon finding a .TAG it compares the symbol following with the symbol after the .GO . If equal the macro scan pointer is left following the symbol after the .TAG . If not equal it continues to scan the rest of the current macro level for a match. If not found the macro processor is popped up one level and the scan is continued. This is continued until either a matching symbol is found or the top level is reached.

.FORMAT The manner in which MIDAS interprets fields is determined by its format table. The entries in this table consist of three twelve bit bytes per word. Each byte consists of two six bit sections. The first section controls how many bits the indicated field is to be shifted left. The second section controls the number of bits to which the field is to be masked. If the specified format allows specification of three fields, the first is specified by the right twelve bits, the second by the middle twelve bits, and the third by the left twelve bits. If the format allows specification of only two fields, the first is specified by the middle twelve bits, and the second by the left twelve bits. If the format allows specification of only one field, it is specified in the left twelve bits. Entries in the MIDAS format table may be changed by the pseudo operation .FORMAT as follows:
.FORMAT fn word
where fn is the number of the format as given in MAC-M-279 and word is the new contents of that entry.

Changes in operators:

<up arrow> With no preceding argument has the value of the ASCII code for the character following masked to its bottom six bits.

! This character is now deleted by the macro processor whenever it detects a dummy variable if it was either of the delimiting break characters.

Changes to DDT (supplementing memo DEC-6-0-UP-DDT-UM-FP-ACT00)¬

The following new commands have been introduced:

|$V Aternate uses of this command turn on and off a display feature which has been added to DDT and which displays the current location and ten locations either side of it.

|$>, |$< Takes a numeric argument and treats it as five fields, the operation code, accumulator, indirect bit, index, and address. It has the value of |@with the nonzero fields in the argument substituted.

n|X Transfers locations 40 to 177777 to the corresponding locations in the n'¬th 200000 word block of memory.

n|Y Transfers locations 40 to 177777 in the n'th 200000 word block of memory to the corresponding locations in lower memory.

n|Z Clears all of the n'th 200000 word block of memory to zero.

|$n" Sets the temporary output mode to ASCII if n is seven or SIXBIT if n is six.

|$|$n" S¬ets the permanent output mode in the same manner as |$n" sets the temporary mode.

|$", |$|$" Set the temporary and permanent output mode to ASCII or SIXBIT depending on the most recent type specified.

n|$T, n|$|$T Set the temporary and permanent output mode respectively to bytes of size n in the current radix.

|$n<close bracket> Takes a following string argument terminated by the first non-SIXBIT character and has the value of that string as SIXBIT or ASCII depending on whether n is six or seven.

|¬$n<open bracket> Takes a following string argument whose first character is used as the terminator and whose value is determined as in |¬$n<close bracket>.

|$<close bracket>, |$<open bracket> Use the most recently set type of character string value.

|$<back slash> Has the value of the number that <backslash> would print out except that indexing and indirect addressing will be used.

The following "commands" are ignored by the top level of DDT and will not terminate a search or multiple proceed:

|B Turns on line printing of DDT's output.

|E Turns off DDT's line printer output (this is the initial state).

|V Turns on teletype output from DDT (this is the initial state).

|W Turns off teletype output form DDT.

Other changes:

The " command will be affected by |$n" and |$|$n" which may set it to either ASCII or SIXBIT.

The n,,m format is now recognized by DDT on input and may also be output in symbolic mode for numbers both halves of which are small negative numbers.

Appendix

Loaders Normally Resident in the PDP-6

```
;SHADOW MODE LOADER

0/      JFCL
1/      MOVSI (JFCL)
2/      CONO 635550
3/      CONO PI,11577
4/      CONO UTC,223110
5/      CONO DC,4010
6/      CONSO DC,1000
7/      JRST 6
10/     DATAI DC,13
11/     AOJGE 13,6
12/     TRNE 13,-1
13/     .
14/     JRST 6

;RIM LOADER

20/     CONO PTR,60
21/     CONSO PTR,10
22/     JRST 21
23/     DATAI PTR,26
24/     CONSO PTR,10
25/     JRST 24
26/     .
27/     JRST 21
```