

# ML/I User's Manual — Appendix H

Implementation on the PDP-11 under DOS

R.C. Saunders  
University of Kent at Canterbury

May 1972

This implementation is based on version AID of ML/I.

Copyright © 1972 R.C. Saunders

Permission is granted to copy and/or modify this document for private use only. Machine readable versions must not be placed on public web sites or FTP sites, or otherwise made generally accessible in an electronic form. Instead, please provide a link to the original document on the official ML/I web site (<http://www.ml1.org.uk>).

## H.1 Restrictions and Additions

This implementation of ML/I contains all the features described in the *ML/I User's Manual*, 4th Edition, August 1970, plus New Features as described in the supplements to that manual (startlines, stop markers, controlled line numbers and optional warning markers).

## H.2 Operating instructions and I/O

Characters typed by the machine are shown in italics.

To run ML/I from the keyboard, the user types the following:

<i>\$RUN ML1</i>	loads ML/I and enters it
<i>ML1 V001</i>	
<i>#KB: ,KB: &lt;KB:</i>	results and messages to keyboard input from keyboard
<i>MCSKIP MT, &lt;&gt;</i>	
<i>MCDEF PIG AS DOG</i>	
<i>PIG</i>	
<i>DOG</i>	
<i>^C</i>	end of file on keyboard is . . .
<i>. EN</i>	. . .EN carriage return linefeed
 <i>AT END OF PROCESS: 3 LINES, 3 CALLS</i>	
<i>VERSION AID</i>	
<i>STOPS ARE</i>	
<i>MACROS ARE</i>	
<i>PIG</i>	
<i>WARNINGS ARE</i>	
<i>INSERTS ARE</i>	
<i>SKIPS ARE</i>	
<i>&lt;</i>	
<i>#^C</i>	ML/I asks for another command string
<i>. KI</i>	To exit, type <b>Ctrl-C KILL</b> carriage return
<i>\$RUN PIP</i>	Load and run a new program

A typical command sequence to read a set of macros, the environment, from a disk file *LOWL.MCR* and the source text from DECTape file *SOURCE.TXT* on unit 1, and to send the value text to DECTape file *VALUE.TXT* on unit 0, and messages to the keyboard would be:

```
$RUN ML1
ML1 V001
#DT: VALUE. TXT, KB: <LOWL. MCR, DT1: SOURCE. TXT
```

At the end of the process, ML/I asks for another command string. The operator may reply: **Ctrl-C, KILL**, carriage return to remove ML/I from core ready for the next program.

The usual conventions regarding command strings apply; note that in the following command:

```
#VALUE. TXT, KB: <KB: , SOURCE. TXT
```

input is taken from the keyboard, and because `SOURCE.TXT` is not preceded by a device name, further input is taken from the keyboard by default. To take input from disk, a `DF:` must be inserted before the filename, unless it is the first file specification or the previous file specification was from disk.

A maximum of four output file specifications may be requested:

1. main results stream
2. the message stream
3. the listing of the results stream
4. secondary results stream

e.g.

```
#DT:VALUE1.TXT,KB:,DT1:VALUE.LST,DF:VALUE2.TXT<SOURCE.TXT
```

If one of the file specifications is null, then that particular output will not be generated. Output may be controlled during processing by changing the values of `S20-S22` as follows:

```
S20 = 0    turns listing stream off
S20 = 1    turns listing stream on
S21 = 0    turns main results stream off
S21 = 1    turns main results stream on
S22 = 0    turns secondary results stream off
S22 = 1    turns secondary results stream on
```

The initial values of these S-variables are set according to the contents of the command string.

A maximum of three input file specifications may be requested. Input is taken from the first file in the list until the end of file (see section H.2.4) is reached. A newline character is added if the last line was not terminated by a newline, and input is then taken from the succeeding files in the list, if any.

Spaces at the end of lines are always deleted.

### H.2.1 Restricting the character positions within a line of input

The S-variables `S11` and `S12` can be set to restrict the character positions within a line of input. This is most generally used for card input to ignore the last eight columns (73 to 80), but it may be used on any input device. ML/I ignores a character unless

```
S11 <= column number <= S12
```

where *column number* means “character position within line” for files other than card decks. Initially `S11` is 1 and `S12` is 118, so that entire lines are read.

(note that if `S12` is less than `S11`, then a newline is the only character sent from that line, and ML/I will repeatedly input newline characters until the end of source text.)

## H.2.2 Input translation facility

It is possible to arrange for one character to be translated into another on input. Thus if a character is not available on a particular device, for example a tab character on cards, S16 is set to the internal code of the character to be translated, e.g. %, and S17 is set to the code for tab.

Initially S16 is set to  $-1$ , so that no translations are effected. The settings of S16 and S17 may be changed dynamically if desired.

## H.2.3 Ordering of input operations

The input routine always deletes spaces from the ends of lines and then translates using S16 and S17.

## H.2.4 End of file for input devices

End of file for various input devices is designated as follows:

- |    |                        |     |                                   |
|----|------------------------|-----|-----------------------------------|
| a. | Fast paper tape reader | PR: | physical end of tape              |
| b. | Card reader            | CR: | EOF card (see below)              |
| c. | Keyboard               | KB: | $\text{^C EN}$ carriage return    |
| d. | Slow paper tape reader | PT: | $\text{^C EN PT}$ carriage return |

An EOF card has 12-11-0 punched in column 1, but note that at least one more card must be in the hopper after this to avoid a hopper not ready (A002) message.

## H.2.5 At end of process

On return to command status, signified by #, the user has three choices:

1. to continue the process with a new command string. The current environment is still in existence with all files from the previous processes closed.
2. to begin ML/I again under a new environment by typing  $\text{^C BE}$  carriage return and the new command string. Note that any MCALTERs from the previous processes will still be active.
3. to call in another program (or to run ML/I again to remove any MCALTERs) by typing  $\text{^C KI}$  carriage return followed by RU or GE and the new program name.

## H.3 Character set

The character set is that which is acceptable to DOS, i.e. the full ASCII character set including lower case letters and control characters.

## H.4 Error messages

Error messages are output to the message stream, the second dataset in the command string. The number 2N of Chapter 6 of the *ML/I User's Manual* is 60; thus if a piece of text in an error message is greater than 60 characters in length, the first 26 and last 26 characters are printed, separated by three dashes and some spaces.

The statistics message of Chapter 6 of the *ML/I User's Manual* is given at the end of every process.

The treatment of illegal input characters is compatible with other DOS programs, e.g. PIP, EDIT. In practice, parity is not checked on paper tape, and the error character, i.e. the character which replaces illegal input characters, does not exist.

The following additional messages are output on the keyboard. In all cases the process is aborted, and an exit is made from ML/I.

### H.4.1 Parity errors

#### *Message*

```
INPUT CHARACTER PARITY!
```

#### *Description*

A character was read which had odd parity.

#### *Message*

```
( INPUT) DEVICE PARITY!  
(OUTPUT)
```

#### *Description*

A hardware error was detected on a bulk storage device.

### H.4.2 Invalid line error

#### *Message*

```
( INPUT) INVALID LINE!  
(OUTPUT)
```

#### *Description*

On input, the maximum byte count was exceeded before a line terminator was found.

On output, the last byte was not a terminator (this message implies a software error).

### H.4.3 Device full

#### *Message*

```
OUTPUT DEVICE FULL!
```

#### *Description*

An output device has no more free blocks.

## H.5 Integer calculations

The initial environment contains ten permanent variables. These are not preset to any particular values. All integers in, or derived from, macro expressions should be less than 32768 in magnitude. Overflow is not detected, except in the case of division by zero, and its effect is undefined.

## H.6 Layout keywords

The following are the layout keywords for this implementation:

SPACE	meaning a space
NL	meaning a newline
TAB	meaning a tab
SL	meaning the imaginary startline character
SPACES	meaning a sequence of one or more spaces

## H.7 S-variables

There are 24 S-variables available on the DOS PDP-11 implementation. S1 to S3 are described in the supplements to the *ML/I User's Manual*. S10 to S24 are described below:

<b>Input Controls</b>		<b>Initial value</b>
S10	not used	0
S11	starting column	1
S12	terminating column	118
S13-S15	not used	0
S16	code to be translated on input	-1
S17	code to replace S16	0
S18-S19	not used	0
<b>Output Controls</b>		<b>Initial value</b>
S20	0 do not list output	*
	1 list output	
S21	0 turn off output to results stream	*
	1 turn on output to results stream	
S22	0 turn off output to secondary results stream	*
	1 turn on output to secondary results stream	
S23-S24	not used	

\* means that these initial values reflect the command string used for the current process.

## Acknowledgement

This implementation of ML/I was supported by a Science Research Council grant to investigate the implementation of machine-independent software using the ML/I macro processor.