ML/I User's Manual — Appendix I

Implementation on the Honeywell DDP-516 under the ADMOS executive

R.D. Eager

June 1973

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

I.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 1 to 4 as described in the supplements to that manual (startlines, stop markers, controlled line numbers and optional warning markers). In addition it contains a facility for suppressing the context print-out following a message produced by a call of the operation macro `MCNOTE`.

I.2 Operating instructions and I/O

In this Section, output from the system is shown in italics.

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

```
*ADMOS
&ML1                                    Loads ML/I and enters it
*ML/I LOADED*
&AC I2 PR O2 TT O3 LP                   Assigns required I/O channels
&                                       Null command starts the process
*ML/I*
MCSKIP MT,<>
MCDEF PIG AS DOG
PIG
DOG
^C                                      Control-C terminates input

AT END OF PROCESS: 3 LINES, 3 CALLS
VERSION AID
STOPS ARE
MACROS ARE
PIG
WARNINGS ARE
INSERTS ARE
SKIPS ARE
(
&                                       Returns to command status
```

I.2.1 Availability

ML/I is available as a self-loading paper tape, also as part of the ADMOS object program library on disc and/or magnetic tape.

I.2.1.1 Hardware requirements

ML/I occupies about 5.8K words of store. The entire area between `A$TL` and `A$FC` (the free store) is used as workspace. At least 2K of workspace is normally required. Adding the

size of the operating system to this, it will be seen that ML/I will just run in 12K but will easily run in 16K.

The actual amount of workspace available (in words) is placed in `S29` at the start of a process for the user's information. However it is not subsequently updated unless ML/I is rerun, and cannot therefore be used during a run to monitor how much workspace remains.

I.2.2 Introduction to I/O

ML/I has quite comprehensive I/O facilities. These are controlled by the settings of the S-variables `S10-S30`. The meanings of the S-variables are described in the sections which follow. In particular, the input medium is controlled by `S10`.

Input/output options may be changed dynamically during a process by changing the values of S-variables. Note, however, that if there exists a construction of form:

```
NL WITHS . . .
```

or

```
NL WITH . . .
```

ML/I needs to look ahead a line in its source text. Hence in this case a switch of input device will not come into effect immediately, and it is necessary to place a blank line after the `MCSET` that effects the change of input device.

The following sections give a comprehensive description of all the I/O options and the S-variables that control them. A summary of the usage of S-variables appears in Section I.7.

I.2.2.1 Terminology

In the sections which follow "8-bit code" means the ISO 8-bit forced-bit-8 character code. "Newline" means the character (value octal 212, in fact) that ML/I uses internally to indicate the end of a line. On output newline is automatically translated back to carriage return, line feed as necessary. On input all characters have bit 8 forced in them before processing.

I.2.3 Control of input

ML/I can use all four ADMOS input channels. The value of `S10` gives the channel number to be used. `S10` is initialised to 1, so input is normally from the teletype at the start of a process.

If a change of input channel is made, the original channel is not "forgotten". Any attempt to input from this channel again will cause ML/I to carry on where it left off. When the end of an input channel is reached, ML/I checks to see if it is the *revert channel*. If it is, the process is terminated; otherwise input is switched to the revert channel.

The revert channel is given by the value of `S30`, and is initialised to 1, so that input normally reverts to the teletype after, say, reading some input from a disc file. It may, however, be changed at will.

The input terminator is the 8-bit character given by the least significant 8 bits of S13. It is initialised to decimal 131 (octal 203, Ctrl-C) but may be changed, for instance, to read a non-standard paper tape.

I.2.3.1 Listing of source text

The source text may be listed, if desired, on channel O3. This is done by setting S14 to 1. If S14 is zero, as it is initially, no source listing is produced. Lines of source listing are preceded by twenty asterisks and a line number. The last line is listed even if it consists only of a terminator, but is not included in the final count of source lines.

I.2.3.2 Input translation facility

It is possible to designate that one character be translated to another on input. This makes it possible to input a character that a device does not support, for example a backarrow from the teletype. However only one character code can be translated in this way.

If it is desired to perform a translation, S16 should be set to the 8-bit ISO code of the character to be translated, and S17 to the 8-bit ISO code of the character that is to replace it. For example, if % (internal code decimal 155) was to represent a @ (internal code decimal 192), S16 and S17 should be set in the following way:

```
MCSET S16 = 155
MCSET S17 = 192
```

Initially, S16 is set to zero, which is the code for runout which is ignored by the input routine anyway. Thus no translations are made.

I.2.3.3 Ordering of input operations

The ordering of input operations is as follows:

a. Ignoring of runout characters.
b. Forcing of bit 8 in the input character.
c. Translation using S16 and S17.
d. Checking for terminator.
e. Checking for other ignorable characters.
f. Checking for illegal characters.

I.2.3.4 Maximum length of input line

No input line should exceed 72 characters in length, including the carriage return at the end. If 71 characters are input and no carriage return is found, then a newline is artificially inserted.

I.2.4 Output

All four ADMOS output channels may be used by ML/I:

`O1`	Default message stream
`O2`	Primary output stream
`O3`	Listing stream
`O4`	Secondary output stream

I.2.4.1 The message stream

The message output channel is given by the value of `S28`. It is initialised to 1 (defaulting to the teletype) but may be changed dynamically if required.

I.2.4.2 Primary and secondary output streams

Output may be sent to either, both or neither of channels `O2` and `O4`. If `S21` is one then output is sent to channel `O2` and if `S21` is zero then output is not sent to this channel. `S22` controls channel `O4` in the same way. The maximum length of an output line on either channel is 72 characters plus carriage return, line feed. If an attempt is made to output a longer line on either channel, then a carriage return followed by a line feed is artificially inserted on that channel. Such a case is not apparent from the listing on channel `O3`, since the line overflow may only occur on one output channel.

If a channel is used for output during a process, then an end-of-file character is written to that channel at the end of the process. This may be suppressed by setting `S26` to 1 in the case of channel `O2`, or `S27` to 1 in the case of channel `O4`. If the channel is not used at all during a process (e.g. because output is suppressed) then no end-of-file is written.

The end-of-file character is determined by the values of `S23` and `S24` for channels `O2` and `O4` respectively, being the character whose 8-bit code is given by the least significant 8 bits of the appropriate S-variable. Both `S23` and `S24` are initialised to decimal 131 (octal 203, `Ctrl-C`).

Both the facilities should be used with extreme caution, especially if the output channels are assigned to disc or magnetic tape files, as a corrupt unterminated file could result. They are primarily intended for producing non-standard paper tapes for use on other machines.

The line number of the output text (even if such text is sent to null or suppressed) is maintained in `S25`, so that it is accessible if required. It may be changed at will.

It can be seen that it is possible, by dynamically switching the values of `S21` and `S22`, to generate two separate sets of output from ML/I; for example when generating assembly code, declarations could be sent to channel `O2` and code to channel `O4`. These two channels could subsequently be merged, if need be, by a second ML/I process, e.g.:

```
&AC O2 DC T 1                    Assign declarations channel
&AC O4 DC T 2                    Assign code channel
&                                Start process
... process to generate output ...
```

`Ctrl-C`	Input terminator
`&AC I2 DC T 1`	Assign declarations to input channel
`&AC I3 DC T 2`	Assign code to input channel
`&`	Start process
`MCSET S10 = 2`	Input from channel `I2`
`MCSET S10 = 4`	Input from channel `I4`
`Ctrl-C`	Input terminator

Messages are omitted from the above for clarity.

I.2.5 Use of output channel 3

Output channel 3 may be used for source and output listings, also for error messages if `S28` is set to 3 during processing. These are all interspersed with each other. If both source and output are listed simultaneously an extra carriage return, line feed is sent to channel `O3` before each line of the source listing in order to separate source from output.

`S20` controls whether the output text is to be listed on channel `O3`: if `S20` is zero, no listing is produced; if `S20` is one, a listing without line numbers is produced; if `S20` is two, a listing with line numbers is produced. These line numbers are a count of output lines and will not, in general, correspond to the count of source lines. Initially `S20` has the value two.

Lines of listing are 91 characters long, plus carriage return, line feed. If a longer line needs to be listed, an artificial carriage return, line feed is inserted after 91 characters have been output on a given line. Note that this insertion will not in general correspond with any artificial insertion on an output channel, since the line length is 72 characters on such a channel. However, the artificial newline insertion on input is listed.

I.2.6 Initialisation

ML/I is serially reusable. This means that after it has been used for one process it can be used for another without being reloaded. All S-variables, etc., are re-initialised at the start of each process.

However, there is one proviso to this. If one process executes a `MCALTER` it will remain in effect for subsequent processes until ML/I is reloaded. This allows, for instance, a string of paper tape processes to be run under some delimiter settings that have been `MCALTER`ed by typing at the teletype in a previous process.

I.2.7 Recovery from error conditions

Usually an error (e.g. a break, whether from the keyboard or forced by the system) causes a return to command status within ML/I. However should control be lost from ML/I it may be restarted from octal 1000, after assigning I/O channels.

I.3 Character set

ML/I uses forced-bit-8 characters throughout. Bit 8 is forced in all characters input, and all characters output have bit 8 set in them. It accepts the full set of ISO code characters with

codes between octal 240 and 337 inclusive. In addition it accepts carriage return (which is taken as "newline"), also linefeed and delete characters which are both, however, ignored.

It will also accept lower case alphabetic characters, which are converted by the input routine into their upper case equivalents. Note that this is in contradiction to the *ML/I User's Manual*, as they are not thereafter treated as separate characters, and if output, will be output as upper case. The reason for this is to allow input to be taken from the VISTA terminal without having to continually shift out when a letter is required.

I.4 Error messages

Error messages are output on the channel given by the value of `S28`, as they occur. With reference to Chapter 6 (of the *ML/I User's Manual*), the number 2N (the maximum number of characters in a piece of text inserted into an error message without being truncated) is 60.

Illegal input characters give rise to the appropriate message given in Section 6 of the *ML/I User's Manual*. The input routine then replaces them by the error character, which is an exclamation mark (`!`).

One addition not mentioned in the *ML/I User's Manual* is the facility for suppressing the context print-out on the message stream after a message produced by a call of the operation macro `MCNOTE`. To suppress the context print-out (but not the message itself), `S4` should be set to 1. To allow the print-out, `S4` should be set to 0. Initially `S4` is zero.

Extra diagnostic information, including a list of the names of all active constructions, is automatically printed at the end of each process. It may be suppressed by setting `S18` to zero. An end-of-file (decimal 131, octal 203) character is written to the message stream at the end of each process. Note that this only applies to the channel in use at the end of the process, and not any intermediate ones. The user should thus be wary of using disc or magnetic tape for error messages unless they are used at the end of the process. However if the same channel is being used for output text the terminator for the output stream will suffice to close the channel, and this does not matter.

There are a number of extra error messages peculiar to this inplementation and these are described in the sections which follow.

I.4.1 Illegal input stream

Message

`ILLEGAL INPUT STREAM`

Description

`S10` has been set to a value not in the range 1 to 4.

System Action

The current process is aborted.

I.4.2 Illegal debugging channel

Message

`ILLEGAL DEBUGGING CHANNEL`

Description

`S28` has been set to a value not in the range 1 to 4.

System Action

`S28` is set to 1, the above message is output using this value, and processing continues.

I.4.3 Process aborted

Message

`PROCESS ABORTED`

Description

Occurs after a non-continuable error such as a break or an attempt to use a non-existent input channel.

System Action

The process is aborted, `A$EP` is set to point to the command input routine within ML/I, and command status within ADMOS is entered without printing the diagnostic information referred to above. No recovery is possible as the run-time stack within ML/I may be corrupt.

I.5 Integer calculations

The initial environment contains ten permanent variables, all set to the value zero. All integers in, or derived from, macro expressions should be less than 32768 in magnitude. Overflow is not always detected, except in the case of division by zero, and its effect is undefined.

I.6 Layout keywords

The following are the layout keywords for this implementation:

`SPACE`	meaning a space.
`NL`	meaning a newline.
`TAB`	meaning the character backslash (octal 334).
`SL`	meaning the imaginary startline character.
`SPACES`	meaning a sequence of one or more spaces.

One implication of the treatment of the `TAB` keyword is that the backslash character is ignored within structure representations, and must be specified by means of `TAB`. In the same way, any occurrence of a backslash within an error message will be replaced by the message `(TAB)`.

The reason for this is that the backslash is often used as a field delimiter in the same way as tab (e.g. by the DAP-16 assembler) and a tab is peripheral-dependent.

I.7 S-variables

The following sections summarise the current uses of S-variables, and their initial values.

If `S10` or `S28` are set to illegal values the appropriate error message of Section I.4 is produced, but if any of the other S-variables have illegal values the effect is undefined. It is dangerous to change the values of S-variables not mentioned below.

I.7.1 Use of S1–S9

`S1` If `S1` is one, the imaginary startline character is inserted on input. If `S1` is zero, no startlines are inserted; this is the initial setting.

`S2` The current source text line number is held in `S2`; it may be changed at any time.

`S3` If `S3` is one, the error message normally generated if a warning marker is not followed by a macro name is suppressed. If `S3` is zero (the initial value), the message is produced.

`S4` If `S4` is one, the context print-out normally given after a call of `MCNOTE` is suppressed. If `S4` is zero, the context print-out is given; this is the initial setting.

`S5` Not currently used.

`S6` Not currently used.

`S7` Not currently used.

`S8` Not currently used.

`S9` Not currently used.

I.7.2 Use of S10–S30

`S10` Controls input selection. Values between 1 and 4 select the appropriate input channel. The initial value is 1.

`S11` Not currently used.

`S12` Not currently used.

`S13` Input terminator character (ISO 8-bit code). The initial value is decimal 131, octal 203.

`S14` If `S14` is zero, the source text is not listed. If `S14` is one, the source text is listed. The initial value is 0.

`S15` Not currently used.

`S16` Used to control character code translation. Characters with the ISO 8-bit code given by `S16` are translated to characters with the ISO 8-bit code given by `S17`, on input. Initially `S16` is -1, so no translations are performed.

S17 See S16 above.

S18 If S18 is one at the end of a process, a list is given of all currently defined constructions. This is output to the message stream. If S18 is zero, the list is not produced. The initial value of S18 is 1.

S19 Not currently used.

S20 The value of S20 controls output to the listing file. See Section I.2.4.2 for details.

S21 The value of S21 controls output to channel O2; see Section I.2.4.2 for details.

S22 The value of S22 controls output to channel O4; see Section I.2.4.2 for details.

S23 S23 contains the ISO 8-bit code for the terminator character to be written to channel O2. It is initially set to decimal 131 (octal 203).

S24 S24 contains the ISO 8-bit code for the terminator character to be written to channel O4. It is initially set to decimal 131 (octal 203).

S25 The current line number of the output text is held in S25. It may be changed if desired.

S26 If S26 is zero, write the terminating character to channel O2 at the end of the process. If S26 is one, do not write the terminating character to the channel. The initial value is 0.

S27 If S27 is zero, write the terminating character to channel O4 at the end of the process. If S27 is one, do not write the terminating character to the channel. The initial value is 0.

S28 Error message channel number, in the range 1–4. The initial value is 1.

S29 Store available at start of process (in words).

S30 S30 contains the current revert channel. See Section I.2.3 for details.