ML/I User's Manual — Appendix S

Implementation on VAX/VMS

R.D. Eager

October 1986

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

S.1 Restrictions and Additions

This implementation of ML/I contains all the features described in the *ML/I User's Manual*, 5th Edition, 1986, including New Features 1 to 6 as described in that manual.

S.2 Operating instructions and I/O

The call of ML/I takes the form:

```
ML1/qualifier/qualifier... input1,input2,input3
```

The following qualifiers are recognised:

`/PRIMARY_OUTPUT:`*`file`*
: This specifies the name of the file to receive the primary output from the ML/I process. If this qualifier is omitted, `SYS$OUTPUT` is used for the primary output.

`/SECONDARY_OUTPUT:`*`file`*
: This specifies the name of the file to receive the secondary output from the ML/I process. If this qualifier is omitted, all output to the secondary output stream is discarded.

`/LISTING_OUTPUT:`*`file`*
: This specifies the name of the file to receive the listing output from the ML/I process. If this qualifier is omitted, all listing output is discarded.

`/DEBUGGING_FILE:`*`file`*
: This specifies the name of the ML/I debugging file. If this qualifier is omitted, `SYS$ERROR` is used.

`/WORKSPACE_SIZE:`*`n`*
: This specifies that ML/I is to be given *n* bytes of workspace; the default value is 5000. The minimum and maximum permissible values for *n* are 1000 and 100000 respectively.

`/VERBOSE` If this qualifier is present, ML/I will output its version number at the start of the process, and the number of processing errors (if any) at the end of the process.

All qualifiers may be abbreviated as long as they remain unique with respect to other qualifiers. Qualifiers may be separated from their values by either a colon (`:`) or an equals sign (`=`).

All output files must be different, and they must also be different from any of the input files. Any legal VAX/VMS filename may be specified for an output file.

Up to three input files (designated as *input1*, *input2* and *input3* above) may be specified. If the first input file is not specified, `SYS$INPUT` is used. An attempt to use an input file (other than *input1*) not specified in the call of ML/I will cause the process to be aborted, after the output of a suitable message.

Any legal VAX/VMS character file may be specified as an input file, as long as the record size does not exceed 240 bytes and it is a CR format file.

S.2.1 Example calls

a. To run ML/I with just input from and output to the terminal, the only command needed is:

```
$ ML1
```

This is sufficient for working through the *Simple Introductory Guide*.

b. To run with two input files called `PIG.TXT` and `DOG.MAR`, and output to `FOO.DAT`, with listing to `BAZ.LIS`:

```
$ ML1/PRI:FOO.DAT/LIS:BAZ.LIS PIG.TXT,DOG.MAR
```

S.2.2 Control of input

Input may be taken from any one of the input files (or *streams*, as they will be referred to from now on). The value of `S10` controls the selection. The possible values are:

`S10 = 1` Input is taken from *input1*.

`S10 = 2` Input is taken from *input2*.

`S10 = 3` Input is taken from *input3*.

If S10 is set to zero, ML/I treats this as "end of file" and ceases processing. If `S10` is set to any illegal value (negative, greater than three, or a value between one and three associated with an input stream which has not been specified) then the process is aborted.

If a change of input stream is made, the original stream is not "forgotten". Any attempt to read from this stream again will cause ML/I to carry on where it left off. When the end of an input stream is reached, ML/I checks to see if it is the *revert stream*. If it is, the process is terminated; otherwise input is switched to the revert stream, and processing continues. The revert stream is initially one; its value is held in `S23` and may be altered by the user if required.

S.2.2.1 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 (e.g. a vertical bar from certain DEC terminals). 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 ASCII code of the character to be translated, and `S17` to the ASCII code of the character that is to replace it. For example, if `%` (ASCII 37) was to represent a tab (ASCII 9), `S16` and `S17` should be set in the following way:

```
MCSET S16 = 37
MCSET S17 = 9
```

Initially, `S16` has the value −1, which since it does not correspond to a valid internal code, will not cause any translations to be made.

S.2.2.2 Ordering of input operations

The ordering of input operations is as follows:

a. Checking for `S10` equal to zero.
b. Checking for invalid values of `S10`.
c. Check for end of file (if the revert stream is selected as a consequence of this check, return to b)).
d. Translation using `S16` and `S17`.
e. Checking for illegal characters.

S.2.3 Control of output

Output may be directed to either, both or neither of the primary output and the secondary output. The values of `S21` and `S22` control the selection; `S21` controls the primary output, and `S22` controls the secondary output. In both cases, a value of zero means that no output is to take place, and a value of one means that output is to take place.

S.2.4 Control of listing

A listing of the output from ML/I may be directed to the listing file specified in the call of ML/I. Listing is controlled by the value of `S20`. If `S20` is zero, no listing is produced at all. If `S20` is one, a listing without line numbers is generated; if `S20` is two, line numbers are included in the listing. `S20` has an initial value of zero.

S.3 Character set

The character set used by ML/I is 7-bit ASCII (codes from 0 to 127 decimal). However, character code 26 decimal (1A hexadecimal) is used internally to indicate end of file, and causes unpredictable effects if it appears in the input to ML/I.

S.4 Error messages

Error messages are output to the debugging file specified in the call of ML/I; this defaults to `SYS$ERROR`. With reference to Chapter 6 of the *ML/I User's Manual*, the number 2N (the maximum number of characters inserted into an error message without truncation) is 64. The error character is question mark (?).

A count of processing errors (i.e. occurrences of the word `Error(s)` on the debugging file) is maintained in `S5`. At the end of a process, ML/I checks this value; if it is nonzero, ML/I generates a standard VMS-style warning message if the `/VERBOSE` qualifier was specified on the command line.

An output lines limit is imposed on the debugging file, to curb excessive output from a process that has gone badly wrong. The limit is implemented by holding a quota of “lines left” in `S12`; if `S12` ever goes negative, the process is aborted. `S12` is initially 500, but this value may be changed by the user at any time.

At the end of a process, a message of the form

```
At end of process: N lines, M calls
```

is output to the debugging file, followed (if `S18` equals one) by a list of the currently defined constructions.

S.4.1 I/O errors

All files are opened as soon as ML/I is entered. Failure to open any file causes an appropriate message to be output, and ML/I immediately exits.

The following run-time messages are peculiar to this implementation. They may be followed by other, advisory, messages which are self-explanatory.

S.4.1.1 Too many lines to the debugging file

Message

```
Debugging file lines quota exhausted
```

Description

The value of `S12` (the quota of remaining lines allowed to the debugging file) has become negative.

System Action

The current process is aborted.

S.4.1.2 Illegal input stream

Message

```
S10 has illegal value, viz n
```

Description

`S10` has been set to the value *n*, which is either outside the range 0–3, or is associated with an input stream that was not specified in the call of ML/I. Note that this error may be caused by `S23` (the revert stream) being set to an illegal value, and end of file then being reached on another input stream.

System Action

The current process is aborted.

S.4.1.3 Primary output failure

Message

```
Primary output failure
```

Description

An error has occurred while writing to the primary output file.

System Action

The current process is aborted.

S.4.1.4 Secondary output failure

Message

```
Secondary output failure
```

Description

An error has occurred while writing to the secondary output file.

System Action

The current process is aborted.

S.4.1.5 Listing output failure

Message

```
Listing output failure
```

Description

An error has occurred while writing to the listing file.

System Action

The current process is aborted.

S.4.1.6 Debugging file output failure

Message

```
Debugging file output failure
```

Description

An error has occurred while writing to the debugging file. This message is output as a standard VMS-style message.

System Action

The current process is aborted.

S.5 Integer calculations

The initial environment contains ten permanent variables, all set to zero. All integers in, or derived from, macro expressions should be less than 2147483647 in magnitude. Overflow is not detected, except in the case of division by zero, and its effect is undefined.

S.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.

S.7 S-variables

There are 24 system variables. S1 to S9 are independent of the implementation, and are used to control and monitor ML/I itself. S10 to S23 are implementation dependent, and are used to control input/output, etc. If an S-variable is set to any value other than those given below, the effect is undefined (except for invalid values of S10, which always cause the process to be terminated).

S.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 Count of processing errors.

S6 Not currently used.

S7 Not currently used.

S8 Not currently used.

S9 Not currently used.

S.7.2 Use of S10–S26

S10 Controls input selection; a value of zero forces end of all input. Values between 1 and 3 select the appropriate input stream; other values cause an error. The initial value of S10 is one.

S11 Contains the exit code from the run of ML/I. It is initially set to the standard VMS value SS$_NORMAL, but may be changed by the user in order to pass back information to DCL.

S12 S12 contains the quota of lines on the debugging file. It is initially 500, and every time ML/I outputs a line to the debugging file (whether via an error message or a MCNOTE) it decreases S12 by one. If S12 ever becomes negative, the process is aborted. The user is at liberty to adjust the value of S12 at any time.

S13 Not currently used.

S14 Not currently used.

`S15` Not currently used.

`S16` Used to control character code translation. Characters with the code given by `S16` are translated to characters with the 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 debugging file (and is not subject to the quota of lines imposed by `S12`). If `S18` is zero, the list is not produced. The initial value of `S18` is one.

`S19` The current line number of the output text is held in `S19`, and is updated when the first character is output to each new line. It may be changed if desired.

`S20` The value of `S20` controls output to the listing file. See Section S.2.4 for details.

`S21` The value of `S21` controls output to the primary output stream, and is initially set to one. See Section S.2.3 for details. Its initial value is 1.

`S22` The value of `S22` controls output to the secondary output stream, and is initially set to zero. See Section S.2.3 for details.

`S23` `S23` contains the current revert stream, and is initially set to one. See Section S.2.2 for details.

`S24` `S24` is initialised to the size of the workspace available to ML/I (in bytes). It thus defaults to 5000 in the absence of the `/WORKSPACE_SIZE` qualifier, and takes the specified value if that qualifier is used. The contents of `S24` are never subsequently used or changed by ML/I; its sole purpose is to provide a means whereby standard macro packages can check that sufficient space is available before starting a potentially lengthy run that may fail if not enough space is available. Of course, the amount of space needed still needs to be determined in a previous run.