

# ML/I User's Manual — Appendix Q

Implementation on ICL 1900 under GEORGE 3/4

R.D. Eager

June 1985

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

Copyright © 1985 R.D. Eager

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>).

## Q.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 supplements to that Manual.

The program has been implemented on GEORGE 3, but in such a way that it will also run under GEORGE 4 without change.

## Q.2 Operating instructions and I/O

ML/I is usually run by calling the ML1 job control macro. This macro takes several different types of parameter; each type is identified by a different keyword. The possible keywords are:

- |       |   |
|-------|---|
| IN    | The value of the parameter specifies an input file for ML/I. Up to three such files may be specified (each being introduced by IN). The first such file is designated the <i>primary input stream</i> ; the second is designated the <i>secondary input stream</i> , and the third is designated the <i>tertiary input stream</i> . If the IN keyword is given without a value, or omitted, the appropriate ML/I file is ONLINED. |
| OUT   | The value of the parameter specifies an output file for ML/I. Up to two such files may be specified (each being introduced by OUT). The first such file is designated the <i>primary output stream</i> , and the second is designated the <i>secondary output stream</i> . If the OUT keyword is given without a value, or omitted, then the appropriate ML/I file is ONLINED.  |
| DEBUG | The value of the parameter specifies the name of the <i>debugging file</i> for ML/I (the destination of all diagnostic output). If the DEBUG keyword is given without a value, or omitted, then the debugging file is ONLINED. See Chapter 6 of the <i>ML/I User's Manual</i> for a description of the uses of this file.   |
| CORE  | The value of the parameter specifies the core allocation for ML/I, and is used to increase the amount of available workspace. It is optional; the default setting is 8000.  |
| ENV   | If this parameter is present, the system variable S18 is set nonzero at the start of the run; this enables the context print-out at the end of processing. See Section Q.7.2 for further details.   |

From the above, it can be seen that ML/I has three possible *input streams* and two possible *output streams*. There is also a stream for debugging messages.

### Q.2.1 Example calls

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

```
ML1
```

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

2. To run with two input files called PIG and DOG, and output to FOO:

```
ML1 INPIG,INDOG,OUTFOO
```

## Q.2.2 Control of input

Input may be taken from any one of the input streams. The value of S10 controls the selection. The possible values are

S10 = 1     Input is taken from the primary input stream.

S10 = 2     Input is taken from the secondary input stream.

S10 = 3     Input is taken from the tertiary input stream.

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, or greater than three) 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 1; its value is held in S23 and may be altered by the user if required.

No line of input may exceed 160 characters.

### Q.2.1.1 Handling of end of file

Normally, no special end of file marker is necessary. However, it may sometimes be useful to be able to force end of file from (for instance) a MOP terminal. A record consisting of four asterisks will have the same effect as setting S10 to zero, but will be recognised even when ML/I is searching for a secondary delimiter (in this situation, a MCSET to change the value of S10 would not be actioned until the construction was completed).

Sometimes, this facility may be a nuisance. To stop any terminator from being recognised, S25 should be set to zero (its initial value is one). To select an alternative terminator, S26 should be set to the decimal value of a word containing the four characters required as a terminator.

### Q.2.2.2 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 (e.g. an underline or backarrow from a MOP terminal). 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 @ (code 32) was to represent underline or backarrow (code 63), S16 and S17 should be set in the following way:

```
MCSET S16 = 32
```

```
MCSET S17 = 63
```

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.

### Q.2.2.3 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.

### Q.2.3 Control of output

Output may be directed to either, both or neither of the primary and secondary output streams. The values of S21 and S22 control the selection; S21 controls output to the *primary* output stream, and S22 controls output to the *secondary* output stream. In both cases, a value of 0 means that no output is to take place, and a value of 1 means that output is to take place.

No output line may exceed 160 characters in length.

### Q.2.4 Workspace

ML/I uses all additional core allocated to it for workspace. The CORE parameter to the ML1 macro is usually used to set this; there is no practical limit to the amount of core that ML/I can use.

## Q.3 Character set

ML/I uses the standard 1900 GRAPHIC character set. Since all 64 character codes represent valid characters, the error character is not used.

## Q.4 Error handling

Error messages are output to the debugging file specified in the call of ML/I; this defaults to the MOP terminal or job journal. 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.

A count of processing errors (i.e. occurrences of the word **ERROR(S)** on the debugging file) is maintained in S11. At the end of a process, ML/I checks this value; if it is zero, ML/I terminates with the message:

DELETED: OK

If S11 is nonzero, the message given on deletion is:

DELETED: ER

In both cases, a suitable message is output by the ML1 macro. The user is free to adjust the value of S11 whilst ML/I is processing, in order to obtain special effects (e.g. forcing the ER deletion).

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 may be changed by the user.

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 1) by a list of the currently defined constructions.

### Q.4.1 Error messages

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

#### Q.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.

#### Q.4.1.2 Illegal input stream

*Message*

S10 HAS ILLEGAL VALUE, VIZ *n*

*Description*

S10 has been set to the value *n*, which is outside the range 1–3. 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.

#### Q.4.1.3 Error on writing to primary output file

*Message*

PRIMARY OUTPUT FAILURE

*Description*

An error has occurred while writing to the primary output file (\*FH1).

*System Action*

The current process is aborted.

### Q.4.1.4 Error on writing to secondary output file

*Message*

SECONDARY OUTPUT FAILURE

*Description*

An error has occurred while writing to the secondary output file (\*FH2).

*System Action*

The current process is aborted.

## Q.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 8388607 in magnitude. Overflow is not detected, except in the case of division by zero, and its effect is undefined.

## Q.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.

## Q.7 S-variables

There are 26 S-variables. S1 to S9 are independent of the implementation, and are used to control and monitor ML/I itself. S10 to S26 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).

### Q.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.

### Q.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 1.
S11	Count of processing errors.
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 determined by the setting of bit 18 in the switch word, but is normally zero.
S19	The current line number of the output text is held in S19. It may be changed if desired.
S20	Not currently used.
S21	The value of S21 controls output to the primary output stream; see Section Q.2.4 for details. Its initial value is 1.
S22	The value of S22 controls output to the secondary output stream; see Section Q.2.4 for details. Its initial value is 0.

- S23        S23 contains the current revert stream. See Section Q.2.3 for details.
- S24        S24 contains the number of words of workspace available to ML/I at the start of processing. It is *not* subsequently updated, and is provided only so that macro packages may make decisions based on the availability of workspace.
- S25        If S25 has the value one, then the four characters represented by the contents of S26 are treated as an input terminator. If S25 is zero, this facility is disabled. The initial value of S25 is 1.
- S26        S26 contains a number which represents the character codes of exactly four characters which are to be treated as an input terminator if they appear as the contents of an input record. Their effect is controlled by the value of S25. The initial value of S26 represents four asterisks (\*\*\*\*).

## Q.8 Program specification

Although ML/I is normally run by using the ML1 macro, users may wish to write their own job control macros. This Section gives a full specification of the interface to the ML/I program itself.

### Q.8.1 Program name

The program is conventionally called KML1.

### Q.8.2 Entry points

There is only one valid entry point, entry point 0 (location 20).

### Q.8.3 Input and output

#### Q.8.3.1 File usage

All input/output is performed using GEORGE “file handlers”. The allocation of units is as follows:

- \*FH0        The debugging file.
- \*FH1        The primary output stream.
- \*FH2        The secondary output stream.
- \*FH3        The primary input stream.
- \*FH4        The secondary input stream.
- \*FH5        The tertiary input stream.

### Q.8.3.2 Buffer sizes

All input/output record buffers are 160 characters in length, except for the debugging file (the limit here being 72 characters).

### Q.8.4 Switches

The following program switches control initial settings of certain S-variables:

**Switch 18** If this switch is ON, then S18 is initialised to 1; the default setting is that the switch is OFF, and in this case S18 is initialised to zero.

### Q.8.5 Program displays

The following display is output at the start of each run of ML/I. It gives the version number of the 1900 implementation of the program; the version number of the main ML/I logic is given as part of the environment printout controlled by S18 (see Section Q.7.2).

```
DISPLAY: ML/I (VERSION x.x) PROCESSING
```

### Q.8.6 Program halts

ML/I may halt for several reasons. The meaning of the various halts (usually trapped and interpreted by job control macros) is as follows:

**HALTED: ST**

There is insufficient core for ML/I to do any useful work (less than 512 words). This should never occur unless the core allocation of ML/I is actually *reduced* after loading.

**HALTED: ER**

There has been a fatal error while writing to the debugging file. Other input/output errors are reported to the debugging file; clearly this is not possible with the debugging file itself. In practice, this error should not occur.

**HALTED: IE**

An attempt has been made to read a record which is longer than 160 characters.

**HALTED: Fn**

ML/I has attempted to use an unassigned file handler; *n* gives the unit number (see above). After the unit has been assigned, ML/I may safely be **RESUMED**.

**HALTED: Tn**

ML/I has attempted to use a file handler which is assigned to a file of an unacceptable type; *n* gives the unit number (see above). The offending unit is **RELEASED**; the user may assign another file and **RESUME** the run.

**HALTED: En**

A fatal error has occurred while trying to read the attributes of the file assigned to unit *n*. It is not possible to continue the run.

### Q.8.7 Program deletions

ML/I deletes itself at the end of a run. One of two messages may be given:

DELETED: OK

The run was completed, and the value of S11 was zero at the end of the run.

DELETED: ER

The run was completed, and the value of S11 was nonzero at the end of the run.

See Section Q.4 for details of the use of S11 in detecting errors.