ML/I User's Manual — Appendix E

Third Edition

Implementation on the ICL 4130 under the NICE executive
(for KOS version see separate document)

P.J. Brown

July 1974

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

E.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 5 as described in supplements to that manual.

E.2 Operating instructions and I/O

E.2.1 Streams

The initial input stream is set by the call of ML/I, but it can thereafter be changed by re-setting the variable `S10`.

There are up to three output streams, which are controlled by `S20`, `S21` and `S22`.

`S20` controls whether the output is to be listed on the lineprinter: 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 with the count of source lines. Initially `S20` has the value two.

`S21` and `S22` control the two main output streams. In each case the value zero inhibits output to the stream and the value one allows it. Thus if `S21` and `S22` are both one, two copies of the output are made. Alternatively `S21` and `S22` can be alternately switched on and off thus sending some output to one stream and some to the other. Initially `S21` is one and `S22` is zero.

The following is an example of the use of output streams:

```
TEXT1
MCSET S22=1
MCSET S21=0
TEXT2
MCSET S22=0
MCSET S21=1
MCSET S20=0
TEXT3
```

In this case `TEXT1` and `TEXT3` go to the stream controlled by `S21`, `TEXT2` goes to the stream controlled by `S22`, and `TEXT1` and `TEXT2` are listed on the line-printer.

The way in which streams are tied to physical devices and files is described in later Sections.

E.2.2 Interface with operating system

This implementation of ML/I runs under the NICE executive under any of the operating systems found on the ICL 4120/4130. It is entered by the command `&ML1` followed by a parameter list. The majority of users will be interested in the special features for the disc-based DES operating systems, so these will be described first.

E.2.3 Interface with DES

If the first parameter of ML/I is `DES`, all I/O is performed through `ASSIGN`ed channels (using the ACIO routine). Input is from channel 36 and the streams corresponding to `S21` and `S22` go to channels 31 and 33, respectively. The ACIO routine is such that by default channel 36 is the card reader, channel 31 is paper tape punch 1 and channel 33 is the control teleprinter. Only the first of these defaults is likely to be used in practice, the other two channels normally being explicitly assigned.

Although input initially comes from channel 36 it can subsequently be switched to any other assigned channel. If `S10` is negative, this means use an assigned channel, the value giving the channel number (e.g. setting `S10` to -97 would cause subsequent input to be taken from channel 97).

The following example shows the use of assigned channels:

```
&ASSIGN;31;DV;0;WORKFILE 5;
&ASSIGN;50;DC;2;MY,FILE;
&ML1;DES;
MCDEF PIG AS COW
MCSET S20=0
MCSET S10=-50
^^^
&NEAT;
&ASSIGN;1;DV;0;WORKFILE 5;
```

This would cause every occurrence of `PIG` in `MY FILE` to be replaced by `COW`. The file would then be assembled by NEAT. The ML/I output listing is inhibited (by `MCSET S20=0`) and channel 33 is not used and therefore not assigned.

If the input channel is assigned to cards spaces are normally deleted from the end of each card before processing. A line containing three uparrows acts as a terminator (there are some special options for "unusual" card forms, see Section E.2.4.4).

When the end of an input channel is reached, if the channel is 36 (i.e. the initial input channel) ML/I quits; otherwise the channel is closed, a newline fed to ML/I and input is resumed from where it left off in channel 36 (the exact mechanism uses the "revert device" as described in Section E.2.4.7.2). Thus if channel 36 is a card stream it should always have a terminator (3 uparrows) at the end, even if input is switched to another channel.

The ACIO routine requires a buffer area, the size of which depends on the number and type of channels that are open at any one time. By default ML/I allocates 1068 words for this, which is ample for normal work. However in exceptional circumstances the message:

```
ACIO OVERFLOW
```

will be given and the buffer area should then be increased by specifying the size of buffer required after the `DES` parameter to `ML1`, e.g.:

```
&ML1;DES;2000;
```

E.2.3.1 Listing of source text

The source text may, if desired, be listed on the lineprinter. This is done by setting `S14` to one. If `S14` is zero, as it is initially, no source listing is produced. Lines of source listing are

preceded by twenty-two asterisks and a line number. For card input the terminating card is included in the listing and for paper tape or teleprinter input the last line is listed even if it consists only of a halt code (which will appear as blank). Thus the number of source lines actually processed is normally one fewer than the count of source lines.

If S14 has the value two, lines are only listed from the original input channel (specifically when S30 equals S10 — see Section E.2.4.7.2).

E.2.3.2 Input translation facility

It is possible to designate that one character be translated into another on input. This makes it possible to input a character that a device does not support, for example a tab from cards. 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 7-bit internal code of the character to be translated and S17 to the 7-bit internal code that is to replace it. For example if % (internal code 5) was to represent a tab (internal code 1), S16 and S17 should be set in the following way:

```
MCSET S16=5
MCSET S17=1
```

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.

Note that the settings of S16 and S17 apply to all input devices and the settings may be changed dynamically if required. The translation facility is applied before the source listing is produced and is applied even to the newlines that are, in some cases, artificially introduced by ML/I at the end of a piece of input (e.g. at the end of a channel).

E.2.3.3 Ordering of input operations

The order in which the input routines perform the following operations:

a. translate using S16 and S17;

b. delete spaces (when card input);

c. check for terminator;

is not generally defined as it depends on the device used. It is, of course, only in very unusual cases that the ordering of these operations makes any difference.

E.2.3.4 Use of lineprinter

The lineprinter is used for error messages and may also be used for source and output listings. These are all interspersed with one another. If both source and output are listed simultaneously an extra newline is sent to the printer before each line of the source listing in order to separate source from output.

E.2.3.5 The DESS and DESC parameters

Two minor extra facilities are available to save a little writing.

a. The command form

```
&ML1;DESS;
```

is equivalent to

```
&ML1;DES;
```

except that S14 is initialised to one and S20 to zero (i.e. source is listed rather than output).

b. The command form

```
&ML1;DESC;
```

is similar to the above except that S14 is initialised to two, not one. This is often useful for listing only that source text that comes from cards.

E.2.4 Further facilities

(The reader may skip to Section 3 if he a) wishes to use ML/I only under DES and b) is happy to confine himself to the facilities described above).

This Section:

a. describes extra I/O facilities, most (but not all) of which are useful only when the ASSIGNed channel facilities are not available.

b. goes into more detail about storage requirements, unusual I/O situations, etc.

All the facilities described here are available under the DES operating system, and similarly all the facilities of Section E.2.3 are available under other operating systems except those facilities concerned with assigned channels (which require ACIO, CFILE and DFILE).

E.2.4.1 Hardware and software needs

ML/I needs a NICE-based operating system including LP and the device routines and supporting software for any other types of I/O that are used (e.g. DRIP, MT, LIBRARY). ML/I occupies about 5K words and requires at least 2K of workspace. It uses all the area between LOWADD and TOPADD for workspace.

E.2.4.2 Direct specification of devices

When channels cannot be ASSIGNed it is necessary to have alternative means of specifying I/O devices. ML/I does this by means of S-variables, parameters and fixed assignments. In addition to the use of physical devices, it allows, on input, the use of items from a standard 4130 library file. The library is accessed, exactly as in ALGOL, using the LIBRARY module. The library can be on disc or magnetic tape, depending on which version of LIBRARY is used.

E.2.4.3 Specification of input device

The initial input device is specified by the parameter to ML/I as follows:

`C`	means cards
`P`	means paper tape
`T`	means control teleprinter (note that this version of ML/I does not support remote teleprinter I/O).
`M` *`decimal digit`*	means library item from the magnetic tape (or disc) handler (or volume) given by the digit.
`DES`	means channel 36.

The first parameter to ML/I may be followed by a list of names of up to five library items that are to be inserted.

If the parameter list to ML/I is null, card input is assumed. The following are therefore some sample calls of ML/I (written in card batch command format rather than teletype format):

```
&ML1;
&ML1;C;
&ML1;M2,LIB1,LIB2,LIB3;
&ML1;DES,LIB1;
```

The parameter to ML/I specifies the initial input (i.e. the initial value of `S10`). Dynamic changes can be made by resetting `S10`. The relevant codes are listed in Section E.7. In many cases the input routines add an extra newline at a switch of stream or at the start or end of input, thus causing line numbers sometimes to differ from those on other listings.

The following Sections give further information about the individual devices.

E.2.4.4 Card input

A card may contain any character that is acceptable by the card reader device routine. The following optional facilties are available with card input:

a. restricting the columns that are recognised.

b. deleting spaces at the end.

c. changing the terminator.

The input routine adds a newline character to the end of each card.

E.2.4.4.1 Restricting card columns

`S11` and `S12` can be set to restrict the columns of a card that ML/I takes its input from. The rule is that ML/I ignores a character on a card unless

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

The columns of the card satisfying this relation are called the *significant part* of the card. Column numbers go from 1 to 80, and initially `S11` is 1 and `S12` is 80, thus causing the entire card to be examined.

If input is being listed the entire card is always printed, even if some of the columns are being ignored.

E.2.4.4.2 Deleting spaces from cards

It is possible to cause the input routine to remove all spaces at the end of the significant part of a card. The deletion of spaces is controlled by the value of `S15` as follows:

`S15 = 0`	means delete all spaces at end of card.
`S15 = 1`	means delete all but one space at end of card.
`S15 = 2`	means leave card untouched.

Of course if there are no spaces the the end of the card the value of `S15` is immaterial.

Initially `S15` is 0, and this is the recommended option.

E.2.4.4.3 The terminator for cards

When card input is via ACIO a terminator card consisting of three uparrows must be supplied. When input is not via ACIO the standard terminator is a card containing three uparrows in successive columns (normally the first three columns as for ACIO); in addition the user may, using `S13`, specify the 7-bit code of any character which, when found in column 1, is to act as a terminator. Initially `S13` is set to the code for `&`. However '`&`' cards should *not* be used as terminators under batch operating systems since such cards would not be fed to ML/I but taken as batch commands, thus prematurely ending an ML/I run.

In all cases a terminator card acts as a "halt code" to ML/I and the card itself is not subject to ML/I processing.

E.2.4.5 Direct paper tape input

(This Section does not apply to `ASSIGN`ed input).

A paper tape may contain any character that is acceptable by the paper tape input device routine (DRIP). Carriage returns and run out are ignored and line-feeds are treated as newline. If an illegal character is encountered the message

`ERCHAR1` *n*

of DRIP is typed, and the character is ignored.

ML/I starts reading a paper tape from the first character, i.e., it does not, in contrast, for example, to MTEDIT, ignore characters up to the first line feed. The paper tape should end with a halt code. On encountering the halt code, ML/I closes the paper tape reader and outputs, on the control teleprinter, the message

`TYPE C, H or T:`

The user must then type one of the above letters. Their meanings are as follows:

1. `C` means continue. The user should reload the reader before typing `C`. On typing `C` the paper tape reader is re-opened (thus, on the interrupt version of DRIP, causing any characters read beyond the previous halt-code to be ignored), and processing continues with the next paper tape.
2. `H` means halt. This causes ML/I to end the current process.
3. `T` means switch to teleprinter input. A newline is fed to ML/I and `S10` is set to 2 thus causing subsequent input to be taken from the control teleprinter.

If during paper tape input the input device is switched by changing S10, the paper tape reader is not closed and, if input is subsequently switched back to paper tape, it resumes where it left off.

E.2.4.6 Direct control teleprinter input

(This Section does not apply to ASSIGNed input).

Teleprinter input is taken a line at a time. When ML/I wants a line of input it types a newline, the current line number and a colon. The user then types his line of input. Any character acceptable by the device routine EXTYPE (in single character mode) may be typed. The following characters have a special meaning:

a. Carriage return terminates the input line. ML/I types a line feed and processes the line. The carriage return is translated into newline.

b. Line feed is ignored.

c. Back-arrow causes the current line to be cancelled. ML/I types back the same line number and the user can then re-commence the typing of his line.

d. Halt code terminates the current ML/I process.

E.2.4.7 Library item input

ML/I can, if desired, take its input from items in the ALGOL library (SBL5) on magnetic tape or on disc. In particular, when magnetic tape is used, the source code can be placed in the ALGOL library and subsequently edited using the program MTEDIT (available from the Applications Group library) before being fed to ML/I.

The value of S10 used to select library item input is one hundred plus the required handler number.

ML/I uses the standard program LIBRARY to access library items. It loads LIBRARY dynamically if it is needed and it is not already in core. Library items come from magnetic tape or from disc according to which version of the program LIBRARY is supplied.

Library items may be inserted within source coming from another device or the source text may consist exclusively of library item(s).

If ML/I is to take its input from the library at any point during processing (even if input is initially from some other device) the names of the library items to be inserted, in the order in which insertion is to take place, should be supplied as parameters at the end of the parameter list in the call of ML/I, e.g.:

```
&ML1;C,LIB1,LIB2;
```

If input is initially from the library ML/I inserts the library items in its parameter list one by one until it reaches the last one and then stops.

If input is initially from some other device and library item input is selected by resetting S10, ML/I inserts *one* library item and then switches input back to the device that was used initially. Thus library items can easily be interspersed with other text.

At the end of each library item an extra newline is fed to ML/I.

E.2.4.7.1 Example of library item input

If the call of ML/I is

```
&ML1;C,LIB1,LIB2;
```

where `LIB1` and `LIB2` are both library items on handler 3, and the text on cards is of form

```
XXX
MCSET S10 = 103
YYY
MCSET S10 = 103
ZZZ
```

then the contents of `LIB1` will be inserted between `XXX` and `YYY` and the contents of `LIB2` between `YYY` and `ZZZ`.

E.2.4.7.2 Advanced library item facilities

The previous description should be adequate for most uses of library item input, but in special circumstances the user might like to have a more exact description of the mechanisms involved so that he may modify them to suit his own needs. This Section therefore provides a detailed description.

The library item facilities use the S-variables `S30` to `S41`.

`S30` specifies the *revert device*. At the end of each library item ML/I resets `S10` to the value of `S30`. Initially `S30` is set to the initial value of `S10` but its value can be changed dynamically if desired (the same mechanism is used for the insertion of ACIO streams).

On the first request for input from a handler, `S31` and `S32` are examined. If `S31` is zero the current process is ended. Otherwise the library item named by `S31` and `S32` is found and the list of library items is shifted down one item. Thus `S31` becomes `S33`, ..., `S39` becomes `S41`. Hence the name of the next library item (if any) will then be in `S31` and `S32`. At the start of each subsequent library item on the same handler, or if input is switched to a different handler, the above process is repeated. Note, however, that if, within a library item, input is switched to, say, cards and then back again to the same handler, input will remain where it left off in the original library item.

It is quite legitimate for a process to end without using all the library items named as parameters.

By suitably modifying `S30` to `S41`, it is possible for the user to effect such facilities as the dynamic inclusion of library items, the inclusion of more than five items, various unusual switches of input device, etc.

E.2.4.7.3 Acknowledgement

The library item routines in ML/I are based on routines developed by W.H. Purvis of the University of North Wales.

E.2.4.8 Output

When not running under DES there is a fixed designation of output streams; the channel controlled by `S21` goes to paper tape punch one and the channel controlled by `S22` goes to the control teleprinter.

E.2.4.9 Initialisation

ML/I is serially re-usable. 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.

E.2.5 Dumping the environments

Under the DES operating systems, it is possible to dump an ML/I environment to backing store so that it can be restored on later runs. The dump takes the form of core images in a special relocatable form. The main advantage of using dumps is speed. If a set of macros is to be used frequently, it may be best to "compile" them and dump them to backing store so that the resultant core images can quickly be retrieved on every subsequent run.

E.2.5.1 Creating a dump

A dump is made at the end of an ML/I process, and is controlled by bits 2 to 4 of the variable `S18` (bit 1 of `S18` is already spoken for — see Section E.4). It is possible to dump selective parts of the environment, though in practice, no doubt, it will most often be required to dump a complete environment. If any of bits 2 to 4 of `S18` is one a dump is made; the individual bits control what is to go into the dump, as follows:

- if bit 2 is one the S-variables are dumped.
- if bit 3 is one the permanent variables are dumped.
- if bit 4 is one all (local or global) macros, skips, inserts and warnings are dumped.

The dump is always made to channel 200.

A sample deck to create a dump is as follows:

```
&ASSIGN;200;DC;2;LISTDUMP,BSR001;
&DELETE;200;                              (in case it exists already)
&ML1;DES;
MCSET . . .
MCDEF . . .
etc.
MCSET S18 = -1                            (to set all bits in S18)
^^^
```

The dump is created as an SBL8 C-file, but, not being a character file, it cannot be subjected to the usual SBL8 operations, such as editing.

Note that two constituents of the environment are not dumped. These are the delimiters of operation macros (which can be MCALTERed) and the effects of MCNODEF, MCNOINS, MCNOSKIP and MCNOWARN. MCALTERs can be used in creating the material to be dumped, e.g.

```
MCALTER OR TO OC
MCDEF OR AS . . .
```

In this case the definition of OR as a macro would be in the environment when it was restored, but the effect of MCALTERing OR to OC would not be restored. The effect is exactly as if all delimiters and keywords were MCALTERed back to their standard values before the dump was made.

When S-variables are dumped, some of them are omitted from the dump. These are the S-variables that are set directly by the parameters of the call of ML/I, i.e. S10 and S30 to S41. Moreover all the bits of S18 except bit 1 are reset to zero before the dump is made (otherwise, on restoring a dump, the extra bits in S18 would cause it to be re-dumped at the end of the run).

E.2.5.2 Multiple dumps

Dumps can be restored at the start of an ML/I run. It is possible to restore more than one dump. For example if one dump contained an environment to add list processing to ALGOL and another dump contained an environment to add character manipulation to ALGOL, then, provided the two were mutually compatible, both could be restored for the same run. If the same construction is defined in more than one restored environment the normal ML/I overriding rules apply. It does not matter if the same construction is defined in several environments (e.g. MCINS %.), except that it would slightly slow down processing.

To cater for the problem of restoring values of the permanent variables for several different dumps, the following rules have been adopted:

a. all permanent variables are initially set to zero at the start of a process.

b. when a dump containing permanent variables is restored only those permanent variables whose dumped values are non-zero are reset.

c. if a dump contains more than the standard number (ten) of permanent variables, then the number of permanent variables is increased accordingly, and the newly created variables are initially zeroised before the dumped variables are restored (if a dump contains, say 10 permanent variables and a previous dump has restored 20 then the new dump does not, of course, decrease the number to 10 but leaves it at 20).

As a consequence of this, if two dumps use different permanent variables then both can be restored simultaneously with suitable initial values of their own permanent variables.

E.2.5.3 Restoring dumps

To restore some dumps ML/I should be called with the parameter REST. The dumps are assigned to channels 100, 101, 102, etc., an unassigned channel indicating the end of the list. If any of the assigned channels corresponds to anything other than a properly constructed dump, the message:

```
REST ERR
```

is printed and the run abandoned. In detail the action of ML/I when the `REST` parameter is given is as follows:

a. Zeroise all permanent variables.

b. Set S-variables to their appropriate initial values.

c. Set "restoring channel" as 100.

d. If restoring channel is not assigned go to step g).

e. Restore the dump from the restoring channel.

f. Increase the restoring channel by 1 and go to step d).

g. Reset `S2` (the line count) to zero and proceed as with a normal `&ML1;DES;` run.

A simple example of the deck to restore a single dump is as follows:

```
&ASSIGN;100;DC;2;LISTDUMP,BSR001;
&ASSIGN;31;...
&ML1;REST;
. . .
```

In the above case the user should check that channel 101 was not assigned (e.g. as a result of previous activity in the job).

E.3 Character set

ML/I accepts any character that the appropriate device routine accepts.

E.4 Error messages

Error messages are output on the lineprinter as they occur (with reference to Chapter 6 of the User's Manual, the number 2N, the maximum number of characters inserted into an error message without truncation, is 64).

Illegal input characters ordinarily give rise to messages from the appropriate device routines but ML/I itself detects errors from cards, giving the message:

```
ILLEGAL INPUT CHARACTER
```

and substituting a dot for the wrong character.

If the variable `S18` is one, then a list of all the construction names in the environment is printed at the end of a process (this is also done if a dump of the environment is made).

There is a number of extra error messages peculiar to this implementation and these are described in the Sections which follow.

E.4.1 Parameter error

Message

`PARAMERR`

(this message is output by `OUTNAME` both to the printer and to the control teleprinter.)

Description

The call of ML/I has illegal parameters, or more than five library items have been named in the parameter list.

System Action

No processing takes place.

E.4.2 Illegal input stream

Message

`ILLEGAL INPUT STREAM`

Description

`S10` has been set to a value not corresponding to any permissible input stream, or library item input has been demanded when no library item names were supplied in the call of ML/I.

System Action

The process is aborted.

E.4.3 Library item not found

Message

`LIBRARY ITEM` *`name`* `NOT FOUND ON HANDLER` *`number`*

Description

Self-explanatory.

System Action

The process is aborted.

E.4.4 ACIO Overflow

Message

`ACIO OVERFLOW`

Description

The ACIO buffer has become too full.

System Action

The process is aborted.

E.5 Integer calculations

All macro variables and constants used in macro expressions should be less in absolute value than $2^{23} - 1$. Overflow is not detected and its effect is undefined.

The initial environment contains ten permanent variables, which are all set to zero.

E.6 Layout keywords

There are no extra layout keywords, beyond those described in the *ML/I User's Manual*.

E.7 S-variables

The following table summaries the current uses of S-variables and their initial values. The indication *char* means "any legal 7-bit internal code character".

If `S10` is set to an illegal value the error message of Section E.4.2 is produced, but if any of the other S-variables have illegal values the effect is undefined.

E.7.1 Control of input

Variable	Permissible values	Meaning	Initial value
`S10`	0	card input	\
	1	paper tape input	| Set by
	2	teleprinter input	| parameters
	100+*n*	input from library on handler *n*	| to ML/I
	-*n*	input from ACIO channel *n*	/
`S11`	any	starting card column	1
`S12`	any	closing card column	80
`S13`	any	card terminating column	code for &
`S14`	0	don't list source text	0
	1	list source text	
	2	list source text if `S10 = S30`	
`S15`	0	delete all spaces at end of card	0
	1	delete all but one space at end of card	
	2	delete no spaces at end of card	
`S16`	any	code to be translated on input	−1
`S17`	*char*	character to replace `S16`	0
`S18`	any	controls environment listing and dumps	1

E.7.2 Control of output

Variable	Permissible values	Meaning	Initial value
S20	0	don't list output	2
	1	list output	
	2	list output with line numbers	
S21	0	Direct case: don't punch output ACIO case: don't output to channel 31	1
	1	Direct case: punch output ACIO case: output to channel 31	
S22	0	Direct case: don't output to CTP ACIO case: don't output to channel 33	0
	1	Direct case: output to CTP ACIO case: output to channel 33	

E.7.3 Inclusion of source from library

Variable	Permissible values	Meaning	Initial value
S30	as for S10	revert device	same as S10
S31 and S32	see below	name of first library item	zero or parameter to ML/I
S33 to S40	ditto	names of subsequent items	zero or parameter to ML/I
S41	0	terminator of list	0

Permissible values for the pairs of S-variables S31-S32, S33-S34, ..., are zero, and the name of any permissible library item.