

# ML/I User's Manual — Appendix B

Implementation on Titan as Part of the MLS System

P.J. Brown

10th April 1968

Copyright © 1968 P.J. Brown

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>).

## B.1 Restrictions and Additions

The Titan implementation of ML/I contains all the features described in the manual. In addition, it contains two *switch* variables, referred to as **S1** and **S2**. These switch variables are macro variables with global scope which are used and set in a similar way to the permanent macro variables. They are used to control the output produced by the macro processor and its treatment of carriage control characters. See Sections B.2.2 and B.3.2 for details.

Due to typographical restrictions, the ‘section’ character used in MLS is represented by  $\emptyset$  throughout this document.

## B.2 Using ML/I via the MLS System

ML/I is available as part of the MLS system. This system provides stream-handling and phase-changing facilities which make it easy to run multi-phase jobs and to send copies of output streams to several different destinations. A complete description of the MLS system may be found in the document *Titan Mixed Language System* — MLS. This section only describes some of the relevant facilities.

### B.2.1 The MACRO command

The MLS command:

```
(MACRO, arg1, arg2, . . . , argn,  $\emptyset$ arg)
```

enters ML/I with input streams *arg1*, *arg2*, . . . , *argn* in that order. On entry to ML/I the environment always consists of the operation macros, the fifty permanent variables P1 to P50 and the two switch variables **S1** and **S2**. Output will normally be set up on the *input* stream specified by the  $\emptyset$ -type argument. The output is then ready to be used as input to another phase or to be sent to one or more destinations using E3. Examples of simple jobs are given in Section B.2.4.

### B.2.2 Output

All output, other than error messages and text produced by the MCNOTE macro will normally be set up on the input stream specified by the  $\emptyset$ -type argument. However, the switch variable **S1** may be used to send output directly to a specified output stream or to inhibit output altogether. The value of **S1** controls the output as follows:

<b>S1</b> < 0	No output is produced
0 <= <b>S1</b> <= 255	Output is to output stream number <b>S1</b>
<b>S1</b> > 255	Output is to the ‘normal’ stream, i.e. the input stream specified by the $\emptyset$ -type argument

The initial value of **S1** is 256, so output will be on the normal stream until **S1** is specifically MCSET to a different value. If **S1** is set to 0 then all output, including error messages, will be on stream 0. If an attempt is made to MCSET **S1** to a value between 0 and 255, and the output stream corresponding to that number does not exist then the MCSET will be aborted and no change of output stream will take place.

S1 is useful for splitting up the output and sending parts of it to different streams or for inhibiting it altogether while setting up macro definitions. The following examples illustrate the use of S1.

- a. MCSET S1=3; *text* MCSET S1=256;  
This outputs *text* on stream 3 and then returns to the normal stream.
- b. MCSET P1=S1; MCSET S1=3; *text* MCSET S1=P1;  
Similar to a) except that after *text* has been output on stream 3 output reverts to the stream that was previously being used (which may or may not be the normal stream).
- c. MCSET S1=-1; *defs* MCSET S1=256;  
This causes output to be suppressed during the processing of *defs* and then reverts to the normal stream.

### B.2.3 Store

If no STORE command is used, a store limit of 4096 words is assumed. This should be sufficient for jobs with a fairly small number of macro definitions unless the definitions are very long. Macro definitions are held in store one character to a halfword. A store limit of 4096 words allows about 2000 halfwords for macro definitions. If necessary users should allow extra space for definitions at one halfword per character plus about 10%. If more than 4096 words of store are required the MLS command:

```
(STORE,n)
```

will give a store limit of *n* words. This command should come immediately before the MACRO command.

### B.2.4 Examples of Simple Jobs

The following are examples of simple jobs. The comments on the right hand side of the examples simply explain the action of the various sections and do not form part of the program.

- a. The following job simply macro-generates the input streams (HB30/DEFS) and (HB30/TEXT) and sends one copy of the output to magnetic tape and one copy to the printer with line numbers.

```
RUN MLS(HB30/MACRO GENERATE)
INPUT D1 (/TEXT)           List of all the input documents.
D2 (/DEFS)                 Stream numbers can be chosen arbitrarily.
OUTPUT 2 PRINTER
3 TO TAPE 9/80 20 BLOCKS
TAPE 9 (HB11)
***U
(MACRO,2,1,Ø1)             Macro-generate streams 2 and 1 onto stream 1.
(E3,1,F*)                  Edit stream 1 with the follows document.
0 2 L
0 3
*
***Z
```

- b. The following job macro-generates the documents (IIT1) and (IIT2) using the macro definitions obtained by editing (DEFS) with (EDITS). If any errors occur the output is printed with line numbers, otherwise the output is renamed as input P5 and IIT is entered.

```

RUN MLS(HB30/GENERATE IIT)
LIMOUT 500
LIMSTORE 8K
COMP 2 MINUTES
INPUT D1 (IIT1)           List of all the input documents.
D2 (IIT2)                Stream numbers can be chosen arbitrarily.
D3 (DEFS)
D4 (EDITS)
OUTPUT 1 PRINTER
***U
(MEDIUM)                Allow for streams up to 30 blocks long.
(E3,3,4)                 Edit (DEFS) onto input stream 3.
(STORE,8192)             Allow 8K of store for large macro definitions.
(MACRO,3,1,2,05)        Macro-generate onto input stream 5.
(IFERRORS,F*)           Test for errors.
(E3,5,F*)               Print output with line numbers if any errors.
O 1 L
*
*
(PSTREAMS,5)            Rename input stream 5 as P-type.
(IIT)                   Enter IIT.
***Z

```

## B.3 Character Set

All currently assigned internal code characters are allowed except backspace, fault, LC/LS and UC/FS, which are illegal. Erase and run-out are ignored.

Upper and lower case letters are considered to be distinct characters. See Section 7 of the manual for the implications of this. Note that throughout the document where 'spaces' are referred to, this means spaces produced by the space bar. Tab is a character in its own right and is in no way equivalent to two or more spaces. No special provisions are made for input from cards.

### B.3.1 Carriage Control Characters

Different carriage control characters are normally treated as different characters for the purpose of matching macro names and delimiters. The original carriage control character is copied over to the output text where appropriate. However, the switch variable S2 may be used to cause all carriage control characters to be translated to a single newline (2.1) on input. The value of S2 controls the treatment of carriage control characters as follows:

S2 = 0                      carriage control characters not changed.

S2 <> 0 carriage control characters translated to a single newline (2.1) on input.

The initial value of S2 is zero.

## B.4 Error Messages

Error messages are output on stream 0 as errors are detected. There is no limit on the permissible number of error messages other than the output limit on stream 0. The IFERRORS command in MLS allows special action to be taken if any errors have occurred. The use of the MCNOTE macro to produce a message on stream 0 is not considered to be an error as far as the IFERROR command is concerned.

With reference to Chapter 6, the number 2N is 60 and the error character is the question mark (inner set 1.4).

### B.4.1 Implementation-Defined Messages

The special messages and error messages for this implementation are listed below.

#### B.4.1.1 Statistics

##### *Message*

```
AT END OF MACRO PROCESSING
  number LINES number CALLS number INSERTS
```

##### *Description*

This message is output when the end of the input text has been reached or after the current process has been aborted. The message gives the number of lines of the source text which have been processed, and the total number of calls and inserts performed.

##### *System Action*

After this message has been output, the output text is set up on the input stream specified by the Ø-type argument in the MACRO command and control is returned to MLS to process the next command.

#### B.4.1.2 Time expired

##### *Message*

```
(COMPUTATION) TIME EXPIRED - PROCESS ABORTED
(EXECUTION )
```

##### *Description*

The job has exceeded its time limit.

##### *System Action*

The job is allowed a further five seconds and the current process is aborted.

### B.4.1.3 Output limit exceeded

*Message*

```
(ERROR MESSAGE HAVE) EXCEEDED THE OUTPUT LIMIT - PROCESS ABORTED
(OUTPUT TEXT HAS    )
```

*Description*

The output limit on stream 0 has been exceeded or the output limit on the pending stream for the output text has been exceeded (see MLS documentation for details of output limits).

*System Action*

The current process is aborted.

### B.4.1.4 Fault

*Message*

```
FAULT number IN MACRO - PROCESS ABORTED
```

*Description*

Fault *number* has occurred (see *Titan Machine-Code Programming Manual*, Chapter 9, for details of faults).

*System Action*

The job is allowed a further five seconds and the current process is aborted.

## B.5 Integer Calculations

All integers, including intermediate results, must be less than  $2^{20}$  in absolute value. Overflow is not detected and its effect is undefined.

There are 50 permanent variables, P1 to P50, in this implementation.