UNIVERSITY OF EDINBURGH
Edinburgh Regional Computing Centre

User Note 37

(March 1984)

Title:

INLAY

Author: John M. Murison

Contact: Advisory Service

Software Support Category: C

Synopsis

INLAY is a program to assist users who wish to format a document but are not familiar with the conventions of the formatting program LAYOUT.

Keywords

Document formatting, INLAY, LAYOUT

Edinburgh Regional Computing Centre

James Clerk Maxwell Building, The King's Buildings, Mayfield Road, Edinburgh, EH9 3JZ. Telephone 031-667 1081

INLAY vsn 2

INLAY is a program held in EMAS library ERCLIB.GENERAL. Its function is to produce, from a text file, a file suitable for input to the document formatting program LAYOUT (i.e. to create a file IN LAYout form). It is thus in a sense doing the reverse of LAYOUT. LAYOUT is described in a manual which may be obtained from the Advisory Service.

The main purpose of providing such a program is to enable typists not familiar with the conventions of LAYOUT to assist in the preparation of files to be processed by LAYOUT. The text of the document would be typed into EMAS using any text editor and in a format close to what was required for the final document. This file could then be processed by INLAY to create a LAYOUT-input file, which could then be processed by LAYOUT and edited as necessary until the LAYOUT output was satisfactory.

INLAY does not in general produce an exact LAYOUT-input image of the given text file, merely a close approximation, and some editing of its output may be necessary. INLAY does not perform particularly well when dealing with tables: in this case use of the '&' form (described below) should be considered. Details of the file produced by INLAY are given below. An example of INLAY's performance is given at the end of this description.

Input

The file which INLAY is given to work with should contain the text of the document, as near as possible in its final form. It is important to observe regular margins, sentence gaps and paragraph indentations in the input file - otherwise INLAY may use inappropriate LAYOUT directives. The input file may be right-justified, however, i.e. extra spaces may have been added between words in order to make each line the same length (apart from title lines and end of paragraph lines of course). Normally such a file would have been produced by a text formatter, and is presumably being processed by INLAY in order to obtain a LAYOUT-input version of the document.

The input file could have been typed in on-line using a text editor, or prepared off-line using a Flexowriter or other device capable of generating paper tape. Note that the file may contain underlining, obtained by using the backspace (BS) or carriage return (CR) characters, and may also contain tab characters (HT). Form feed (FF) characters are treated as newline characters. Other characters which are not normally permitted to be input on-line to EMAS will be ignored if encountered by INLAY in the input; e.g. the 'Stop' code available on Flexowriters.

If tabs are used in the input file, it is necessary to specify the tab settings in force when the text was prepared. The user is prompted at his terminal the first time a tab character is encountered by INLAY.

If a line in the file is found to consist of a solitary & symbol, then INLAY generates a $L0 directive and causes the lines following to be treated as $L lines (i.e. no directives used in them). A subsequent & on a line by itself causes a reversion to normal processing.

Various details of the format of the document, such as those noted above, are required by INLAY. This information is requested by a series of prompts, output on the user's terminal when INLAY is invoked, as follows:

```
Command:INLAY(input file, output file)
Please give the following information about file INPUT FILE
(Default values in brackets)
Right justified (No): n
Left margin (0): 10
Line length excluding left margin (72): 68
Indent at start of paragraphs (3):
Spaces between sentences (2): 1
```

Notes

* If no reply is given to a particular question the default value is assumed; e.g. see "Indent at start of paragraphs" above.

* The information is requested in the order given above. If the response given to any question is '*', then no further questions are asked, the default values being assumed.

* The line length is taken only as an approximation to the line length used in the input file. The value given can be exceeded by an input line, but any input line greater than 250 characters is truncated. The line length value is used by INLAY when determining whether an input line has been curtailed deliberately (it is assumed to have been curtailed if it is less than four fifths of its expected length).

* If a reply is found to be faulty, the question is repeated.

Output

The file generated by INLAY always starts with the following LAYOUT assignment directives:

```
$A INVERT=0; INVO=0; PAGE=0
$A CAP=0; CAPSH=0; CAPO=0; CAPSHO=0
$A TAB=3,6,9,12,15,...,33,36,39,44,49,54,59,64,69,...,94,99
```

In addition, if values other than the defaults have been specified in the initial set of questions then corresponding assignments are made to the LAYOUT parameters JUST, LEFT, LINE, PGAP and SGAP, in the first $A directive at the start of the output file. No other $A directives are used in the output file, except to change the INDENT parameter. It should be noted that the PAGE parameter is set to 0, thus suppressing all page turns. The setting of the TAB parameter is not changed from the values given above, and whenever INLAY has to place text at a particular column, the tab value closest to that column is used. This is why INLAY sometimes causes text to be displaced slightly.

In other ways the design of INLAY has concentrated on keeping the output file as straightforward as possible. The following points should be noted:

* Where the input file is not right justified, if there are two spaces instead of one between two words in the input file, INLAY treats them as a single space; no action is taken to preserve the two spaces. If there are more than two contiguous space characters in the input then a $T directive will probably be generated.

* If the input file has been described as right justified, INLAY checks that each line is exactly the given line length, then permits up to 5 space characters between two words on the line before using a $T directive to represent it. For sentence gaps, up to 5 spaces plus the given sentence gap are allowed. If the line is not the given line length then it is treated as an unjustified line (see above).

* INLAY tries to detect centred headings, and represents them by use of a $LM directive. The conditions for an input line being a centred heading are as follows: a) the line starts at least two positions away from the current left margin; b) the text length is less than four fifths of the specified line length; c) the text is centred with respect to the specified line length, to within two positions. The $L modifier U is also used if the complete text is underlined.

* INLAY assumes that a line has been deliberately curtailed if it is less than four fifths of its expected length. The next output line will then be preceded by $BO.

* The specified number of spaces should be given between sentences in the input file; otherwise INLAY will insert a '$' before the '.' or '?' or '!' terminating the first of the two sentences.

* INLAY tries to minimize the breaking of running text by directives. It may appear to generate a '$A INDENT' line earlier than necessary because of this.

* Where a tab has to be generated in an output line, INLAY checks that that line starts with a directive. If it does not, then a '$BO' or '$J' is inserted at the start of the line.

Calling INLAY

Library ERCLIB.GENERAL must have been nominated as a SEARCHDIR:

```
Command:OPTION(SEARCHDIR=ERCLIB.GENERAL)
```

This need only be done once. INLAY can then be called:

```
Command:INLAY(input file, output file)
```

'input file' is the text file from which INLAY generates 'output file'. Both files must be specified.

Once INLAY has been called, the questions described above appear on the user's terminal. When these have been replied to, INLAY proceeds to generate the output file.

Suggested Use

The text file should be prepared by whatever means are suitable. A listing of it should be examined and major errors, such as text omitted or misplaced, corrected by editing. It should then be processed by INLAY as described above, the file generated by INLAY processed by LAYOUT, and the result listed. This final file can then be examined and the modifications required to the LAYOUT input file established.

Pagination can also be worked out at this stage. The changes can then be carried out using a text editor on the file produced by INLAY, and the modified file put through LAYOUT again. The original file is no longer required.

For example, suppose that file A has been created from a Flexowriter tape. The following sequence could be used to prepare a suitable LAYOUT input file:

```
Command:INLAY(A, ALI)                          LI for 'LAYOUT Input'
Please give the following ...
 :
 :

Command:LAYOUT(ALI, .LP)

 (The line printer output is then examined)

Command:ECCE(ALI)                              Edit the LAYOUT input file,
>.....                                         not the original file.
 :
 :
>%C

Command:LAYOUT(ALI, AFD)                       FD for 'final document'

Command:LIST(AFD, .LP)
```

Example

An example of the use of INLAY is given on the following pages. Note that the rightmost left margin position has occasionally been displaced by INLAY. This could be corrected by changing the 6th TAB value from 18 to 17 when editing the LAYOUT input file.

The original file input to INLAY:

READ PROFILE PARAMETERS

The meanings of the parameters to read profile are as follows:

key — The keyword relating to the particular profile information; chosen by the program writer.

info — A scalar variable (NOT an array) of any type; usually a %record or a %string.
read profile copies the information from SS#PROFILE to info. If the number of bytes held for the specified keyword in SS#PROFILE is greater than the size of info, then only as many bytes as info can hold are passed; truncation of the file information is from the right. If the number of bytes held in SS#PROFILE is less than the size of info (this includes the case where no information is held for the specified keyword) then the rightmost bytes of info are cleared to 0. See also flag, described below.

version — read profile returns the version number appropriate to the information currently stored in SS#PROFILE. This is 0 if no information is stored for the given keyword (or if SS#PROFILE does not exist).

flag — Set by read profile as follows:

0 Success.
1 Info held in SS#PROFILE is larger than size of info parameter passed. Some rightmost bytes of file info not transferred.
2 Info held in SS#PROFILE is smaller than size of info parameter passed. Some rightmost bytes of info parameter set to zero.
3 SS#PROFILE does not exist.
4 Keyword not found in SS#PROFILE.
5 Failed to connect SS#PROFILE. The attempts to connect SS#PROFILE are as follows. If the program is running in background mode, the program will attempt to connect SS#PROFILE a total of five times, with a delay of 20 seconds of cpu time between each attempt, before returning this flag. If the program is running in foreground mode, up to 3 attempts will be made, with a delay of 1 second of cpu time between each.
6 File SS#PROFILE corrupt.
7 Keyword is null.

If flag=2 on return then the info parameter has been partially padded with zeros, as described above. If flag>2 on return then the info parameter has been cleared to zeros and version has been set to 0. Note that only flag values of 5 or more indicate a definite fault.

The 'LAYOUT input', i.e. the file generated by INLAY:

(All requested values left at their defaults)

```
$A INVERT=0; INVO=0; PAGE=0
$A CAP=0; CAPSH=0; CAPO=0; CAPSHO=0
$A TAB=3,6,9,12,15,18,21,24,27,30,33,36,39,44,49,54,59,64,69,74,79,84,89,94,99
$LM
READ PROFILE PARAMETERS
$B1 The meanings of the parameters to read profile are as follows:
$A INDENT=4
$B1$T0 key$T4 The keyword relating to the particular profile information;
chosen by the program writer.
$B1$T0 info$T4 A scalar variable (NOT an array) of any type; usually a $%record
or a $%string.
$B0 read profile copies the information from SS#PROFILE to info.
If the number of bytes held for the specified keyword in
SS#PROFILE is greater than the size of info, then only as
many bytes as info can hold are passed; truncation of the
file information is from the right. If the number of bytes
held in SS#PROFILE is less than the size of info (this
includes the case where no information is held for the
specified keyword) then the rightmost bytes of info are
cleared to 0. See also flag, described below.
$B1$T0 version$T4 read profile returns the version number appropriate to the
information currently stored in SS#PROFILE. This is 0 if no
information is stored for the given keyword (or if SS#PROFILE
does not exist).
$A INDENT=5
$B1$T0 flag$T4 Set by read profile as follows:
$B0 0 Success.
$B0 1 Info held in SS#PROFILE is larger than size of info
$A INDENT=6
parameter passed. Some rightmost bytes of file info
not transferred.
$B0$T5 2$T6 Info held in SS#PROFILE is smaller than size of info
parameter passed. Some rightmost bytes of info
parameter set to zero.
$B0$T5 3 SS#PROFILE does not exist.
$B0$T5 4 Keyword not found in SS#PROFILE.
$B0$T5 5 Failed to connect SS#PROFILE. The attempts to connect
SS#PROFILE are as follows. If the program is running
in background mode, the program will attempt to connect
SS#PROFILE a total of five times, with a delay of 20
seconds of cpu time between each attempt, before
returning this flag. If the program is running in
foreground mode, up to 3 attempts will be made, with a
delay of 1 second of cpu time between each.
$B0$T5 6 File SS#PROFILE corrupt.
$B0$T5 7 Keyword is null.
$A INDENT=5
$B1 If flag=2 on return then the info parameter has been
partially padded with zeros, as described above. If
flag>2 on return then the info parameter has been cleared
to zeros and version has been set to 0. Note that only
flag values of 5 or more indicate a definite fault.
$E
```

The 'final document', i.e. the file generated by LAYOUT from INLAY's output:

READ PROFILE PARAMETERS

The meanings of the parameters to read profile are as follows:

key — The keyword relating to the particular profile information; chosen by the program writer.

info — A scalar variable (NOT an array) of any type; usually a %record or a %string.
read profile copies the information from SS#PROFILE to info. If the number of bytes held for the specified keyword in SS#PROFILE is greater than the size of info, then only as many bytes as info can hold are passed; truncation of the file information is from the right. If the number of bytes held in SS#PROFILE is less than the size of info (this includes the case where no information is held for the specified keyword) then the rightmost bytes of info are cleared to 0. See also flag, described below.

version — read profile returns the version number appropriate to the information currently stored in SS#PROFILE. This is 0 if no information is stored for the given keyword (or if SS#PROFILE does not exist).

flag — Set by read profile as follows:

0 Success.
1 Info held in SS#PROFILE is larger than size of info parameter passed. Some rightmost bytes of file info not transferred.
2 Info held in SS#PROFILE is smaller than size of info parameter passed. Some rightmost bytes of info parameter set to zero.
3 SS#PROFILE does not exist.
4 Keyword not found in SS#PROFILE.
5 Failed to connect SS#PROFILE. The attempts to connect SS#PROFILE are as follows. If the program is running in background mode, the program will attempt to connect SS#PROFILE a total of five times, with a delay of 20 seconds of cpu time between each attempt, before returning this flag. If the program is running in foreground mode, up to 3 attempts will be made, with a delay of 1 second of cpu time between each.
6 File SS#PROFILE corrupt.
7 Keyword is null.

If flag=2 on return then the info parameter has been partially padded with zeros, as described above. If flag>2 on return then the info parameter has been cleared to zeros and version has been set to 0. Note that only flag values of 5 or more indicate a definite fault.