

February 1984

Hamish Dewar  
Computer Science Department  
Edinburgh University

LAYOUT 1.5 USER'S GUIDE

Provisional -- under development

1. INTRODUCTION TO TEXT FORMATTING
2. INPUT FILE FORMAT
3. PRINTING EFFECTS
4. FONTS
5. DIRECTIVES
6. FORMATTING PARAMETERS
7. CHECKLISTS

## 1. INTRODUCTION TO TEXT FORMATTING

### The need for formatting programs

For run-of-the-mill purposes, text files prepared on a computer system can be listed directly to a printer. This is quite adequate for documents which have no special layout requirements or where the desired layout can be created by use of a standard context editor. However, when the layout requirements are more demanding and when the printing device has greater capability than an ordinary printer, such as multiple font and proportional spacing facilities, this approach is unsatisfactory, since it does not permit the full capability of the printer to be utilised. In addition, it is not the most convenient approach for documents that are likely to go through several drafts or require regular updating. Assistance in meeting both of these requirements is provided by a type of computer program known as a paginator or formatting program. This document provides a description of one such program, called LAYOUT.

### Main formatting operations

The basic aspect of document production which is handled by text formatters is that of laying the text out on the page. The main functions which can be automatically carried out are the following:

line-filling, that is making lines up to a standard length, or as near as can be achieved without splitting a word between two lines;

justification, that is inserting extra spacing to produce an even right margin;

page-filling, that is making pages up to a standard length, and including headings and footings;

applying selected conventions for the layout of paragraphs and sub-sections;

indentation of sections of the text;

formatting of tabular information, so that the columns line up;

creating special printing effects such as underlining.

### Line-filling

When line-filling is in force, the formatter largely ignores the line structure of the original text file, hereafter called the source file. A line-break separating two words is treated as exactly equivalent to a space between two words. Lines for the final document are assembled by taking words in sequence until the next word cannot be fitted onto the current line. At that point, a line-break is made and justification is applied if selected.

Two advantages stem from this mode of operation. Firstly, the source file can be prepared -- and, more importantly, changed -- without too much concern for preserving uniform line lengths. Secondly, a decision to use a different line-length in the final document can be implemented simply by altering one statement in the source file. All the consequential changes in the final document, which may be substantial, come about automatically.

### Justification

If the justification option is selected, LAYOUT inserts additional spacing as necessary into a line to bring its length up to the prevailing maximum line length. This creates a flush right margin in running text.

Justification is applied only to lines which have been automatically filled, unless specially requested.

#### Avoiding line-filling

Obviously it is inappropriate to have line-filling applied to the whole of a document; there are points at which line breaks are wanted in any case (irrespective of whether the next word could be fitted on the previous line). As line-filling is applied only to what can be regarded as running text, in many cases it is unnecessary to take any special action to avoid undesired line-filling. In particular, LAYOUT does not attempt to apply line-filling across a blank line, nor will it move words to the previous line from a line starting with a space. However, there is also a way of indicating that one or more lines in the source text are to be left as they are without being re-formatted.

#### Printing effects

The standard character set available on a computer terminal and ordinary line printers (known as ASCII) is a limited one. It cannot handle special effects such as:

- underlining, bolding and italicising
- subscripts and superscripts
- delicately variable spacing horizontally and vertically
- fonts with proportionally spaced characters
- multiple fonts

By employing special conventions in the preparation of the source text, it is possible to obtain access to all these facilities within the confines of the standard character set.

#### Tables and Indentation

In designing the layout of a document it is often a requirement to cause text to be aligned to a particular position on a line, perhaps to give a standard appearance to header lines or when preparing a table in which columns have to be lined up under each other. For some purposes it may be satisfactory to create the desired effect by preparing the text with additional spacing to put the parts in the desired place. Particularly with a screen editor, this may be a straightforward and convenient operation.

However, there are two considerations which may make it more appropriate to use the facilities provided by a formatter to create the effect indirectly. The first is that, where several lines are laid out according to the same format, it is possible to adjust the format by altering a single definition statement rather than having to modify each case individually. The second is that, when proportional-spacing fonts are to be used in the final document, columns lined up by the addition of spaces on a terminal will not necessarily be lined up in the final document. In general, only leading spaces on a line can be guaranteed to produce consistent positioning of the following text.

## 2. INPUT FILE FORMAT

#### Formatting commands

Files to be processed by LAYOUT are standard text files in which the material which is to appear in the final document is interspersed with format control information which directs LAYOUT to create special effects. The format control sequences are called directives. Directives are not printed in the final document but are interpreted by the formatter to produce the required effect.

All directives are introduced by a single reserved character called the escape marker. By default the dollar-sign ('\$') is used for this purpose, but there is a way of selecting an alternative character if that choice should be troublesome.

The escape marker is followed by the directive name, which is a sequence of letters. It is immaterial whether directive names are typed in upper-case or lower-case. Most of the main directives are given single letter names, for brevity.

For some directives the name may be followed by a numeric value which will be called the directive argument. Examples of arguments are:

1 -2 +3 99 3.4 0.25 0.02" 1/3 1/3"

Arguments are mostly whole numbers, but as illustrated, a decimal point may be present in appropriate cases. Fractional values may also be represented as the ratio of two numbers. When dimensions are being specified, an argument may be terminated by the "" symbol, signifying inches. In a few cases, a minus or plus sign is valid, and means that the value given is relative to some existing value.

Examples of complete directives are

\$B1 \$T+1 \$T4 \$D0 \$TEXT \$GREEK

For those directives which take a numeric argument, the value 1 is assumed if the argument is omitted. So the first two examples above could also be represented by:

\$B \$T+

A complete directive must be terminated by one of the following:

- a space
- a line break
- an escape marker introducing another directive
- a left bracket (see next section)

Apart from the built-in directives which have a pre-defined effect, there is a facility which permits the user to define additional directives. A few of the built-in directives, such as '\$Q', have their own special format.

#### Atoms

When processing running text, LAYOUT applies the operations of line-filling and justification (if selected). The term atom will be used for the basic unit of text which is placed on the line. This is somewhat broader than the concept of a word, since an atom may consist of a word plus a terminating punctuation symbol, for example. The precise definition of an atom is any sequence of characters delimited fore and aft by one of the following:

- a space
- a line-break
- an atom-breaking directive

## Sentences

A sentence is defined to be a sequence of atoms, the first starting with a capital letter and the last ending with any of the following punctuation symbols: '.', '?', or '!'. The only relevance of the sentence as a unit is that LAYOUT inserts extra space at sentence boundaries, the amount being user-selectable.

## User-defined directives

The user is free to define additional directives to achieve effects which can be specified in terms of the built-in directives. This is done by including '\$D' (Define) commands in the source file. The directive name may be any sequence of letters and the definition may be a combination of directives and text. The scope of all the directives appearing in the definition of a user-defined directive may be restricted by following the occurrence of the name with a left bracket.

## Single-character directives

Sometimes where a particular effect is to be used very heavily, the full form of a directive, with preceding escape marker and following space, may be rather clumsy. Accordingly the '\$D' command also allows a single character to be defined as a directive. Any printing character other than a letter, digit, escape marker or bracket may be used. For example, the symbol '^' might be employed as a substitute for the superscript directive, '\_' as a substitute for the underline directive, or the symbol '!' for '\$T+'.

These single-character directives are not preceded by an escape marker and do not require to be terminated by a space. However, the effect of such directives (that is, of the directives in their definition) may be limited by use of bracketing, and this means that if it is required to follow them with a left bracket as actual text, the bracket needs to be preceded by an escape marker.

## Spaces and line-breaks

Multiple spaces are treated as equivalent to a single space, except at the beginning of a source line or anywhere in a line governed by a '\$L' directive.

In general, line breaks in the source file are regarded as equivalent to spaces. However, completely blank lines in the source are significant and a line beginning with a space causes the preceding line break to be treated as significant, that is, as terminating a document line.

Among the built-in directives, a number are defined to have an effect which includes causing a line-break -- '\$P' and '\$I', for example. Others have an effect which includes causing an atom-break -- '\$T' for example. The remainder of the built-in directives -- '\$U' for example -- imply neither a line-break nor an atom-break. Similarly, the occurrence of a user-defined directive does not of itself imply any kind of break, though directives in its definition may do so. It is for this reason that it is important to be aware of the following rules:

- a single space terminating an alphabetic directive is ignored
- a space following a non-alphabetic directive is NOT ignored

## The Escape Marker

When the escape marker is followed by a letter, it introduces a directive. When it is followed by a digit, it introduces a font selection. When it is followed by any other character, it causes that character to be treated in its own right as part of the text, even if it is defined to have some significance for formatting. So, for example, a character chosen as a single-character directive can be included in the final document by preceding it by the escape marker. Similarly, to cause the escape marker itself to appear in the text, it is necessary to duplicate it, including '\$\$' to obtain '\$'.

The following special cases of this convention may be noted.

Preceding a space by the escape marker causes the space to be treated as significant, that is just like any other character making up an atom. This can be useful in several cases. It can be used to provide added spacing, when ordinary spaces would be ignored; it can be used to prevent an unwanted line break between two words; it can be used to prevent adjustment of a particular gap when justification is performed.

Preceding a period, or other sentence terminator, by the escape marker causes it not to be seen as a sentence terminator (useful when the period signifies an abbreviation rather than the end of a sentence).

### 3. PRINTING EFFECTS

The built-in directives for invoking special printing effects are the following:

\$U	underlining
\$H	bolding (heavy)
\$R	superscripting and subscripting
\$<number>	font selection

The scope of these directives (that is, the amount of text which they affect) may be indicated in a number of different ways.

They may be terminated by a left bracket. In this case the effect of the directive extends to the next matching right bracket. Neither of the delimiting brackets goes into the final document. Any of the following pairs of brackets may be used:

( ) [ ] < > { }

Brackets associated with directives may be nested, but note that occurrences of left brackets in the text itself are not relevant to nesting. The escape marker may be used to avoid unwanted matching of right brackets.

If these directives are used in text covered by a \$L (Lines) or \$A (Atoms) or \$W (Words) directive, their effect is localised to the number of lines, atoms or words concerned.

Otherwise, their effect continues until explicitly reset, for example by another use of the same directive. In this connection, \$U and \$H have the effect of alternately switching on and off the related effect.

None of the directives for special printing effects implies an atom break. If they are required to coincide with an atom boundary, they should be preceded by a line-break or space, or followed by a line-break or an EXTRA space.

#### Underlining

The directive '\$U' is used to cause text to be underlined. For example, the effect illustrated in the following line

he will insist in pronouncing it Forfar

could be achieved by

he \$U(will) insist in pronouncing it For\$U(far)

or, without bracketing but utilising the toggle effect, by

he \$U will\$U insist in pronouncing it For\$U far\$U

or, with '\_' defined as equivalent to '\$U', by

he \_will\_ insist in pronouncing it For\_far\_

or, using the \$W (Words) directive to restrict the effect to a single word (or remainder of a word), by

he \$W\$U will insist in pronouncing it For\$W\$U far

or, with '%' defined as '\$W\$U',

he %will insist in pronouncing it For%far

Note the differences in the use of spaces in these examples.

#### Bolding

The directive '\$H' (for Heavy) is used to cause part of the text to appear in bold print. It is used in similar fashion to the underline directive. For example, the effect

for further news watch this space

could be achieved by

for further news \$H(watch this space)

or, using the toggle effect, by

for further news \$H watch this space\$H

#### Superscripting and subscripting

The effects of subscripting and superscripting are obtained by means of the Row directive ('\$R'). The form '\$R-' causes following text to be raised above the current line position, and the form '\$R+' causes text to be lowered below the current line position. In either case, the minus or plus is followed by the amount of the displacement in terms of the height of the font. Thus a value of 1/3 would indicate one third of the font size. This is one of the directives for which it is particularly convenient to introduce a single character substitute. For example, with '\' defined as '\$A\$R-1/3', the following effect

this permits selection of 2<sup>n</sup> colours out of a palette of 2<sup>24</sup>

could be achieved by

this permits selection of 2\ncolours out of a palette of 2\24

More elaborate definitions would select an alternative font as well as altering the printing position.

#### 4. FONTS

Depending on the printer involved and the font library which is available, a certain selection of fonts may be accessed through LAYOUT. In the case of some printers there is the possibility of adding to the existing set of fonts by defining additional ones using the '\$F' command.

A font consists of a set of character shapes, each shape corresponding to a single character in the ordinary ASCII character set employed in the source file. For conventional fonts, the correspondence is as natural as the particular selection of letters and other characters permits, but for special graphical fonts it may well be arbitrary, so that the appearance of the character in the source file bears no resemblance to the way it looks in the final document. The attributes of character shapes which are relevant to formatting are their width in the horizontal direction, and their height (and depth below the line) in the vertical direction. For each font there is a font height which is used as a basis for inter-line spacing and is related to the maximum character height within the font. There is also a font width which is used to establish the character pitch (for fixed pitch fonts) and as an average character width (for variable pitch fonts).

##### Units of measurement

The units of measurement used internally by LAYOUT to determine how to arrange the text on the page are specific to the printer concerned. To avoid introducing unnecessary device dependencies into the source file, heights are usually specified in terms of lines (that is, as multiples of the font height) and widths in terms of columns (that is, as multiples of the font width). Alternatively, either kind of measurement may be specified as an absolute measurement in inches. The second form is distinguished from the first by the use of the double-quote symbol for inch (as in 0.2" or, equivalently, 1/5")

##### Font selection

At any time there is a numbered set of fonts available for selection, as a result of font definitions, pre-defined or user-created. Initially, a standard default font is selected, so that if no other specification is made, this is the font that will be used. At any point in the text file another font may be selected by means of a font selection directive.

This takes the form of an occurrence of the escape marker immediately followed by the number of the font. The extent of the text to which the font selection is to apply can be indicated in the same way as for the special-effect directives.

Font selections neither imply a line-break nor terminate an atom; as with alphabetic directives, a single space following a font selection is ignored. For example, assuming the prevailing font to be number 16 and that a gothic font has been defined as font 23, the effect

as a recent article in *Die Welt* states

could be achieved by

as a recent article in \$23 Die Welt \$16 states

or, using bracketing, by

as a recent article in \$23(Die Welt) states

#### 5. DIRECTIVES

##### The \$L directive

The 'L' stands for Lines and this directive is used to indicate that a certain number of lines in the source file are to be taken as they stand. The number involved is specified after the letter 'L'. For example '\$L5' means 'leave the next 5 lines alone' and '\$L1' means 'leave the next line alone'. As a special convention for this directive, a value of '0' (zero) may be specified, and this means 'leave all following lines alone until another line-breaking LAYOUT directive is reached'. The zero convention is useful when the number of lines is quite large, since it saves counting them.

The lines concerned start with the line following the one on which the \$L directive appears. The \$L directive may be followed on the same line, or subsequently, by special-effect directives; the scope of all of these will be limited to the number of lines specified. For example, the following source lines:

\$L2\$H

First line of pair

Second (and last) line of pair

would produce the following output:

First line of pair

Second (and last) line of pair

##### The \$A directive

The '\$A' (Atoms) directive is used to restrict the scope of special effects to a specified number of following atoms. The effect of any of the relevant directives occurring within the number of atoms indicated after the '\$A' directive expires at the end of the group of atoms.

##### The \$W directive

The '\$W' (Word) directive is similar to the '\$A' directive, except that its effect is also cancelled by the appearance in an atom of a character other than a letter.

##### The \$B directive

While it is possible to cause blank lines to appear in the final document simply by including blank lines in the source text, it can be more convenient in some cases to make use of the '\$B' directive. The 'B' stands for Blank and this directive is used to cause a specified number of blank lines to be included at the point at which the directive occurs. Alternatively, the amount of space to be left may be specified as an absolute amount in inches. Thus '\$B4' means 'leave 4 blank lines' and '\$B0.5"' means 'leave blank space measuring half an inch'. Note that '0' (zero) has its natural significance for this directive, so that '\$B0' causes a line break but does not introduce any blank space.

If the amount of space specified in a '\$B' directive is not available on the current page, a page break is made but no residual blank space is left at the top of the next page.

For printers which have the relevant capability, a negative number is also valid, though reverse movement is always limited to the current page.

### The \$P directive

At a point in the text where it is required to start a new paragraph, the user may find it most convenient simply to leave a number of blank lines (typically two, one or none) and to indent the first line of the paragraph by including a few spaces. Again, however, for some purposes, it may be preferable to use a specific LAYOUT marker to mark the start of a paragraph. This is the '\$P' directive.

The effect is:

- (a) a line break is made
- (b) a specified number of blank lines are left
- (c) a page break is made if fewer than two lines are available on the current page
- (c) the first line of the paragraph is indented by the amount specified by the LAYOUT parameter PGAP (paragraph gap).

The number of blank lines to be inserted is indicated by a number immediately after the 'P'; '0' (zero) is permissible, as is an amount in inches.

### The \$N directive

As with line-filling, the normal action of LAYOUT is to put as many lines on a page as will fit within the defined page size (as defined by the LAYOUT parameter PAGE). The '\$N' (Newpage) directive may be included in the source at any point where a page break is wanted whether or not further lines would fit on the current page.

### The \$D directive

The Define directive is used to introduce a new directive. It takes the form

```
$D newname = definition
```

The name being defined ( newname ) may be any sequence of letters. This is followed by an equals sign and the text of the definition which extends up the end of the source line. If it is required to include line breaks within the definition, these should be preceded by a hyphen (minus) which is not treated as part of the definition.

After such a definition the name defined is available for use as a directive, preceded by the escape marker.

### The \$V directive

If pagination is performed in a simple-minded fashion, it can sometimes lead to unfortunate page breaks, with, for example, a heading separated from the start of the text to which it applies. For this reason, it can be prudent to include in the source file at appropriate points, '\$V' (Verify) directives which cause a page break to be made if less than a certain amount of space remains unused on the current page. The 'V' is followed by the number of lines to be checked for, or the absolute amount in inches. Often, Verify directives are combined with other directives in a stereotyped group to give effect to the user's preferred way of, for example, starting a new section of the text. The following source line might be used to introduce a section heading, to be underlined and preceded by two blank lines:

```
$B2 $V4 $L1U
```

Note that in this context the Verify directive should follow the Blank directive, whereas a Verify followed by a Blank directive may be used to guarantee that a certain number of blank lines should be left and these should not be broken by a page boundary.

### The \$T directive

The way in which fixed positioning effects are achieved in LAYOUT depends on the familiar concept of 'tab' settings. There are two aspects to this: setting the tab positions and using them. Setting tabs is achieved by including in the source file an assignment like the following:

```
$TAB=10, 20, 40 (positions in columns)
```

```
or $TAB=1", 1.6", 2.0", 3.2" (positions in inches)
```

which has the effect of establishing tabs at the column positions or absolute positions specified. Any desired number of tab positions may be established (up to a maximum of 25).

Subsequently, a particular tab position may be selected by including a '\$T' directive at the appropriate point in the text. The 'T' is followed by the tab number, counting from 1. -- there is a tab 0, but it is set immutably to column 0. For example, the following effect

```
1.      New equipment  11.30
2.      Maintenance   12.15
3.      LUNCH          12.45
```

could be achieved by

```
$A TAB=14, 34
```

```
$LO
```

```
1. $T1 New equipment $T2 11.30
2. $T1 Maintenance $T2 12.15
3. $T1 LUNCH $T2 12.45
```

Tabbing can also be specified relative to the current position using the forms '\$T+' (forward tabbing) and '\$T-' (reverse tabbing). It is often convenient to provide a single character substitute for '\$T+1' (ie move to next tab stop), so that for example, the above effect could be achieved by

```
$D |= $T+1
```

```
$LO
```

```
1. |New equipment |11.30
2. |Maintenance |12.15
3. |LUNCH |12.45
```

### The \$I directive

The method used to cause a section of text to be indented also makes use of tab settings. That is, the amount of indentation required is expressed as 'to a specified tab setting'. This is done by means of the '\$I' directive followed by the required tab number. The scope of this directive may be indicated by bracketing. Its scope may alternatively be limited by its occurring within the number of lines specified by a '\$L' directive, or it may be cancelled by another occurrence of '\$I', typically '\$I0' to stop indentation altogether. The forms '\$I+' and '\$I-' are also valid for this directive, indicating selection of a tab setting so many away from the current indentation selection.

Indentation is often used in order to present a sequence of enumerated points. It may be a requirement that the numerals or letters or whatever used to label the individual sections of text should not be indented, or should be indented less than the body of the text. For example:

1. This is main point number one. Like other main points it is indented to tab setting 1, but the identifying number '1.' is tabbed back to the start of the line.
2. This is main point number two. It starts off just like the previous one, but now, to make life interesting, we decide to include a couple of sub-points or sub-sections here.
  - (a) Sub-point one. The text part is indented to tab setting 2, but again the label '(a)' is tabbed back.
  - (b) Sub-point two. Same again.

3. This is main point number three, so we have returned to the same pattern as for main points one and two.

To achieve this effect, the source file could be prepared as follows:

```

$A TAB=6,14
$!1
$T- 1. $T+ This is main point number one. Like other main points
it is indented to tab setting 1, but the identifying number '1.'
is tabbed back to the start of the line.
$B1$T- 2. $T+ This is main point number two. It starts off just like
the previous one, but now, to make life interesting, we decide to
include a couple of sub-points or sub-sections here.
$!2
$T- (a) $T+ Sub-point one. The text part is indented to tab setting
3, but again the label '(a)' is tabbed back.
$B0$T- (b) $T+ Sub-point two. Same again.
$!1
$B1$T- 3. $T+ This is main point number three, so we have returned to
the same pattern as for main points one and two.

```

Apart from the use of multiple indentation levels, the main point to note is the use of the tab selections to position the labels.

#### The \$F directive

This directive is used for font definition. The form currently supported is

\$Fn = m S=scale? H=height? W=width?

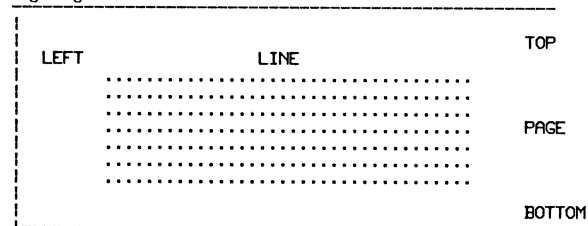
n is the number of the font being defined  
m is the number of a Gimms font (see charts)  
scale is a scaling factor: if it is greater than 1 the font  
is scaled up from the default size, less than 1 down.  
height is an imposed font height superseding the internal font height  
width is an imposed character width superseding the internal  
character widths, and producing a fixed pitch font (looking ghastly).

#### 6. FORMATTING PARAMETERS

There is a small set of parameters which control the way text is formatted on a continuing basis. There are default values to which the parameters are initially set, which are intended to be convenient for the most common requirements. The default values shown in the checklist below are typical ones; but they may differ for different printers and paper formats.

Some of the parameters define the physical format and dimensions of the final document and the registration of the image on the physical page. Typically any required alternative assignment of these would be made at the start of the source file and not subsequently altered. Other parameters, like the level of indentation, are more volatile, and are likely to be re-defined at regular intervals.

The parameters which define the physical format of the output document are indicated in the following diagram:



LEFT defines the left margin, while LINE defines the number of actual printing positions. Note that the intention is that LEFT should be used solely to control the placing of the text on the physical page; it should not be used for indentation. TOP and BOTTOM define the number of lines to be left at the top and bottom of the physical page respectively, and again PAGE defines the number of actual printing lines. TOP plus PAGE plus BOTTOM should add up to the intended physical page size (66 lines for standard line-printers).

There is a parameter PAGENO (page number) which controls whether (and how) pages are numbered. Its default value is zero, which means no page numbering. If it is assigned a non-zero value, pages are numbered starting with the given value, and as successive page breaks are made, the number is increased by one. There is an additional parameter SECTNO (section number) which can be used to number sections (here understood to be a major unit of the document) and pages within sections. When page-numbering is requested, the number appears midway along the middle line of the bottom margin.

The parameter JUST controls whether lines are to be justified or not. Assigning the value 1 to it enables justification, while the value zero (the default) disables justification.

## 7. CHECKLISTS

### Directives

L:line-breaker A:atom-breaker

<b>\$An</b>	-	restrict scope of following directive(s) to next n atoms
<b>\$Bn</b>	L	insert n blank lines
<b>\$Bk"</b>	L	insert k inches of blank space
<b>\$B0</b>	L	start a new line
<b>\$Cn</b>	A	align to column position n
<b>\$Ck"</b>	A	align to position within line specified by k inches
<b>\$D id=...</b>	L	define new directive id
<b>\$E</b>	L	end of source file
<b>\$Fn=...</b>	L	font definition
<b>\$G ...</b>	L	get specified file and include in source text
<b>\$H(...)</b>	-	print bracketed text in bold (heavy)
<b>\$H</b>	-	start/stop bold (heavy) effect
<b>\$In(...)</b>	L	indent all text within brackets to tab setting n
<b>\$In</b>	L	set indentation level for subsequent text
<b>\$J</b>	L	terminate line with justification
<b>\$K</b>		
<b>\$Ln</b>	L	leave the next n source lines alone
<b>\$L0</b>	L	leave source lines alone until next line-breaking directive
<b>\$M</b>		
<b>\$N</b>	L	start a new page
<b>\$O</b>		
<b>\$Pn</b>	L	start a new paragraph, inserting n blank lines
<b>\$Q ...</b>	A	ignore rest of source line if condition false
<b>\$R-f</b>	-	negative displacement by fraction f for superscripting
<b>\$R+f</b>	-	positive displacement by fraction f for subscripting
<b>\$S(...)</b>	-	print bracketed text in italics (slanted)
<b>\$S</b>	-	start/stop italic (slanted) printing
<b>\$Tn</b>	A	align to tab setting n
<b>\$U(...)</b>	-	underline bracketed text
<b>\$U</b>	-	start/stop underlining
<b>\$Vn</b>	L	start a new page if within n lines of end of present page
<b>\$W</b>	-	restrict effect of following directive(s) to next word

### Control parameters

<b>TOP</b>	margin at top of page
<b>PAGE</b>	usable page length
<b>BOTTOM</b>	margin at bottom of page
<b>NLS</b>	line spacing
<b>LEFT</b>	left margin
<b>LINE</b>	line length
<b>SGAP</b>	spaces between sentences
<b>PGAP</b>	spaces at start of paragraph
<b>TAB</b>	current tab settings
<b>PAGENO</b>	current page number
<b>SECTNO</b>	current section number
<b>JUST</b>	justification selector
<b>ESCAPE</b>	escape character



## Changes from LAYOUT 1.0

The capability defined in this document is an interim stage in the enhancement of LAYOUT to provide greater flexibility of control over page make-up. Users of older versions of LAYOUT will find that existing files are acceptable to the new version with little modification, though naturally some changes will be required to access multiple fonts and a number of effects which were 'built-in' to the old version (such as %) require to be defined in the new. A set of stock pre-definition files are available in the Filestore directory DOC for use in creating commonly required environments. One general-purpose one is DOC:STYLE1.LAY. These may be incorporated using the '\$G' directive.

Several features have been retained specifically to provide compatibility although they are not now documented and will probably be phased out.

The most likely source of incompatibility arises from the fact that the new version supports user-defined directives, which may be arbitrary sequences of letters. This leads to a stricter requirement for termination of directives which disallows such usages as \$lu or \$p immediately followed by text. This may mean having to insert spaces in existing files to avoid 'Unknown directive' reports.

Although the actual incompatibilities are small, there are now better ways of achieving certain effects than were available before and it is hoped that users will graduate to these through time.

Specific modifications:-

1. Stricter requirement for directive termination
2. Columns numbered from zero, not 1.
3. Blank lines treated as significant.
4. Spaces at start of line treated as significant.
5. Lines starting with space treated as implying line break.
6. Underline etc. markers no longer built in
7. INVERT defaulted off and no default capital conventions.
8. No 'updated source' facility.
9. \$I re-defined (but source files using INDENT will still work).
10. \$S re-defined.

## Availability

This version of LAYOUT is still under development. There are one or two outstanding issues of definition and implementation to be resolved; constructive suggestions would be welcome. For those prepared to be guinea pigs, it is available on the Laser Printer for which the Filestore directory LP1 is the spool directory. All files should have the extension .LAY. Bear in mind that the cost per page using this device is comparable to that of daisy-wheel printers, about 5 pence.

A preview capability is available on APMS with level 1 graphics. This is invoked by the command DOC:LG1. The program prompts for a file name and then displays the result of formatting on the graphics screen at a default reduction of 4 compared with the laser printer. The reduction factor can be altered by typing a number in response to the file-name prompt. After creating each page image, the preview program pauses until RETURN is pressed. The cursor keys may be used to alter the part of the page image which is displayed (but only that part of the page which fits in the graphics framestore is available).