

Layout

Layout is a text formatting language defining commands (called *directives*) which are embedded in the source text (the *manuscript*), and which control the way that the output (the *formatted text*) is produced.

This note describes *Layout* version 1.5, which is an intermediate version but the best for which the department has the source code. The only available document on this version of *Layout* ('Layout 1.5 Provisional User's Guide' by Hamish Dewar, Edinburgh University Computer Science Department, February 1984) is incomplete and slightly inaccurate, as well as not being widely publicised. It is hoped that this note will serve as a manual for *Layout* 1.5 and so make it easier to use.

A sample *Layout* 1.5 manuscript file (the manuscript for this document) is in the file Text:Lay.lay on the APMs.

1. Escape Marker

A basic characteristic of *Layout* is that the manuscript, in general, represents the text to be printed unless explicitly flagged otherwise. The method of flagging text as other than text to be printed (i.e. as a directive) is to precede it with the *escape marker*. By default, this is '\$', the dollar sign, but if this is an awkward choice it can be redefined in the document.

2. Atoms, Words and Sentences

The *atom* is the basic unit of text recognised by *Layout*. Atoms are not split when line filling, and are not stretched when justifying the line. An atom is defined as a sequence of characters, delimited on both sides by one of the following:

- a space
- a newline
- an atom breaking directive

A word is similar to an atom, but it consists entirely of letters; that is, it is terminated also by any character other than a letter. It is relevant only to the *explicit words* directive.

A sentence consists of a sequence of atoms starting with a capital letter and finishing with a full stop, a question mark or an exclamation mark. The only relevance of sentences in *Layout* is that extra space is inserted between them, the actual amount of space being user-definable.

3. Spaces and Newlines

Within the scope of an *explicit lines* directive, both spaces and newlines within the text are treated as significant, and are preserved in the formatted text. Outside any such scope, line filling is enabled and, in general, spaces and newlines serve only to separate words and are treated equivalently; multiple spaces, for example, are treated as a single space. Any additional spacing effects are produced with directives. However, some commonly occurring requirements are covered by letting spaces and newlines be significant in certain situations:

- Blank lines (formed by two consecutive newlines) are preserved in the text; thus blank space can be produced simply by leaving blank lines in the manuscript.
- Spaces at the start of a line of the manuscript are significant, and are preserved in the formatted text. They also make the preceding newline significant, that is, they force a line break. Thus, a new paragraph can be started by leaving the required number of blank lines, then starting the first line of the new paragraph with several spaces.

Spaces are also treated as significant after explicit positioning directives, such as moving to a tab stop.

4. Directives

Directives can take several forms, as follows:

- Escape marker, followed by a single letter. This invokes a basic (built in) directive. The directive may be followed by one or more arguments, depending on the particular directive. The case of the letter is not significant.
- Escape marker, followed by a number. This selects a previously defined font. There can be up to 64 fonts, numbered from 0 to 63, of which 0 and 1 are predefined as horizontal and vertical fixed pitch (and rather grotty) fonts.
- Escape marker, followed by two or more letters. This invokes a previously defined macro. The case of the letters is again not significant. The invocation is implemented as straight text substitution; thus the macro itself needn't be syntactically valid as long as it becomes valid in context, and nested macro calls are allowed (up to a maximum of 16 levels).
- A single non-alphanumeric symbol. This invokes a previously defined

single-character macro (the term applies to the name of the macro, rather than its body); the invocation is implemented as above.

Note the possible ambiguity between a built in directive followed by text, and a macro call. This means that some way of terminating directive names is required. In the first three cases above, a single space after the directive will be ignored.

5. Arguments

Some directives take a numeric argument, usually specifying a measurement but sometimes specifying other items such as fonts. There is a default unit (such as lines, pixels, tab settings and so on) for the argument, dependent on the directive; the argument is specified as a multiple of this unit. Where non-integer values are relevant, the multiple can be expressed as a decimal, or as a fraction with the numerator and denominator separated by an oblique, '/'; the units may be changed to inches by following the argument with '"'.

Some directives allow the argument to be specified relative to some current value; this is done by preceding the multiple with either a '+' or a '-'.

6. Scoping

Layout has facilities for *scoping* of directives, i.e. for restricting their effect to a small part of the document. These scopes can be nested. At the end of the scope, the state returns to that before the start of the scope; that is, the effect of any directives invoked within the scope does not persist beyond the end of the scope.

Scoping can be done in one of two ways.

- Some directives can be followed by an opening bracket, signifying the start of the scope. The corresponding closing bracket ends the scope. The closing bracket does not imply an atom break, regardless of whether or not the directive that caused it does. Changing font, invoking a macro and some built-in directives come into this category. Four different pairs of brackets are allowed: round brackets '()', square brackets '[]', curly brackets '{} ' and angle brackets '<> '.
- The directives *explicit lines*, *explicit atoms* and *explicit words* each cause a new scope lasting for the stated number of lines, atoms or words to be produced. This scope is just like any other; thus, the effect of any directives invoked within the scope doesn't persist beyond the end of the scope.

In the absence of any such scoping, the effect of the directive persists until explicitly stopped.

7. Layout Parameters

Several parameters, such as line length and the tab settings, control the layout of the formatted text. Each of these has a default value, given below. Some of the parameters, such as page length and margin sizes, are normally set no more than once within the document; others, such as the level of indentation, are likely to change frequently. The parameters can be changed with the *assign* directive. The list of parameters is given below.

- *top* (default $\frac{2}{3}$ "), the top margin
- *bottom* (default $\frac{4}{3}$ "), the bottom margin
- *page* (default $9\frac{2}{3}$ "), the height of the page on which to allow printing. *Top* + *page* + *bottom* should always equal the physical height of the page, $11\frac{2}{3}$ ".
- *left* (default 0), the left margin
- *line* (default 8"), the line length
- *nls* (default $\frac{1}{8}$ "), the height of each line
- *tab* (default at $\frac{2}{3}$ " intervals), settings of up to 25 tab stops, numbered 1 to 25. Tab stop 0 is at the left margin and can't be changed.
- *indent* (default 0), the current level of indentation, in tab stops
- *sgap* (default $\frac{1}{6}$ "), the amount of extra space inserted between sentences
- *pgap* (default $\frac{1}{4}$ "), the amount of indentation at the start of each paragraph, as defined by the *new paragraph* directive
- *just* (default 0), whether or not justification is enabled (zero means disabled)
- *pageno* (default 0), the page number. If zero, then page numbering is disabled. The page number is printed in the middle of the bottom margin.
- *sectno* (default 0), the section number. If zero, or if *pageno* is zero, the section numbering is disabled. If enabled, page and section numbers are given in the form '1-2', meaning page 2 of section 1.
- *start* (default 1), the first page to be printed
- *finish* (default 9999), the last page to be printed. Together with *start*, this provides the capability of printing only a part of the document.
- *escape* (default '\$'), the escape marker

8. Macros

It is possible to define directives to perform frequently required tasks. This provides some of the advantages that routines provide for programming languages; it reduces the size of the manuscript and makes it more legible.

Two different sorts of name for user-defined directives are available:

- Single-character names, for extreme brevity. The character can be any non-alphanumeric character except for the escape marker and brackets. For example, the definition

$\$d / = \$8\$r+0.3$

where font number 8 is smaller than the normal font, will allow the symbol ‘/’ to be used for subscripts; the manuscript text

$x/(1)$

will produce the formatted text

x_1

- Arbitrary names, consisting of a sequence of two or more letters. The definition

$\$d \text{ dot} = \$b0 \$y+0.09" \$x+0.3" \$x+9 \$o9 \$y-0.09" \$x0 \$t1$

produces the bullet character at the beginning of each of these paragraphs.

9. Other Uses of the Escape Marker

When the escape marker is followed by a letter or a number, it denotes a directive. Following it with any other character forces *Layout* to treat that character as normal text, regardless of any special meaning it may have. Useful cases of this are as follows:

- The escape marker followed by a space causes the space to be treated as a normal character. This can be used to prevent a line break between two words; when combined with a normal (un-escaped) space, it provides extra spacing between words.
- The escape marker followed by a full stop (or a question or exclamation mark) prevents the character from being recognised as a sentence terminator, for example when it signifies an abbreviation instead.
- Preceding a valid closing bracket, a character defined as the name of a directive or the escape marker itself with the escape marker causes that character to be included in the text as a normal character.

10. Description of Built-In Directives

There follows a description of each of the built-in directives provided by *Layout*. A table is given afterwards, summarising the directives and their various properties.

Assign (\$A)

The *assign* directive is used to change the values of *Layout* parameters. For example,

```
$A just = 1
```

Several assignments can be made at once, separated by semicolons:

```
$A left = 2"; line = 4"
```

The default unit for the parameters *left*, *line*, *sgap*, *pgap* and *tab* is the width of a space in the current font; for the parameters *top*, *bottom*, *page* and *nls* it is the line height (the current value of *nls*); for the others, the values must be integers. There is a special case for the parameter *escape*; the form

```
$A escape = '%'
```

sets the escape marker to '%'.

The new value can be given relative to the old value, by preceding it with '+' or '-':

```
$A indent = +1
```

The old value of the parameter can be saved on a stack by replacing the '=' with '<':

```
$A indent < 1
```

saves the current value of *indent*, then sets *indent* to 1. A new value needn't be given:

```
$A indent <
```

saves the current value of *indent*. The saved value can be retrieved by replacing the '=' with '>':

```
$A indent >
```

retrieves the previously saved value of *indent*. It can even be retrieved and then changed:

```
$A indent > +1
```

though this feature is of dubious value.

The *assign* directive must stretch to the end of the line; no other directive or text can follow it.

There is an alternative version of this directive. The 'A' and the intervening space can be left out, but only one assignment can be made. For example:

```
$Tab = 1", 2", 3"
```

The '\$A' version was going to be phased out and replaced by this alternative, to allow the name to be used for the next directive.

Explicit Atoms (\$A)

The *explicit atoms* directive is used to set up a scope which lasts for a specified number of atoms.

Note the ambiguity of names between this and the *assign* directive. If the '\$A' is followed directly (with no intervening spaces) by another escape marker (which is presumably the start of another directive) or a number, then it means *explicit atoms*, otherwise it means *assign*.

Blank (\$B)

The *blank* directive has three uses. Firstly, it inserts blank space into the document. The amount of space is determined by its argument; the default unit is in lines, but the amount can be specified in inches too. Secondly, the directive causes a line break. The directive '\$B0' thus starts a new line without inserting any blank space. Thirdly, if given a negative argument, it causes a movement up the page after the line break.

The *blank* directive has no effect if called at the top of the page, if the new page is not explicit. *New page* and *verify* are the only directives that cause an explicit new page; any other cause is implicit. The very start of the document is treated as an explicit new page.

Move to Column (\$C)

This directive can be used in two ways. When used within or at the end of an atom (but not at the beginning of one), it changes the spacing between characters. It can thus be used for overprinting one character with another. The most useful way of specifying the argument in this case is relative to the current position; the default unit is the width of a space character. When used between atoms, or at the start of an atom, it acts like moving to a tab stop at the specified position.

Define (\$D)

This directive is used to define macros. The name of the macro is given (either a single character or two or more letters), followed by an equals sign, and then the body of the macro. The body extends to the end of the line. The macro can be forgotten by redefining it with an empty body.

End (\$E)

The *end* directive terminates the manuscript file. It is not obligatory; *Layout* will insert one itself if there is none there.

Font (\$F)

This directive is used to define fonts. Its argument is the number of the font to be defined, which can be any number from 2 to 63 (fonts 0 and 1 are predefined); this is followed by an equals sign.

The definition can be given in two ways. The first way is with a font name, which can refer either to a raster font read in with the *get file* directive, or to a Times Roman font. The latter case gives access to various sizes of normal, bold, italic and scientific fonts. The names of these fonts are of the form 'TimesXnn'. 'X' must be one of 'R', 'B', 'I' and 'S', meaning 'Roman' (normal), 'Bold', 'Italic' and 'Scientific', respectively. 'nn' is the point size; available point sizes are 6, 7, 8, 9, 10, 12, 13, 14, 16, 18 and 24, with 5 point Roman, 4 and 5 point Bold and 5 point Scientific fonts also available.

The second way is to define in terms of a vector font. There are 31 Gimms fonts and 5 Times fonts available; by default they are read in, but they can be left out if they are not needed. The number of the desired font is given first. The font can be adjusted by following the definition with one of the following:

- 'S', followed by a number, gives the scale factor (magnification). 256 gives no scaling, and other numbers scale accordingly.
- 'H', followed by a size, specifies the ascender height for the font.
- 'D', followed by a size, gives a descender height for the font.
- 'W', followed by a size, makes the font fixed pitch by specifying a width for all the characters.

The last three use inches as the default unit.

Get File (\$G)

This reads in a file, for example one of the files containing Times font definitions. It can also be used to read in and display a *pseudo display file*, which has been created with a graphics editor like *Edwin Draw*.

Heavy (\$H)

This directive switches on and off ‘emboldening’ of text. This is implemented by printing characters several times, each time displaced by one pixel. The default amount is two extra pixels, but any number between 1 and 7 can be used. The directive can be followed by a bracket, restricting its scope.

Indent (\$I)

The *indent* directive is an alternative to the *Layout* parameter *indent*. If the parameter has been used in the document, then the directive acts like the *tab* directive, but the argument is given relative to the current level of indentation (as opposed to the nearest tab stop). If the parameter has not been used, then the directive sets the level of indentation.

Justify (\$J)

This directive causes a line break, like the ‘\$B0’ directive, but also forces justification of the completed line.

Explicit Lines (\$L)

The *explicit lines* directive is used to set up a scope that lasts for the specified number of lines. If the argument is zero, then the scope will last for an indefinite number of lines; it is terminated by any line-breaking directive. Indentation is normally switched off for the extent of the scope.

An *assign* directive, calls to macros and font changes can be placed on the line after the *explicit lines* directive; they will be executed at the start of the scope, and their effect restricted to within it. There can also be several other parameters on the line. These are:

- ‘M’ to centre the lines
- ‘I’ to indent the lines
- ‘H’ or ‘B’ to make the lines bold; these can be followed by a number specifying the degree of boldness, as for the *heavy* directive
- ‘U’ to underline the lines; this can be followed by a number giving the depth of the underline, as for the *underline* directive

New Page (\$N)

This directive closes the current page and starts a new one.

Orb (\$O)

This draws a solid circle of the specified radius (in pixels) at the current position. Note that because the directive is a line-breaker, if there are no other positioning commands before it then the current position will be at the start of the line.

New Paragraph (\$P)

The *new paragraph* takes as its argument a number of lines. If the current position is below the top of the page, then this number of lines is left. If there are less than two lines of space left on the page, a new page is taken. The following text is then indented by the *Layout* parameter *pgap*.

Row (\$R)

This directive behaves in two different ways, depending on whether its argument is absolute or relative. The default unit is the height of the current font.

If the argument is relative, then the directive causes the vertical displacement of text on the line to be adjusted. It can take brackets to delimit its scope. In this case it doesn't cause an atom break, so it can be used within an atom.

If the argument is absolute, the directive causes a move to that position down the page, and breaks the atom. It can't be followed by brackets.

Tab (\$T)

The *tab* directive causes a move to the specified tab stop. The tab stop can be given relative to the current position, in which case it causes a move by the specified number of tab positions.

Underline (\$U)

This directive switches underlining on and off. Its scope can be delimited with brackets. The default distance of the underline below the base line of the text is 5 pixels, but any number from 1 up can be specified with the directive.

Verify (\$V)

The *verify* directive is used to check that a certain amount of room is left on the page. If there is not, then a new page is taken. The directive can be used to ensure that a chapter of the document is kept on one page; it can also be used to check that a new section, say, is not started too close to the foot of the page.

Explicit Words (\$W)

This directive is similar in effect to the *explicit atoms* directive, but using words rather than atoms.

Set X Coordinate (\$X)

This directive moves to a specified position across the page. It can be followed by a comma and another position, which is treated as a y coordinate to move to as well. The positions can be given relative to the current position. If the pen width (see below) is not zero, then a line is drawn from the current position to the new position.

The same comment about the current position holds as for the *orb* directive.

Set Y Coordinate (\$Y)

This directive is analogous to the *set x coordinate* directive, even to the extent of allowing an x coordinate to be specified after the y coordinate.

Set Pen Width (\$Z)

The *set pen width* directive sets the pen width for the *set x coordinate* and *set y coordinate* directives.

Directive	Breaker ¹	Brackets allowed	Default units	Meaning
A	-	no	atoms	Explicit atoms
A	L	no	-	Assign
B	L	no	lines	Blank
C	- ²	no	spaces	Move to Column
D	A	no	-	Define
E	L	no	-	End
F	L	no	-	Font
G	A	no	-	Get file
H	-	yes	steps	Heavy
I	L	yes	tab stops	Indent
J	L	no	-	Justify
L	L	no	lines	Explicit lines
N	L	no	-	New page
O	L	no	pixels	Orb
P	L	no	lines	New paragraph
R	- ³	yes ⁴	font height	Row
T	A	no	tab stops	Tab
U	-	yes	pixels	Underline
V	L	no	lines	Verify
W	-	no	words	Explicit words
X	L	no	pixels	Set x coordinate
Y	L	no	pixels	Set y coordinate
Z	L	no	pixels	Set pen width

1: 'A' for atom-, 'L' for line-breaker

2: Line-breaker if at start of atom

3: Atom-breaker if argument is absolute

4: Brackets not allowed if absolute

Table 1. Summary of *Layout* directives and their properties

```

%a page=10.3" top=0.5" bottom=0.5" line=6.25" left=0.6"
%a nle=2" just=1" page=0.3" page=0.2"
%a tab=0.5" 0.5" 1.1.25" 1.5" 2.5" 3" 3.5" 4" 4.5" 5" 5.5" 6"
%a pager=0.1
%a 12=1" line=0.12
%a 22=1" line=0.12
%a 32=1" line=0.12
%a 724=1" line=0.14
%a 725=1" line=0.16
%a ~-880-0.5
%a esc=88
%a dot=800 %y=0.69" %x=0.3" %x+y %y=0.69" %x0 %x1
%a section=526
%a subsection=901 %x0 %x1 %x2
%a unnumbered=901 %x0 %x1 %x2
%a sdy="5c-0.3"
%a eddy="5c-0.3"
%a pred=532(f=Editor)
%a minus=5c-0.3-
%a num=530-0.6
%a denom=530-1.26-0.5( ) %c-1.26-0.4
%a 12
%a 11a
%a section(Layout)
%a 532(Layout) is a text formatting language
defining (called 532(directive)) which are embedded in
the source text (the 532(manuscript)), and which control
the way that the output (the 532(formatted text)) is produced.
%a 0 This note describes 532(Layout) version 15.5, which is an intermediate
version but the best for which the department has the source code.
The only available document on this version of 532(Layout)
('Layout 15.5 Provisional User's Guide') by Keith Dwyer, Edinburgh University
Computer Science Department, February 1984)
is incomplete and slightly inaccurate, as well as not being widely published.
It is hoped that this note will serve as a manual for 532(Layout) 15.5
and so make it easier to use.
%a 0 A sample 532(Layout) 15.5 manuscript file (the manuscript for this
document) is in the file Text:Layout on the FPHs.
%a subsection(1). Escape Marker)
%a 0 A basic characteristic of 532(Layout) is that the manuscript, in general,
represents the text to be printed unless explicitly flagged otherwise.
The method of flagging text as other than text to be printed (15.08, as
a directive) is to precede it with the 532(escape marker). By default, this
is 'esc', the dollar sign, but if this is an awkward choice it can be
redefined in the document.
%a subsection(2). Atoms, Words and Sentences)
%a 0 The 532(atom) is the basic unit of text recognised by 532(Layout). Atoms
are not split when line filling, and are not stretched when justifying the
line. An atom is defined as a sequence of characters, delimited on both sides
by one of the following:
%a indent(1)
%a 0 A space
%a 0 A non-breaking directive
%a indent(1)
%a 0 A word is similar to an atom, but it consists entirely of letters; that is,
it is terminated also by any character other than a letter. It is relevant
only to the 532(explicit words) directive.
%a 0 A sentence consists of a sequence of atoms starting with a capital letter
and finishing with a full stop, a question mark, or an exclamation mark. The
only relevance of sentences in 532(Layout) is that extra space is inserted
between them, the actual amount of space being user-definable.
%a subsection(3). Spaces and Neolines)
%a 0 Within the scope of an 532(explicit lines) directive, both spaces and
neolines within the text are treated as significant, and are preserved in the
formatted text. Outside any such scope, the filling is enabled and, in general,
spaces and neolines serve only to separate words and are treated equivalently;
multiple spaces, for example, are treated as a single space. Any additional
spacing effects are produced by letting spaces and neolines be significant in
certain situations:
%a indent(1)
%a 0 Blank lines (formed by two consecutive neolines) are preserved in the text;
these blank spaces can be produced simply by leaving blank lines in the manuscript.
%a 0 Spaces at the start of a line of the manuscript are significant, and are
preserved in the formatted text. They also make the preceding neoline

```

```

significant, that is, they force a line break. Thus, a new paragraph can be
started by leaving the required number of blank lines, then starting the
first line of the new paragraph with several spaces.
%a indent(1)
%a 0 Spaces are also treated as significant after explicit positioning directives,
such as moving to a tab stop.
%a subsection(4). Directives)
%a 0 Directives can take several forms, as follows:
%a indent(1)
%a 0 Escape marker, followed by a single letter. This invokes a basic (built
in) directive. The directive may be followed by one or more arguments, depending
on the particular directive. The case of the letter is not significant,
%a 0 Escape marker, followed by a number. This selects previously defined
font. There can be up to 64 fonts, numbered from 0 to 63, of which 0 and 1
are predefined as horizontal and vertical fixed pitch (and rather gross)
fonts.
%a 0 Escape marker, followed by two or more letters. This invokes a previously
defined macro. The case of the letters is again not significant.
%a 0 The invocation is implemented as straight text substitution,
%a 0 thus the macro itself needn't be syntactically valid as long as it becomes
valid in context, and nested macro calls are allowed (up to a maximum of
16 levels).
%a 0 A single non-alphanumeric symbol. This invokes a previously defined
single-character macro (the term applies to the name of the macro, rather
than its body); the invocation is implemented as above.
%a indent(1)
%a 0 Note the possible ambiguity between a built in directive followed
by text, and a macro call. This means that some use of terminating directive
names is required. In the first three cases above, a single space after the
directive will be ignored.
%a subsection(5). Arguments)
%a 0 Some directives take a numeric argument, usually specifying a measurement
but sometimes specifying other items such as fonts. There is a default
unit (such as lines, pixels, tab settings and so on) for the argument, dependent
on the directive; the argument is specified as a multiple of this unit.
%a 0 Where non-integer values are relevant, the multiple can be expressed as a
decimal, or as a fraction with the numerator and denominator separated by
an oblique, '5c-0.35/5c+0.3'; the units may be changed to inches by following the argument
with '5c+0.50"5c+0.5".
%a 0 Some directives allow the argument to be specified relative to some
current value; this is done by preceding the multiple with either a '+' or
a 'minus'.
%a subsection(6). Scoping)
%a 0 532(Layout) has facilities for 532(scoping) of directives, 15.08, for
restricting their effect to a small part of the document. These scopes can
be nested. At the end of the scope, the state returns to that before the start
of the scope; that is, the effect of any directives invoked within the scope
does not persist beyond the end of the scope.
%a 0 Scoping can be done in one of two ways.
%a indent(1)
%a 0 Some directives can be followed by an opening bracket, signifying the
start of the scope.
%a 0 The corresponding closing bracket ends the scope.
%a 0 The closing bracket does not imply an atom break, regardless of whether or not the
directive that caused it does.
%a 0 Changing font, invoking a macro and some built-in directives come into
this category.
%a 0 Four different pairs of brackets are allowed: round brackets '()', square
brackets '[ ]', curly brackets '{ }', and angle brackets '< >'.
%a 0 The directives 532(explicit lines), 532(explicit atoms) and 532(explicit
words) each cause a new scope lasting for the stated number of lines,
%a 0 atoms or words to be produced. This scope is just like any other; thus, the
effect of any directives invoked within the scope doesn't persist beyond the
end of the scope.
%a indent(1)
%a 0 In the absence of any such scoping, the effect of the directive persists
until explicitly stopped.
%a subsection(7). Layout Parameters)
%a 0 Several parameters, such as line length and the tab settings, control
the layout of the formatted text. Each of these has a default value, given
below. Some of the parameters, such as page length and margin sizes, are
normally set no more than once within the document; others, such as the
level of indentation, are likely to change frequently. The parameters can be
changed with the 532(assign) directive. The list of parameters is given below.
%a indent(1)
%a 0 inch=5c+0.48-0.2c"5c+0.4

```

```

select $32(top) (default $sum(2)$denom(3)$inch), the top margin
select $32(bottom) (default $sum($dc+$3)$denom(3)$inch()), the bottom margin
select $32(page) (default $sum(2)$denom(3)$inch()), the height of the
page on which to allow printing. $32(top) is +$ $32(page) is +$ $32(bottom) should always equal the physical he
of the page. $1$sum(2)$denom(3)$inch().
select $32(left) (default 0), the left margin
select $32(right) (default 0), the line length
select $32(nline) (default $sum(1)$denom(8)$inch()), the height of each line
select $32(tab) (default at $sum(2)$denom(8)$inch()) intervals, settings of up to
25 tab stops, numbered 1 to 25. Tab stop 0 is at the left margin and can't be
changed.
select $32(indent) (default 0), the current level of indentation, in tab stops
select $32(egap) (default $sum(1)$denom(6)$inch()), the amount of extra space inserted
between sentences
select $32(pgap) (default $sum(1)$denom(4)$inch()), the amount of indentation at the
start of each paragraph, as defined by the $32(new paragraph) directive
select $32(just) (default 0), whether or not justification is enabled (zero means
disabled)
select $32(pageno) (default 0), the page number. If zero, then page numbering is
disabled. The page number is printed in the middle of the bottom margin.
select $32(section) (default 0), the section number. If zero, or if $32(fengno)
is zero, the section numbering is disabled. If enabled, page and section
numbers are given in the form "1-2", meaning page 2 of section 1.
select $32(letter) (default 1), the first page to be printed
select $32(finish) (default 5555), the last page to be printed.
Together with $32(selector), this provides the capability of printing only a
part of the document.
select $32(escape) (default '\esc('), the escape marker
to indent)
endsection(B, Pageno)
endsection(B, Pageno)

```

```
%d8 Two different sorts of name for user-defined directives are available:
%e indent{1
%def Single-character names, for extreme brevity. The character can be any
normal-alphanumeric character except for the escape marker and brackets.
For example, the definition
%def $+1 spec(d % / = %sec(18sec)(x-10.3
%e8 where font number 8 is smaller than the normal font, will allow the
symbol '%' to be used for subscript; the manuscript text
%def $+1 x/(1
%e8 will produce the formatted text
%def $+1 MSB(9-16.3(1))
%def {floating-mem, consisting of a sequence of two or more letters. The
definition
%def $+1 spec(d dot = %sec(16d %sec)(y-10.05' %sec(x-10.3' %sec(x+y %sec(105 %sec(y-8.05' %sec(x)8 %sec(13
%e8 produces the bullet character at the beginning of each of these
line-ignores.
```

`sgd` When the escape marker is followed by a letter or a number, it denotes a directive. Following it with any other character forces `$2(Laout)` to treat that character as normal text, regardless of any special meaning it may have. Useful cases of this are as follows:

- `$a indent<`
The escape marker followed by a space causes the space to be treated as a normal character. This can be used to prevent a line break between two words when combined with a normal (unescaped) space.
- `it provides extra spacing between words.`
- `$dot The escape marker followed by a full stop (or a question or exclamation mark)` prevents the character from being recognised as a sentence terminator, for example when it signifies an abbreviation, a character defined as the name of a dot preceding a valid closing bracket, a character defined as the name of a directive or the escape marker itself with the escape marker causes that character to be included in the text as a normal character.
- `$a indent<`

subsection(18. Description of Built-In Directives)

```

960 There follows a description of each of the built-in directives provided
961 by S32(LAYOUT). A table is given afterwards, summarising the directives and
962 the ir version properties.
963 Summarised(Header) (S32(1))
964 The S32(LAYOUT) directive is used to change the values of S32(LAYOUT)
965 parameters. For example,
966 S32(5+1 S32(1) Just = 1
967 S32(Several assignments can be made at once, separated by semicolons:

```

So! The default unit for the parameter `$SZ(left)`, `$SZ(right)`, `$SZ(upper)` and `$SZ(sub)` is the width of a space in the current font for the parameter `$SZ(top)`, `$SZ(bottom)`, `$SZ(page)` and `$SZ(mis)` it is the line height

24, with 5 point Roman, 4 and 5 point Bold and 5 point Scientific fonts also available.

\$g0 The second way is to define in terms of a vector font. There are 31 Glims fonts and 5 Times fonts available by default they are read in, but they can be left out if they are not needed. The number of the desired font is given first. The font can be adjusted by following the definition with one of the following:

\$a indent{1} followed by a number, gives the scale factor (magnification). 255 gives no scaling, and other numbers scale accordingly.

\$dot 'y', followed by a size, specifies the ascender height for the font.

\$dot 'd', followed by a size, gives a descender height for the font.

\$dot 'x', followed by a size, makes the font fixed pitch by specifying a width for all the characters.

\$a indent{}

The last three use inches as the default unit.

Summbered(Get File (Spec(16))

\$g0 This reads in a file, for example one of the files containing Times font definitions. It can also be used to read in and display a **\$GZ(pseudo display file)**, which has been created with a graphics editor like **\$GZ(Edwin Draw)**.

Summbered(heavy (Spec(14))

\$g0 This directive switches on and off 'boldfacing' of text.

This is implemented by printing characters several times, each time displaced by one pixel. The default amount is two extra pixels, but any number between 1 and 7 can be used.

The directive can be followed by a bracket, restricting its scope.

Summbered(indent (Spec(11))

\$g0 The **\$GZ(indent)** directive is an alternative to the **\$GZ(Layout)** parameter **\$GZ(indent)**. If the parameter has been used in the document, then the directive acts like the **\$GZ(tab)** directive, but the argument is given relative to the current level of indentation (as opposed to the nearest tab stop). If the parameter has not been used, then the directive sets the level of indentation.

Summbered(Justify (Spec(13))

\$g0 This directive causes a line break, like the **'Spec(18)'** directive, but also forces justification of the completed line.

Summbered(Explicit Lines (Spec(12))

\$g0 The **\$GZ(explicit lines)** directive is used to set up a scope that lasts for the specified number of lines. If the argument is zero, then the scope will last for an indefinite number of lines; it is terminated by end line-breaking directive. Indentation is normally switched off for the extent of the scope.

\$g0 **\$GZ(casign)** directive, calls to macros and font changes can be placed on the line after the **\$GZ(explicit lines)** directive; they will be executed at the start of the scope, and their effect restricted to within it. There can also be several other parameters on the line. These are:

\$a indent{1}

\$dot 'y' to center the lines

\$dot 'y' to indent the lines

\$dot 'y' or **'g'** to make the lines bold; these can be followed by a number specifying the degree of boldness, as for the **\$GZ(heavy)** directive

\$dot 'u' to underline the lines; this can be followed by a number giving the depth of the underline, as for the **\$GZ(underline)** directive

\$a indent{}

Summbered(New Page (Spec(19))

\$g0 This directive closes the current page and starts a new one.

Summbered(Orb (Spec(10))

\$g0 This draws a solid circle of the specified radius (in pixels) at the current position. Note that because the directive is a line-breaker, if there are no other positioning commands before it then the current position will be at the start of the line.

Summbered(New Paragraph (Spec(17))

\$g0 The **\$GZ(new paragraph)** takes as its argument a number of lines. If the current position is below the top of the page, then this number of lines is left. If there are less than two lines of space left on the page, a new page is taken. The following text is then indented by the **\$GZ(layout)** parameter **\$GZ(marg)**.

Summbered(Row (Spec(18))

\$g0 This directive behaves in two different ways, depending on whether its argument is absolute or relative. The default unit is the height of the current font.

\$g0 If the argument is relative, then the directive causes the vertical displacement of text on the line to be adjusted. It can take brackets to delimit its scope. In this case it doesn't cause an atom break, so it can be

used within an atom.

\$g0 If the argument is absolute, the directive causes a move to that position down the page, and breaks the atom. It can't be followed by brackets.

Summbered(Tab (Spec(17))

\$g0 The **\$GZ(tab)** directive causes a move to the specified tab stop. The tab stop can be given relative to the current position, in which case it causes a move by the specified number of tab positions.

Summbered(underline (Spec(11))

\$g0 This directive switches underlining on and off. Its scope can be delimited with brackets. The default distance of the underline below the base line of the text is 5 pixels, but any number from 1 up can be specified with the directive.

Summbered(Verify (Spec(15))

\$g0 The **\$GZ(verify)** directive is used to check that a certain amount of room is left on the page. If there is not, then a new page is taken. The directive can be used to ensure that a chapter of the document is kept on one page; it can also be used to check that a new section, say, is not started too close to the foot of the page.

Summbered(Explicit Words (Spec(14))

\$g0 This directive is similar in effect to the **\$GZ(explicit atoms)** directive, but using words rather than atoms.

Summbered(Set X Coordinate (Spec(16))

\$g0 This directive moves to a specified position across the page. It can be followed by a comma and another position, which is treated as a **\$GZ(y)** coordinate to move to as well. The positions can be given relative to the current position. If the pen width (see below) is not zero, then a line is drawn from the current position to the new position.

\$g0 The same comment about the current position holds as for the **\$GZ(orb)** directive.

Summbered(Set Y Coordinate (Spec(17))

\$g0 This directive is analogous to the **\$GZ(set x coordinate)** directive, even to the extent of allowing an **\$GZ(x)** coordinate to be specified after the **\$GZ(y)** coordinate.

Summbered(Set Pen Width (Spec(12))

\$g0 The **\$GZ(set pen width)** directive sets the pen width for the **\$GZ(set x coordinate)** and **\$GZ(set y coordinate)** directives.

\$12 Jerry Gibbons

\$12 July 1987

\$GZ

\$a line{1}

\$10a

Directive

a

b

c

d

e

f

g

h

i

j

k

l

m

n

o

p

q

r

s

t

u

v

w

x

y

z

\$g-25

\$a left{1}

\$10a

\$-18.3(\$-2/388 (Breaker(1)))

```

1  L
2  L
3  L
4  L
5  L
6  L
7  L
8  L
9  L
10 L
11 L
12 L
13 L
14 L
15 L
16 L
17 L
18 L
19 L
20 L
21 L
22 L
23 L
24 L
25 L
26 L
27 L
28 L
29 L
30 L
31 L
32 L
33 L
34 L
35 L
36 L
37 L
38 L
39 L
40 L
41 L
42 L
43 L
44 L
45 L
46 L
47 L
48 L
49 L
50 L
51 L
52 L
53 L
54 L
55 L
56 L
57 L
58 L
59 L
60 L
61 L
62 L
63 L
64 L
65 L
66 L
67 L
68 L
69 L
70 L
71 L
72 L
73 L
74 L
75 L
76 L
77 L
78 L
79 L
80 L
81 L
82 L
83 L
84 L
85 L
86 L
87 L
88 L
89 L
90 L
91 L
92 L
93 L
94 L
95 L
96 L
97 L
98 L
99 L
100 L
101 L
102 L
103 L
104 L
105 L
106 L
107 L
108 L
109 L
110 L
111 L
112 L
113 L
114 L
115 L
116 L
117 L
118 L
119 L
120 L
121 L
122 L
123 L
124 L
125 L
126 L
127 L
128 L
129 L
130 L
131 L
132 L
133 L
134 L
135 L
136 L
137 L
138 L
139 L
140 L
141 L
142 L
143 L
144 L
145 L
146 L
147 L
148 L
149 L
150 L
151 L
152 L
153 L
154 L
155 L
156 L
157 L
158 L
159 L
160 L
161 L
162 L
163 L
164 L
165 L
166 L
167 L
168 L
169 L
170 L
171 L
172 L
173 L
174 L
175 L
176 L
177 L
178 L
179 L
180 L
181 L
182 L
183 L
184 L
185 L
186 L
187 L
188 L
189 L
190 L
191 L
192 L
193 L
194 L
195 L
196 L
197 L
198 L
199 L
200 L
201 L
202 L
203 L
204 L
205 L
206 L
207 L
208 L
209 L
210 L
211 L
212 L
213 L
214 L
215 L
216 L
217 L
218 L
219 L
220 L
221 L
222 L
223 L
224 L
225 L
226 L
227 L
228 L
229 L
230 L
231 L
232 L
233 L
234 L
235 L
236 L
237 L
238 L
239 L
240 L
241 L
242 L
243 L
244 L
245 L
246 L
247 L
248 L
249 L
250 L
251 L
252 L
253 L
254 L
255 L
256 L
257 L
258 L
259 L
260 L
261 L
262 L
263 L
264 L
265 L
266 L
267 L
268 L
269 L
270 L
271 L
272 L
273 L
274 L
275 L
276 L
277 L
278 L
279 L
280 L
281 L
282 L
283 L
284 L
285 L
286 L
287 L
288 L
289 L
290 L
291 L
292 L
293 L
294 L
295 L
296 L
297 L
298 L
299 L
300 L
301 L
302 L
303 L
304 L
305 L
306 L
307 L
308 L
309 L
310 L
311 L
312 L
313 L
314 L
315 L
316 L
317 L
318 L
319 L
320 L
321 L
322 L
323 L
324 L
325 L
326 L
327 L
328 L
329 L
330 L
331 L
332 L
333 L
334 L
335 L
336 L
337 L
338 L
339 L
340 L
341 L
342 L
343 L
344 L
345 L
346 L
347 L
348 L
349 L
350 L
351 L
352 L
353 L
354 L
355 L
356 L
357 L
358 L
359 L
360 L
361 L
362 L
363 L
364 L
365 L
366 L
367 L
368 L
369 L
370 L
371 L
372 L
373 L
374 L
375 L
376 L
377 L
378 L
379 L
380 L
381 L
382 L
383 L
384 L
385 L
386 L
387 L
388 L
389 L
390 L
391 L
392 L
393 L
394 L
395 L
396 L
397 L
398 L
399 L
400 L
401 L
402 L
403 L
404 L
405 L
406 L
407 L
408 L
409 L
410 L
411 L
412 L
413 L
414 L
415 L
416 L
417 L
418 L
419 L
420 L
421 L
422 L
423 L
424 L
425 L
426 L
427 L
428 L
429 L
430 L
431 L
432 L
433 L
434 L
435 L
436 L
437 L
438 L
439 L
440 L
441 L
442 L
443 L
444 L
445 L
446 L
447 L
448 L
449 L
450 L
451 L
452 L
453 L
454 L
455 L
456 L
457 L
458 L
459 L
460 L
461 L
462 L
463 L
464 L
465 L
466 L
467 L
468 L
469 L
470 L
471 L
472 L
473 L
474 L
475 L
476 L
477 L
478 L
479 L
480 L
481 L
482 L
483 L
484 L
485 L
486 L
487 L
488 L
489 L
490 L
491 L
492 L
493 L
494 L
495 L
496 L
497 L
498 L
499 L
500 L
501 L
502 L
503 L
504 L
505 L
506 L
507 L
508 L
509 L
510 L
511 L
512 L
513 L
514 L
515 L
516 L
517 L
518 L
519 L
520 L
521 L
522 L
523 L
524 L
525 L
526 L
527 L
528 L
529 L
530 L
531 L
532 L
533 L
534 L
535 L
536 L
537 L
538 L
539 L
540 L
541 L
542 L
543 L
544 L
545 L
546 L
547 L
548 L
549 L
550 L
551 L
552 L
553 L
554 L
555 L
556 L
557 L
558 L
559 L
560 L
561 L
562 L
563 L
564 L
565 L
566 L
567 L
568 L
569 L
570 L
571 L
572 L
573 L
574 L
575 L
576 L
577 L
578 L
579 L
580 L
581 L
582 L
583 L
584 L
585 L
586 L
587 L
588 L
589 L
590 L
591 L
592 L
593 L
594 L
595 L
596 L
597 L
598 L
599 L
600 L
601 L
602 L
603 L
604 L
605 L
606 L
607 L
608 L
609 L
610 L
611 L
612 L
613 L
614 L
615 L
616 L
617 L
618 L
619 L
620 L
621 L
622 L
623 L
624 L
625 L
626 L
627 L
628 L
629 L
630 L
631 L
632 L
633 L
634 L
635 L
636 L
637 L
638 L
639 L
640 L
641 L
642 L
643 L
644 L
645 L
646 L
647 L
648 L
649 L
650 L
651 L
652 L
653 L
654 L
655 L
656 L
657 L
658 L
659 L
660 L
661 L
662 L
663 L
664 L
665 L
666 L
667 L
668 L
669 L
670 L
671 L
672 L
673 L
674 L
675 L
676 L
677 L
678 L
679 L
680 L
681 L
682 L
683 L
684 L
685 L
686 L
687 L
688 L
689 L
690 L
691 L
692 L
693 L
694 L
695 L
696 L
697 L
698 L
699 L
700 L
701 L
702 L
703 L
704 L
705 L
706 L
707 L
708 L
709 L
710 L
711 L
712 L
713 L
714 L
715 L
716 L
717 L
718 L
719 L
720 L
721 L
722 L
723 L
724 L
725 L
726 L
727 L
728 L
729 L
730 L
731 L
732 L
733 L
734 L
735 L
736 L
737 L
738 L
739 L
740 L
741 L
742 L
743 L
744 L
745 L
746 L
747 L
748 L
749 L
750 L
751 L
752 L
753 L
754 L
755 L
756 L
757 L
758 L
759 L
760 L
761 L
762 L
763 L
764 L
765 L
766 L
767 L
768 L
769 L
770 L
771 L
772 L
773 L
774 L
775 L
776 L
777 L
778 L
779 L
780 L
781 L
782 L
783 L
784 L
785 L
786 L
787 L
788 L
789 L
790 L
791 L
792 L
793 L
794 L
795 L
796 L
797 L
798 L
799 L
800 L
801 L
802 L
803 L
804 L
805 L
806 L
807 L
808 L
809 L
810 L
811 L
812 L
813 L
814 L
815 L
816 L
817 L
818 L
819 L
820 L
821 L
822 L
823 L
824 L
825 L
826 L
827 L
828 L
829 L
830 L
831 L
832 L
833 L
834 L
835 L
836 L
837 L
838 L
839 L
840 L
841 L
842 L
843 L
844 L
845 L
846 L
847 L
848 L
849 L
850 L
851 L
852 L
853 L
854 L
855 L
856 L
857 L
858 L
859 L
860 L
861 L
862 L
863 L
864 L
865 L
866 L
867 L
868 L
869 L
870 L
871 L
872 L
873 L
874 L
875 L
876 L
877 L
878 L
879 L
880 L
881 L
882 L
883 L
884 L
885 L
886 L
887 L
888 L
889 L
890 L
891 L
892 L
893 L
894 L
895 L
896 L
897 L
898 L
899 L
900 L
901 L
902 L
903 L
904 L
905 L
906 L
907 L
908 L
909 L
910 L
911 L
912 L
913 L
914 L
915 L
916 L
917 L
918 L
919 L
920 L
921 L
922 L
923 L
924 L
925 L
926 L
927 L
928 L
929 L
930 L
931 L
932 L
933 L
934 L
935 L
936 L
937 L
938 L
939 L
940 L
941 L
942 L
943 L
944 L
945 L
946 L
947 L
948 L
949 L
950 L
951 L
952 L
953 L
954 L
955 L
956 L
957 L
958 L
959 L
960 L
961 L
962 L
963 L
964 L
965 L
966 L
967 L
968 L
969 L
970 L
971 L
972 L
973 L
974 L
975 L
976 L
977 L
978 L
979 L
980 L
981 L
982 L
983 L
984 L
985 L
986 L
987 L
988 L
989 L
990 L
991 L
992 L
993 L
994 L
995 L
996 L
997 L
998 L
999 L
1000 L

```

Meaning

Explicit atoms
Align
Blank
Move to Column
Def line
End
Font
Get file
Heavy
Indent
Justify
Explicit lines
New page
Orb
New paragraph
Row
Tab
Underline
Verify
Explicit words
Set x coordinate
Set y coordinate
Set pen width
See left; line>

515

514 1: 'q' for atom, 'L' for line-breaker
514 2: Line-breaker if at start of atom
514 3: Line-breaker if argument is absolute
514 4: Brackets not allowed if absolute
515a

Table 1. Summary of SEZ(Layout) directives and their properties

515
515a
515b
515c
515d
515e
515f
515g
515h
515i
515j
515k
515l
515m
515n
515o
515p
515q
515r
515s
515t
515u
515v
515w
515x
515y
515z
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

Font TimesR5

!"#\$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`
'abcdefghijklmnopqrstuvwxyz{|}~

Font TimesR6

!"#\$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`
'abcdefghijklmnopqrstuvwxyz{|}~

Font TimesR7

!"#\$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`
'abcdefghijklmnopqrstuvwxyz{|}~

Font TimesR8

!"#\$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`
'abcdefghijklmnopqrstuvwxyz{|}~

Font TimesR9

!"#\$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`
'abcdefghijklmnopqrstuvwxyz{|}~

Font TimesR10

!"#\$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`
'abcdefghijklmnopqrstuvwxyz{|}~

Font TimesR12

!"#\$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`
'abcdefghijklmnopqrstuvwxyz{|}~

Font TimesR13

!"#\$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`
'abcdefghijklmnopqrstuvwxyz{|}~

Font TimesR14

!"#\$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`
'abcdefghijklmnopqrstuvwxyz{|}~

Font TimesR16

!"#\$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`
'abcdefghijklmnopqrstuvwxyz{|}~

Font TimesR18

!"#\$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`
'abcdefghijklmnopqrstuvwxyz{|}~

Font TimesR24

!"#\$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`
'abcdefghijklmnopqrstuvwxyz{|}~

Font TimesI6

! \$%&'()* ,-./0123456789:; ?
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]
'abcdefghijklmnopqrstuvwxyz

Font TimesI7

! \$%&'()* ,-./0123456789:; ?
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]
'abcdefghijklmnopqrstuvwxyz

Font TimesI8

! \$%&'()* ,-./0123456789:; ?
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]
'abcdefghijklmnopqrstuvwxyz

Font TimesI9

! \$%&'()* ,-./0123456789:; ?
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]
'abcdefghijklmnopqrstuvwxyz

Font TimesI10

! \$%&'()* ,-./0123456789:; ?
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]
'abcdefghijklmnopqrstuvwxyz

Font TimesI12

! \$%&'()* ,-./0123456789:; ?
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]
'abcdefghijklmnopqrstuvwxyz

Font TimesI13

! \$%&'()* ,-./0123456789:; ?
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]
'abcdefghijklmnopqrstuvwxyz

Font TimesI14

! \$%&'()* ,-./0123456789:; ?
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]
'abcdefghijklmnopqrstuvwxyz

Font TimesI16

! \$%&'()* ,-./0123456789:; ?
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]
'abcdefghijklmnopqrstuvwxyz

Font TimesI18

! \$%&'()* ,-./0123456789:; ?
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]
'abcdefghijklmnopqrstuvwxyz

Font TimesI24

! \$%&'()* ,-./0123456789:; ?
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]
'abcdefghijklmnopqrstuvwxyz

Font TimesB4

! \$%&'()* ,-. /0123456789:; ?
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]
'abcdefghijklmnopqrstuvwxyz

Font TimesB5

! \$%&'()* ,-. /0123456789:; ?
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]
'abcdefghijklmnopqrstuvwxyz

Font TimesB6

! \$%&'()* ,-. /0123456789:; ?
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]
'abcdefghijklmnopqrstuvwxyz

Font TimesB7

! \$%&'()* ,-. /0123456789:; ?
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]
'abcdefghijklmnopqrstuvwxyz

Font TimesB8

! \$%&'()* ,-. /0123456789:; ?
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]
'abcdefghijklmnopqrstuvwxyz

Font TimesB9

! \$%&'()* ,-. /0123456789:; ?
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]
'abcdefghijklmnopqrstuvwxyz

Font TimesB10

! \$%&'()* ,-. /0123456789:; ?
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]
'abcdefghijklmnopqrstuvwxyz

Font TimesB12

! \$%&'()* ,-. /0123456789:; ?
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]
'abcdefghijklmnopqrstuvwxyz

Font TimesB13

! \$%&'()* ,-. /0123456789:; ?
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]
'abcdefghijklmnopqrstuvwxyz

Font TimesB14

! \$%&'()* ,-. /0123456789:; ?
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]
'abcdefghijklmnopqrstuvwxyz

Font TimesB16

! \$%&'()* ,-. /0123456789:; ?
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]
'abcdefghijklmnopqrstuvwxyz

Font TimesB18

! \$%&'()* ,-. /0123456789:; ?
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]
'abcdefghijklmnopqrstuvwxyz

Font TimesB24

! \$%&'()* ,-. /0123456789:; ?
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]
'abcdefghijklmnopqrstuvwxyz

Gimms font 26

abcdefghijklmnopqrstuvwxyz{|}~

Gimms font 32

!"#\$%&'()*+,-./0123456789:;<=>?
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz

Gimms font 33

!"#\$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNPQRSTUVWXYZ[]~
abcdefghijklmnopqrstuvwxyz{|}~

Gimms font 41

!"#\$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNPQRSTUVWXYZ[]~
abcdefghijklmnopqrstuvwxyz{|}~

Gimms font 42

!"#\$%&'()*+,-./0123456789:;<=>?
ABHΔEΦΓXIJKΛMNOPΘPΣΤΤVΩΞΨΖ
|

Gimms font 43

!"#\$%&'()*+,-./0123456789:;<=>?
@ABHΔEΦΓXIJKΛMNOPΘPΣΤΤVΩΞΨΖ[]~
αβγδεφγχικλμνοπθρστυφωξψζ{|}~

Gimms font 44

!"#\$%&'()*+,-./0123456789:;<=>?
@ABHΔEΦΓXIJKΛMNOPΘPΣΤΤVΩΞΨΖ[]~
αβγδεφγχικλμνοπθρστυφωξψζ{|}~

Gimms font 48

!"#\$%&'()*+,-./0123456789:;<=>?
@ABHΔEΦΓXIJKΛMNOPΘPΣΤΤVΩΞΨΖ[]~
αβγδεφγχικλμνοπθρστυφωξψζ{|}~

Gimms font 51

!"#еъяцыЕЪЯЫЦЫ0123456789:;<=>?
@АБЭДЙФГЖИЧКЛМНОПШРСТЮВЩХУЗ[]~
абэдйфгжичклмнопшрстювщхуз{|}~

Gimms font 52

!"#\$%&'()*+,-./0123456789:;<=>?
uBcDeFgHijKlMnOpQrStUvWxYz
abcdefghijklmnopqrstuvwxyz

Gimms font 53

!"#\$%&'()*+,-./0123456789:;<=>?
ABcDeFgHijKlMnOpQrStUvWxYz
abcdefghijklmnopqrstuvwxyz

Gimms font 60

!"#\$%&'()*+,-./0123456789:;<=>?
HBCDEFGHIJKLMPQNRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz

Gimms font 61

!"#\$%&'()*+,-./0123456789:;<=>?
ABCDEFGHIJKLMNPQRSTUVWXYZ[\]~
`abcdefghijklmnopqrstuvwxyz{|}~

Gimms font 62

!"#\$%&'()*+,-./0123456789:;<=>?
ABCDEFGHIJKLMNPQRSTUVWXYZ \ _
`abcdefghijklmnopqrstuvwxyz{|}~

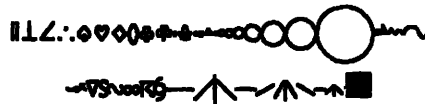
Gimms font 63

!"#\$%&'()*+,-./0123456789:;<=>?
ABCDEFGHIJKLMNPQRSTUVWXYZ \ _
`abcdefghijklmnopqrstuvwxyz{|}~

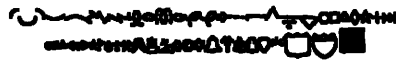
Gimms font 64

!"#\$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNPQRSTUVWXYZ[\]~
`abcdefghijklmnopqrstuvwxyz{|}~

Gimms font 2



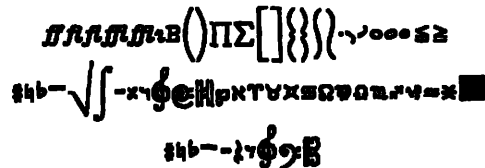
Gimmis font 3



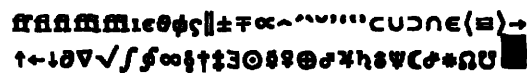
Gimmms font 4



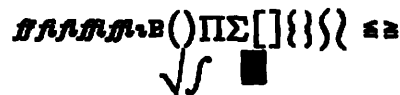
Gimms font 5



Gimmms font 6



Gimme font ?



Gimmms font 11



Gimms font 12



Gimmms font 13



Gimms font 14



Gimmys font 15



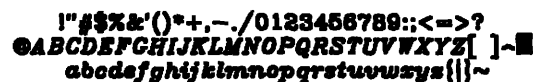
Gimms font 16



Gimme font 17



Gimmys font 23



Gimms font 24

