The layouts package: User manual

Peter R. Wilson
Catholic University of America
(Now at: peter.r.wilson@boeing.com)

July 2002*

Abstract

The LaTeX layouts package enables the display of various elements of a document's layout. The elements include the positioning of text on a page, the disposition of floats on a page, the geometrical layout of lists and footnotes, the design of section headers and their display in a table of contents. It also enables document designers to experiment with potential layout designs.

Contents

1 Introduction **1**
1.1 Command types . . . 1
1.2 General facilities . . . 2
1.2.1 Scaling . . . 2
1.2.2 Layout types . . . 2
1.2.3 Length units . . . 3
1.3 Usage . . . 4

2 Page spread **4**

3 Page layout **7**
3.1 Changing the page layout in your document . . . 13

4 The memoir class page layout **14**

*Previous versions were dated March 2002, November 2001, July 2001 and January 1999.

5 Paragraph layout **18**

6 Float layouts **21**
6.1 Float and text page layout 21
6.2 Detailed float layout 24
6.3 Changing the float layout in your document 27

7 List layout **29**
7.1 Changing lists 34

8 Sectional heading layout **35**

9 Footnote layout **39**
9.1 Changing the footnote layout 41

10 Table of Contents layout **43**
10.1 Changing the Table of Contents, etc 45

11 Font boxing **47**

List of Tables

1 Page spread symbols 5
2 Commands for setting trial page parameters 11
3 Commands for setting trial memoir page parameters 16
4 Commands for setting trial paragraph parameters 18
5 Commands for setting trial float page parameters 22
6 Commands for setting trial float parameters 25
7 Commands for setting trial list parameters 31
8 Commands for setting trial heading parameters 36
9 Commands for setting trial footnote parameter values 40
10 Commands for setting trial values for ToC parameters 44

List of Figures

1 (Left) The LaTeX book spread; (Right) Spread for many of Gutenberg's books 5
2 Left-hand two-column page layout parameters 8
3 Page layout for this document 10
4 An experimental page layout 12
5 Right-hand page major layout parameters for the memoir class 15
6 An experimental memoir class page layout 17
7 Paragraph parameters 18
8 Paragraphs in this document 19
9 An outset paragraph 19
10 Float and text page parameters 22
11 The standard LaTeX float and text page settings 23
12 Float page layout for decreasing likelihood of float-only pages 23
13 Float parameters 25
14 Float layout with rules 26
15 List parameters 30
16 Layout of an `enumerate` list 31
17 The layout of the `listX` environment 33
18 Display heading parameters 36
19 Run-in heading parameters 36
20 Subsubsection heading layout parameters for this document 37
21 Subsubsection layout parameters for a run-in heading 38
22 The footnote parameter layout 39
23 A footnote layout 40
24 A user-specified footnote layout 41
25 Table of Contents entry parameters 43
26 Typical Table of Contents entry for this document 44
27 Table of Contents entry with a large value for `\@dotsep` 45

1 Introduction

LaTeX has several pre-defined styles for the layout of typeset documents [Lam94]. Authors using LaTeX sometimes wish to understand how these layouts are parameterised (or controlled). The `layouts` package enables the display of certain of these parameterised layouts, showing what the parameters control. It also provides facilities for experimenting with different values of the typesetting parameters, showing the results graphically.

The layouts package was developed as an aid to the author when he was developing a new LaTeX class for typsetting ISO standards. The development of the package has benefitted from Kent McPherson's layout.sty [McP88].

1.1 Command types

The majority of the commands provided fall into one of several categories.

Drawing Commands of the form `\draw`*L*, `\`*L*`diagram`, and `\`*L*`design` generate a LaTeX `picture` of the layout *L*.

Current layout Commands of the form `\current`*L* collect, as far as possible, the values of the layout parameters for the current *L* layout environment for use by the corresponding `\draw`*L* command. If a parameter value is not directly accessible, then a typical value is provided.

Layout values Commands of the form `\`*L*`values` produce a tabular listing of the current values of the parameters controlling the layout *L*. As far as possible these are the actual values. In some cases the values are hard-wired in the body of a macro, and in these cases reasonable guesstimates are made of a 'typical' value. In the printed table, guesstimates are indicated by appending two question marks (??) to the printed value. Note that the tabular listing produced by a `\`*L*`values` command is not the same as a listing that might be produced by the corresponding `\draw`*L* command. The appearance of the table may be controlled as follows.

- The declaration `\printheadingstrue` will cause a short title to be placed above the tabular; the tabular is untitled when `\printheadingsfalse` is in effect.
- The font size used for the table is specified via the `\setvaluestextsize`{⟨*fontsize*⟩} macro, where ⟨*fontsize*⟩ is one of LaTeX's font size commands (e.g., `\footnotesize`, which is the default value.).

User-specified parameter values Commands of the form `\try`*P* can be used to change the value of the layout parameter *P*. Note that these commands do *not* affect the the layout of the document, they only modify the values used in displaying a layout.

Control Commands of the form `\`*C*`true` or `\`*C*`false` control certain aspects of the kinds of layouts pictured.

1.2 General facilities

In order to initialize the layouts package correctly for your document, the command `\setuplayouts` must be issued immediately after the `\begin{document}`, or at least before any following command that changes font sizes or suchlike. Fortunately you don't have to concern youself with this as it is called automatically by the layouts package. However, you may call it yourself at some later point in the document to pick up the current value of the `\baselineskip` or `\parskip` if these have been modified. The `\setuplayouts` command also sets the `\setlayoutscale` to its default value of 0.5.

1.2.1 Scaling

A few layout pictures are drawn full size. Most are drawn less than full size, typically to half-scale. To make the pictures fit your document neatly, you may wish to change the scale. The command `\setlayoutscale{`⟨*frac*⟩`}`, where ⟨*frac*⟩ is a decimal number (e.g., 0.75) sets the scaling. For example, `\setlayoutscale{0.5}` will produce pictures at half full size, and `\setlayoutscale{1.0}` will make the pictures full size.

1.2.2 Layout types

In general, two kinds of layout pictures can be produced.

1. A picture showing a generic layout and its controlling parameters is produced by a command of the form `\`*L*`diagram`. The same effect may be achieved by using a `\draw`*L* after the declaration `\drawparameterstrue`.
2. A picture showing an 'actual' layout is produced by a command of the form `\`*L*`design`. The same effect may be achieved by using a `\draw`*L* after the declaration `\drawparametersfalse`.

> **Warning:** If an actual layout differs too much from a 'normal' layout then LaTeX may hiccup a time or two. The typical problem is when a dimension that is expected to be positive is actually negative; LaTeX will respond with an error message like 'Bad `\line` or `\vector` argument'.

Note: The `\`*L*`diagram` and `\`*L*`design` commands were introduced in version 2.5 of the package. The previous method of generating pictures (via `\draw`*L* and `\drawparameters...`) is mainly retained for compatibility reasons.

Examples of both kinds of layout are shown later.

Some aspects of the appearance of the pictures can be controlled. The following apply to both generic and actual layouts.

- The size of the font used for displaying the parameters is specified via the command: `\setparameterstextsize`{⟨*size*⟩}, where ⟨*size*⟩ is one of LaTeX's font size commands. The initial setting is `\setparameterstextsize{\footnotesize}`.
- Some parts of the layouts are labelled, for example the footer area in the page layout is labelled 'footer'. The macro `\setlabelfont`{⟨*font*⟩} can be used to control the font used for typesetting these labels. The default is `\setlabelfont{\normalsize}`, but can be, for example, `\setlabelfont{\large\itshape}` to use a large italic font instead. Note that ⟨*font*⟩ must be specified as a declaration(s).

The following controls only apply to actual layouts (i.e., when using `\`*L*`design`, or the earlier `\drawparametersfalse` `\draw`*L* pairing is in effect).

- Some layouts have some explanatory header text. Putting `\printheadingsfalse` prohibits these while `\printheadingstrue` lets them be printed.
- In the generic layouts, the lengths controlled by the various parameters are indicated by dimension lines; these are not normally shown in actual layouts. Using `\drawdimensionstrue` will let these be shown in actual layouts. The default setting is `\drawdimensionsfalse`.

  > **Warning:** If you draw actual layouts with `\drawdimensionstrue` the drawings may be overburdened and if the layout differs too much from a 'normal' layout then LaTeX may hiccup a time or two (typically with an error message like 'Bad `\line` or `\vector` argument').

- When `\printparameterstrue` is in effect, a table of the parameter values will be printed below the layout. Unsurprisingly, `\printparametersfalse` prohibits the table. The default is `\printparameterstrue`.

1.2.3 Length units

The tabular listings of the parameter length values produced by the `\`*L*`values` macros give the lengths in TeX points by default. The length units may be set by the `\printinunitsof`{⟨*unit*⟩} command. The ⟨*unit*⟩ argument may be any of TeX's length units, except for scaled point. That is, the allowable values are `pt`, `pc`, `in`, `mm`, `cm`, `bp`, `dd`, or `cc`; an unkown unit is treated as though it were `pt`. When printing in pt units any stretch or shrink values are printed as well; using `PT` instead of `pt` eliminates these 'rubber' values. The initial setting is `\printinunitsof{pt}`.

The macro `\prntlen`{⟨*length*⟩} prints the value of ⟨*length*⟩, where ⟨*length*⟩ is a LaTeX length, using the units specified by `\printinunitsof`. For example, the following code

```
The textwidth is \printinunitsof{pt}\prntlen{\textwidth} which is also
```

```
\printinunitsof{in}\prntlen{\textwidth} or
\printinunitsof{mm}\prntlen{\textwidth}.
```

prints out:
The textwidth is 433.62pt which is also 6.00117in or 152.39821mm.

1.3 Usage

The drawing commands (i.e., `\...diagram`, `\...design` and `\draw...`) are typically used within a `figure` environment, although they can be used in running text. Internally they generate a `picture` and `tabular` environment, both enclosed in their own `center` environments.

The `\setlayoutscale` command can be used anywhere in a document after the initial `\setuplayouts` command. Used in running text it will alter the scale factor for all succeeding drawing commands. However, if `\setlayoutscale` is called within an environment, such as the `figure` environment, it will only affect the scale factors for succeeding drawing commands in that environment.

The layouts package uses many LaTeX counters and lengths. If it is used with too many other packages, the available number of counters and lengths may be exhausted. Essentially, the layouts package is intended to be used in short documents as an exploratory tool when designing the typesetting rules to be embodied in a new package or class file.

2 Page spread

Book designers often start by determining the proportions of a page, the proportions of the text block, and the position of the text block on the page. Often the width of a page is taken as the starting point.

In the following let w be the width of a single page; that is, the distance from the spine to the outside of the fore edge. I also use the other symbols given in Table 1.

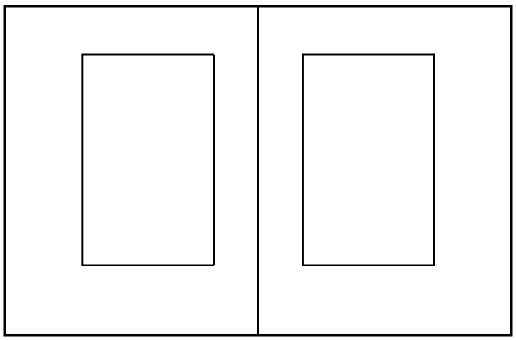
Given values for w, P, T, S, and any two of H, E and F, it is possible to calculate the page size, the text block size and the position of the text. The command
`\drawaspread[`$\langle F\rangle$`]{`$\langle w\rangle$`}{`$\langle P\rangle$`}{`$\langle T\rangle$`}{`$\langle S\rangle$`}{`$\langle H\rangle$`}{`$\langle E\rangle$`}{`$\langle G\rangle$`}`
where the first parameter is optional, draws a two page spread with the given page width and proportions. If the optional $\langle F\rangle$ is not provided, then the bottom margin is calculated from the values of other parameters. If $\langle G\rangle$ is zero or negative, then a single column layout is shown, otherwise a double column layout is presented.

Figure 1 shows two different spreads. This was produced from the code below:

```
\newlength{\pwlayi}
```

Table 1: Page spread symbols

Symbol	Meaning
w	Page width
P	Ratio of page height to width
T	Ratio of text block height to width
S	Ratio of spine width to page width
H	Ratio of top margin height to spine width
E	Ratio of fore edge margin width to spine width
F	Ratio of bottom margin height to spine width
G	Ratio of width of two-column gutter to spine width



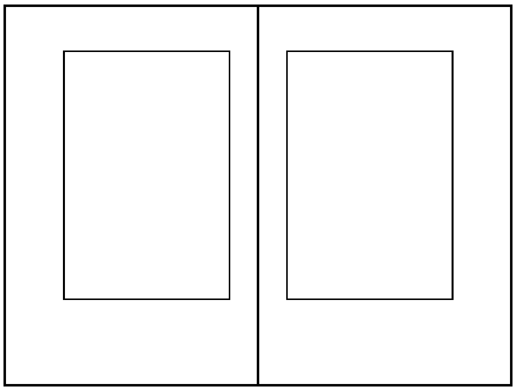


Figure 1: (Left) The LaTeX book spread; (Right) Spread for many of Gutenberg's books

```
\setlength{\pwlayi}{0.4375\textwidth}
\newlength{\pwlayii}
\setlength{\pwlayii}{0.5\pwlayi}

\begin{figure}
\centering
\begin{minipage}[b]{\pwlayi}
\drawaspread{\pwlayii}{1.294}{1.618}{0.176}{1.037}{1.685}{0}
\end{minipage}
\hfill
\begin{minipage}[b]{\pwlayi}
\drawaspread{\pwlayii}{1.5}{1.5}{0.111}{1.5}{2.0}{0}
\end{minipage}
\caption{(Left) The \LaTeX{} book spread;
         (Right) Spread for many of Gutenberg's books}
\label{fig:spread}
\end{figure}
```

The `\drawaspread` command does not scale via the `\setlayoutscale` command. The size of the illustration is controlled by the value given for the width of the page.

3 Page layout

The layout of a single page is produced by the `\pagediagram` or `\pagedesign` (or `\drawpage`) command. The kind of page drawn is specified via three control commands[1].

Right- or left-hand page layouts are specified by the commands `\oddpagelayout`*true/false* (`\oddpagelayouttrue` for an odd-numbered (i.e., right-hand) page, or `\oddpagelayoutfalse` for an even-numbered (i.e., left-hand or verso) page). The default is `\oddpagelayouttrue`.

Double or single column layouts are controlled by the commands `\twocolumnlayout`*true/false*. Using `\twocolumnlayouttrue` will result in a picture of a double column page layout and using `\twocolumnlayoutfalse` will result in a picture of a single column page layout. The default is `\twocolumnlayoutfalse`.

The display of marginal notes is controlled by the commands `\drawmarginpars`*true/false*. Using `\drawmarginparstrue` rusults in representations of marginal notes being drawn and using `\drawmarginparsfalse` hides the marginal notes. The default is `\drawmarginparstrue`.

Normally LaTeX places any marginal notes in the outside margin; the standard LaTeX command `\reversemarginpar` causes notes to be placed in the opposite margin to the default one, and `\normalmarginpar` causes the notes to be put in the default margin. The package commands `\reversemarginpar`*true/false* may be used to control the marginal note locations in the pictures. `\reversemarginpartrue` corresponds to `\reversemarginpar` and `\reversemarginparfalse` corresponds to `\normalmarginpar`. The package default is `\reversemarginparfalse`.

A LaTeX class may always place marginal notes into the same margin. The package command `\marginparswitch`*true/false*, is supplied to control the placement in the diagrams. `\marginparswitchtrue` will place marginal notes according to the class desires, while `\marginparswitchfalse` will always draw marginal notes in one margin.

As an example, Figure 2 is produced by the following code.

```
\begin{figure}
\oddpagelayoutfalse
\twocolumnlayouttrue
\pagediagram
\caption{Left-hand two-column page layout parameters} \label{fig:pplt}
\end{figure}
```

The two lines forming an inverted 'L' represent the left-hand and top of the printed sheet. By default, LaTeX makes all page layout measurements from a point located one inch in from the left-hand side of the printed sheet and one inch down from the top of the

[1] Remember that all drawing commands of the form `\draw...`, except for `\drawaspread`, are also controlled by the `\drawparameters...` command.

The circle is at 1 inch from the top and left of the page. Dashed lines represent (`\hoffset + 1 inch`) and (`\voffset + 1 inch`) from the top and left of the page.

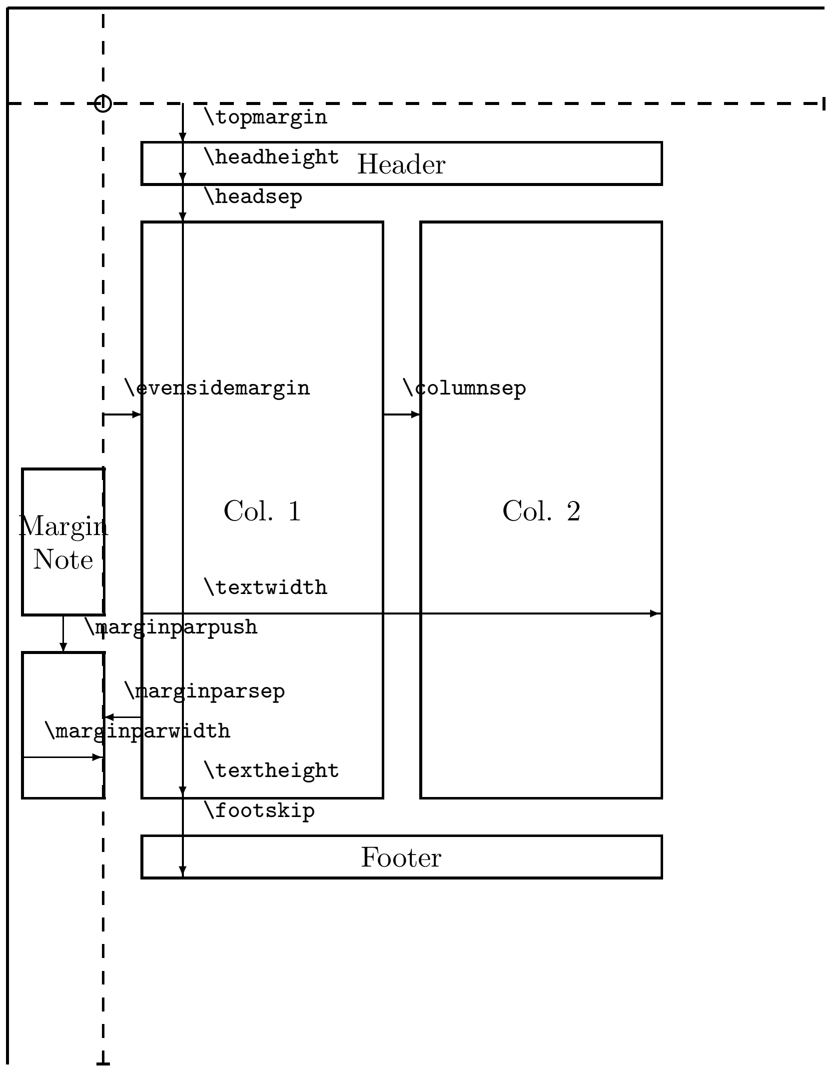


Figure 2: Left-hand two-column page layout parameters

sheet. This point is marked with a circle. The dashed lines correspond to the vertical and horizontal offsets from this point.

The command `\currentpage` collects together the page parameter settings for the current document for the page on which it is issued. For example, it determines whether the current page is odd or even and in which margin any marginal notes would be placed. Figure 3 shows the odd page layout for the document you are now reading. It was produced by the following commands:

```
\begin{figure}
\currentpage
\oddpagelayouttrue
\pagedesign
\caption{Odd page layout for this document} \label{fig:ptrs}
\end{figure}
```

The resulting picture is correctly proportioned, both horizontally and vertically.

When layouts are drawn with `...diagram`[2] the actual values of the parameters used to produce the layout are normally listed at the bottom of the picture. This can be used, as in Figure 3, to find out what the settings are for the current document.

Among other parameters that are set by `\currentpage` are values for the page width and height. If these have been specified by the `\paperwidth` and `\paperheight` commands, these values are used. Otherwise the width and height are set to the size of American letter paper, which is eight by eleven and a half inches.

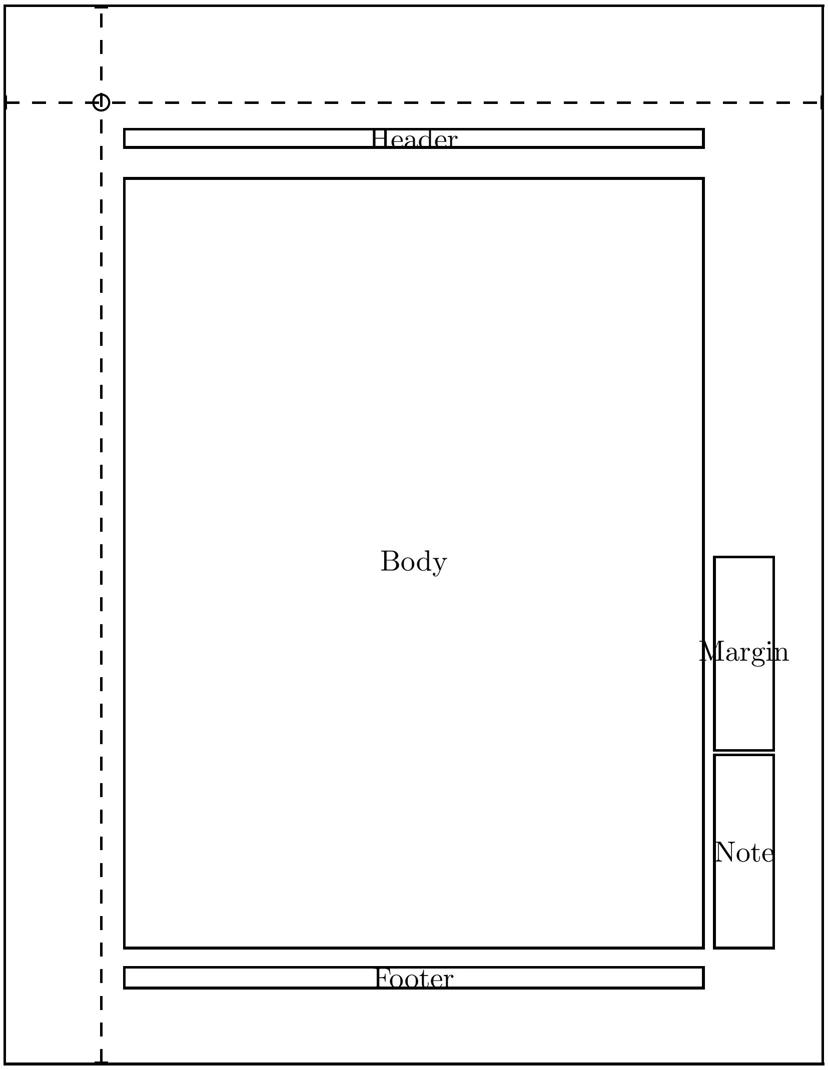
There are a set of commands for experimenting with the page layout parameters. They are listed in Table 2. Each of these commands takes one argument which is a length value. There is, in addition, an odd command, `\setfootbox{`⟨*height*⟩`}{`⟨*depth*⟩`}`, which may be used to specify the expected ⟨*height*⟩ and ⟨*depth*⟩ of the footer. Footers are typically a single line but may require more space than that. Note that there is no way to actually specify the vertical size of the footer in LaTeX, which will use as much space as it needs. If a footer is given a ⟨*depth*⟩ then a dashed line is drawn at `\footskip` below the main text box.

The following code provides an example of the use of some of these commands when trying an experimental page layout. The result is shown in Figure 4. Note that `\currentpage` is used to initialize all the appropriate parameter values before setting any trial values.

```
\begin{figure}
\currentpage
\trypaperwidth{11in}
\trypaperheight{8.5in}
```

[2] Or longhandedly with `\drawparametersfalse` and `\draw...`.

The circle is at 1 inch from the top and left of the page. Dashed lines represent (`\hoffset + 1 inch`) and (`\voffset + 1 inch`) from the top and left of the page.



Lengths are to the nearest pt.

page height = `795pt`	page width = `614pt`
`\hoffset = 0pt`	`\voffset = 0pt`
`\oddsidemargin = 18pt`	`\topmargin = 21pt`
`\headheight = 12pt`	`\headsep = 25pt`
`\textheight = 578pt`	`\textwidth = 434pt`
`\footskip = 30pt`	`\marginparsep = 10pt`
`\marginparpush = 5pt`	`\columnsep = 30pt`
`\columnseprule = 0.0pt`	

Figure 3: Page layout for this document

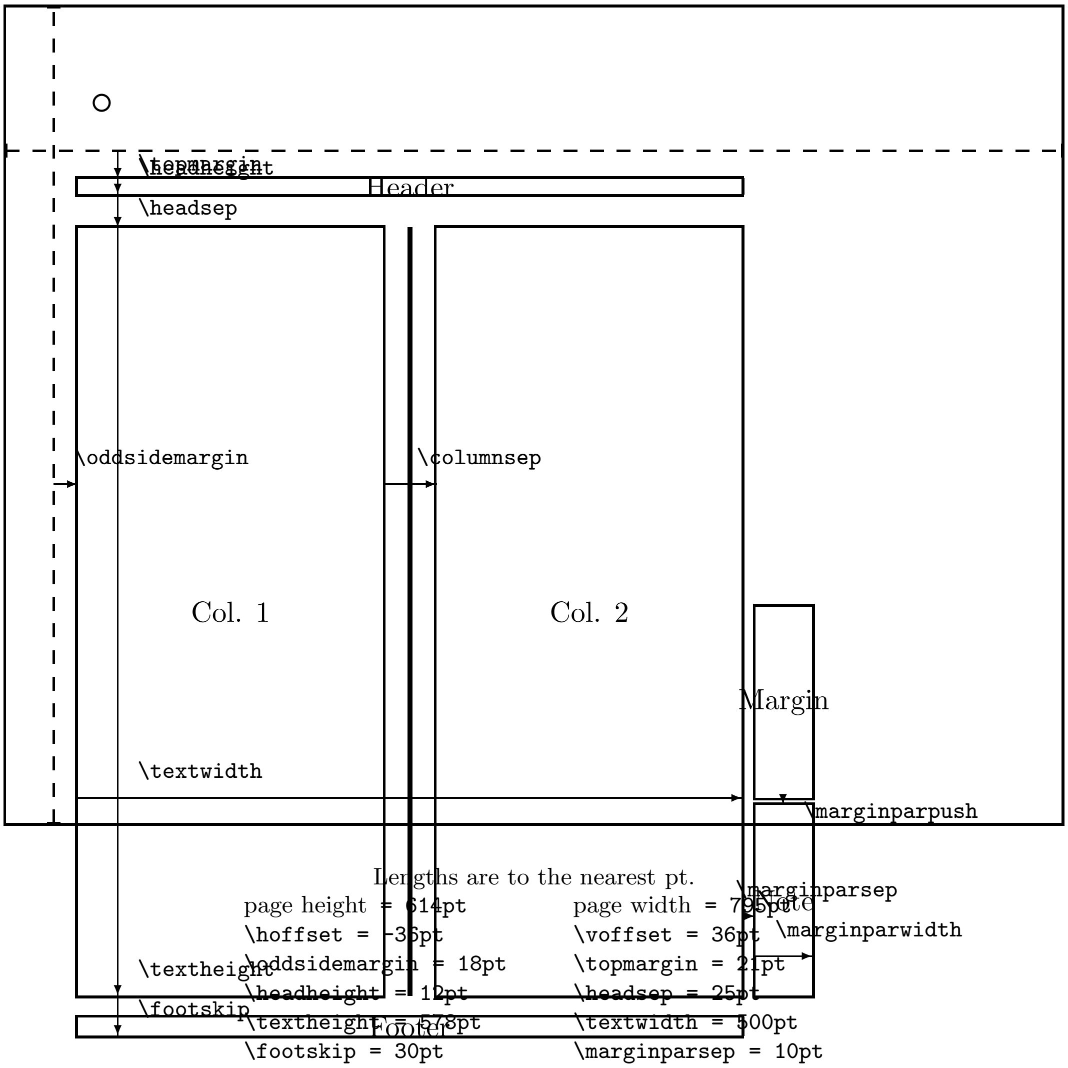
Table 2: Commands for setting trial page parameters

Command	Parameter
\trypaperwidth	sets the \paperwidth value
\trypaperheight	sets the \paperheight value
\tryhoffset	sets the \hoffset value (usually 0pt)
\tryvoffset	sets the \voffset value (usually 0pt)
\tryoddsidemargin	sets the \oddsidemargin (usually 21–63pt)
\tryevensidemargin	sets the \evensidemargin value (usually 39–82pt)
\trymarginparwidth	sets the \marginparwidth value (usually 68–107pt)
\trymarginparsep	sets the \marginparsep value (usually 10–11pt)
\trymarginparpush	sets the \marginparpush value (usually 5–7pt)
\trytopmargin	sets the \topmargin value (usually 27pt)
\tryheadheight	sets the \headheight value (usually 12pt)
\tryheadsep	sets the \headsep value (usually 25pt)
\tryfootskip	sets the \footskip value (usually 30pt)
\trytextheight	sets the \textheight value (usually 36–43 times the \baselineskip)
\trytextwidth	sets the \textwidth value (usually 345–390pt)
\trycolumnsep	sets the \columnsep value (usually 10pt)
\trycolumnseprule	sets the \columnseprule value (usually 0pt)

```
\trytextwidth{500pt}
\trycolumnsep{40pt}
\trycolumnseprule{3pt}
\tryhoffset{-0.5in}
\tryvoffset{0.5in}
\printheadingsfalse
\drawdimensionstrue
\twocolumnlayouttrue
\pagedesign
\caption{An experimental page layout}\label{fig:pudf}
\end{figure}
```

As can be seen, the resulting layout runs off the bottom of the specified physical page; either because the page dimensions were incorrectly set (the designer getting a landscape page when a portrait page was desired), or because the text height and width parameters were set incorrectly for the physical page (probably satisfactory for a portrait page but certainly wrong for the landscape page).

Note that \printheadingsfalse was used to delete the explanatory heading text, and dimensions were drawn because of \drawdimensionstrue.



Lengths are to the nearest pt.

page height = 614pt	page width = 795pt
\hoffset = -36pt	\voffset = 36pt
\oddsidemargin = 18pt	\topmargin = 21pt
\headheight = 12pt	\headsep = 25pt
\textheight = 578pt	\textwidth = 500pt
\footskip = 30pt	\marginparsep = 10pt
\marginparpush = 5pt	\columnsep = 40pt
\columnseprule = 3.0pt	

Figure 4: An experimental page layout

3.1 Changing the page layout in your document

You can only specify the page layout in a document's preamble. That is, it is set by the class, or by a package or per document via code you write in the preamble. To be on the safe side and avoid strange error messages like `You can't use '\spacefactor' in vertical mode`, enclose all your preamble commands within the command pair `\makeatletter` and `\makeatother`.

The page layout parameters can all be changed using the LaTeX `\setlength` command. For example, to set the width of the text to 3 inches, put this into the preamble:

```
\makeatletter
\setlength{\textwidth}{3in}
\makeatother
```

The `\pagevalues` command can be used to produce a table, as shown here, of the values of the current document's page layout parameters. This is the default tabulation.

```
\pagevalues
```

Actual page layout values.

`\paperheight` = 794.96999pt	`\paperwidth` = 614.295pt
`\hoffset` = 0.0pt	`\voffset` = 0.0pt
`\evensidemargin` = 18.06749pt	`\oddsidemargin` = 18.06749pt
`\topmargin` = 21.0pt	`\headheight` = 12.0pt
`\headsep` = 25.0pt	`\textheight` = 578.15999pt
`\textwidth` = 433.62pt	`\footskip` = 30.0pt
`\marginparsep` = 10.0pt	`\marginparpush` = 5.0pt
`\columnsep` = 30.0pt	`\columnseprule` = 0.0pt
1em = 10.95003pt	1ex = 4.71457pt

Note: Prior to version 2.5 the tabulations were set in a `center` environment. Starting with version 2.5 there is no surrounding environment.

4 The memoir class page layout

The memoir class [Wil01] provides a different set of page layout parameters than the standard LaTeX classes.

The layout of a single page for this class is produced by the `\stockdiagram` and `\stockdesign` (or `\drawstock`) commands. The kind of page drawn is specified via the same control commands that are used by the page drawing commands. That is:
`\oddpagelayout`*true/false*,
`\twocolumnlayout`*true/false*,
`\drawmarginpars`*true/false*,
`\reversemarginpar`*true/false*, and
`\marginparswitch`*true/false*.

Generally speaking, the commands for displaying and experimenting with the layout of a memoir class page are the same as, or analagous to, those for a standard page layout.

As an example, Figure 5 is produced by the following code.

```
\begin{figure}
\oddpagelayouttrue
\twocolumnlayoutfalse
\stockdiagram
\caption{Right-hand page major layout parameters for
         the \file{memoir} class} \label{fig:mempplt}
\end{figure}
```

Like the `\currentpage` command, `\currentstock` collects together the page parameter settings for the current memoir class document.

For information on how to design a page layout the class documentation provides all the details [Wil01] (and uses the layouts package to illustrate them).

There are a set of commands for experimenting with the page layout parameters. They are listed in Table 3. Each of these commands takes one argument which is a length value. The `\setfootbox` command may be used as well.

The following code provides an example of the use of some of these commands when trying an experimental page layout. The result is shown in Figure 6.

```
\begin{figure}
\currentstock
\trystockheight{11in}
\trystockwidth{8.5in}
\trytrimtop{1in}
\trytrimedge{1in}
```

Dashed lines represent the actual page size after trimming the stock.

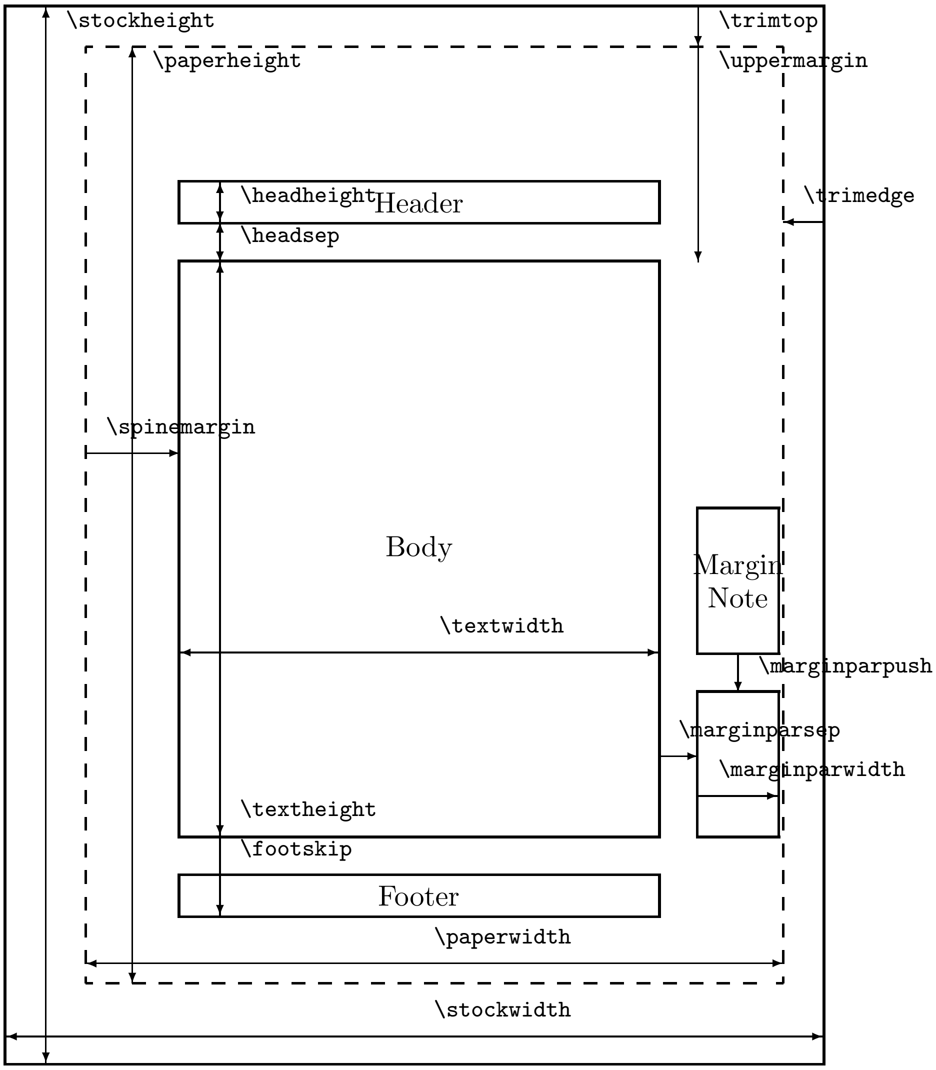


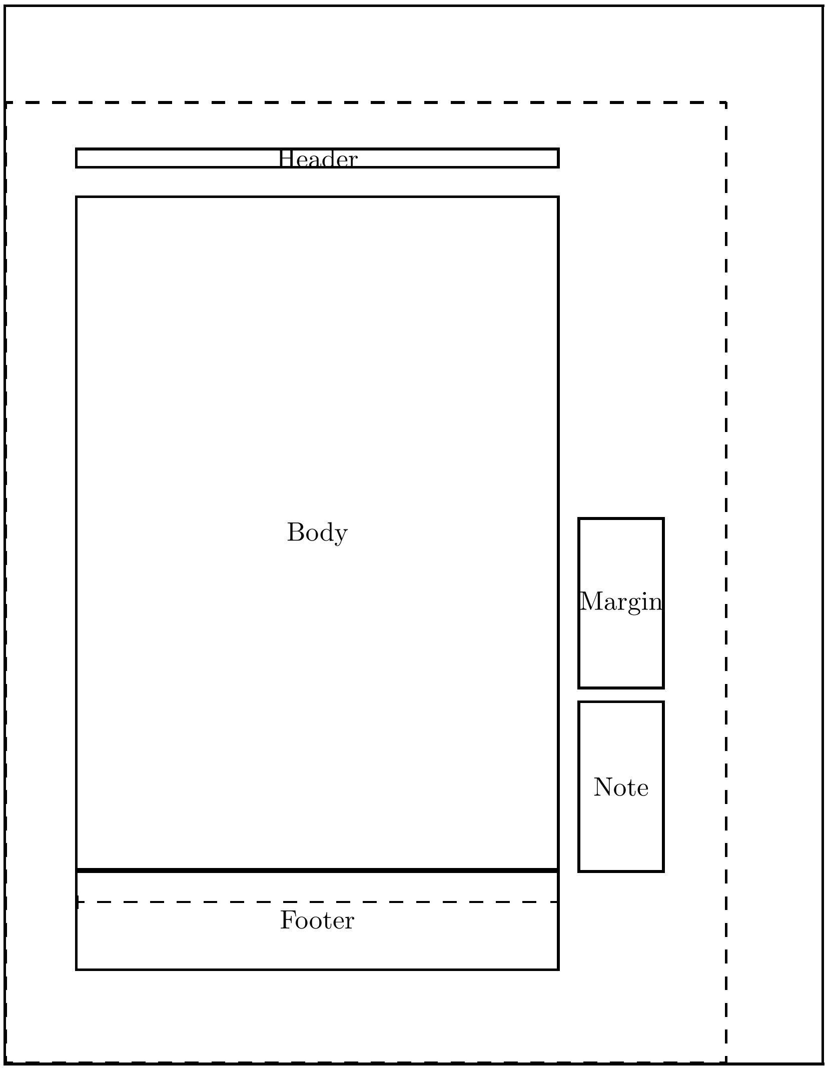
Figure 5: Right-hand page major layout parameters for the memoir class

Table 3: Commands for setting trial memoir page parameters

Command	Parameter
\trystockwidth	sets the \stockwidth value
\trystockheight	sets the \stockheight value
\trypaperwidth	sets the \paperwidth value
\trypaperheight	sets the \paperheight value
\trytrimedge	sets the \trimedge value
\trytrimtop	sets the \trimtop value
\tryspinemargin	sets the \spinemargin (usually 21–63pt)
\trymarginparwidth	sets the \marginparwidth value (usually 68–107pt)
\trymarginparsep	sets the \marginparsep value (usually 10–11pt)
\trymarginparpush	sets the \marginparpush value (usually 5–7pt)
\tryuppermargin	sets the \uppermargin value (usually 27pt)
\tryheadheight	sets the \headheight value (usually 12pt)
\tryheadsep	sets the \headsep value (usually 25pt)
\tryfootskip	sets the \footskip value (usually 30pt)
\trytextheight	sets the \textheight value (usually 36–43 times the \baselineskip)
\trytextwidth	sets the \textwidth value (usually 345–390pt)
\trycolumnsep	sets the \columnsep value (usually 10pt)
\trycolumnseprule	sets the \columnseprule value (usually 0pt)

```
\trypaperheight{10in}
\trypaperwidth{7.5in}
\tryspinemargin{0.75in}
\tryuppermargin{1.0in}
\tryheadheight{12pt}
\tryheadsep{24pt}
\trytextwidth{5in}
\trytextheight{7in}
\tryfootskip{24pt}
\setfootbox{50pt}{50pt}
\trymarginparsep{17pt}
\trymarginparwidth{62pt}
\trymarginparpush{12pt}
\setlabelfont{\small}
\stockdesign
\caption{An experimental \file{memoir} class page layout}\label{fig:pmemudf}
\end{figure}
```

There is also a \stockvalues command to print a tabulation of the current stock specifications.



Lengths are to the nearest pt.

```
\stockheight = 795pt       \stockwidth = 614pt
\pageheight = 723pt        \pagewidth = 542pt
\textheight = 506pt        \textwidth = 361pt
\trimtop = 72pt            \trimedge = 72pt
\uppermargin = 72pt        \spinemargin = 54pt
\headheight = 12pt         \headsep = 24pt
\footskip = 24pt           \marginparsep = 17pt
\marginparpush = 12pt      \columnsep = 30pt
\columnseprule = 0.0pt
```

Figure 6: An experimental memoir class page layout

5 Paragraph layout

The commands `\paragraphdiagram` and `\paragraphdesign` (or `\drawparagraph`) can be used to visualize the parameters that affect paragraphing. This is illustrated in Figure 7 which was produced by the following code:

```
\begin{figure}
\paragraphdiagram
\caption{Paragraph parameters}\label{fig:fpara}
\end{figure}
```

The command `\currentparagraph` sets the drawing parameters to the settings for the current document. The commands listed in Table 4 can be used to set trial values for the relevant parameters. These commands take one argument which must be a length.

Figure 8 shows the paragraphing as used in this document. It was produced with this code:

```
\currentparagraph
\begin{figure}
\paragraphdesign
\caption{Paragraphs in this document}\label{fig:dpara}
\end{figure}
```

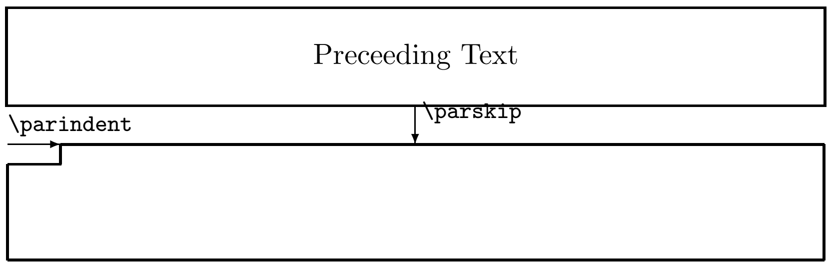
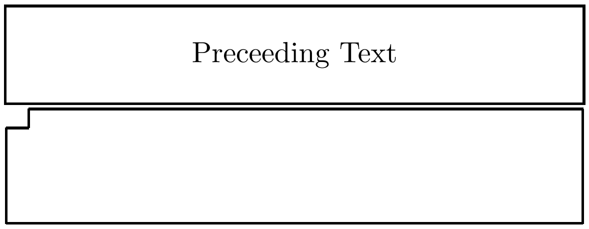


Figure 7: Paragraph parameters

Table 4: Commands for setting trial paragraph parameters

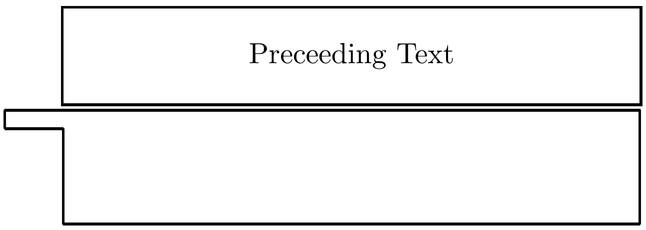
Command	Parameter
`\tryparindent`	sets the `\parindent` value
`\tryparskip`	sets the `\parskip` value
`\tryparlinewidth`	sets the `\linewidth` value
`\tryparbaselineskip`	sets the `\baselineskip` value



Lengths are to the nearest pt.

```
\parindent = 17pt        \parskip = 5pt
\baselineskip = 14pt     \linewidth = 434pt
```

Figure 8: Paragraphs in this document



Lengths are to the nearest pt.

```
\parindent = -44pt       \parskip = 5pt
\baselineskip = 14pt     \linewidth = 434pt
```

Figure 9: An outset paragraph

Interestingly, I found that LaTeX is happy even if `\parindent` is set to a negative value. It isn't demonstrated in this typescript but you can see the effect in Figure 9 which was produced from:

```
\currentparagraph
\begin{figure}
\tryparindent{-4em}
\paragraphdesign
\caption{An outset paragraph}\label{fig:mpara}
\end{figure}
```

The `\paragraphvalues` command can be used to produce a table, shown below, of the values of the current document's paragraph layout parameters (as set at the time that the command is used). The tabulation was typeset via

```
\begin{center}
\printinunitsof{mm}
\paragraphvalues
\end{center}
```

Note the use of `\printinunitsof` to set millimetres as the unit of length.

Actual paragraph layout values.

`\parindent` = 0mm	`\parskip` = 1.75728mm
`\baselineskip` = 4.77979mm	`\linewidth` = 152.39821mm
1em = 3.84843mm	1ex = 1.65695mm

6 Float layouts

Two sets of commands are provided for displaying the layouts of LaTeX floats (e.g., the `figure` and `table` environments). One set is for a macro view of floats and the other is for a more detailed view.

6.1 Float and text page layout

This set of commands is for displaying the general parameters for the location of floats on a page and proportioning the available space between the floats and textual material.

The commands `\floatpagediagram` and `\floatpagedesign` (or `\drawfloatpage`) are used to visualize the general parameters. These are illustrated in Figure 10 which was produced by the following code:

```
\begin{figure}
\floatpagediagram
\caption{Float and text page parameters}\label{fig:fpp}
\end{figure}
```

The command `\currentfloatpage` sets the drawing parameters to the settings for the current document. The 'standard' LaTeX settings are shown in Figure 11, produced by the code below.

```
\begin{figure}
\currentfloatpage
\trytotalnumber{3}
\trytopnumber{2}
\trytopfraction{0.7}
\trytextfraction{0.2}
\trybottomfraction{0.3}
\trybottomnumber{1}
\setlayoutscale{0.25}
\floatpagedesign
\caption{The standard \LaTeX{} float and text page settings}
\label{fig:fpstd}
\end{figure}
```

The commands listed in Table 5 can be used to set trial values for the relevant parameters. The commands take one argument, which is either an integer number or a decimal fraction, depending on the particular command.

Figure 12 illustrates float page settings that increase the likelihood of a float remaining

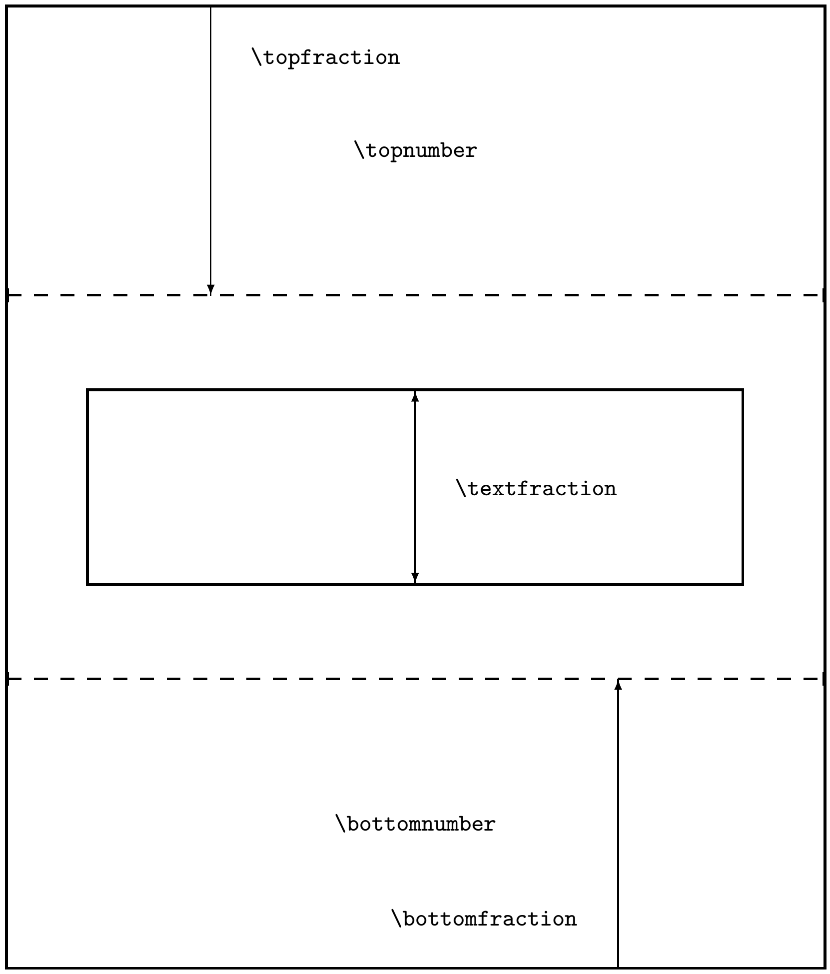
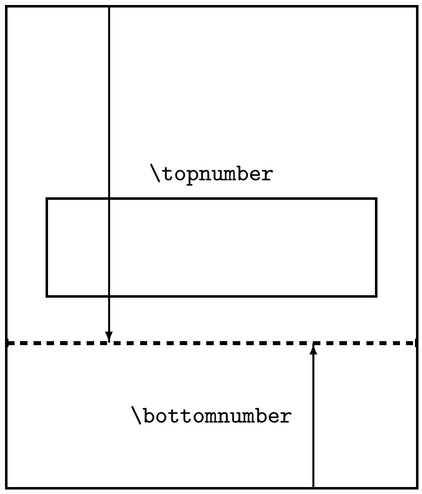


Figure 10: Float and text page parameters

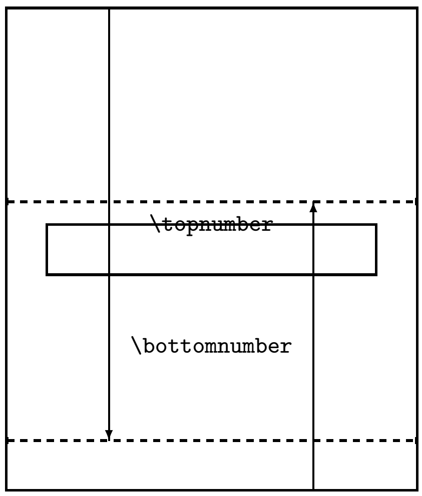
Table 5: Commands for setting trial float page parameters

Command	Parameter
`\trytotalnumber`	(integer) sets the `\totalnumber` (usually 3)
`\trytopnumber`	(integer) sets the `\topnumber` (usually 2)
`\trytopfraction`	(decimal) sets the `\topfraction` (usually 0.7)
`\trytextfraction`	(decimal) sets the `\textfraction` (usually 0.2)
`\trybottomnumber`	(integer) sets the `\bottomnumber` (usually 1)
`\trybottomfraction`	(decimal) sets the `\bottomfraction` (usually 0.3)




```
\topnumber = 2        \topfraction = 0.699
\bottomnumber = 1     \bottomfraction = 0.300
\totalnumber = 3      \textfraction = 0.199
```

Figure 11: The standard LaTeX float and text page settings




```
\topnumber = 2        \topfraction = 0.899
\bottomnumber = 2     \bottomfraction = 0.600
\totalnumber = 4      \textfraction = 0.100
```

Figure 12: Float page layout for decreasing likelihood of float-only pages

on a text page without being put on a page by itself. The figure was produced from the following code:

```
\begin{figure}
\currentfloatpage
\trytotalnumber{4}
\trytopnumber{2}
\trytopfraction{0.9}
\trytextfraction{0.1}
\trybottomnumber{2}
\trybottomfraction{0.6}
\setlayoutscale{0.25}
\floatpagedesign
\caption{Float page layout for decreasing likelihood of float-only pages}
\label{fig:fpudf}
\end{figure}
```

6.2 Detailed float layout

The other view of floats concentrates on the spacing between text and floats on a page. The `\floatdiagram` and `\floatdesign` (or `\drawfloat`) commands are used to visualize this aspect.

The relevant parameters are shown in Figure 13, produced from the following code:

```
\begin{figure}
\setlayoutscale{0.9}
\floatdiagram
\caption{Float parameters}\label{fig:flp}
\end{figure}
```

As usual, the command `\currentfloat` sets the trial float parameters to those in effect for the current document.

The commands listed in Table 6, all of which take a length argument, set trial values for the float parameters.

The following code, used to produce Figure 14, shows the use of some of these commands:

```
\begin{figure}
\currentfloat
\tryintextsep{\intextsep}
\trytopfigrule{0.5pt}
\trybotfigrule{1pt}
```

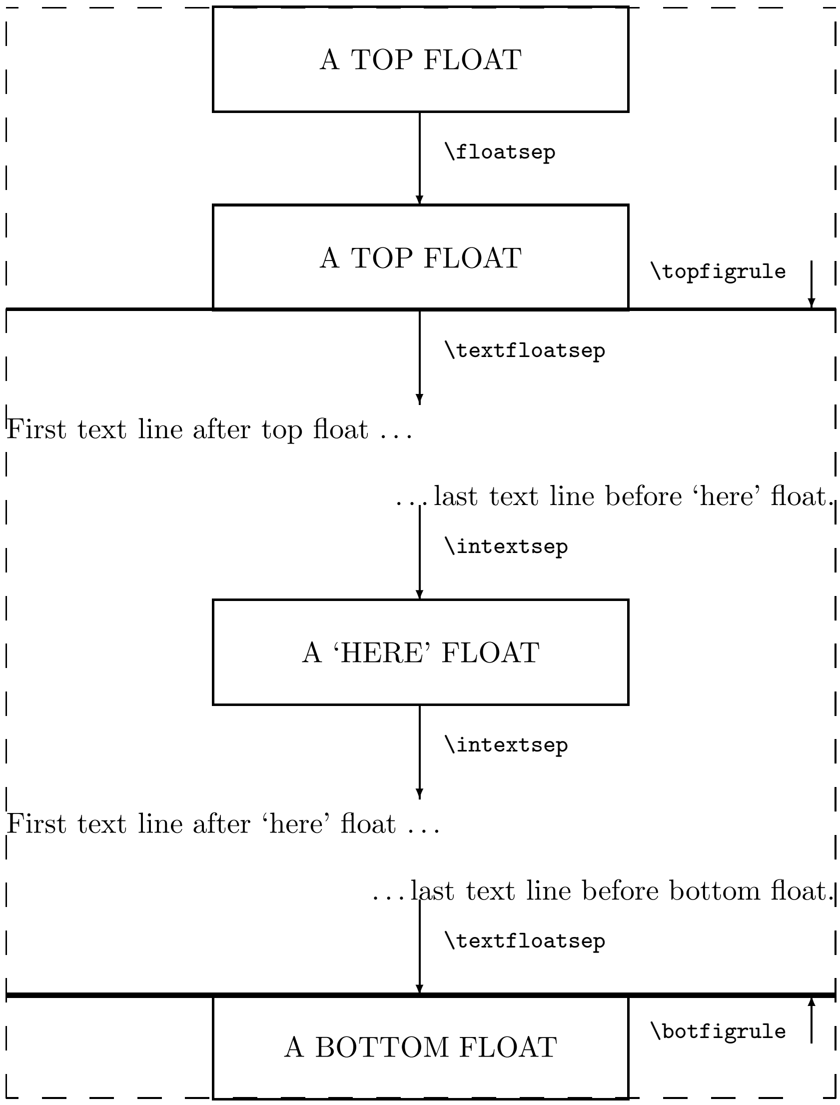
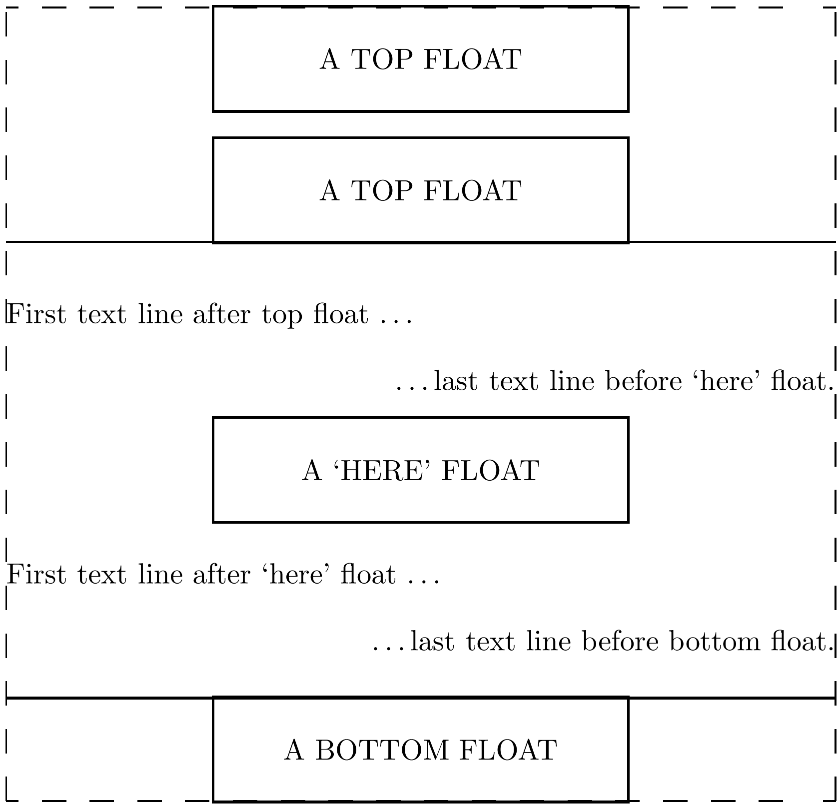


Figure 13: Float parameters

Table 6: Commands for setting trial float parameters

Command	Parameter
`\trytextfloatsep`	sets the `\textfloatsep` value (usually 20pt)
`\tryfloatsep`	sets the `\floatsep` value (usually 12pt)
`\tryintextsep`	sets the `\intextsep` value (usually 12pt)
`\trytopfigrule`	sets the thickness of a `\topfigrule` (usually 0pt)
`\trybotfigrule`	sets the thickness of a `\botfigrule` (usually 0pt)



Lengths are to the nearest pt.

`\floatsep = 12pt` `\textfloatsep = 20pt`
`\intextsep = 12pt` topfigrule thickness `= 0.5pt`
botfigrule thickness `= 1.0pt`

Figure 14: Float layout with rules

```
\setlayoutscale{0.9}
\floatdesign
\caption{Float layout with rules}\label{fig:fludf}
\end{figure}
```

The `\topfigrule` and `\botfigrule` are little known LaTeX commands; they are not discussed by Lamport [Lam94] but are described by Goossens *et al* [GMS94]. They are like the `\footnoterule` command in that they draw a rule, or other decoration, below floats at the top of a page (`\topfigrule`) and above floats at the bottom of a page (`\botfigrule`). Both these commands have been defined in the preamble to this document as:

```
\makeatletter
\newlength{\figrulesep}
\setlength{\figrulesep}{0.5\textfloatsep}
\newcommand{\topfigrule}{\vspace*{-1pt}%
  \noindent\rule[-\figrulesep]{\columnwidth}{1pt}}
\newcommand{\botfigrule}{\vspace*{-2pt}%
  \noindent\rule[\figrulesep]{\columnwidth}{2pt}}
\makeatother
```

Their effect can be seen throughout the printed result. The typical thickness for a `\rule` is 0.4pt; the thickness of these rules has been exagerated in order to make them more noticeable. The `\topfigrule` is drawn immediately after the top floats before the `\textfloatsep` spacing is applied. Similarly the `\botfigrule` is drawn after the `\textfloatsep` is applied for the bottom floats. Whatever is drawn as a `...figrule` should take no vertical space, hence the use of negative vertical space in their definitions above. Note that the rules have been given either positive or negative vertical offsets to ensure some space between a float and the rule.

6.3 Changing the float layout in your document

The `\...number` commands are changed with the LaTeX `\setcounter` command, while the `\...fraction` commands have to be changed via the `\renewcommand`. For example, the preamble to this document contains the following (enclosed within `\makeatletter` and `\makeatother` commands):

```
\setcounter{topnumber}{2}
\setcounter{bottomnumber}{2}
\setcounter{totalnumber}{4}
\renewcommand{\topfraction}{0.9}
\renewcommand{\bottomfraction}{0.6}
\renewcommand{\textfraction}{0.1}
```

Note that there are also the commands:

- `\dbltopnumber` for setting the maximum number of two-column floats at the top of a two-column page (typically 2);
- `\dbltopfraction` for setting the maximum fraction of a two column page that can be occupied by the top two-column floats (typically 0.7); and
- `\dblfloatpagefraction` for setting the minimum fraction of a page that has to be occupied by two-column floats before a 'float only' page is produced (typically 0.5).

The `\...sep` commands are changed using the `\setlength` command. The separation values should have a little bit of give in them, that is, they should be rubber lengths. A typical set of values might be specified as:

```
\setlength{\floatsep}{12pt plus 2pt minus 2pt}
\setlength{\textfloatsep}{20pt plus 2pt minus 4pt}
\setlength{\intextsep}{\floatsep}
```

There are corresponding separation commands for two-column floats at the top or bottom of a page. These are `\dblfloatsep` for inter-float separation and `\dbltextfloatsep` for spacing between a two-column float and the text area.

Actual float layout values.

`\floatsep` = 12.0pt plus 2.0pt minus 2.0pt	`\textfloatsep` = 20.0pt plus 2.0pt minus 4.0pt
`\intextsep` = 12.0pt plus 2.0pt minus 2.0pt	
topfig rule thickness = ??	botfig rule thickness = ??
`\topnumber` = 2	`\topfraction` = 0.9
`\bottomnumber` = 2	`\bottomfraction` = 0.6
`\totalnumber` = 4	`\textfraction` = 0.1
`\dblfloatsep` = 12.0pt plus 2.0pt minus 2.0pt	`\dbltextfloatsep` = 20.0pt plus 2.0pt minus 4.0pt
`\dbltopnumber` = 2	`\dbltopfraction` = .7
`\dblfloatpagefraction` = .5	`\floatpagefraction` = .5
1em = 10.95003pt	1ex = 4.71457pt

The `\floatvalues` command can be used to produce a table, as shown here, of the values of the current document's float layout parameters (as set at the time that the command is used).

7 List layout

The commands \listdiagram and \listdesign (or \drawlist), as their names suggest, display the layout of list environments. The list parameters are shown in Figure 15, which was produced by the following code:

```
\begin{figure}
\listdiagram
\caption{List parameters} \label{fig:lstp}
\end{figure}
```

The list layout may be controlled by the \listasparа*true/false* commands. Use \listasparatrue for displaying the list when it is being treated as a paragraph, otherwise use \listasparafalse. The default is \listasparatrue.

The command \currentlist extracts the list parameters from the current environment for display via \drawlist. Figure 16 graphically illustrates the layout for an enumerate type list. The figure was generated by the following code:

```
\begin{enumerate}
\item Figure~\ref{fig:lstenum} illustrates the layout of an
      \texttt{enumerate} list.
  \currentlist
  \begin{figure}
  \listdesign
  \caption{Layout of an \texttt{enumerate} list} \label{fig:lstenum}
  \end{figure}
\end{enumerate}
```

1. Figure 16 illustrates the layout of an enumerate list.

Note that \currentlist was called within the list environment in order to pick up the desired parameter values.

Table 7 gives a listing of the commands that are provided for experimenting with the list layout parameters. Each of these commands takes a length as its argument.

Figure 17 shows the layout of a user-defined list. An example of the list as it would appear in a document is shown by the list that follows this paragraph.

ListX, first item: This is an example of a user-defined list. The appearance is somewhat different from that normally seen in LaTeX lists. Note that the

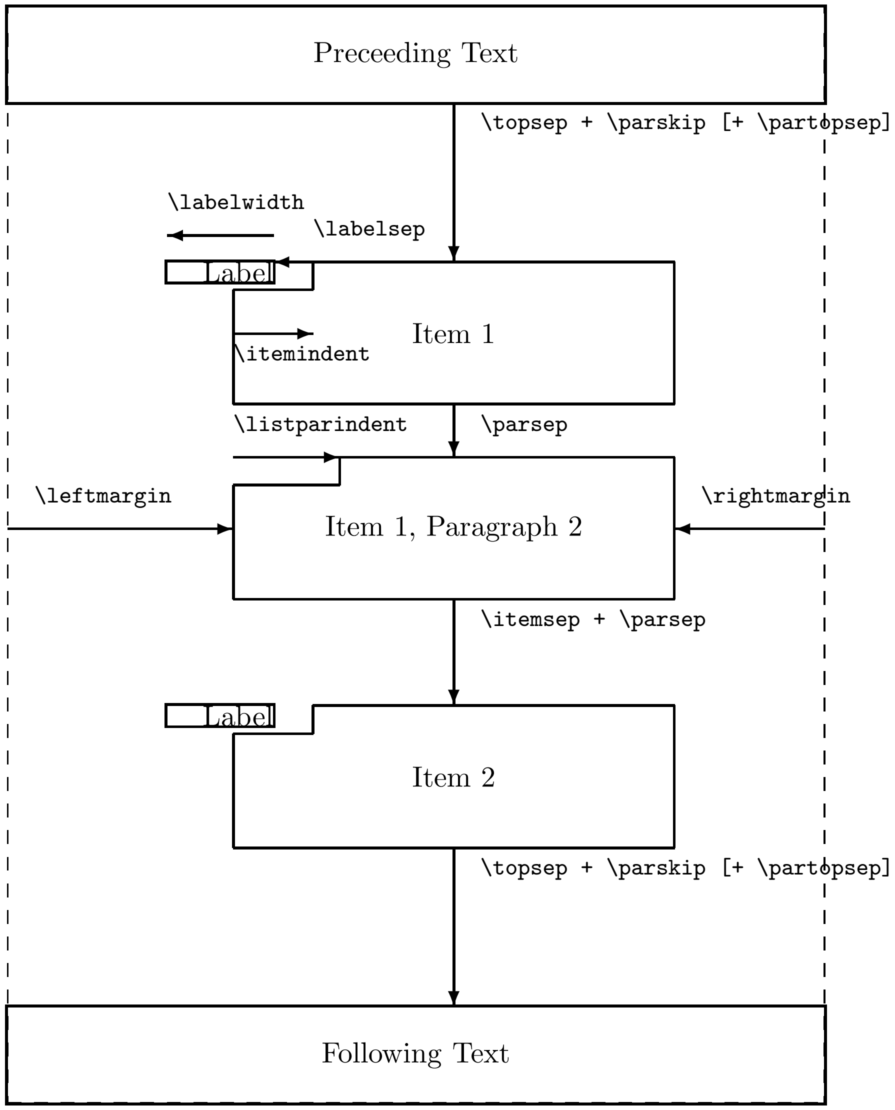
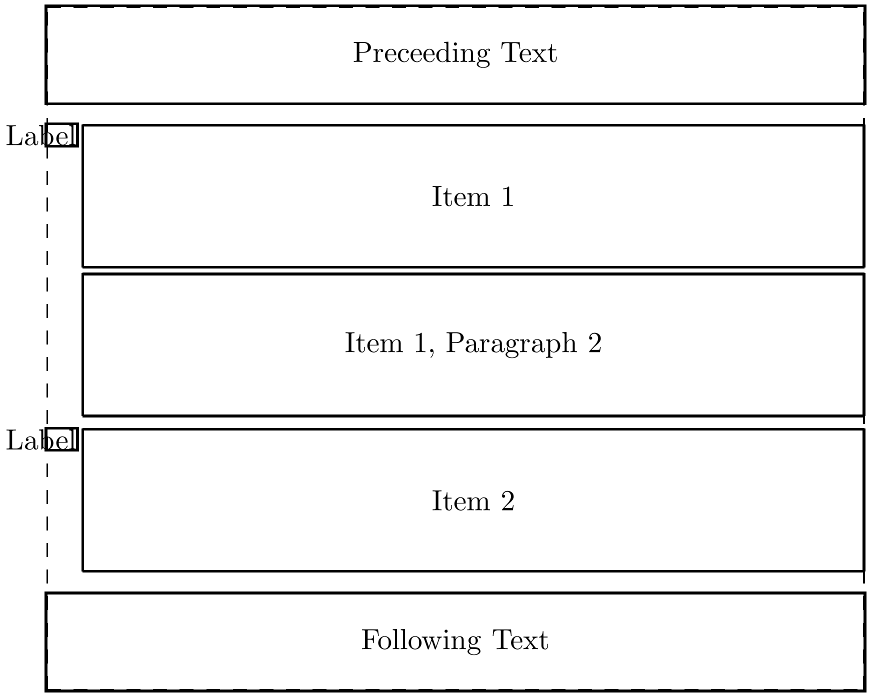


Figure 15: List parameters



Lengths are to the nearest pt.

```
\leftmargin = 27pt      \rightmargin = 0pt
\itemindent = 0pt       \labelwidth = 22pt
\labelsep = 5pt         \listparindent = 0pt
\topsep = 9pt           \parskip = 5pt
\partopsep = 3pt        \parsep = 5pt
\itemsep = 5pt
```

Figure 16: Layout of an `enumerate` list

Table 7: Commands for setting trial list parameters

Command	Parameter
`\tryitemindent`	sets the `\itemindent` value
`\trylabelwidth`	sets the `\labelwidth` value
`\trylabelsep`	sets the `\labelsep` value
`\tryleftmargin`	sets the `\leftmargin` value
`\tryrightmargin`	sets the `\rightmargin` value
`\trylistparindent`	sets the `\listparindent` value
`\trytopsep`	sets the `\topsep` value
`\tryparskip`	sets the `\parskip` value
`\trypartopsep`	sets the `\partopsep` value
`\tryparsep`	sets the `\parsep` value
`\tryitemsep`	sets the `\itemsep` value

text extends into the normal right-hand margin. Also, the body of the list text is indented from the label.

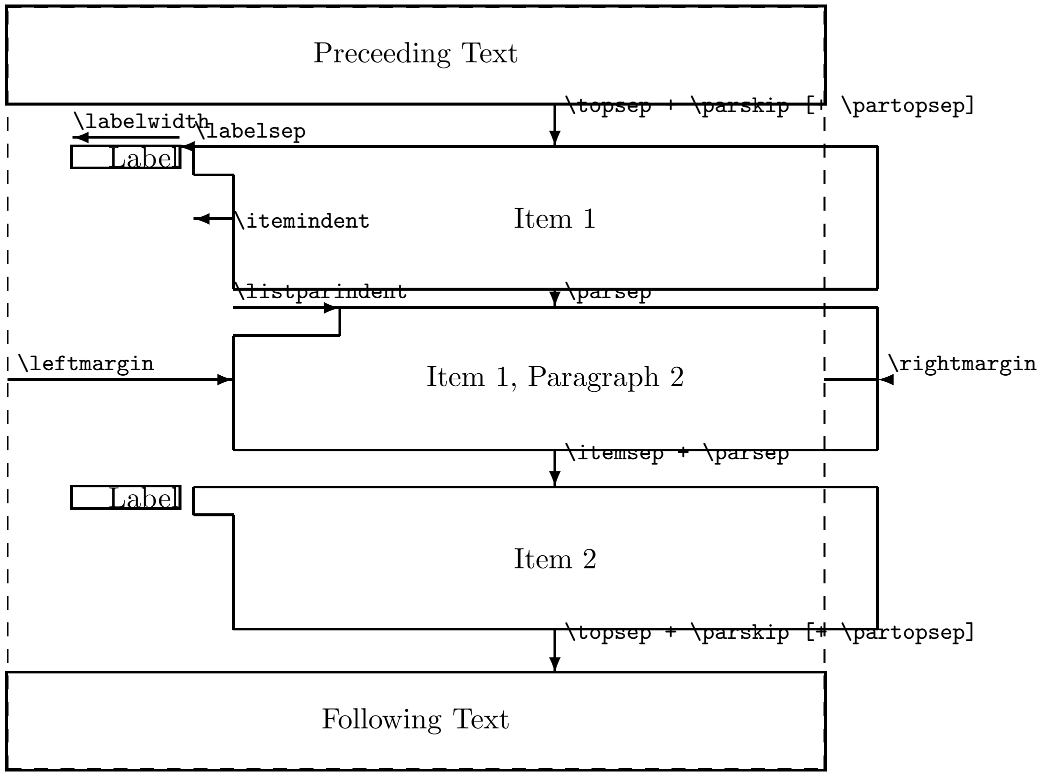
No claim is made that there is anything aesthetic about the design of the list. In fact, I think that it is pretty awful. It is merely provided as an example of a non-standard list and so that the `\drawlist` command can be shown off.

ListX, second item: Now we will draw the layout of this list from within itself (see Figure 17 for the result). The code used is:

```
\currentlist
\begin{figure}
\drawdimensionstrue
\listdesign
\caption{The layout of the \texttt{listX} environment}
\label{fig:lstudf}
\end{figure}
```

ListX, third item: The definition of this list environment is:

```
\newenvironment{listX}%
  {\begin{list}{item}%
    {\setlength{\itemindent}{-30pt}%
     \setlength{\labelwidth}{80pt}%
     \setlength{\labelsep}{1em}%
     \setlength{\leftmargin}{170pt}%
     \setlength{\rightmargin}{-40pt}%
     \setlength{\listparindent}{80pt}%
     \setlength{\topsep}{3ex}%
     \setlength{\partopsep}{\topsep}%
     \setlength{\parsep}{\topsep}%
     \setlength{\itemsep}{\topsep}%
    }%
  }%
  {\end{list}}
```



Lengths are to the nearest pt.

`\leftmargin = 170pt`	`\rightmargin = -40pt`
`\itemindent = -30pt`	`\labelwidth = 80pt`
`\labelsep = 11pt`	`\listparindent = 80pt`
`\topsep = 14pt`	`\parskip = 5pt`
`\partopsep = 14pt`	`\parsep = 14pt`
`\itemsep = 14pt`	

Figure 17: The layout of the `listX` environment

In LaTeX many display environments, such as the `quotation` environment, are defined in terms of the generic `list` environment, so the settings for these may also be explored with `\currentlist`. For example, the following code shows how to determine the 'list' settings for the `thebibliography` environment (with apologies to Leslie Lamport).

```
\begin{thebibliography}{Dillo 83}
\bibitem[Knud 66]{kn:gnus} D. E. Knudson. \emph{1996 World Gnus Almanac.}
\currentlist
\begin{figure}
\listdesign
\caption{Bibliography list}
\end{figure}
\end{thebibliography}
```

Running this code is left as an exercise for the reader.

7.1 Changing lists

Many of LaTeX's environments are defined in terms of lists, most noticeably the `description`, `enumerate` and `itemize` environments. To change any of LaTeX's predefined list environments it is probably best to examine their definitions in the appropriate class file (e.g., classes.dtx) and then put your modified definitions into a package file.

Actual list layout values.

`\leftmargin` = 2.28111pc	`\rightmargin` = 2.28111pc
`\itemindent` = 0pc	`\labelwidth` = 1.82489pc
`\labelsep` = 0.45622pc	`\listparindent` = 0pc
`\topsep` = 0.74995pc	`\parskip` = 0.37497pc
`\partopsep` = 0.24998pc	`\parsep` = 0.37497pc
`\itemsep` = 0.37497pc	
1em = 0.91245pc	1ex = 0.39285pc

The `\listvalues` command can be used to produce a table, as shown here, of the values of the current document's list layout parameters (as set at the time that the command is used). This table was produced by:

```
\begin{quote}
\printinunitsof{pc}
\listvalues
\end{quote}
```

8 Sectional heading layout

In LaTeX a few headings, like `\part` and `\chapter` are defined using special definition code. The majority, though, are defined via the internal LaTeX `\@startsection` command. This command takes 6 arguments:
`\@startsection{`⟨*name*⟩`}{`⟨*level*⟩`}{`⟨*indent*⟩`}{`⟨*beforeskip*⟩`}{`⟨*afterskip*⟩`}{`⟨*font style*⟩`}`.

The commands `\headingdiagram{`⟨*font style*⟩`}` and `\headingdesign{`⟨*font style*⟩`}` (or `\drawheading{`⟨*font style*⟩`}`) draw a picture of the layout for sectional headings based on the `\@startsection` command. Unlike the other drawing commands it takes one parameter, which is the specification of the size and/or font of the heading. For example, `\drawheading{\large\itshape}`. For `\headingdiagram` (or `\drawheading` with `\drawparameterstrue` is in effect) the parameter has no effect.

There are two kinds of headings:

1. Display headings, where the heading is set off from the text body, and
2. Run-in headings, where the text body starts on the same line as the heading.

When `\headingdiagram` is called (or `\drawheading` with `\drawparameterstrue` in effect) the pair of commands `\runinheadtrue` and `\runinheadfalse` control whether a run-in head or a display head will be illustrated. The default is `\runinheadfalse`.

Figures 18 and 19 show the parameters of the two kinds of headings. They were produced by the following code:

```
\begin{figure}
\setlayoutscale{1}
\headingdiagram{ }
\caption{Display heading parameters}\label{fig:hdp}
\end{figure}

\begin{figure}
\setlayoutscale{1}
\runinheadtrue
\headingdiagram{ }
\caption{Run-in heading parameters}\label{fig:hrp}
\end{figure}
```

The command `\currentheading` sets up default parameter values for the illustration of a heading. These values are based on guesstimates of the values of the arguments of the `\@startsection` command. The commands given in Table 8 can be used to explicitly set heading parameters. Each of these commands takes a length for its parameter value.

. . . end of last line of preceeding text.

$\|$*beforeskip*$\|$ + `\parskip` (of text font) + `\baselineskip` (of heading font)

indent 3.5 Heading Title

afterskip + `\parskip` (of heading font) + `\baselineskip` (of text font)

This is the start of the after-heading text which continues on . . .
second line of text following the heading . . .

Figure 18: Display heading parameters

. . . end of last line of preceeding text.

$\|$*beforeskip*$\|$ + `\parskip` (of text font) + `\baselineskip` (of heading font)

indent 3.5 Heading Title *afterskip* (< 0) Start of text . . .
second line of text following the heading . . .

Figure 19: Run-in heading parameters

Table 8: Commands for setting trial heading parameters

Command	Parameter
`\trybeforeskip`	sets the *beforeskip* value
`\tryafterskip`	sets the *afterskip* value
`\tryindent`	sets the *indent* value

. . . end of last line of preceeding text.

3.5 Heading Title

This is the start of the after-heading text which continues on . . .
second line of text following the heading . . .

Lengths are to the nearest pt.

beforeskip = -14pt	*afterskip* = 7pt
indent = 0pt	(heading font) \baselineskip = 0pt
\parskip = 5pt	heading font = \large

Figure 20: Subsubsection heading layout parameters for this document

The regular reader of LaTeX documents will have noticed that the `\subsubsection` headings in this manual do not conform to the usual LaTeX style. In fact, the preamble to this document contains the following definition:

```
\makeatletter
\renewcommand{\subsubsection}{\@startsection%
  {subsubsection}%       name
  {3}%                   level
  {0mm}%                 indent
  {-\baselineskip}%      beforeskip
  {0.5\baselineskip}%    afterskip
  {\large\itshape}}%     style
\makeatother
```

Note that the `\makeatletter` and `\makeatother` commands are required because of the use of the @ character in the name of the `\@startsection` command.

Figure 20 illustrates the layout for this heading, and was produced by the code below.

```
\currentheading
\trybeforeskip{-\baselineskip}
\tryafterskip{0.5\baselineskip}
\tryindent{0mm}
\begin{figure}
\setlayoutscale{1}
\headingdesign{\large\itshape}
\caption{Subsubsection layout parameters for this document}
\label{fig:hdudf}
```

. . . end of last line of preceeding text.

3.5 Heading Title Start of text . . .
second line of text following the heading . . .

Lengths are to the nearest pt.

beforeskip = -14pt	*afterskip* = -7pt
indent = 0pt	(heading font) \baselineskip = 0pt
\parskip = 5pt	heading font = \large

Figure 21: Subsubsection layout parameters for a run-in heading

```
\end{figure}
```

The same heading, but specified as a run-in heading by making the value of *afterskip* negative rather than positive, is illustrated in Figure 21.

9 Footnote layout

Footnote layouts are produced by the commands `\footnotediagram` and `\footnotedesign` (or `\drawfootnote`). The relevant footnote parameters are shown in Figure 22, which was produced by the following code:

```
\begin{figure}
\setlayoutscale{0.4}
\setlabelfont{\huge\itshape}
\footnotediagram
\caption{The footnote parameter layout} \label{fig:fp}
\end{figure}
```

Note the use of `\setlabelfont` to change the font used for labelling (the 'MAIN TEXT' in this case).

Some of the current settings for the footnote parameters are set by the `\currentfootnote` command. Guesstimates are provided for the likely value of the `\baselineskip` that is used within a footnote, and also for the dimension of the footnote rule.

Figure 23 shows the default footnote layout, and was produced by the following code:

```
\begin{figure}
\currentfootnote
```

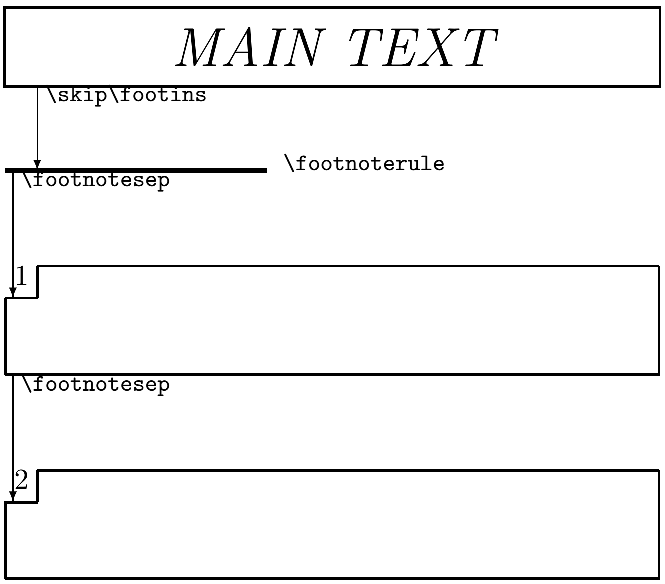


Figure 22: The footnote parameter layout

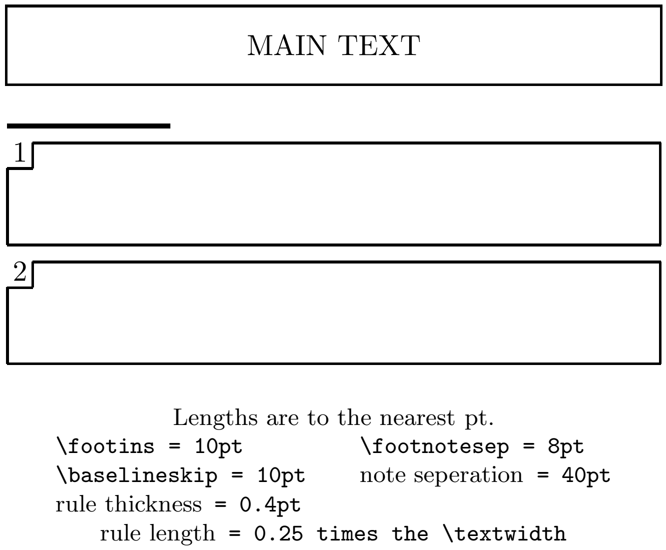


Figure 23: A footnote layout

Table 9: Commands for setting trial footnote parameter values

Command	Parameter
`\tryfootins`	sets the `\skip\footins` value (usually 10pt)
`\tryfootnotesep`	sets the `\footnotesep` value (usually 7pt)
`\tryfootnotebaseline`	sets the footnote `\baselineskip` value
`\tryfootruleheight`	sets the `\footnoterule` height value (usually 0.4pt)
`\tryfootrulefrac`	sets the length of the `\footnoterule`

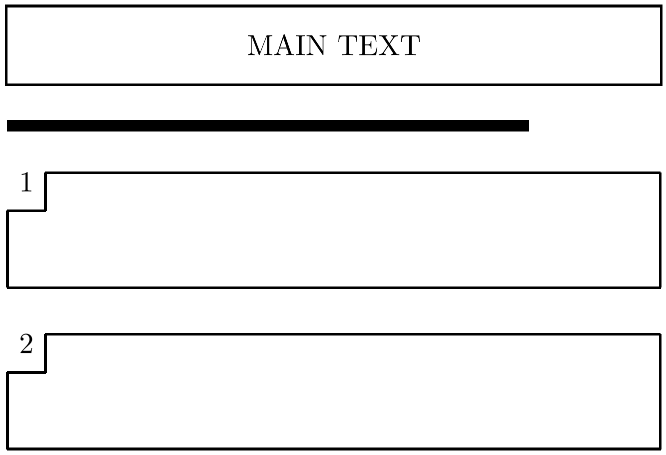
```
\setlayoutscale{0.4}
\footnotedesign
\caption{The current footnote layout}\label{fig:ftry}
\end{figure}
```

The resulting picture has all vertical dimensions magnified by a factor of 4 with respect to the horizontal dimensions.

Commands for individually setting trial values for footnote parameters are given in Table 9. Except for `\tryfootrulefrac`, these all take a length as their parameter. The parameter value for `\tryfootrulefrac` is a decimal number representing a fraction of the `\textwidth`. The trial length of the footnote rule is set to this fraction of the width of the text block.

Figure 24, produced from the following code, shows an experimental layout for footnotes.

```
\begin{figure}
```



Lengths are to the nearest pt.

`\footins = 10pt` `\footnotesep = 20pt`
`\baselineskip = 15pt` note seperation `= 80pt`
rule thickness `= 1.0pt`
rule length `= 0.8 times the \textwidth`

Figure 24: A user-specified footnote layout

```
\currentfootnote
\tryfootins{10pt}
\tryfootnotesep{20pt}
\tryfootnotebaseline{15pt}
\tryfootruleheight{1pt}
\tryfootrulefrac{0.8}
\setlayoutscale{0.4}
\footnotedesign
\caption{A user-specified footnote layout}
\label{fig:fudf}
\end{figure}
```

9.1 Changing the footnote layout

The value of `\footnotesep` is changed via the `\setlength` command. Spacing between the bottom of the text area and the first footnote is normally a rubber length. The following is a typical set of values:

```
\setlength{\footnotesep}{7pt}
\setlength{\skip\footins}{10pt plus 4pt minus 2pt}
```

For changing other values, see the documented class file classes.dtx.

Actual footnote layout values.

`\footins` = 10.0pt plus 4.0pt minus 2.0pt	`\footnotesep` = 7.7pt
rule thickness = ??	rule length = ??
1em = 10.95003pt	1ex = 4.71457pt

The `\footnotevalues` command can be used to produce a table, as shown here, of the values of the current document's footnote layout parameters (as set at the time that the command is used).

10 Table of Contents layout

The format of an entry in the Table of Contents (ToC) is usually specified by the internal LaTeX `\@dottedtocline` command:
`\@dottedtocline{`⟨*level*⟩`}{`⟨*indent*⟩`}{`⟨*numwidth*⟩`}`
where ⟨*indent*⟩ and ⟨*numwidth*⟩ relate to the formatting, and an entry will be typeset only if ⟨*level*⟩ is less than or equal to the value of the `tocdepth` counter.

The `\tocdiagram` and `\tocdesign` (or `\drawtoc`) commands are used for visualizing the layout of section titles in a *Table of Contents* listing based on the `\@dottedtocline` command, as shown in Figure 25. This was produced by the following code:

```
\begin{figure}
\setlayoutscale{0.8}
\tocdiagram
\caption{Table of Contents entry parameters}\label{fig:tocp}
\end{figure}
```

Figure 26 was produced by the following code:[3]

```
\begin{figure}
\setlayoutscale{0.8}
\currenttoc
\tocdesign
\caption{Typical Table of Contents entry for this document} \label{fig:thistoc}
```

[3]The drawing of the dotted leader is not completely accurate due to scaling and rounding within the drawing algorithm.

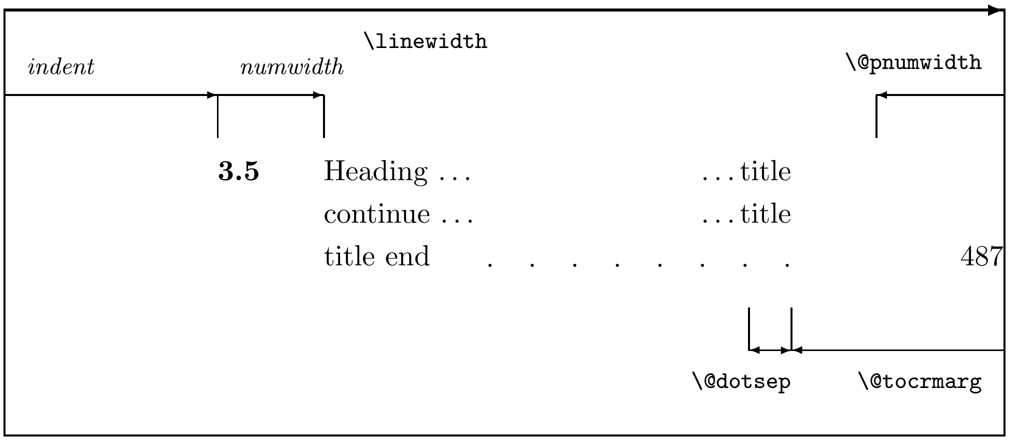


Figure 25: Table of Contents entry parameters

3.5 Heading . . . . . . title
continue . . . . . . title
title end 487

Lengths are to the nearest pt.

```
indent = 16pt          numwidth = 25pt
\@tocrmarg = 28pt      \@pnumwidth = 17pt
\@dotsep = 4.5
```

Figure 26: Typical Table of Contents entry for this document

Table 10: Commands for setting trial values for ToC parameters

Command	Parameter
`\trytocindent`	sets the *indent* value
`\trytocnumwidth`	sets the *numwidth* value
`\trytoclinewidth`	sets the `\linewidth` value
`\trytocrmarg`	sets the `\@tocrmarg` value
`\trytocpnumwidth`	sets the `\@pnumwidth` value
`\trytocdotsep`	sets the `\@dotsep` value

```
\end{figure}
```

The command `\currenttoc` sets values for the ToC parameters based on those for the current document. The paramaters can all be individually adjusted via the commands listed in Table 10. These commands all require a length as their parameter, except for the `\trytocdotsep` command which takes a number (integer or decimal) as its parameter value. This sets the trial value for the separation between dots in the leader between the sectional title and the page number. The parameter is the separation value in *mu* (math units).[4]

Figure 27, which was produced by the following code, shows that the effect of having a large value for `\@dotsep` is to eliminate the dotted leader line between the title text and

[4]There are *18mu* units to *1em* unit.

3.5 Heading . . . . . . title
continue . . . . . . title
title end 487

Lengths are to the nearest pt.

```
indent = 16pt          numwidth = 25pt
\@tocrmarg = 28pt      \@pnumwidth = 17pt
\@dotsep = 1000
```

Figure 27: Table of Contents entry with a large value for `\@dotsep`

the page number.

```
\begin{figure}
\setlayoutscale{0.8}
\currenttoc
\trytocdotsep{1000}
\tocdesign
\caption{Table of Contents entry with a large value for \texttt{\bs @dotsep}}
\label{fig:tocudf}
\end{figure}
```

10.1 Changing the Table of Contents, etc

The methods of specifying the typesetting of the various entries in a Table of Contents or a List of Figures (or Tables) varies from one kind of entry to another. For details consult the documented class specification file classes.dtx.

However, the `\@dotsep`, `\@pnumwidth` and `\@tocrmarg` values can all be set with the `\renewcommand`. For example, in the preamble:

```
\makeatletter
\renewcommand{\@dotsep}{4.5}
\renewcommand{\@pnumwidth}{1.55em}
```

```
\renewcommand{\@tocrmarg}{2.55em}
\makeatother
```

Typically, for figure and table captions, and for sectioning commands that are defined using the `\@startsection` command, table entry typesetting is specified via an `\l@X` command, where `X` is `figure`, `table`, `subsection`, etc., as appropriate. In turn, these commands call the `\@dottedtocline` command. For example, to change the typesetting for a paragraph entry in the Table of Contents, and for a table caption you should do something like (changing the lengths as appropriate):

```
\makeatletter
\renewcommand*{\l@paragraph}{\@dottedtocline{4}{7em}{4em}}
\renewcommand*{\l@table}{\@dottedtocline{1}{0em}{3.0em}}
\makeatother
```

Actual ToC layout values.

`\@tocrmarg` = 2.55em	`\@pnumwidth` = 1.55em
`\@dotsep` = 4.5	*indent* = ??
numwidth = ??	
1em = 10.95003pt	1ex = 4.71457pt

The `\tocvalues` command can be used to produce a table, as shown here, of the values of the current document's Table of Contents layout parameters (as set at the time that the command is used).

11 Font boxing

Sometimes it is useful to see the size of the box enclosing some text. Two commands are provided for this purpose.

The `\drawfontframe{`⟨*text*⟩`}` produces a drawing of ⟨*text*⟩ together with its surrounding box drawn. A bullet is placed at the position of the reference point and a horizontal dotted line is drawn along the baseline. The command can be used in normal mode or can be `\put` in a picture environment.  The framed g was produced by the following code:

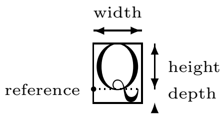
```
\drawfontframe{\Huge\textbf{g}}
```

The command places the box inside a `picture` environment that is just big enough to enclose the box.

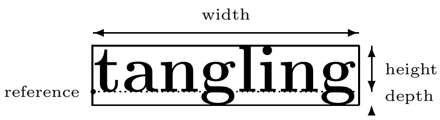
Sometimes it is useful to see the size of the box enclosing some text. Two commands are provided for this purpose.

The `\drawfontframelabel{`⟨*text*⟩`}` is similar, except that it labels the reference point, and the width height and depth of the box. The box is placed inside a picture environment that is just big enough to enclose the box. This means that the labels extend beyond the `picture`. If `\printparameterstrue` is set before issuing the command, the actual values for the width, height and depth of the box are printed in a `center` environment following the drawing.

Here is a simple example created by `\drawfontframelabel{\Huge Q}` with `\printparametersfalse`.



This time with `\printparameterstrue` and `\drawfontframelabel{\Huge\textbf{tangling}}`



width `= 99.51997pt` height `= 17.27776pt` depth `= 4.83775pt`

If either of the `\drawfont...` commands are used inside a `picture` environment, then the `\unitlength` must be set to 1pt, as the drawings are meant to be exact size and the commands assume that all drawing lengths are in terms of pts.

Internally, the commands typeset their argument inside a 'save box' called `\layoutsbox` and then perform the size measurements on `\layoutsbox`. You can use `\layoutsbox` for your own purposes, but if you do so any subsequent use of a `\drawfont...` command will overwrite anything that you might have saved in `\layoutsbox`.

References

[GMS94] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Addison-Wesley Publishing Company, 1994.

[Lam94] Leslie Lamport. *LaTeX: A Document Preparation System*. Addison-Wesley Publishing Company, second edition, 1994.

[McP88] Kent McPherson. 'Page Layout in LaTeX'. *TUGboat*, 9(1):78–82, April 1988.

[Wil01] Peter Wilson. *The memoir class for configurable book typesetting*. June, 2001. (Available from CTAN in `macros/latex/contrib/supported/memoir`)

Index

\@dotsep 44–45
\@dottedtocline 43, 46
\@pnumwidth 44–45
\@startsection 35, 37, 46
\@tocrmarg 44–45

B
\baselineskip 2, 11, 16, 18, 39–40
\botfigrule 25, 27
\bottomfraction 22
\bottomnumber 22

C
center (environment) 4, 13
\chapter 35
classes.dtx (file) 34, 42, 45
\columnsep 11, 16
\columnseprule 11, 16
\currentfloat 24
\currentfloatpage 21
\currentfootnote 39
\currentheading 35
\currentlist 29, 34
\currentpage 9, 14
\currentparagraph 18
\currentstock 14
\currenttoc 44

D
\dblfloatpagefraction 28
\dblfloatsep 28
\dbltextfloatsep 28
\dbltopfraction 28
\dbltopnumber 28
description (environment) 34
\drawaspread 4, 6
\drawdimensionsfalse 3
\drawdimensionstrue 3, 11
\drawfloat 24
\drawfloatpage 21
\drawfontframe 47
\drawfontframelabel 47
\drawfootnote 39
\drawheading 35
\drawlist 29, 32
\drawmarginparsfalse 7
\drawmarginparstrue 7
\drawpage 7
\drawparagraph 18
\drawparametersfalse 2–3
\drawparameterstrue 2, 35
\drawstock 14
\drawtoc 43

E
enumerate (environment) 29, 34
\evensidemargin 11

F
figure (environment) 4, 21
\floatdesign 24
\floatdiagram 24
\floatpagedesign 21
\floatpagediagram 21
\floatsep 25
\floatvalues 28
\footins 40
\footnotedesign 39
\footnotediagram 39
\footnoterule 27, 40
\footnotesep 40–41
\footnotesize 1
\footnotevalues 42
\footskip 9, 11, 16

H
\headheight 11, 16
\headingdesign 35
\headingdiagram 35
\headsep 11, 16
\hoffset 11

I
\intextsep 25
\itemindent 31
itemize (environment) 34
\itemsep 31

L
\labelsep 31
\labelwidth 31
layout.sty (file) 1
layoutsbox (save box) 47
\leftmargin 31
\linewidth 18, 44
list (environment) 29, 34
\listasparafalse 29
\listasparatrue 29
\listdesign 29
\listdiagram 29
\listparindent 31
\listvalues 34

M
\makeatletter 13, 27, 37
\makeatother 13, 27, 37
\marginparpush 11, 16
\marginparsep 11, 16
\marginparswitchfalse 7
\marginparswitchtrue 7
\marginparwidth 11, 16

N
\normalmarginpar 7

O
\oddpagelayoutfalse 7
\oddpagelayouttrue 7
\oddsidemargin 11

P
\pagedesign 7
\pagediagram 7
\pagevalues 13
\paperheight 9, 11, 16
\paperwidth 9, 11, 16
\paragraphdesign 18
\paragraphdiagram 18
\paragraphvalues 20
\parindent 18, 20
\parsep 31
\parskip 2, 18, 31
\part 35
\partopsep 31
picture (environment) 4
\printheadingsfalse 1, 3, 11
\printheadingstrue 1, 3
\printinunitsof 3, 20
\printparametersfalse 3
\printparameterstrue 3
\prntlen 3

Q
quotation (environment) 34

R
\renewcommand 27, 45
\reversemarginpar 7
\reversemarginparfalse 7
\reversemarginpartrue 7
\rightmargin 31
\rule 27
\runinheadfalse 35
\runinheadtrue 35

S
\setcounter 27
\setfootbox 9, 14
\setlabelfont 3, 39
\setlayoutscale 2, 4, 6
\setlength 13, 28, 41
\setparameterstextsize 3
\setuplayouts 2, 4
\setvaluestextsize 1
\skip 40
\spacefactor 13
\spinemargin 16
\stockdesign 14
\stockdiagram 14

\stockheight 16
\stockvalues 16
\stockwidth 16
\subsubsection 37

T

table (environment) 21
tabular (environment) 4
\textfloatsep 25, 27
\textfraction 22
\textheight 11, 16
\textwidth 11, 16, 40
thebibliography (environment) 34
tocdepth (counter) 43
\tocdesign 43
\tocdiagram 43
\tocvalues 46
\topfigrule 25, 27
\topfraction 22
\topmargin 11
\topnumber 22
\topsep 31
\totalnumber 22
\trimedge 16
\trimtop 16
\tryafterskip 36
\trybeforeskip 36
\trybotfigrule 25
\trybottomfraction 22
\trybottomnumber 22
\trycolumnsep 11, 16
\trycolumnseprule 11, 16
\tryevensidemargin 11
\tryfloatsep 25
\tryfootins 40
\tryfootnotebaseline 40
\tryfootnotesep 40
\tryfootrulefrac 40
\tryfootruleheight 40
\tryfootskip 11, 16
\tryheadheight 11, 16
\tryheadsep 11, 16
\tryhoffset 11
\tryindent 36
\tryintextsep 25
\tryitemindent 31
\tryitemsep 31
\trylabelsep 31
\trylabelwidth 31
\tryleftmargin 31
\trylistparindent 31
\trymarginparpush 11, 16
\trymarginparsep 11, 16
\trymarginparwidth 11, 16
\tryoddsidemargin 11
\trypaperheight 11, 16
\trypaperwidth 11, 16
\tryparbaselineskip 18
\tryparindent 18
\tryparlinewidth 18
\tryparsep 31
\tryparskip 18, 31
\trypartopsep 31
\tryrightmargin 31
\tryspinemargin 16
\trystockheight 16
\trystockwidth 16
\trytextfloatsep 25
\trytextfraction 22
\trytextheight 11, 16
\trytextwidth 11, 16
\trytocdotsep 44
\trytocindent 44
\trytoclinewidth 44
\trytocnumwidth 44
\trytocpnumwidth 44
\trytocrmarg 44
\trytopfigrule 25
\trytopfraction 22
\trytopmargin 11
\trytopnumber 22
\trytopsep 31
\trytotalnumber 22
\trytrimedge 16
\trytrimtop 16

\tryuppermargin 16
\tryvoffset 11
\twocolumnlayoutfalse 7
\twocolumnlayouttrue 7

U
\uppermargin 16

V
\voffset 11