

The `layouts` package: User manual

Peter R. Wilson
CUA and NIST
(Now at: `peter.r.wilson@boeing.com`)

January 1999*

Abstract

The L^AT_EX `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	1
1.2.1	Scaling	2
1.2.2	Layout types	2
1.3	Usage	2
2	Page spread	3
3	Page layout	5
3.1	Changing the page layout in your document	9
4	Paragraph layout	11

*The previous version was dated November 1998.

5	Float layouts	14
5.1	Float and text page layout	14
5.2	Detailed float layout	17
5.3	Changing the float layout in your document	20
6	List layout	22
6.1	Changing lists	27
7	Sectional heading layout	28
8	Footnote layout	32
8.1	Changing the footnote layout	34
9	Table of Contents layout	36
9.1	Changing the Table of Contents, etc	38
10	Font boxing	40

List of Tables

1	Page spread symbols	3
2	Commands for setting trial page parameters	8
3	Commands for setting trial paragraph parameters	11
4	Commands for setting trial float page parameters	15
5	Commands for setting trial float parameters	18
6	Commands for setting trial list parameters	24
7	Commands for setting trial heading parameters	29
8	Commands for setting trial footnote parameter values	33
9	Commands for setting trial values for ToC parameters	37

List of Figures

1	(Left) The L ^A T _E X book spread; (Right) Spread for many of Gutenberg's books	3
2	Left-hand two-column page layout parameters	6

3	Page layout for this document	7
4	An experimental page layout	10
5	Paragraph parameters	11
6	Paragraphs in this document	12
7	An outset paragraph	12
8	Float and text page parameters	15
9	The standard L ^A T _E X float and text page settings	16
10	Float page layout for decreasing likelihood of float-only pages	16
11	Float parameters	18
12	Float layout with rules	19
13	List parameters	23
14	Layout of an <code>enumerate</code> list	24
15	The layout of the <code>listX</code> environment	26
16	Display heading parameters	29
17	Run-in heading parameters	29
18	Subsubsection heading layout parameters for this document	30
19	Subsubsection layout parameters for a run-in heading	31
20	The footnote parameter layout	32
21	A footnote layout	33
22	A user-specified footnote layout	34
23	Table of Contents entry parameters	36
24	Typical Table of Contents entry for this document	37
25	Table of Contents entry with a large value for <code>\@dotsep</code>	38

1 Introduction

L^AT_EX has several pre-defined styles for the layout of typeset documents [Lam94]. Authors using L^AT_EX 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 L^AT_EX class for typesetting 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 `\drawL` generate a L^AT_EX **picture** of the layout *L*.

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

Layout values Commands of the form `\Lvalues` produce a table 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 a table produced by a `\Lvalues` command is not the same as a table that might be produced by the corresponding `\drawL` command.

User-specified parameter values Commands of the form `\tryP` 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 `\Ctrue` or `\Cfalse` 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 yourself

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. These are controlled by setting a boolean variable.

1. A picture showing a generic layout and its controlling parameters; drawings following the command `\drawparameterstrue` will be of this form.
2. A picture showing an ‘actual’ layout together with a table of the parameter values; drawings following the command `\drawparametersfalse` will be of this form.

Examples of both kinds of layout are shown later.

1.3 Usage

The `\draw...` commands 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 `\draw...` commands. However, if `\setlayoutscale` is called within an environment, such as the `figure` environment, it will only affect the scale factors for succeeding `\draw...` commands in that environment.

The `layouts` package uses a lot of \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.

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

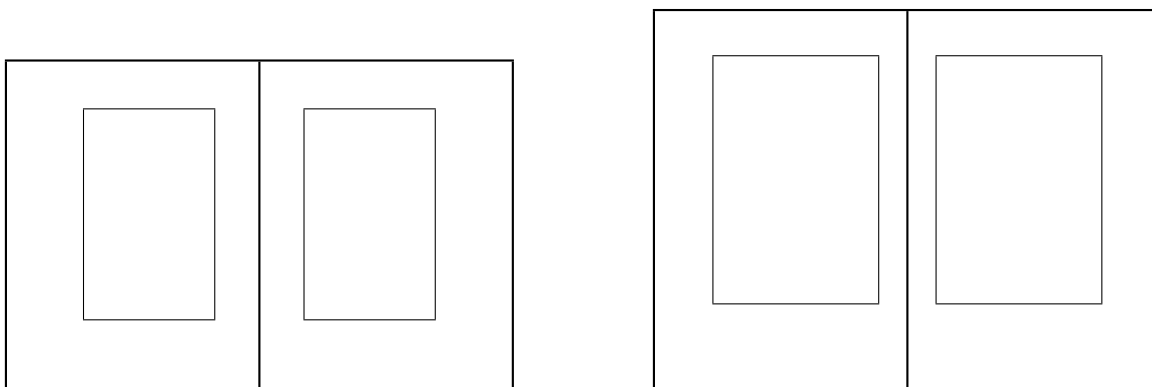


Figure 1: (Left) The L^AT_EX book spread; (Right) Spread for many of Gutenberg's books

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

`\drawspread[$\langle F \rangle$]{ $\langle w \rangle$ }{ $\langle P \rangle$ }{ $\langle T \rangle$ }{ $\langle S \rangle$ }{ $\langle H \rangle$ }{ $\langle E \rangle$ }{ $\langle G \rangle$ }`

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{\pwwidth}
\setlength{\pwwidth}{0.4375\textwidth}
\newlength{\pwwidth}
\setlength{\pwwidth}{0.5\pwwidth}

\begin{figure}
\centering
\begin{minipage}[b]{\pwwidth}
\drawspread{\pwwidth}{1.294}{1.618}{0.176}{1.037}{1.685}{0}
\end{minipage}
\hfill
\begin{minipage}[b]{\pwwidth}
\drawspread{\pwwidth}{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 `\drawspread` 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 `\drawpage` command. The kind of page drawn is specified via two control commands¹.

Right- or left-hand page layouts are specified by the commands `\oddpagelayouttrue/false` (`\oddpagelayouttrue` for an odd-numbered (i.e., right-hand or recto) 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 `\twocolumnlayouttrue/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`.

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

```
\begin{figure}
\oddpagelayoutfalse
\twocolumnlayouttrue
\drawpage
\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, L^AT_EX 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 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. Figure 3 shows the page layout for the document you are now reading. It was produced by the following commands:

```
\begin{figure}
\currentpage
\drawparametersfalse
\drawpage
\caption{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 `\drawparametersfalse`, the actual values of the parameters used to produce the layout are listed at the bottom of the picture. This can be used,

¹Remember that all drawing commands, except for `\drawspread`, are also controlled by the `\drawparameters...` command.

Dashed lines represent $(\backslash\text{hoffset} + 1\text{ inch})$ and $(\backslash\text{voffset} + 1\text{ inch})$ from the top left hand corner of the page.

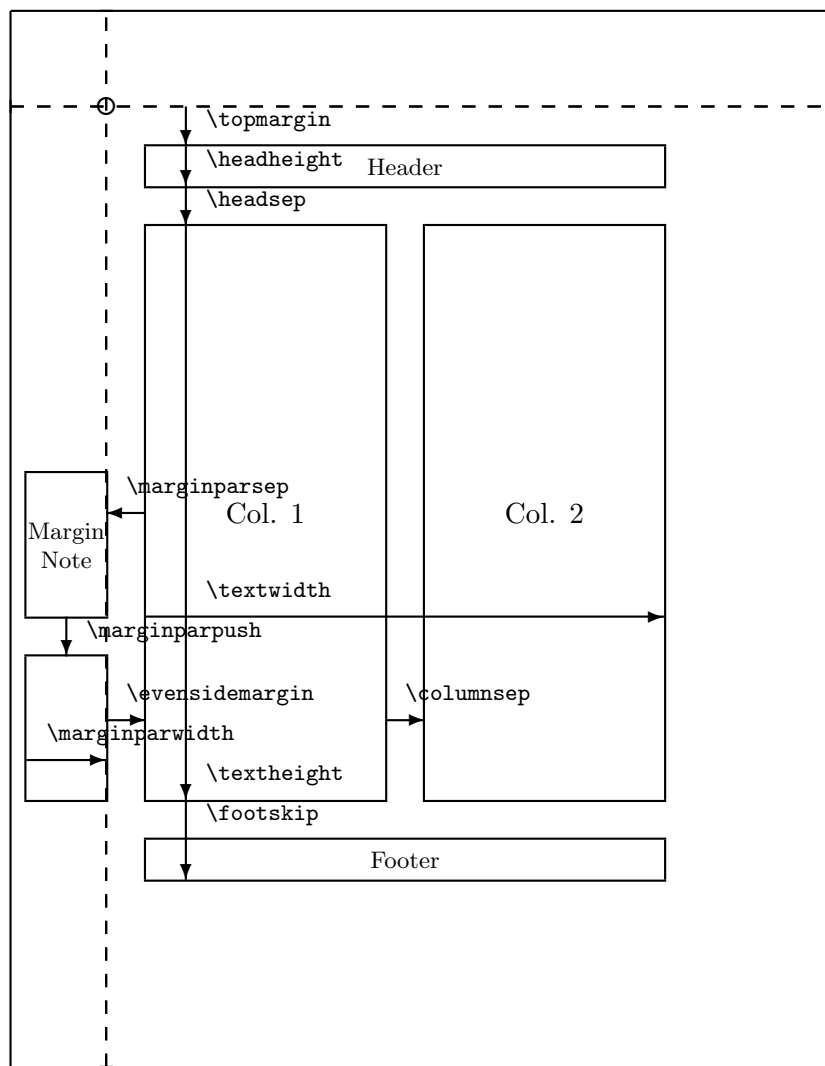
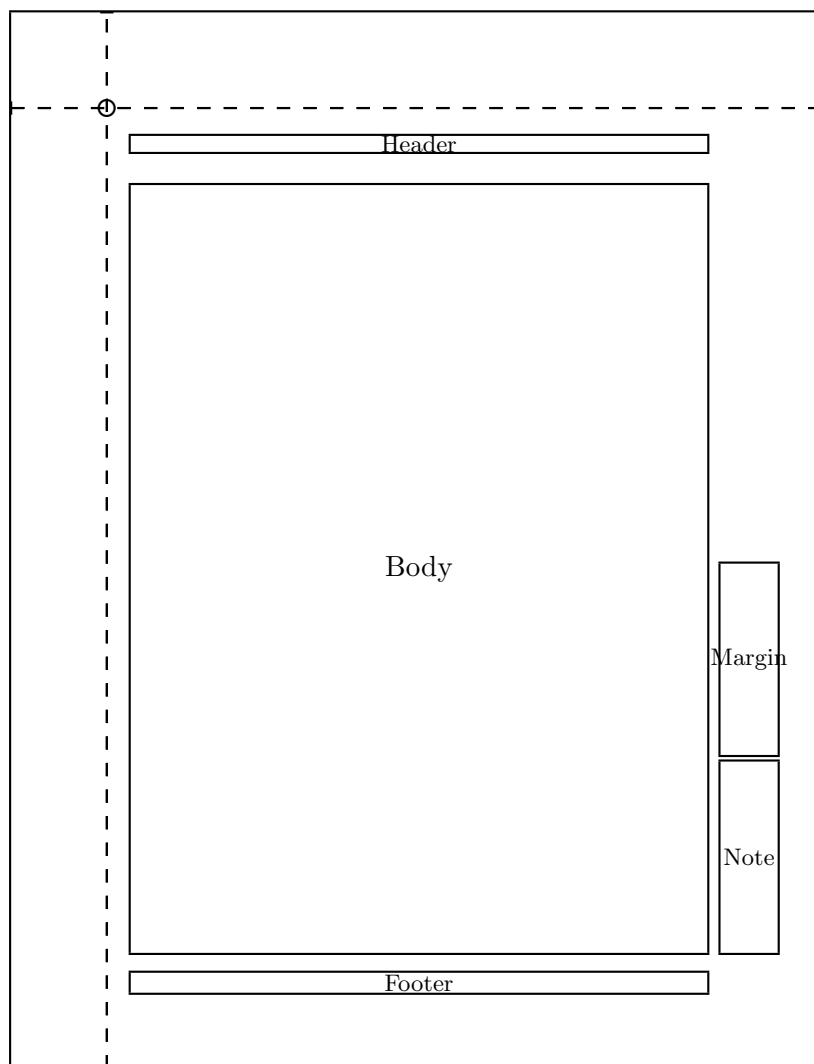


Figure 2: Left-hand two-column page layout parameters

Dashed lines represent (`\hoffset + 1 inch`) and (`\voffset + 1 inch`) from the top left hand corner of the page.



Lengths are to the nearest pt.

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

Figure 3: Page layout for this document

Table 2: Commands for setting trial page parameters

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

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.

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}
\trytextwidth{500pt}
\trycolumnsep{40pt}
\trycolumnseprule{3pt}
\tryhoffset{-0.5in}
\tryvoffset{0.5in}

```

```

\drawparametersfalse
\twocolumnlayouttrue
\drawpage
\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 alright for a portrait page but certainly wrong for the landscape page).

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 L^AT_EX `\setlength` command. For example, to set the width of the text to 3 inches, put this into the preamble:

```

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

```

Actual page layout values.

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

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

Dashed lines represent $(\backslash\text{hoffset} + 1\text{ inch})$ and $(\backslash\text{voffset} + 1\text{ inch})$ from the top left hand corner of the page.

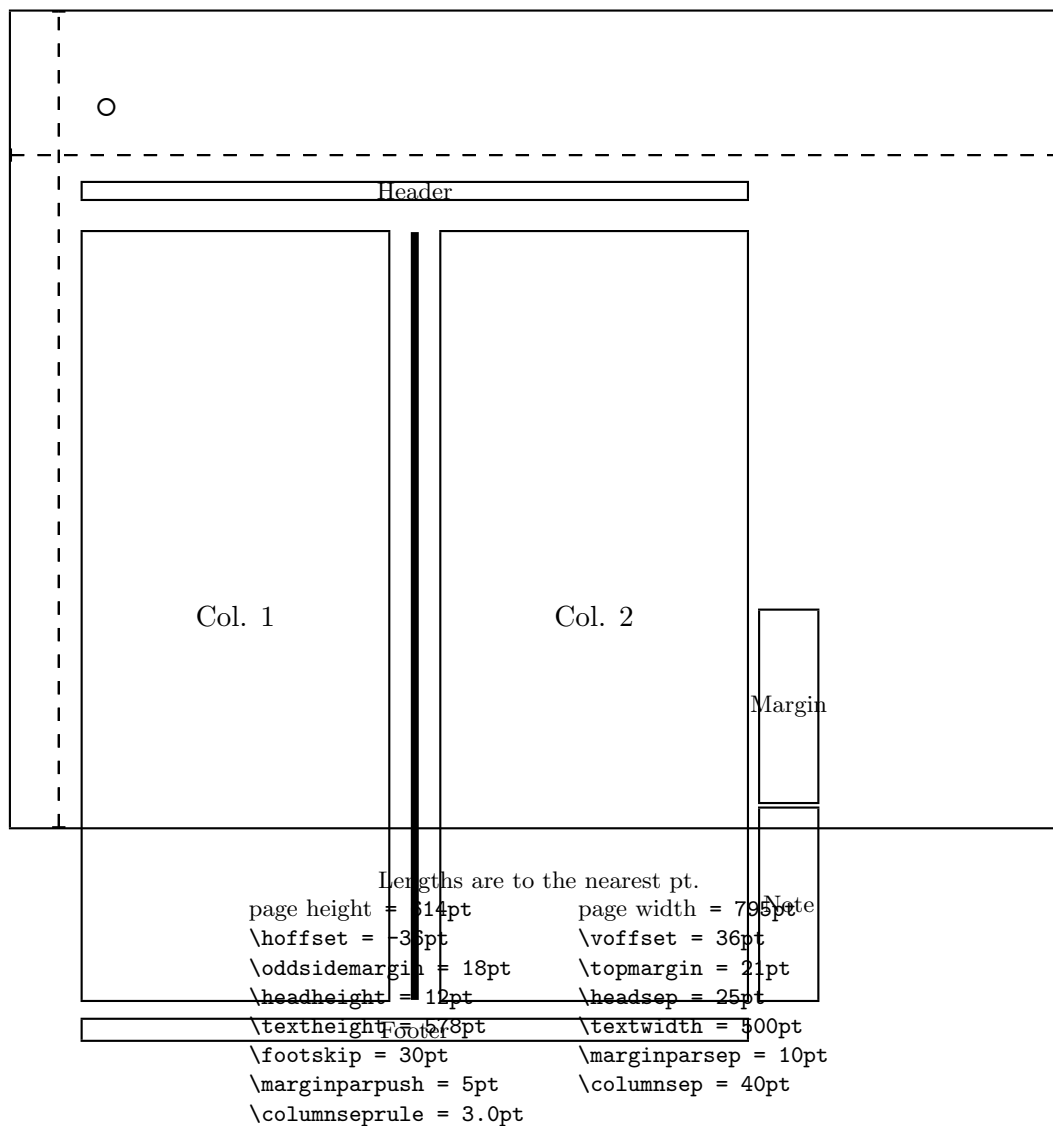


Figure 4: An experimental page layout

4 Paragraph layout

The command `\drawparagraph` is used to visualize the parameters that affect paragraphing. This is illustrated in Figure 5 which was produced by the following code:

```
\begin{figure}
\drawparagraph
\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 3 can be used to set trial values for the relevant parameters. These commands take one argument which must be a length.

Figure 6 shows the paragraphing as used in this document. It was produced with this code:

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

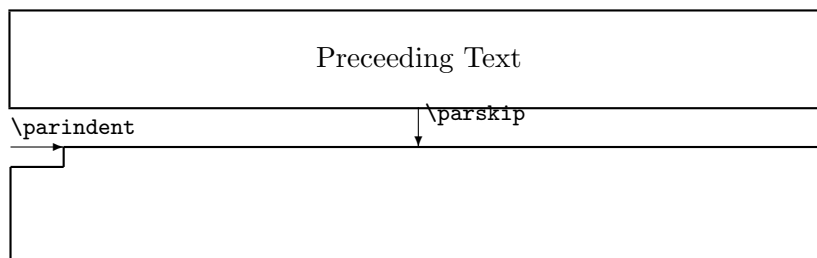
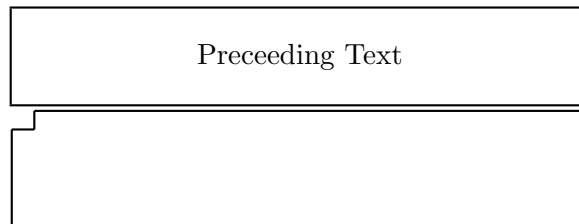


Figure 5: Paragraph parameters

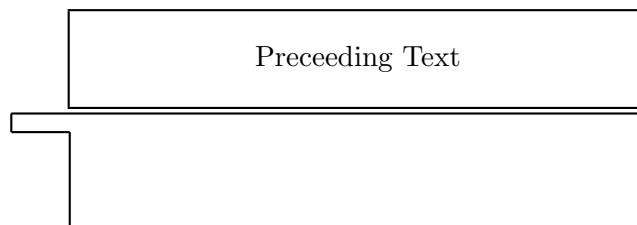
Table 3: Commands for setting trial paragraph parameters

Command	Parameter
<code>\tryparindent</code>	sets the <code>\parindent</code> value
<code>\tryparskip</code>	sets the <code>\parskip</code> value
<code>\tryparlinewidth</code>	sets the <code>\linewidth</code> value
<code>\tryparbaselineskip</code>	sets the <code>\baselineskip</code> value



Lengths are to the nearest pt.
`\parindent = 17pt` `\parskip = 5pt`
`\baselineskip = 14pt` `\linewidth = 434pt`

Figure 6: Paragraphs in this document



Lengths are to the nearest pt.
`\parindent = -44pt` `\parskip = 5pt`
`\baselineskip = 14pt` `\linewidth = 434pt`

Figure 7: An outset paragraph

Interestingly, I found that L^AT_EX 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 7 which was produced from:

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

Actual paragraph layout values.

<code>\parindent</code> = 17.0pt	<code>\parskip</code> = 5.0pt
<code>\baselineskip</code> = 13.6pt	<code>\linewidth</code> = 433.62pt
<code>lem</code> = 10.95003pt	<code>lex</code> = 4.71457pt

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

5 Float layouts

Two sets of commands are provided for displaying the layouts of L^AT_EX 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.

5.1 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 command `\drawfloatpage` is used to visualize the general parameters. These are illustrated in Figure 8 which was produced by the following code:

```
\begin{figure}
\drawfloatpage
\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’ L^AT_EX settings are shown in Figure 9, produced by the code below.

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

The commands listed in Table 4 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 10 illustrates float page settings that increase the likelihood of a float remaining

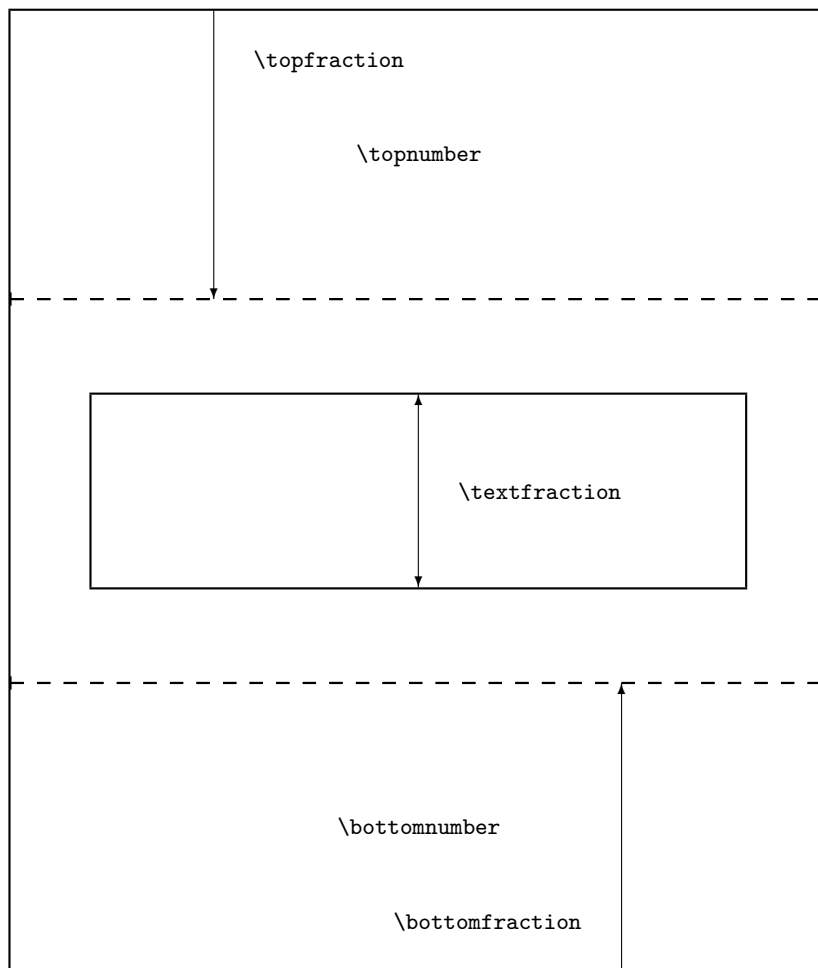


Figure 8: Float and text page parameters

Table 4: Commands for setting trial float page parameters

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

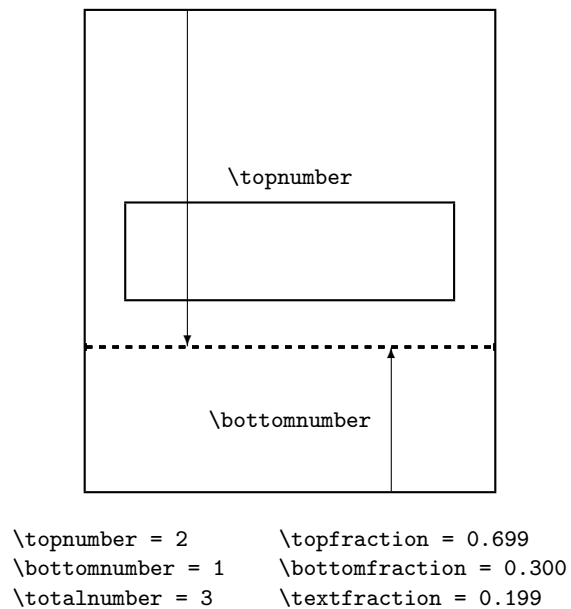


Figure 9: The standard L^AT_EX float and text page settings

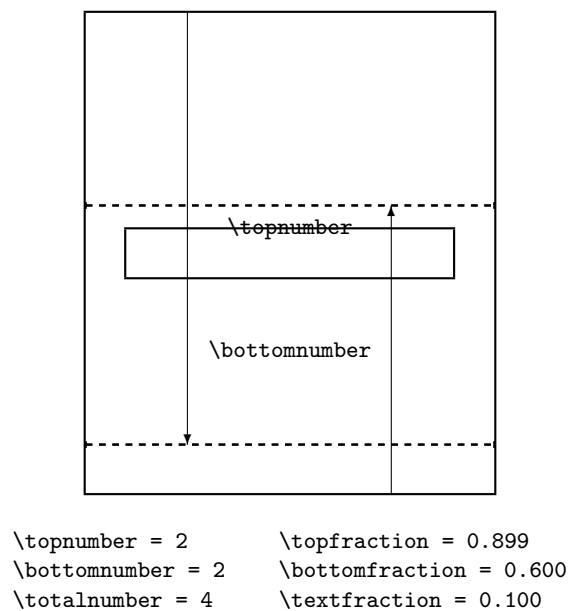


Figure 10: Float page layout for decreasing likelihood of float-only pages

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}
\drawparametersfalse
\setlayoutscale{0.25}
\drawfloatpage
\caption{Float page layout for decreasing likelihood of float-only pages}
\label{fig:fpudf}
\end{figure}
```

5.2 Detailed float layout

The other view of floats concentrates on the spacing between text and floats on a page. The `\drawfloat` command is used to visualize this aspect.

The relevant parameters are shown in Figure 11, produced from the following code:

```
\begin{figure}
\setlayoutscale{0.9}
\drawfloat
\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 5, all of which take a length argument, set trial values for the float parameters.

The following code, used to produce Figure 12, shows the use of some of these commands:

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

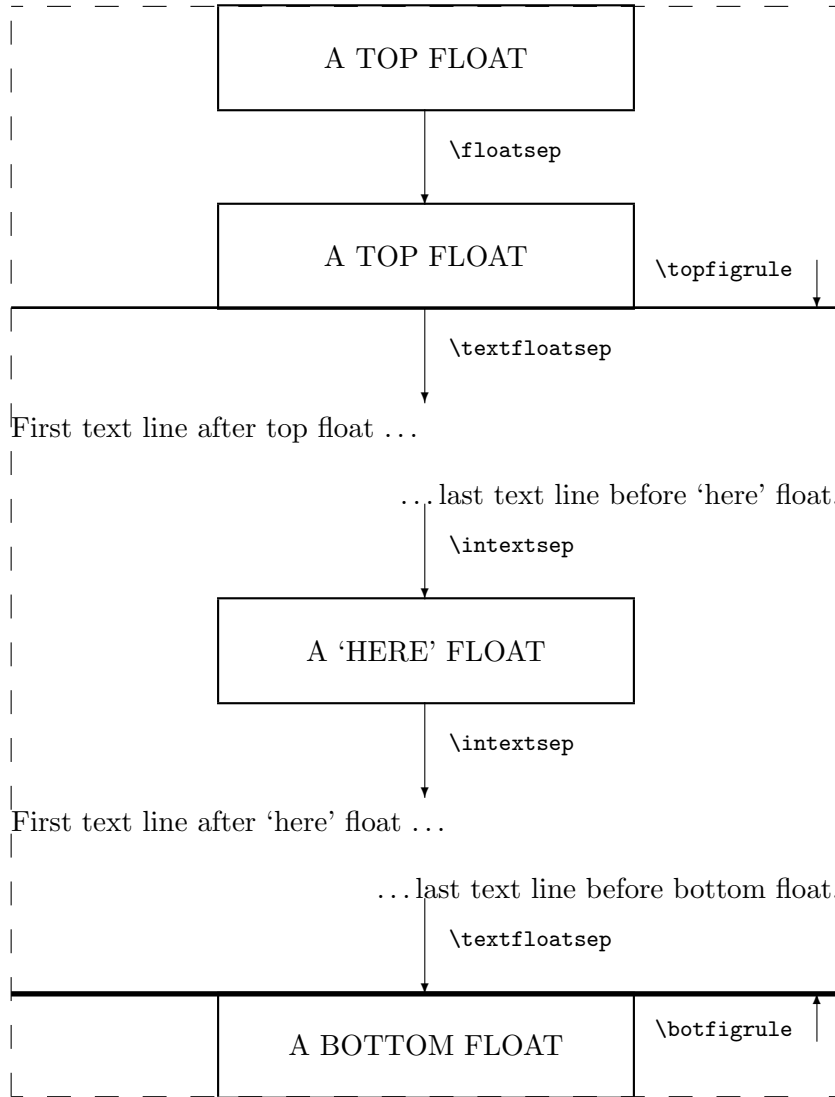
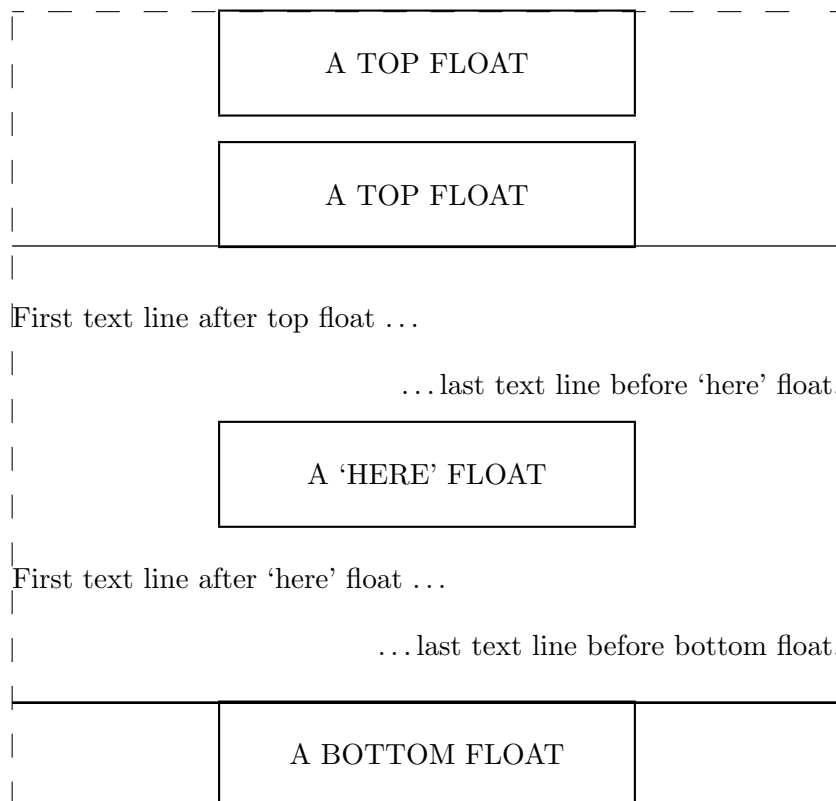


Figure 11: Float parameters

Table 5: Commands for setting trial float parameters

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



Lengths are to the nearest pt.
`\floatsep = 12pt` `\textfloatsep = 20pt`
`\intextsep = 12pt` `topfigrule thickness = 0.5pt`
`botfigrule thickness = 1.0pt`

Figure 12: Float layout with rules

```

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

```

The `\topfigrule` and `\botfigrule` are little known L^AT_EX commands; they are not discussed by Lamport [Lam86, 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 exaggerated 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.

5.3 Changing the float layout in your document

The `\...number` commands are changed with the L^AT_EX `\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.

<code>\floatsep = 12.0pt plus 2.0pt minus 2.0pt</code>	<code>\textfloatsep = 20.0pt plus 2.0pt minus 4.0pt</code>
<code>\intextsep = 12.0pt plus 2.0pt minus 2.0pt</code>	
topfig rule thickness = 0pt ??	botfig rule thickness = 0pt ??
<code>\topnumber = 2</code>	<code>\topfraction = 0.9</code>
<code>\bottomnumber = 2</code>	<code>\bottomfraction = 0.6</code>
<code>\totalnumber = 4</code>	<code>\textfraction = 0.1</code>
<code>\dblfloatsep = 12.0pt plus 2.0pt minus 2.0pt</code>	<code>\dbltextfloatsep = 20.0pt plus 2.0pt minus 4.0pt</code>
<code>\dbltopnumber = 2</code>	<code>\dbltopfraction = .7</code>
<code>\dblfloatpagefraction = .5</code>	<code>\floatpagefraction = .5</code>
<code>1em = 10.95003pt</code>	<code>1ex = 4.71457pt</code>

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).

6 List layout

The command `\drawlist`, as its name suggests, displays the layout of list environments. The list parameters are shown in Figure 13, which was produced by the following code:

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

The list layout may be controlled by the `\listasparatrue/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 14 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}
    \drawparametersfalse
    \drawlist
    \caption{Layout of an \texttt{enumerate} list} \label{fig:lstenum}
    \end{figure}
\end{enumerate}
```

1. Figure 14 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 6 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 15 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 L^AT_EX lists. Note that the

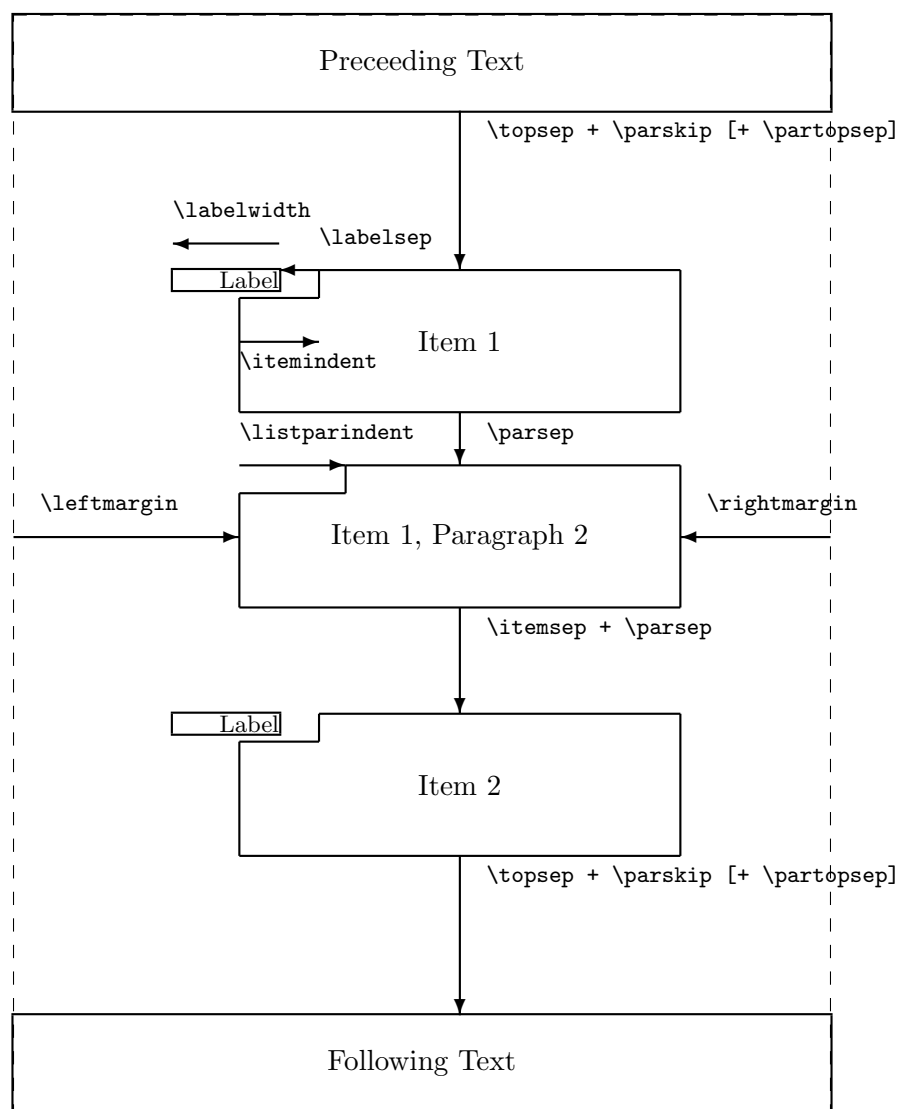
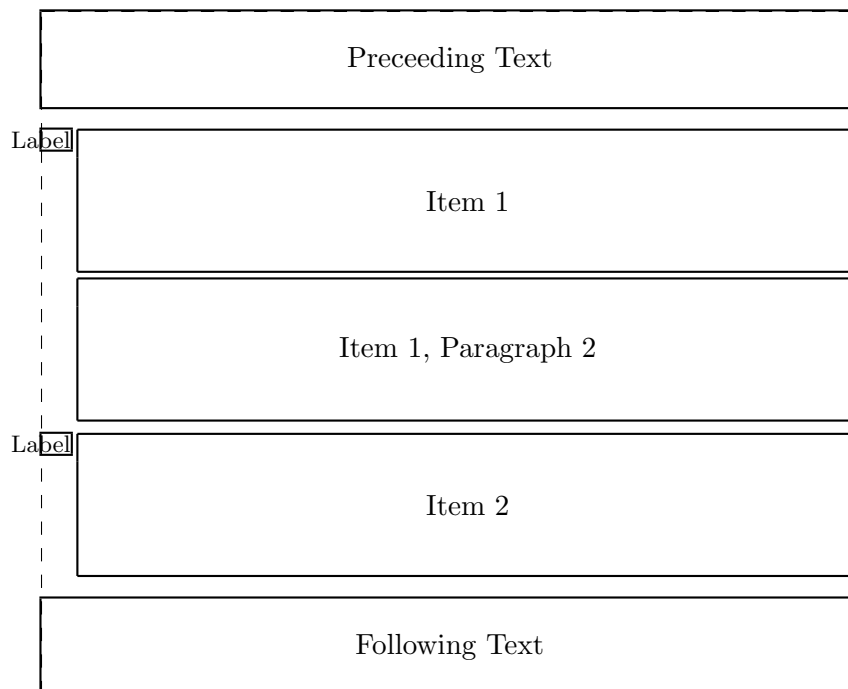


Figure 13: List parameters



Lengths are to the nearest pt.

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

Figure 14: Layout of an `enumerate` list

Table 6: Commands for setting trial list parameters

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

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

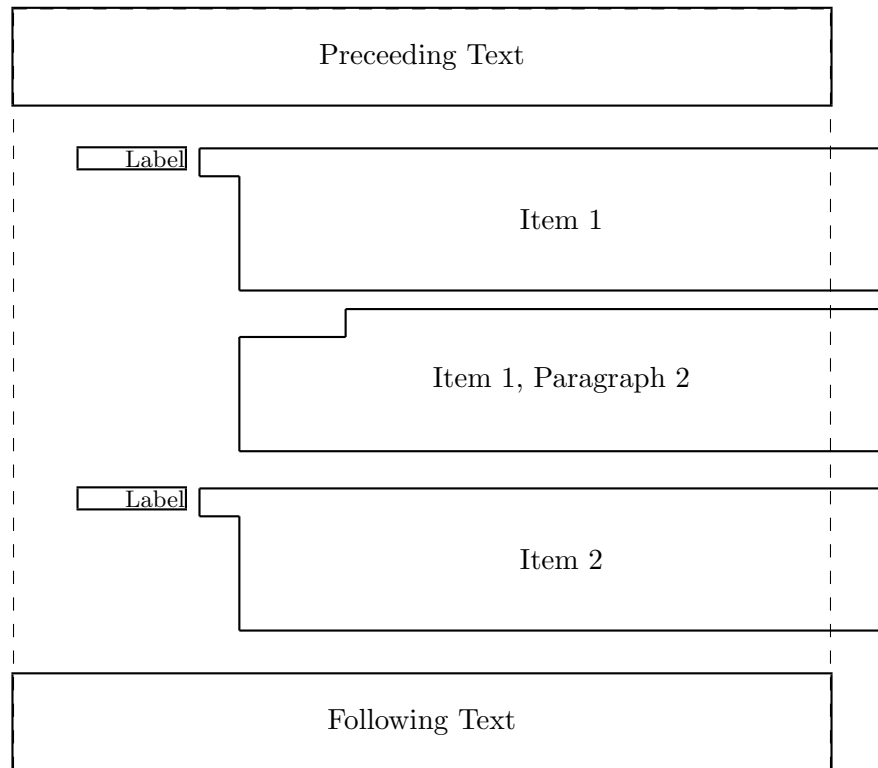
No claim is made that there is anything aesthetic about the design of the list. In fact, I think that it is pretty awful. It is merely provided as an example of a non-standard list and so that the `\drawlist` command can be shown off.

ListX, second item: Now we will draw the layout of this list from within itself (see Figure 15 for the result). The code used is:

```
\currentlist
\begin{figure}
\drawparametersfalse
\drawlist
\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.

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

Figure 15: The layout of the `listX` environment

In L^AT_EX 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. {\em 1996 World Gnus Almanac.}
\currentlist
\begin{figure}
\drawparametersfalse
\drawlist
\caption{Bibliography list}
\end{figure}
\end{thebibliography}
```

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

6.1 Changing lists

Many of L^AT_EX’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.

<code>\leftmargin = 27.37506pt</code>	<code>\rightmargin = 27.37506pt</code>
<code>\itemindent = 0.0pt</code>	<code>\labelwidth = 21.90005pt</code>
<code>\labelsep = 5.475pt</code>	<code>\listparindent = 0.0pt</code>
<code>\topsep = 9.0pt plus 3.0pt minus 5.0pt</code>	<code>\parskip = 4.5pt plus 2.0pt minus 1.0pt</code>
<code>\partopsep = 3.0pt plus 1.0pt minus 1.0pt</code>	<code>\parsep = 4.5pt plus 2.0pt minus 1.0pt</code>
<code>\itemsep = 4.5pt plus 2.0pt minus 1.0pt</code>	
<code>lem = 10.95003pt</code>	<code>lex = 4.71457pt</code>

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}
\listvalues
\end{quote}
```

7 Sectional heading layout

In L^AT_EX a few headings, like `\part` and `\chapter` are defined using special definition code. The majority, though, are defined via the internal L^AT_EX `\@startsection` command. This command takes 6 arguments:

`\@startsection{<name>}{<level>}{<indent>}{<beforekip>}{<afterskip>}{}`.

The command `\drawheading{}` draws a picture of the layout for sectional headings based on the `\@startsection` command. Unlike the other `\draw...` commands it takes one parameter, which is the specification of the size and/or font of the heading. For example, `\drawheading{\large\itshape}`. When `\drawparameterstrue` is in effect then 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 `\drawheading` is called 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 16 and 17 show the parameters of the two kinds of headings. They were produced by the following code:

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

```
\begin{figure}
\setlayoutscale{1}
\runinheadtrue
\drawheading{ }
\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 7 can be used to explicitly set heading parameters. Each of these commands takes a length for its parameter value.

The regular reader of L^AT_EX documents will have noticed that the `\subsubsection`

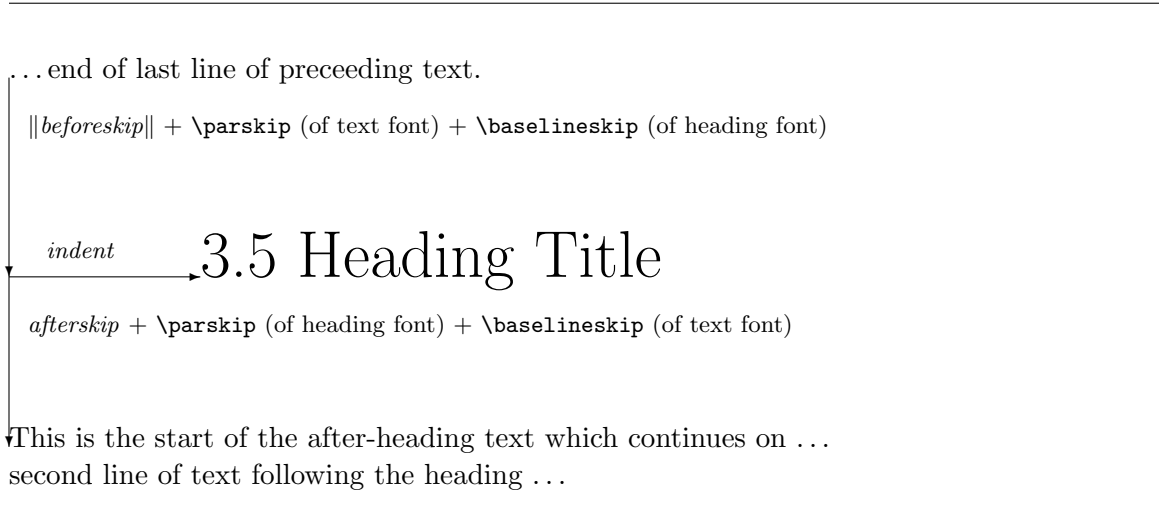


Figure 16: Display heading parameters

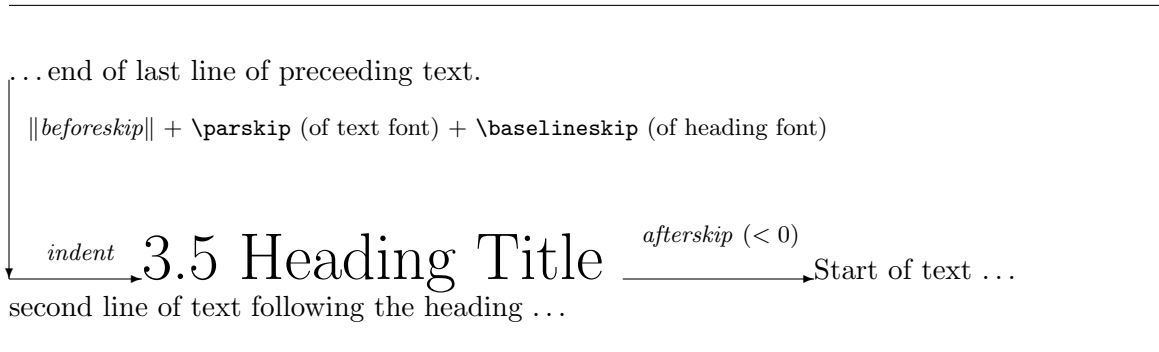


Figure 17: Run-in heading parameters

Table 7: Commands for setting trial heading parameters

Command	Parameter
$\backslash trybeforeskip$	sets the <i>beforeskip</i> value
$\backslash tryafterskip$	sets the <i>afterskip</i> value
$\backslash tryindent$	sets the <i>indent</i> 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.	
<i>beforeskip</i> = -14pt	<i>afterskip</i> = 7pt
<i>indent</i> = 0pt	(heading font) \baselineskip = 0pt
\parskip = 5pt	heading font = \large

Figure 18: Subsubsection heading layout parameters for this document

headings in this manual do not conform to the usual L^AT_EX 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 18 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}
\drawparametersfalse
\drawheading{\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 19: 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 19.

8 Footnote layout

Footnote layouts are produced by the command `\drawfootnote`. The relevant footnote parameters are shown in Figure 20, which was produced by the following code:

```
\begin{figure}
\setlayoutscale{0.4}
\drawfootnote
\caption{The footnote parameter layout} \label{fig:fp}
\end{figure}
```

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

```
\begin{figure}
\currentfootnote
\drawparametersfalse
\setlayoutscale{0.4}
\drawfootnote
\caption{The current footnote layout}\label{fig:ftry}
```

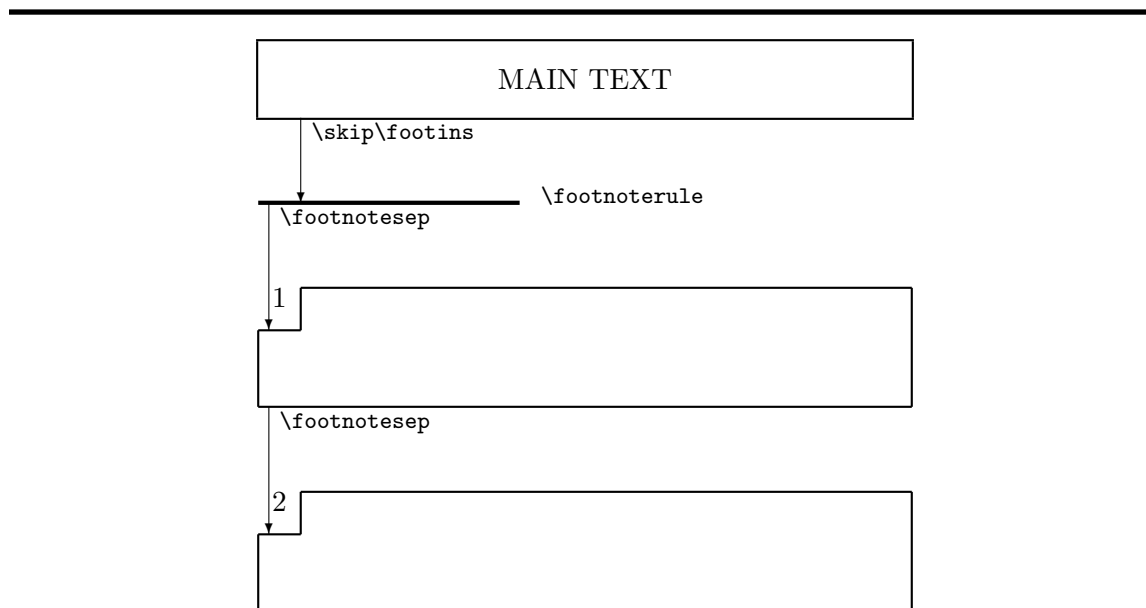
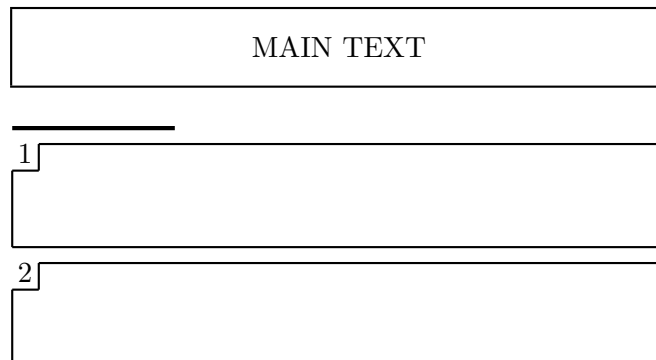


Figure 20: The footnote parameter layout



Lengths are to the nearest pt.
`\footins = 10pt` `\footnotesep = 8pt`
`\baselineskip = 10pt` note separation = 40pt
rule thickness = 0.4pt
rule length = 0.25 times the `\textwidth`

Figure 21: A footnote layout

Table 8: Commands for setting trial footnote parameter values

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

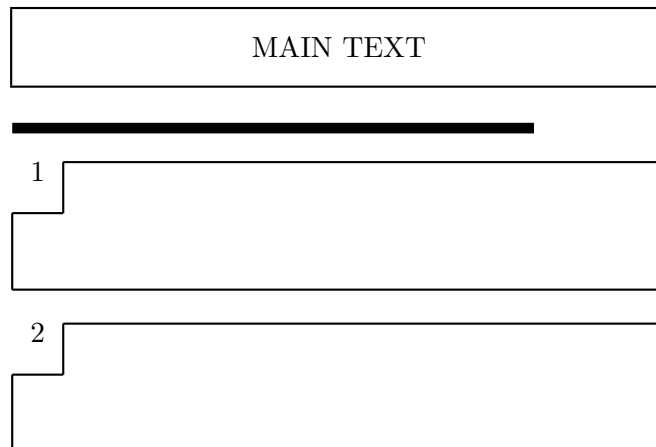
`\end{figure}`

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 8. 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 22, produced from the following code, shows an experimental layout for footnotes.

```
\begin{figure}
\currentfootnote
\tryfootins{10pt}
\tryfootnotesep{15pt}
```



Lengths are to the nearest pt.
`\footins = 10pt` `\footnotesep = 15pt`
`\baselineskip = 20pt` note separation = 80pt
`rule thickness = 1.0pt`
`rule length = 0.8 times the \textwidth`

Figure 22: A user-specified footnote layout

```
\tryfootnotebaseline{20pt}
\tryfootruleheight{1pt}
\tryfootrulefrac{0.8}
\drawparametersfalse
\setlayoutscale{0.4}
\drawfootnote
\caption{A user-specified footnote layout}
\label{fig:fudf}
\end{figure}
```

8.1 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.

<code>\footins</code> = 10.0pt plus 4.0pt minus 2.0pt	<code>\footnotesep</code> = 7.7pt
rule thickness = 0.4pt ??	rule length = 173.44534pt ??
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).

9 Table of Contents layout

The format of an entry in the Table of Contents (ToC) is usually specified by the internal L^AT_EX `\@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 `\drawtoc` command is used for visualizing the layout of section titles in a *Table of Contents* listing based on the `\@dottedtocline` command, as shown in Figure 23. This was produced by the following code:

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

Figure 24 was produced by the following code:²

```
\begin{figure}
\setlayoutscale{0.8}
\currenttoc
\drawparametersfalse
\drawtoc
```

²The drawing of the dotted leader is not completely accurate due to scaling and rounding within the drawing algorithm.

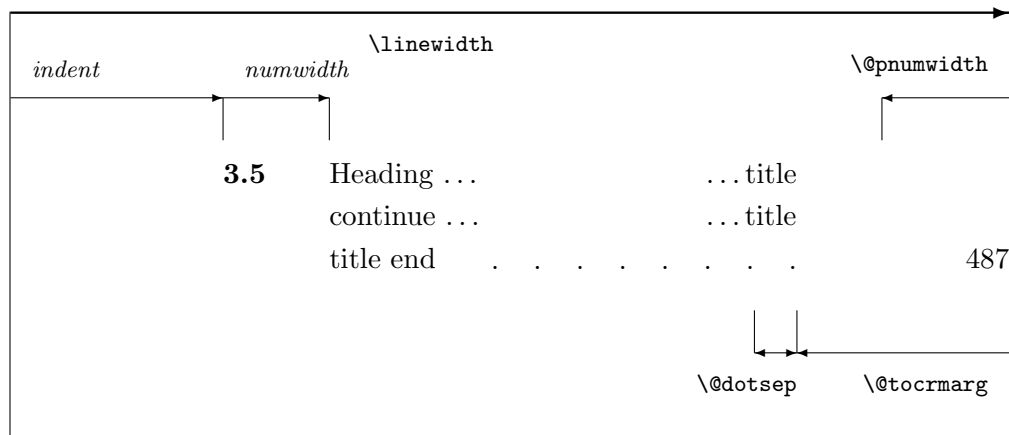


Figure 23: Table of Contents entry parameters

3.5 Headingtitle	
continuetitle	
title end		487

Lengths are to the nearest pt.
`indent` = 16pt `numwidth` = 25pt
`\@tocrmarg` = 28pt `\@pnumwidth` = 17pt
`\@dotsep` = 4.5

Figure 24: Typical Table of Contents entry for this document

Table 9: Commands for setting trial values for ToC parameters

Command	Parameter
<code>\trytocindent</code>	sets the <i>indent</i> value
<code>\trytocnumwidth</code>	sets the <i>numwidth</i> value
<code>\trytoclinewidth</code>	sets the <code>\linewidth</code> value
<code>\trytocrmarg</code>	sets the <code>\@tocrmarg</code> value
<code>\trytocpnumwidth</code>	sets the <code>\@pnumwidth</code> value
<code>\trytocdotsep</code>	sets the <code>\@dotsep</code> value

```
\caption{Typical Table of Contents entry for this document} \label{fig:thistoc}
\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 9. 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 seperation between dots in the leader between the sectional title and the page number. The parameter is the seperation value in *mu* (math units).³

Figure 25, 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

³There are *18mu* units to *1em* unit.

3.5 Headingtitle	
continuetitle	
title end		487

Lengths are to the nearest pt.
`indent = 16pt` `numwidth = 25pt`
`\@tocrmarg = 28pt` `\@pnumwidth = 17pt`
`\@dotsep = 1000`

Figure 25: Table of Contents entry with a large value for `\@dotsep`

the page number.

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

9.1 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.

<code>\@tocrmarg</code> = 2.55em	<code>\@pnumwidth</code> = 1.55em
<code>\@dotsep</code> = 4.5	<code>indent</code> = 1.5em ??
<code>numwidth</code> = 2.3em ??	
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).

10 Font boxing

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

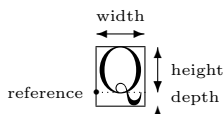
The `\drawfontframe{⟨text⟩}` produces a drawing of `⟨text⟩` together with its surrounding box drawn. A bullet is placed at the position of the reference point and a horizontal dotted line is drawn along the baseline. The command can be used in normal mode or can be `\put` in a picture environment.  The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

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 `\drawparameterstrue` 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 `\drawparametersfalse`.



This time with `\drawparameterstrue` 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.
- [Lam86] Leslie Lamport. *LaTeX: A Document Preparation System*. Addison-Wesley Publishing Company, 1986.
- [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.

Index

\@dotsep 37–38
\@dottedtocline 36, 39
\@pnumwidth 37–38
\@startsection 28, 30, 39
\@tocrmarg 37–38

B

\baselineskip 2, 8, 11, 32–33
\botfigrule 18, 20
\bottomfraction 15
\bottomnumber 15

C

center (environment) 2
\chapter 28
classes.dtx (file) 27, 34, 38
\columnsep 8
\columnseprule 8
\currentfloat 17
\currentfloatpage 14
\currentfootnote 32
\currentheading 28
\currentlist 22, 27
\currentpage 5, 8
\currentparagraph 11
\currenttoc 37

D

\dblfloatpagefraction 21
\dblfloatsep 21
\dbltextfloatsep 21
\dbltopfraction 21
\dbltopnumber 21
description (environment) 27
\drawspread 3–4
\drawfloat 17
\drawfloatpage 14
\drawfontframe 40
\drawfontframelabel 40
\drawfootnote 32
\drawheading 28

\drawlist 22, 25
\drawpage 5
\drawparagraph 11
\drawparametersfalse 2, 5
\drawparameterstrue 2, 28
\drawtoc 36

E

enumerate (environment) 22, 27
\evensidemargin 8

F

figure (environment) 2, 14
\floatsep 18
\floatvalues 21
\footins 33
\footnoterule 20, 33
\footnotesep 33–34
\footnotevalues 35
\footskip 8

H

\headheight 8
\headsep 8
\hoffset 8

I

\intextsep 18
\itemindent 24
itemize (environment) 27
\itemsep 24

L

\labelsep 24
\labelwidth 24
layout.sty (file) 1
layoutsbox (save box) 40
\leftmargin 24
\linewidth 11, 37
list (environment) 22, 27
\listasparafalse 22

`\listasparatrue` 22
`\listparindent` 24
`\listvalues` 27

M

`\makeatletter` 9, 20, 30
`\makeatother` 9, 20, 30
`\marginparpush` 8
`\marginparsep` 8
`\marginparwidth` 8

O

`\oddpagelayoutfalse` 5
`\oddpagelayouttrue` 5
`\oddsidemargin` 8

P

`\pagevalues` 9
`\paperheight` 8
`\paperwidth` 8
`\paragraphvalues` 13
`\parindent` 11, 13
`\parsep` 24
`\parskip` 2, 11, 24
`\part` 28
`\partopsep` 24
`picture` (environment) 2

Q

`quotation` (environment) 27

R

`\renewcommand` 20, 38
`\rightmargin` 24
`\rule` 20
`\runinheadfalse` 28
`\runinheadtrue` 28

S

`\setcounter` 20
`\setlayoutscale` 2, 4
`\setlength` 9, 21, 34
`\setuplayouts` 1–2
`\skip` 33

`\spacefactor` 9
`\subsubsection` 28

T

`table` (environment) 14
`tabular` (environment) 2
`\textfloatsep` 18, 20
`\textfraction` 15
`\textheight` 8
`\textwidth` 8, 33
`thebibliography` (environment) 27
`tocdepth` (counter) 36
`\tocvalues` 39
`\topfigrule` 18, 20
`\topfraction` 15
`\topmargin` 8
`\topnumber` 15
`\topsep` 24
`\totalnumber` 15
`\tryafterskip` 29
`\trybeforeskip` 29
`\trybotfigrule` 18
`\trybottomfraction` 15
`\trybottomnumber` 15
`\trycolumnsep` 8
`\trycolumnseprule` 8
`\tryevensidemargin` 8
`\tryfloatsep` 18
`\tryfootins` 33
`\tryfootnotebaseline` 33
`\tryfootnotesep` 33
`\tryfootrulefrac` 33
`\tryfootruleheight` 33
`\tryfootskip` 8
`\tryheadheight` 8
`\tryheadsep` 8
`\tryhoffset` 8
`\tryindent` 29
`\tryintextsep` 18
`\tryitemindent` 24
`\tryitemsep` 24
`\trylabelsep` 24
`\trylabelwidth` 24

`\tryleftmargin` 24
`\trylistparindent` 24
`\trymarginparpush` 8
`\trymarginparsep` 8
`\trymarginparwidth` 8
`\tryoddsidemargin` 8
`\trypaperheight` 8
`\trypaperwidth` 8
`\tryparbaselineskip` 11
`\tryparindent` 11
`\tryparlinewidth` 11
`\tryparsep` 24
`\tryparskip` 11, 24
`\trypartopsep` 24
`\tryrightmargin` 24
`\trytextfloatsep` 18
`\trytextfraction` 15
`\trytextheight` 8
`\trytextwidth` 8
`\trytocdotsep` 37
`\trytocindent` 37
`\trytoclinewidth` 37
`\trytocnumwidth` 37
`\trytocpnumwidth` 37
`\trytocrmarg` 37
`\trytopfigrule` 18
`\trytopfraction` 15
`\trytopmargin` 8
`\trytopnumber` 15
`\trytopsep` 24
`\trytotalnumber` 15
`\tryvoffset` 8
`\twocolumnlayoutfalse` 5
`\twocolumnlayouttrue` 5

V

`\voffset` 8