

# Standard Letter Document Class for L<sup>A</sup>T<sub>E</sub>X version 2e

Leslie Lamport and Frank Mittelbach and Rainer Schöpf

March 6, 2020

This file is maintained by the L<sup>A</sup>T<sub>E</sub>X Project team.  
Bug reports can be opened (category `latex`) at  
<https://latex-project.org/bugs.html>.

## Contents

<b>1</b>	<b>Initial Code</b>	<b>2</b>
1.1	Setting Paper Sizes . . . . .	2
1.2	Choosing the type size . . . . .	3
1.3	Two-side or one-side printing . . . . .	3
1.4	Draft option . . . . .	3
1.5	Equation numbering on the left . . . . .	3
1.6	Flush left displays . . . . .	4
<b>2</b>	<b>Executing Options</b>	<b>4</b>
<b>3</b>	<b>Loading Packages</b>	<b>4</b>
<b>4</b>	<b>Document Layout</b>	<b>4</b>
4.1	Paragraphing . . . . .	4
4.2	Page Layout . . . . .	5
4.2.1	Vertical spacing . . . . .	5
4.2.2	The dimension of text . . . . .	5
4.2.3	Margins . . . . .	6
4.2.4	Footnotes . . . . .	6
4.3	Page Styles . . . . .	7
4.3.1	Marking conventions . . . . .	7
4.3.2	Defining the page styles . . . . .	7
<b>5</b>	<b>Document Markup</b>	<b>8</b>
5.1	Global Declarations . . . . .	8
5.2	The generic letter commands . . . . .	9
5.2.1	Page breaking control . . . . .	10
5.3	Customizing the labels . . . . .	13
5.4	Lists . . . . .	14

5.4.1	General List Parameters	14
5.4.2	Enumerate	16
5.4.3	Itemize	16
5.4.4	Description	16
5.5	Defining new environments	17
5.5.1	Verse	17
5.5.2	Quotation	17
5.5.3	Quote	17
5.5.4	Theorem	17
5.6	Setting parameters for existing environments	17
5.6.1	Array and tabular	17
5.6.2	Tabbing	18
5.6.3	Minipage	18
5.6.4	Framed boxes	18
5.6.5	Equation and eqnarray	18
5.7	Font changing	19
5.8	Footnotes	19
5.9	Words	20
5.10	Date	20
5.11	Two column mode	21
5.12	The page style	21
5.13	Single or double sided printing	21

## 1 Initial Code

In this part we define a few commands that are used later on.

`\@ptsize` This control sequence is used to store the second digit of the pointsize we are typesetting in. So, normally, it's value is one of 0, 1 or 2.

```
1 \let\letter
2 \newcommand\@ptsize{}
```

### 1.1 Setting Paper Sizes

The variables `\paperwidth` and `\paperheight` should reflect the physical paper size after trimming. For desk printer output this is usually the real paper size since there is no post-processing.

```
3 \DeclareOption{a4paper}
4   {\setlength\paperheight {297mm}%
5    \setlength\paperwidth  {210mm}}
6 \DeclareOption{a5paper}
7   {\setlength\paperheight {210mm}%
8    \setlength\paperwidth  {148mm}}
9 \DeclareOption{b5paper}
10  {\setlength\paperheight {250mm}%
11   \setlength\paperwidth  {176mm}}
12 \DeclareOption{letterpaper}
13   {\setlength\paperheight {11in}%
14    \setlength\paperwidth  {8.5in}}
15 \DeclareOption{legalpaper}
```

```

16   {\setlength\paperheight {14in}%
17    \setlength\paperwidth  {8.5in}}
18 \DeclareOption{executivepaper}
19   {\setlength\paperheight {10.5in}%
20    \setlength\paperwidth  {7.25in}}

```

The option `landscape` switches the values of `\paperheight` and `\paperwidth`, assuming the dimensions were given for portrait paper.

```

21 \DeclareOption{landscape}
22   {\setlength\@tempdima  {\paperheight}%
23    \setlength\paperheight {\paperwidth}%
24    \setlength\paperwidth  {\@tempdima}}

```

## 1.2 Choosing the type size

The type size options are handled by defining `\@ptsize` to contain the last digit of the size in question and branching on `\ifcase` statements. This is done for historical reasons to stay compatible with other packages that use the `\@ptsize` variable to select special actions. It makes the declarations of size options less than 10pt difficult, although one can probably use 9 and 8 assuming that a class won't define both 8pt and 18pt options.

```

25 \DeclareOption{10pt}{\renewcommand\@ptsize{0}}
26 \DeclareOption{11pt}{\renewcommand\@ptsize{1}}
27 \DeclareOption{12pt}{\renewcommand\@ptsize{2}}

```

## 1.3 Two-side or one-side printing

Two-sided printing was not supported in the L<sup>A</sup>T<sub>E</sub>X 2.09 version of this document class.

```

28 \if@compatibility
29   \DeclareOption{twoside}{\@latexerr{No 'twoside' layout for letters}%
30                               \@eha}
31 \else
32   \DeclareOption{twoside}{\@twoside true \@mparswitch true}
33 \fi
34 \DeclareOption{oneside}{\@twoside false \@mparswitch false}

```

## 1.4 Draft option

If the user requests `draft` we show any overfull boxes. We could probably add some more interesting stuff to this option.

```

35 \DeclareOption{draft}{\setlength\overfullrule{5pt}}
36 \DeclareOption{final}{\setlength\overfullrule{0pt}}

```

## 1.5 Equation numbering on the left

The option `leqno` can be used to get the equation numbers on the left side of the equation.

```

37 \DeclareOption{leqno}{\input{leqno.clo}}

```

## 1.6 Flush left displays

The option `fleqn` redefines the displayed math environments in such a way that they come out flush left, with an indentation of `\mathindent` from the prevailing left margin.

```
38 \DeclareOption{fleqn}{\input{fleqn.clo}}
```

## 2 Executing Options

Here we execute the default options to initialize certain variables.

```
39 \ExecuteOptions{letterpaper,10pt,oneside,onecolumn,final}
```

The `\ProcessOptions` command causes the execution of the code for every option `foo` which is declared and for which the user typed the `foo` option in his `\documentclass` command. For every option `bar` he typed, which is not declared, the option is assumed to be a global option. All options will be passed as document options to any `\usepackage` command in the document preamble.

```
40 \ProcessOptions
```

Now that all the options have been executed we can load the chosen class option file that contains all size dependent code.

```
41 \input{size1\@ptsize.clo}
```

## 3 Loading Packages

The standard class files do not load additional packages.

## 4 Document Layout

In this section we are finally dealing with the nasty typographical details.

### 4.1 Paragraphing

`\lineskip` These parameters control T<sub>E</sub>X's behavior when two lines tend to come too close together.  
`\normallineskip`

```
42 \setlength\lineskip{1\p@}
```

```
43 \setlength\normallineskip{1\p@}
```

`\baselinestretch` This is used as a multiplier for `\baselineskip`. The default is to *not* stretch the baselines.

```
44 \renewcommand\baselinestretch{}
```

`\parskip` `\parskip` gives extra vertical space between paragraphs and `\parindent` is the width of the paragraph indentation. Letters are typeset without paragraph indentation.

```
45 \setlength\parskip{0.7em}
```

```
46 \setlength\parindent{0\p@}
```

`\@lowpenalty` The commands `\nopagebreak` and `\nolinebreak` put in penalties to discourage these breaks at the point they are put in. They use `\@lowpenalty`, `\@medpenalty` or `\@highpenalty`, dependent on their argument.

```
47 \@lowpenalty 51
48 \@medpenalty 151
49 \@highpenalty 301
```

`\clubpenalty` These penalties are used to discourage club and widow lines. Because we use their default values we only show them here, commented out.

```
50 % \clubpenalty 150
51 % \widowpenalty 150
```

`\displaywidowpenalty` Discourage (but not so much) widows in front of a math display and forbid breaking directly in front of a display. Allow break after a display without a penalty.  
`\predisplaypenalty`  
`\postdisplaypenalty` Again the default values are used, therefore we only show them here.

```
52 % \displaywidowpenalty 50
53 % \predisplaypenalty 10000
54 % \postdisplaypenalty 0
```

`\interlinepenalty` Allow the breaking of a page in the middle of a paragraph.

```
55 % \interlinepenalty 0
```

`\brokenpenalty` We allow the breaking of a page after a hyphenated line.

```
56 % \brokenpenalty 0
```

## 4.2 Page Layout

All margin dimensions are measured from a point one inch from the top and lefthand side of the page.

### 4.2.1 Vertical spacing

`\headheight` The `\headheight` is the height of the box that will contain the running head. The `\headsep` is the distance between the bottom of the running head and the top of the text. `\topskip` is the `\baselineskip` for the first line on a page, its value depends on the size option that was specified. Therefore it is initialized in one of the `size1x.clo` files.

```
57 \setlength\headheight{12\p@}
58 \setlength\headsep {45\p@}
```

`\footskip` The distance from the baseline of the box which contains the running footer to the baseline of last line of text is controlled by the `\footskip`. Bottom of page:

```
59 \setlength\footskip{25\p@}
```

### 4.2.2 The dimension of text

`\textwidth` When we are in compatibility mode we have to make sure that the dimensions of the printed area are not different from what the user was used to see.

```
60 \if@compatibility
61 \setlength\textwidth{365\p@}
```

`\textheight` Now that we have computed the width of the text, we have to take care of the height. The `\textheight` is the height of text (including footnotes and figures, excluding running head and foot).

```
62 \setlength\textheight{505\p@}
63 \fi
```

In native mode we use the dimensions as they were computed by one of the `xxpt` options, together with one of the `..paper` options.

### 4.2.3 Margins

`\oddsidemargin` The values of `\oddsidemargin` and `\evensidemargin` are computed from those of `\paperwidth` and `\textwidth`.  
`\evensidemargin`  
`\marginparwidth`

```
64 \ifcompatibility
65 \setlength\oddsidemargin{53pt}
66 \setlength\evensidemargin{53pt}
67 \setlength\marginparwidth{90pt}
68 \else
69 \setlength\@tempdima{\paperwidth}
70 \addtolength\@tempdima{-2in}
71 \addtolength\@tempdima{-\textwidth}
72 \setlength\oddsidemargin {.5\@tempdima}
73 \setlength\evensidemargin {\oddsidemargin}
74 \setlength\marginparwidth {90\p@}
75 \fi
```

`\marginparsep` The horizontal space between the main text and marginal notes is determined by `\marginparsep`, the minimum vertical separation between two marginal notes is controlled by `\marginparpush`.

```
76 \setlength\marginparsep {11\p@}
77 \setlength\marginparpush{5\p@}
```

`\topmargin` The `\topmargin` is the distance between the top of ‘the printable area’ –which is 1 inch below the top of the paper– and the top of the box which contains the running head.

It can now be computed from the values set above.

```
78 \setlength\topmargin{27pt}
```

### 4.2.4 Footnotes

`\footnotesep` `\footnotesep` is the height of the strut placed at the beginning of every footnote. It equals the height of a normal `\footnotesize` strut in this class, thus no extra space occurs between footnotes.

```
79 \setlength\footnotesep{12\p@}
```

`\footins` `\skip\footins` is the space between the last line of the main text and the top of the first footnote.

```
80 \setlength{\skip\footins}{10\p@ \@plus 2\p@ \@minus 4\p@}
```

### 4.3 Page Styles

The page style *foo* is defined by defining the command `\ps@foo`. This command should make only local definitions. There should be no stray spaces in the definition, since they could lead to mysterious extra spaces in the output (well, that's something that should be always avoided).

`\@evenhead` The `\ps@...` command defines the macros `\@oddhead`, `\@oddfoot`, `\@evenhead`,  
`\@oddhead` and `\@evenfoot` to define the running heads and feet—e.g., `\@oddhead` is the  
`\@evenfoot` macro to produce the contents of the heading box for odd-numbered pages. It is  
`\@oddfoot` called inside an `\hbox` of width `\textwidth`.

#### 4.3.1 Marking conventions

To make headings determined by the sectioning commands, the page style defines the commands `\chaptermark`, `\sectionmark`, ..., where `\chaptermark{<text>}` is called by `\chapter` to set a mark, and so on.

The `\...mark` commands and the `\...head` macros are defined with the help of the following macros. (All the `\...mark` commands should be initialized to no-ops.)

L<sup>A</sup>T<sub>E</sub>X extends T<sub>E</sub>X's `\mark` facility by producing two kinds of marks, a 'left' and a 'right' mark, using the following commands:

`\markboth{<left>}{<right>}`: Adds both marks.  
`\markright{<right>}`: Adds a 'right' mark.  
`\leftmark`: Used in the `\@oddhead`, `\@oddfoot`, `\@evenhead` or `\@evenfoot` macros, it gets the current 'left' mark. `\leftmark` works like T<sub>E</sub>X's `\botmark` command.  
`\rightmark`: Used in the `\@oddhead`, `\@oddfoot`, `\@evenhead` or `\@evenfoot` macros, it gets the current 'right' mark. `\rightmark` works like T<sub>E</sub>X's `\firstmark` command.

The marking commands work reasonably well for right marks 'numbered within' left marks—e.g., the left mark is changed by a `\chapter` command and the right mark is changed by a `\section` command. However, it does produce somewhat anomalous results if two `\markboth`'s occur on the same page.

Commands like `\tableofcontents` that should set the marks in some page styles use a `\@mkboth` command, which is `\let` by the `pagestyle` command (`\ps@...`) to `\markboth` for setting the heading or to `\gobbletwo` to do nothing.

#### 4.3.2 Defining the page styles

The page styles *empty* and *plain* are defined in the L<sup>A</sup>T<sub>E</sub>X kernel (`ltpage.dtx`), but these definitions are changed to a simpler version for this document class.

`\ps@headings` The definition of the page style *headings* has to be different for two sided printing than it is for one sided printing.

```
81 \if@twoside
82   \def\ps@headings{%
```

The running feet are empty in this page style.

```
83     \let\@oddfoot\@empty\let\@evenfoot\@empty
```

The running head contains some information about this letter. The head is the same for even and odd pages.

```

84      \def\@oddhead{\slshape\headtoname{} \ignorespaces\toname
85              \hfil \@date
86              \hfil \pagename{} \thepage}%
87      \let\@evenhead\@oddhead

```

For one sided printing we don't need to define `\@evenhead` so the definition is somewhat simpler.

```

88 \else
89   \def\ps@headings{%
90       \let\@oddfoot\@empty
91       \def\@oddhead{\slshape\headtoname{} \ignorespaces\toname
92               \hfil \@date
93               \hfil \pagename{} \thepage}}
94 \fi

```

`\ps@empty` The definition of the page style *empty* is simple: No running head or foot at all.

```

95 \def\ps@empty{%
96     \let\@oddfoot\@empty\let\@oddhead\@empty
97     \let\@evenfoot\@empty\let\@evenhead\@empty}

```

`\ps@firstpage` The page style *firstpage* puts the telephone number in the proper place for the letterhead. It should be adapted to site conventions. The size of the number is determined depending on the main size.

```

98 \def\ps@firstpage{%
99     \let\@oddhead\@empty
100    \def\@oddfoot{\raisebox{-45\p@}{\z@}{%
101        \hb@xt@\textwidth{\hspace*{100\p@}%
102            \ifcase \@ptsize\relax
103                \normalsize
104            \or
105                \small
106            \or
107                \footnotesize
108            \fi
109            \fromlocation \hfill \telephonenumber}}\hss}}

```

`\ps@plain` The definition of the page style *plain* is again simple.

```

110 \def\ps@plain{%
111     \let\@oddhead\@empty
112     \def\@oddfoot{\normalfont\hfil\thepage\hfil}%
113     \def\@evenfoot{\normalfont\hfil\thepage\hfil}}

```

## 5 Document Markup

### 5.1 Global Declarations

`\name` The following declarations, shown with examples, give information about the sender:

`\signature`

`\address` • `\name{Dr. L. User}` : to be used for the return address on the envelope.

`\location`

`\telephone` 114 `\newcommand*{\name}[1]{\def\fromname{#1}}`



- `\signature{Larry User}` : goes after the closing.

```

115      \newcommand*{\signature}[1]{\def\fromsig{#1}}

• \address{3245 Foo St.\Gnu York} : used as the return address in the
  letter and on the envelope. If not declared, then an institutional standard
  address is used.

116      \newcommand*{\address}[1]{\def\fromaddress{#1}}

• \location{Room 374} : Acts as modifier to the standard institutional ad-
  dress.

117      \newcommand*{\location}[1]{\def\fromlocation{#1}}

• \telephone{(415)123-4567} : Just in case some style puts it on the letter.

118      \newcommand*{\telephone}[1]{\def\telephonenumber{#1}}

\fromname We make sure that the internal control sequences that are used to store the infor-
\fromsig   mation exist and are empty.
\fromaddress 119 \name{}
\fromlocation 120 \signature{}
\telephonenumber 121 \address{}
               122 \location{}
               123 \telephone{}

\makelabels The \makelabels declaration causes mailing labels to be made.
124 \newcommand*{\makelabels}{%
At the beginning of the document, we need to activate the \@mlabel and
\@startlabels commands, as well as write \@startlabels to the .aux file.
125   \AtBeginDocument{%
126     \let\@startlabels\startlabels
127     \let\@mlabel\mlabel
128     \if@filesw
129       \immediate\write\@mainaux{\string\@startlabels}\fi}%
At the end of the document we need to write \clearpage to the .aux file.
130   \AtEndDocument{%
131     \if@filesw\immediate\write\@mainaux{\string\clearpage}\fi}}
\makelabels is allowed only before the \begin{document} command.
132 \@onlypreamble\makelabels

```

## 5.2 The generic letter commands

**letter** The letter environment creates a new letter, starting from page 1, with footnotes starting from 1 as well. (The first page is unnumbered.) It has a single argument, which is the addressee and his address, as in

```

\begin{letter}{Sam Jones \\\
               Institute for Retarded Study\\
               Princeton, N.J.}

```

Local declarations, such as `\address`, can follow the `\begin{letter}`.

```

133 \newenvironment{letter}[1]
134   {\newpage
135     \if@twoside \ifodd\c@page
136               \else\thispagestyle{empty}\null\newpage\fi
137     \fi
138     \c@page \@ne
139     \c@footnote \z@
140     \interlinepenalty=200 % smaller than the TeXbook value

```

The `\leavevmode` and `\ignorespaces` commands are there for protecting against an empty argument.

```

141   \@processto{\leavevmode\ignorespaces #1}}

```

The end of the environment possibly writes the address information on the `.aux` file.

```

142   {\stopletter\@@par\pagebreak\@@par
143     \if@filesw
144       \begingroup
145         \let\=\relax
146         \let\protect\@unexpandable@protect
147         \immediate\write\@auxout{%
148           \string\@mlabel{\returnaddress}{\toname\\toaddress}}%
149       \endgroup
150     \fi}

```

`\@processto` `\@processto` gets the `\toname` and `\toaddress` from the letter environment's macro argument. `\@xproc` and `\@yproc` are auxiliary macros.

```

\@yproc 151 \long\def\@processto#1{%
152   \@xproc #1\\@@@%
153   \ifx\toaddress\@empty
154     \else
155       \@yproc #1@@@%
156     \fi}
157 \long\def\@xproc #1\\#2@@@{\def\toname{#1}\def\toaddress{#2}}
158 \long\def\@yproc #1\\#2@@@{\def\toaddress{#2}}

```

### 5.2.1 Page breaking control

`\stopbreaks` When the command `\stopbreaks` is issued no page breaks should occur until `\startbreaks` is called.

```

159 \newcommand*{\stopbreaks}{%
160   \interlinepenalty\@M
161   \def\par{\@@par\nobreak}%
162   \let\\\@nobreakcr
163   \let\vspace\@nobreakvspace}

```

`\nobreakvspace` These are needed by `\stopbreaks`

```

\@nobreakvspace 164 \DeclareRobustCommand\@nobreakvspace
\@nobreakcr 165   {\@ifstar\@nobreakvspace\@nobreakvspace}
166 \def\@nobreakvspace#1{%
167   \ifvmode

```

```

168     \nobreak\vskip #1\relax
169   \else
170     \@bsphack\vadjust{\nobreak\vskip #1}\@esphack
171   \fi}

172 \def\@nobreakcr{\@ifstar{\@normalcr*}{\@normalcr*}}

\startbreaks This cancels the effect of \stopbreaks.
173 \newcommand*\startbreaks{%
174   \let\\\@normalcr
175   \interlinepenalty 200%
176   \def\par{\@par\penalty 200\relax}}

\longindentation The size of the indent to use before the closing of the letter.
177 \newdimen\longindentation
178 \longindentation=.5\textwidth

\indentedwidth The width of the closing of the letter.
179 \newdimen\indentedwidth
180 \indentedwidth=\textwidth
181 \advance\indentedwidth -\longindentation

\opening Text is begun with the \opening command, whose argument generates the salu-
tation, as in

\opening{Dear Henry,}

This should produce everything up to and including the ‘Dear Henry,’ and a \par
command that follows. Since there’s a \vfil at the bottom of every page, it can
add vertical fill to position a short letter. It should use the following commands:

```

- \toname : name part of ‘to’ address. Will be one line long.
- \toaddress : address part of ‘to’ address. The lines separated by \.
- \fromname : name of sender.
- \fromaddress : argument of current \address declaration– null if none.  
Should use standard institutional address if null.
- \fromlocation : argument of current \location declaration–null if none.
- \telephonenumber : argument of current \telephone declaration–null if none.

```

182 \newcommand*\opening{[1]{\ifx\@empty\fromaddress
183   \thispagestyle{firstpage}%
184   {\raggedleft\@date\par}%
185   \else % home address
186     \thispagestyle{empty}%

187     {\raggedleft\begin{tabular}{l@{}}\ignorespaces
188       \fromaddress \*[2\parskip]%
189       \@date \end{tabular}\par}%
190   \fi
191   \vspace{2\parskip}%
192   {\raggedright \toname \. \toaddress \par}%
193   \vspace{2\parskip}%
194   #1\par\nobreak}

```

`\closing` The body of the letter follows, ended by a `\closing` command, as in

```
\closing{Yours truly,}
```

This command generates the closing matter, and the signature. An obvious thing to do is to use a `\parbox` for the closing and the signature. Should use the following:

- `\fromsig` : argument of current `\signature` declaration or, if null, the `\fromname`.
- `\stopbreaks` : a macro that inhibits page breaking.

```
195 \newcommand{\closing}[1]{\par\nobreak\vspace{\parskip}%
196   \stopbreaks
197   \noindent
198   \ifx\@empty\fromaddress\else
199     \hspace*{\longindentation}\fi
200   \parbox{\indentedwidth}{\raggedright
201     \ignorespaces #1\\[6\medskipamount]%
202     \ifx\@empty\fromsig
203       \fromname
204     \else \fromsig \fi\strut}%
205   \par}
```

`\smallskipamount` Of these three, only `\medskipamount` is actually used above.

```
\medskipamount 206 %\smallskipamount=.5\parskip
```

```
\bigskipamount 207 \medskipamount=\parskip
```

```
208 %\bigskipamount=2\parskip
```

`\cc` After the `\closing` you can put arbitrary stuff, which is typeset with zero  
`\encl` `\parindent` and no page breaking. Commands designed for use after the clos-  
`\ps` ing are:

```
\cc{Tinker\\Evers\\Chance}
```

which produces:

```
cc:  Tinker
     Evers
     Chance
```

Note the obvious use of `\parbox`.

```
209 \newcommand*{\cc}[1]{%
210   \par\noindent
211   \parbox[t]{\textwidth}{%
212     \@hangfrom{\normalfont\ccname: }%
213     \ignorespaces #1\strut}\par}

\encl{Foo(2)\\Bar}
```

which produces:

```
encl:  Foo(2)
       Bar
```

```
214 \newcommand*{\encl}[1]{%
215   \par\noindent
216   \parbox[t]{\textwidth}{%
217     \@hangfrom{\normalfont\enclname: }%
218     \ignorespaces #1\strut}\par}
```

The only thing `\ps` needs to do is call `\startbreaks`, which allows page breaking again.

```
219 \newcommand*{\ps}{\par\startbreaks}
```

`\stopletter` The `\stopletter` command is called by `\endletter` to do the following:

- Add any desired fill or other material at the end of the letter.
- Define `\returnaddress` to be the return address for the mailing label. More precisely, it is the first argument of the `\mlabel` command described below. It should be defined to null if the return address doesn't appear on the labels. Any command, other than `\`, that should not be expanded until the `\mlabel` command is actually executed must be preceded by `\protect`. Whenever possible, `\protect` commands in the definition of `\returnaddress`—it's much more efficient that way. In particular, when the standard return address is used, you should define `\returnaddress` to something like `\protect\standardreturnaddress`.

```
220 \newcommand*{\stopletter}{}%
```

### 5.3 Customizing the labels

Commands for generating the labels are put on the `.aux` file, which is read in and processed by the `\end{document}` command. You have to define the following two commands:

- `\startlabels` : Should reset the page layout parameters if necessary.
- `\mlabel{<return address>}{<to address>}` : Command to generate a single label.

`\returnaddress` The return address for the mailing labels can be stored in this macro.

```
221 \newcommand*{\returnaddress}{}%
```

`\labelcount` A register to count the labels

```
222 \newcount\labelcount
```

`\startlabels` The following `\startlabels` command sets things up for producing labels in two columns of five 2" × 4-1/4" labels each, suitable for reproducing onto Avery brand number 5352 address labels.

```
223 \newcommand*{\startlabels}{\labelcount\z@
224   \pagestyle{empty}%
225   \let\@texttop\relax
226   \topmargin -50\p@
227   \headsep \z@
228   \oddsidemargin -35\p@
229   \evensidemargin -35\p@
230   \textheight 10in
231   \@colht\textheight \@colroom\textheight \vsize\textheight
232   \textwidth 550\p@
233   \columnsep 26\p@
234   \ifcase \@ptsize\relax
235     \normalsize
```

```

236 \or
237 \small
238 \or
239 \footnotesize
240 \fi
241 \baselineskip \z@
242 \lineskip \z@
243 \boxmaxdepth \z@
244 \parindent \z@
245 \twocolumn\relax}

\@startlabels \@startlabels is the command name that is written to the .aux file. It is a no-op
at first, and defined to be the same as \startlabels in the \begin{document}
hook.
246 \let\@startlabels=\relax

\mlabel This command prints an address label; it is used when the user specified
\makelabels in the preamble of his document. The command \mlabel takes
two arguments; the second argument is supposed to be the address; the first arg-
ument can be used to print a return address. In this document class we ignore
the first argument. Also the labels are supposed to be 2 inch high and 3.6 inch
wide. When your address labels have a different you will have to define your own
\mlabel command.
247 \newcommand*{\mlabel}[2]{%
248 \parbox[b][2in][c]{262\p@}{\strut\ignorespaces #2}%
249 }

\@mlabel The macro \@mlabel is written to the .aux file instead of \mlabel. This allows
us to make it a no-op by default, and then activate it in the \begin{document}
hook.
250 \let\@mlabel=\gobbletwo

```

## 5.4 Lists

### 5.4.1 General List Parameters

The following commands are used to set the default values for the list environ-  
ment's parameters. See the L<sup>A</sup>T<sub>E</sub>X manual for an explanation of the meanings  
of the parameters. Defaults for the list environment are set as follows. First,  
\rightmargin, \listparindent and \itemindent are set to 0pt. Then, for a Kth  
level list, the command \@listK is called, where 'K' denotes 'i', 'ii', ... , 'vi'.  
(I.e., \@listiii is called for a third-level list.) By convention, \@listK should  
set \leftmargin to \leftmarginK.

```

\leftmargin For efficiency, level-one list's values are defined at top level, and \@listi is defined
\leftmargini to set only \leftmargin.
\leftmarginii 251 \setlength\leftmargini {2.5em}
\leftmarginiii The following three are calculated so that they are larger than the sum of
\leftmarginiv \labelsep and the width of the default labels (which are '(m)', 'vii.' and 'M.').
\leftmarginv 252 \setlength\leftmarginii {2.2em}
\leftmarginvi 253 \setlength\leftmarginiii {1.87em}

```

```

254 \setlength\leftmarginiv {1.7em}
255 \setlength\leftmarginv {1em}
256 \setlength\leftmarginvi {1em}
Here we set the top level leftmargin.
257 \setlength\leftmargin {\leftmargini}

\labelsep \labelsep is the distance between the label and the text of an item; \labelwidth
\labelwidth is the width of the label.
258 \setlength \labelsep {5\p@}
259 \setlength \labelwidth{\leftmargini}
260 \addtolength\labelwidth{-\labelsep}

\partopsep When the user leaves a blank line before the environment an extra vertical space
of \partopsep is inserted, in addition to \parskip and \topsep.
261 \setlength\partopsep{0\p@}

\@beginparpenalty These penalties are inserted before and after a list or paragraph environment.
\@endparpenalty They are set to a bonus value to encourage page breaking at these points.

\@itempenalty This penalty is inserted between list items.
262 \@beginparpenalty -\@lowpenalty
263 \@endparpenalty -\@lowpenalty
264 \@itempenalty -\@lowpenalty

\@listI \@listI defines top level and \@listi values of \leftmargin, \parsep, \topsep,
\@listi and \itemsep
265 \def\@listI{\setlength\leftmargin{\leftmargini}%
266 \setlength\parsep {0\p@}%
267 \setlength\topsep {.4em}%
268 \setlength\itemsep{.4em}}
269 \let\@listi\@listI

We have to initialize these parameters.
270 \@listi

\@listii Here are the same macros for the higher level lists.
\@listiii
\@listiv
\@listv
\@listvi
271 \def\@listii {\setlength \leftmargin{\leftmarginii}%
272 \setlength \labelwidth{\leftmarginii}%
273 \addtolength\labelwidth{-\labelsep}}
274 \def\@listiii{\setlength \leftmargin{\leftmarginiii}%
275 \setlength \labelwidth{\leftmarginiii}%
276 \addtolength\labelwidth{-\labelsep}%
277 \setlength \topsep {.2em}%
278 \setlength \itemsep {\topsep}}
279 \def\@listiv {\setlength \leftmargin{\leftmarginiv}%
280 \setlength \labelwidth{\leftmarginiv}%
281 \addtolength\labelwidth{-\labelsep}}
282 \def\@listv {\setlength \leftmargin{\leftmarginv}%
283 \setlength \labelwidth{\leftmarginv}%
284 \addtolength\labelwidth{-\labelsep}}
285 \def\@listvi {\setlength \leftmargin{\leftmarginvi}%
286 \setlength \labelwidth{\leftmarginvi}%
287 \addtolength\labelwidth{-\labelsep}}

```

### 5.4.2 Enumerate

The enumerate environment uses four counters: *enumi*, *enumii*, *enumiii* and *enumiv*, where *enumN* controls the numbering of the Nth level enumeration.

<code>\theenumi</code>	The counters are already defined in the L <sup>A</sup> T <sub>E</sub> X kernel ( <code>ltlists.dtx</code> ), but their
<code>\theenumii</code>	representation is changed here.
<code>\theenumiii</code>	288 <code>\renewcommand\theenumi{\@arabic\c@enumi}</code>
<code>\theenumiv</code>	289 <code>\renewcommand\theenumii{\@alph\c@enumii}</code>
	290 <code>\renewcommand\theenumiii{\@roman\c@enumiii}</code>
	291 <code>\renewcommand\theenumiv{\@Alph\c@enumiv}</code>
<code>\labelenumi</code>	The commands <code>\labelenumi</code> ... <code>\labelenumiv</code> generate the label for each item.
<code>\labelenumii</code>	292 <code>\newcommand\labelenumi{\theenumi.}</code>
<code>\labelenumiii</code>	293 <code>\newcommand\labelenumii{(\theenumii)}</code>
<code>\labelenumiv</code>	294 <code>\newcommand\labelenumiii{\theenumiii.}</code>
	295 <code>\newcommand\labelenumiv{\theenumiv.}</code>
<code>\p@enumii</code>	The expansion of <code>\p@enumN\theenumN</code> defines the output of a <code>\ref</code> command
<code>\p@enumiii</code>	when referencing an item of the Nth level of an enumerated list.
<code>\p@enumiv</code>	296 <code>\renewcommand\p@enumii{\theenumi}</code>
	297 <code>\renewcommand\p@enumiii{\theenumi(\theenumii)}</code>
	298 <code>\renewcommand\p@enumiv{\p@enumiii\theenumiii}</code>

### 5.4.3 Itemize

<code>\labelitemi</code>	Itemization is controlled by <code>\labelitemi</code> , <code>\labelitemii</code> , <code>\labelitemiii</code> , and
<code>\labelitemii</code>	<code>\labelitemiv</code> , which define the labels of the various itemization levels: the sym-
<code>\labelitemiii</code>	bols used are bullet, bold en-dash, asterisk and centered dot.
<code>\labelitemiv</code>	299 <code>\newcommand\labelitemi{\textbullet}</code>
	300 <code>\newcommand\labelitemii{\normalfont\bfseries \textendash}</code>
	301 <code>\newcommand\labelitemiii{\textasteriskcentered}</code>
	302 <code>\newcommand\labelitemiv{\textperiodcentered}</code>

### 5.4.4 Description

<code>description</code>	The description environment is defined here – while the itemize and enumerate environments are defined in the L <sup>A</sup> T <sub>E</sub> X kernel ( <code>ltlists.dtx</code> ).
--------------------------	--

```

303 \newenvironment{description}
304     {\list{}{\labelwidth\z@ \itemindent-\leftmargin
305               \let\makelabel\descriptionlabel}}
306     {\endlist}

```

<code>\descriptionlabel</code>	To change the formatting of the label, you must redefine <code>\descriptionlabel</code> .
--------------------------------	---

```

307 \newcommand*\descriptionlabel[1]{\hspace\labelsep
308                               \normalfont\bfseries #1}

```



## 5.5 Defining new environments

### 5.5.1 Verse

**verse** The verse environment is defined by making clever use of the list environment's parameters. The user types `\` to end a line. This is implemented by letting `\` equal `\@centercr`.

```
309 \newenvironment{verse}
310     {\let\=\@centercr
311      \list{}{\setlength\itemsep{\z@}%
312              \setlength\itemindent{-15\p@}%
313              \setlength\listparindent{\itemindent}%
314              \setlength\rightmargin{\leftmargin}%
315              \addtolength\leftmargin{15\p@}}%
316      \item[]}
317     {\endlist}
```

### 5.5.2 Quotation

**quotation** The quotation environment is also defined by making clever use of the list environment's parameters. The lines in the environment are set smaller than `\textwidth`. The first line of a paragraph inside this environment is indented.

```
318 \newenvironment{quotation}
319     {\list{}{\setlength\listparindent{1.5em}%
320              \setlength\itemindent{\listparindent}%
321              \setlength\rightmargin{\leftmargin}}%
322      \item[]}
323     {\endlist}
```

### 5.5.3 Quote

**quote** The quote environment is like the quotation environment except that paragraphs are not indented.

```
324 \newenvironment{quote}
325     {\list{}{\setlength\rightmargin{\leftmargin}}%
326      \item[]}
327     {\endlist}
```

### 5.5.4 Theorem

This document class does not define its own theorem environments, the defaults, supplied by the L<sup>A</sup>T<sub>E</sub>X kernel (`ltthm.dtx`) are available.

## 5.6 Setting parameters for existing environments

### 5.6.1 Array and tabular

`\arraycolsep` The columns in an array environment are separated by `2\arraycolsep`.

```
328 \setlength\arraycolsep{5\p@}
```

`\tabcolsep` The columns in an tabular environment are separated by `2\tabcolsep`.

```
329 \setlength\tabcolsep{6\p@}
```

`\arrayrulewidth` The width of vertical rules in the array and tabular environments is given by `\arrayrulewidth`.  
 330 `\setlength\arrayrulewidth{.4\p@}`

`\doublerulesep` The space between adjacent rules in the array and tabular environments is given by `\doublerulesep`.  
 331 `\setlength\doublerulesep{2\p@}`

### 5.6.2 Tabbing

`\tabbingsep` This controls the space that the `\'` command puts in. (See L<sup>A</sup>T<sub>E</sub>X manual for an explanation.)  
 332 `\setlength\tabbingsep{\labelsep}`

### 5.6.3 Minipage

`\@minipagerestore` The macro `\@minipagerestore` is called upon entry to a minipage environment to set up things that are to be handled differently inside a minipage environment. In the current styles, it does nothing.

`\@mpfootins` Minipages have their own footnotes; `\skip\@mpfootins` plays same rôle for footnotes in a minipage as `\skip\footins` does for ordinary footnotes.  
 333 `\skip\@mpfootins = \skip\footins`

### 5.6.4 Framed boxes

`\fboxsep` The space left by `\fbox` and `\framebox` between the box and the text in it.

`\fboxrule` The width of the rules in the box made by `\fbox` and `\framebox`.  
 334 `\setlength\fboxsep{3\p@}`  
 335 `\setlength\fboxrule{.4\p@}`

### 5.6.5 Equation and eqnarray

`\theequation` The equation counter will be typeset using arabic numbers.  
 336 `\renewcommand\theequation{\@arabic\c@equation}`

`\jot` `\jot` is the extra space added between lines of an `eqnarray` environment. The default value is used.  
 337 `\setlength\jot{3pt}`

`\@eqnnum` The macro `\@eqnnum` defines how equation numbers are to appear in equations. Again the default is used.  
 338 `\def\@eqnnum{(\theequation)}`

## 5.7 Font changing

Here we supply the declarative font changing commands that were common in L<sup>A</sup>T<sub>E</sub>X version 2.09 and earlier. These commands work in text mode *and* in math mode. They are provided for compatibility, but one should start using the `\text...` and `\math...` commands instead. These commands are redefined using `\@renewfontswitch`, a command with three arguments: the user command to be defined; L<sup>A</sup>T<sub>E</sub>X commands to execute in text mode and L<sup>A</sup>T<sub>E</sub>X commands to execute in math mode.

```
\rm The commands to change the family.
\tt 339 \DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\sff 340 \DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
      341 \DeclareOldFontCommand{\tt}{\normalfont\ttfamily}{\mathtt}

\bf The command to change to the bold series. One should use \mdseries to explicitly
switch back to medium series.
      342 \DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}

\s1 And the commands to change the shape of the font. The slanted and small caps
\it shapes are not available by default as math alphabets, so those changes do nothing
\sc in math mode. One should use \upshape to explicitly change back to the upright
shape.
      343 \DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
      344 \DeclareOldFontCommand{\sl}{\normalfont\slshape}{\relax}
      345 \DeclareOldFontCommand{\sc}{\normalfont\scshape}{\relax}

\cal The commands \cal and \mit should only be used in math mode, outside math
\mit mode they have no effect. Currently the New Font Selection Scheme defines these
commands to generate warning messages. Therefore we have to define them ‘by
hand’.
      346 \DeclareRobustCommand*\cal{\@fontswitch{\relax}{\mathcal}}
      347 \DeclareRobustCommand*\mit{\@fontswitch{\relax}{\mathnormal}}
```

## 5.8 Footnotes

`\footnoterule` Usually, footnotes are separated from the main body of the text by a small rule. This rule is drawn by the macro `\footnoterule`. We have to make sure that the rule takes no vertical space (see `plain.tex`) so we compensate for the natural height of the rule of 0.4pt by adding the right amount of vertical skip.

To prevent the rule from colliding with the footnote we first add a little negative vertical skip, then we put the rule and make sure we end up at the same point where we begun this operation.

```
348 \renewcommand\footnoterule{%
349   \kern-\p@
350   \hrule \@width .4\columnwidth
351   \kern .6\p@}

\c@footnote A counter for footnotes.
352 % \newcounter{footnote}
```

`\@makefntext` The footnote mechanism of L<sup>A</sup>T<sub>E</sub>X calls the macro `\@makefntext` to produce the actual footnote. The macro gets the text of the footnote as its argument and should use `\@makefnmark` to produce the mark of the footnote. The macro `\@makefntext` is called when effectively inside a `\parbox` of width `\columnwidth` (i.e., with `\hsize = \columnwidth`).

An example of what can be achieved is given by the following piece of T<sub>E</sub>X code.

```
\long\def\@makefntext#1{%
  \@setpar{\@par
    \@tempdima = \hsize
    \advance\@tempdima-10pt
    \parshape \@ne 10pt \@tempdima}%
  \par
  \parindent 1em\noindent
  \hb@xt@0\z@{\hss\@makefnmark}#1}
```

The effect of this definition is that all lines of the footnote are indented by 10pt, while the first line of a new paragraph is indented by 1em. To change these dimensions, just substitute the desired value for ‘10pt’ (in both places) or ‘1em’. The mark is flush right against the footnote.

In this document class we use a simpler macro, in which the footnote text is set like an ordinary text paragraph, with no indentation except on the first line of a paragraph, and the first line of the footnote. Thus, all the macro must do is set `\parindent` to the appropriate value for succeeding paragraphs and put the proper indentation before the mark.

```
353 \long\def\@makefntext#1{%
354   \noindent
355   \hangindent 5\p@
356   \hb@xt@5\p@{\hss\@makefnmark}#1}
```

`\@makefnmark` The footnote markers that are printed in the text to point to the footnotes should be produced by the macro `\@makefnmark`. We use the default definition for it.

```
357 %\def\@makefnmark{\hbox{$^{\@thefnmark}\m@th$}}
```

## 5.9 Words

`\ccname` This document class is for documents prepared in the English language. To prepare  
`\enclname` a version for another language, various English words must be replaced. All the  
`\pagename` English words that require replacement are defined below in command names.

```
\headtoname 358 \newcommand*{\ccname}{cc}
359 \newcommand*{\enclname}{encl}
360 \newcommand*{\pagename}{Page}
361 \newcommand*{\headtoname}{To}
```

## 5.10 Date

`\today` This macro uses the T<sub>E</sub>X primitives `\month`, `\day` and `\year` to provide the date of the L<sup>A</sup>T<sub>E</sub>X-run.

```
362 \newcommand*{\today}{\ifcase\month\or
363   January\or February\or March\or April\or May\or June\or
```

```

364 July\or August\or September\or October\or November\or December\fi
365 \space\number\day, \number\year}

```

## 5.11 Two column mode

`\columnsep` This gives the distance between two columns in two column mode.

```

366 \setlength\columnsep{10\p@}

```

`\columnseprule` This gives the width of the rule between two columns in two column mode. We have no visible rule.

```

367 \setlength\columnseprule{0\p@}

```

## 5.12 The page style

We have *plain* pages in this document class by default. We use arabic page numbers.

```

368 \pagestyle{plain}
369 \pagenumbering{arabic}

```

## 5.13 Single or double sided printing

We don't try to make each page as long as all the others.

```

370 \raggedbottom

```

`\@texttop` The document class letter sets `\@texttop` to `\vskip 0pt plus .00006fil` on the first page of a letter, which centers a short letter on the page. This fil value may have to be changed for other letterheads. This setting has to be done after `\raggedbottom` is executed, since the latter sets `\@texttop` to `\relax`.

```

371 \def\@texttop{\ifnum\c@page=1\vskip \z@ plus.00006fil\relax\fi}

```

We always start in one column mode.

```

372 \onecolumn
373 \letletter

```

## Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

Symbols		
<code>\@par</code> . . . . .	<a href="#">142, 161, 176</a>	<code>\@eqnnum</code> . . . . . <a href="#">338</a>
<code>\@auxout</code> . . . . .	<a href="#">147</a>	<code>\@esphack</code> . . . . . <a href="#">170</a>
<code>\@beginparpenalty</code> . . . . .	<a href="#">262</a>	<code>\@evenfoot</code> <a href="#">81, 83, 97, 113</a>
<code>\@bsphack</code> . . . . .	<a href="#">170</a>	<code>\@evenhead</code> . . . . . <a href="#">81, 87, 97</a>
<code>\@colht</code> . . . . .	<a href="#">231</a>	<code>\@fontswitch</code> . . . . . <a href="#">346, 347</a>
<code>\@colroom</code> . . . . .	<a href="#">231</a>	<code>\@gobbletwo</code> . . . . . <a href="#">250</a>
<code>\@date</code> . . . . .	<a href="#">85, 92, 184, 189</a>	<code>\@hangfrom</code> . . . . . <a href="#">212, 217</a>
<code>\@eha</code> . . . . .	<a href="#">30</a>	<code>\@highpenalty</code> . . . . . <a href="#">47</a>
<code>\@endparpenalty</code> . . . . .	<a href="#">262</a>	<code>\@ifstar</code> . . . . . <a href="#">165, 172</a>
		<code>\@itempenalty</code> . . . . . <a href="#">262</a>
		<code>\@latexerr</code> . . . . . <a href="#">29</a>
		<code>\@listI</code> . . . . . <a href="#">265</a>
		<code>\@listi</code> . . . . . <a href="#">265</a>
		<code>\@listii</code> . . . . . <a href="#">271</a>
		<code>\@listiii</code> . . . . . <a href="#">271</a>
		<code>\@listiv</code> . . . . . <a href="#">271</a>
		<code>\@listv</code> . . . . . <a href="#">271</a>
		<code>\@listvi</code> . . . . . <a href="#">271</a>
		<code>\@lowpenalty</code> . . . . .
		. . . . . <a href="#">47, 262, 263, 264</a>



<code>\leftmarginiii</code> . . . . .	<code>\overfullrule</code> . . . 35, 36	<code>\startlabels</code> . . 126, 223
. . . . . 251, 274, 275		<code>\stopbreaks</code> . . . 159, 196
<code>\leftmarginiv</code> . . . . .	<b>P</b>	<code>\stopletter</code> . . . 142, 220
. . . . . 251, 279, 280	<code>\p@enumii</code> . . . . . 296	<code>\string</code> . . . 129, 131, 148
<code>\leftmarginv</code> . . . . .	<code>\p@enumiii</code> . . . . . 296	<code>\strut</code> 204, 213, 218, 248
. . . . . 251, 282, 283	<code>\p@enumiv</code> . . . . . 296	
<code>\leftmarginvi</code> . . . . .	<code>\pagebreak</code> . . . . . 142	<b>T</b>
. . . . . 251, 285, 286	<code>\pagename</code> . . 86, 93, 358	<code>\tabbingsep</code> . . . . . 332
<code>letter</code> (environment) 133	<code>\pagenumbering</code> . . . 369	<code>\tabcolsep</code> . . . . . 329
<code>\lineskip</code> . . . . . 42, 242	<code>\paperheight</code> 4, 7, 10,	<code>\telephone</code> . . . 114, 123
<code>\listparindent</code> . . . . .	13, 16, 19, 22, 23	<code>\telephonenumber</code> . . . . .
. . . . . 313, 319, 320	<code>\paperwidth</code> . . . . .	. . . . . 109, 118, 119
<code>\location</code> . . . . . 114, 122	. . . 5, 8, 11, 14,	<code>\textasteriskcentered</code>
<code>\longindentation</code> . . . . .	17, 20, 23, 24, 69	. . . . . 301
. . . . . 177, 181, 199	<code>\parbox</code> 200, 211, 216, 248	<code>\textbullet</code> . . . . . 299
<b>M</b>	<code>\parindent</code> . . . . . 45, 244	<code>\textendash</code> . . . . . 300
<code>\makelabel</code> . . . . . 305	<code>\parsep</code> . . . . . 266	<code>\textheight</code> 62, 230, 231
<code>\makelabels</code> . . . . . 124	<code>\parskip</code> . . . . . 45,	<code>\textperiodcentered</code> 302
<code>\marginparpush</code> . . . . . 76	188, 191, 193,	<code>\textwidth</code> . . . . .
<code>\marginparsep</code> . . . . . 76	195, 206, 207, 208	60, 71, 101, 178,
<code>\marginparwidth</code> . . . . . 64	<code>\partopsep</code> . . . . . 261	180, 211, 216, 232
<code>\mathbf</code> . . . . . 342	<code>\postdisplaypenalty</code> 52	<code>\theenumi</code> . . . . .
<code>\mathcal</code> . . . . . 346	<code>\predisplaypenalty</code> . 52	. 288, 292, 296, 297
<code>\mathit</code> . . . . . 343	<code>\ps</code> . . . . . 209	<code>\theenumii</code> 288, 293, 297
<code>\mathnormal</code> . . . . . 347	<code>\ps@empty</code> . . . . . 95	<code>\theenumiii</code> 288, 294, 298
<code>\mathrm</code> . . . . . 339	<code>\ps@firstpage</code> . . . . . 98	<code>\theenumiv</code> . . . 288, 295
<code>\mathsf</code> . . . . . 340	<code>\ps@headings</code> . . . . . 81	<code>\theequation</code> . . 336, 338
<code>\mathtt</code> . . . . . 341	<code>\ps@plain</code> . . . . . 110	<code>\thepage</code> 86, 93, 112, 113
<code>\medskipamount</code> 201, 206	<b>Q</b>	<code>\thispagestyle</code> . . . . .
<code>\mit</code> . . . . . 346	<code>quotation</code> (environ-	. . . . . 136, 183, 186
<code>\mlabel</code> . . . . . 127, 247	ment) . . . . . 318	<code>\toaddress</code> . . . 148,
<b>N</b>	<code>quote</code> (environment) . 324	153, 157, 158, 192
<code>\name</code> . . . . . 114, 119	<b>R</b>	<code>\today</code> . . . . . 362
<code>\newcount</code> . . . . . 222	<code>\raggedleft</code> . . . 184, 187	<code>\toname</code> . . . . .
<code>\nobreakvspace</code> . . . . . 164	<code>\raisebox</code> . . . . . 100	84, 91, 148, 157, 192
<code>\normalfont</code> . . . . .	<code>\returnaddress</code> 148, 221	<code>\topmargin</code> . . . . . 78, 226
. 112, 113, 212,	<code>\rm</code> . . . . . 339	<code>\topsep</code> . . . 267, 277, 278
217, 300, 308,	<b>S</b>	<code>\tt</code> . . . . . 339
339, 340, 341,	<code>\sc</code> . . . . . 343	<code>\ttfamily</code> . . . . . 341
342, 343, 344, 345	<code>\scshape</code> . . . . . 345	<code>\twocolumn</code> . . . . . 245
<code>\normallineskip</code> . . . 42	<code>\sf</code> . . . . . 339	<b>V</b>
<code>\normalsize</code> . . . 103, 235	<code>\sffamily</code> . . . . . 340	<code>\vadjust</code> . . . . . 170
<code>\null</code> . . . . . 136	<code>\signature</code> . . . 114, 120	<code>verse</code> (environment) . 309
<b>O</b>	<code>\sl</code> . . . . . 343	<code>\vsize</code> . . . . . 231
<code>\oddsidemargin</code> . 64, 228	<code>\small</code> . . . . . 105, 237	<b>W</b>
<code>\onecolumn</code> . . . . . 372	<code>\smallskipamount</code> . . 206	<code>\widowpenalty</code> . . . . . 50
<code>\opening</code> . . . . . 182	<code>\startbreaks</code> . . 173, 219	<code>\write</code> . . . . . 129, 131, 147