Standard Letter Document Class for LaTeX version 2e

Leslie Lamport and Frank Mittelbach and Rainer Schöpf

November 30, 2022

This file is maintained by the LaTeX Project team.
Bug reports can be opened (category `latex`) at
`https://latex-project.org/bugs.html`.

Contents

1 Initial Code 2
- 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

2 Executing Options 4

3 Loading Packages 4

4 Document Layout 4
- 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

5 Document Markup 8
- 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 . . . 15

5.4.1 General List Parameters 15
5.4.2 Enumerate 16
5.4.3 Itemize 16
5.4.4 Description 17
5.5 Defining new environments 17
5.5.1 Verse 17
5.5.2 Quotation 17
5.5.3 Quote 18
5.5.4 Theorem 18
5.6 Setting parameters for existing environments 18
5.6.1 Array and tabular 18
5.6.2 Tabbing 18
5.6.3 Minipage 18
5.6.4 Framed boxes 19
5.6.5 Equation and eqnarray 19
5.7 Font changing 19
5.8 Footnotes 20

6 Initialization 21
6.1 Words 21
6.2 Date 21
6.3 Two column mode 21
6.4 The page style 21
6.5 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.

```
⟨*letter⟩
\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.

```
\DeclareOption{a4paper}
   {\setlength\paperheight {297mm}%
    \setlength\paperwidth  {210mm}}
\DeclareOption{a5paper}
   {\setlength\paperheight {210mm}%
    \setlength\paperwidth  {148mm}}
\DeclareOption{b5paper}
   {\setlength\paperheight {250mm}%
    \setlength\paperwidth  {176mm}}
\DeclareOption{letterpaper}
   {\setlength\paperheight {11in}%
   \setlength\paperwidth  {8.5in}}
\DeclareOption{legalpaper}
  {\setlength\paperheight {14in}%
   \setlength\paperwidth  {8.5in}}
\DeclareOption{executivepaper}
  {\setlength\paperheight {10.5in}%
   \setlength\paperwidth  {7.25in}}
```

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

```
\DeclareOption{landscape}
  {\setlength\@tempdima   {\paperheight}%
   \setlength\paperheight {\paperwidth}%
   \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 wont define both 8pt and 18pt options.

```
\DeclareOption{10pt}{\renewcommand\@ptsize{0}}
\DeclareOption{11pt}{\renewcommand\@ptsize{1}}
\DeclareOption{12pt}{\renewcommand\@ptsize{2}}
```

1.3 Two-side or one-side printing

Two-sided printing was not supported in the LaTeX 2.09 version of this document class.

```
\if@compatibility
  \DeclareOption{twoside}{\@latexerr{No ‘twoside’ layout for letters}%
                                   \@eha}
\else
  \DeclareOption{twoside}{\@twosidetrue  \@mparswitchtrue}
\fi
```

```
\DeclareOption{oneside}{\@twosidefalse \@mparswitchfalse}
```

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.

```
\DeclareOption{draft}{\setlength\overfullrule{5pt}}
\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.

```
\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.

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

2 Executing Options

Here we execute the default options to initialize certain variables.

```
\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.

```
\ProcessOptions
```

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

```
\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`
`\normallineskip`

These parameters control TeX's behavior when two lines tend to come too close together.

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

`\baselinestretch`

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

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

`\parskip`
`\parindent`

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

```
\setlength\parskip{0.7em}
\setlength\parindent{0\p@}
```

`\@lowpenalty` `\@medpenalty` `\@highpenalty` 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.

```
\@lowpenalty   51
\@medpenalty  151
\@highpenalty 301
```

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

```
% \clubpenalty  150
% \widowpenalty 150
```

`\displaywidowpenalty` `\predisplaypenalty` `\postdisplaypenalty` 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. Again the default values are used, therefore we only show them here.

```
% \displaywidowpenalty 50
% \predisplaypenalty   10000
% \postdisplaypenalty  0
```

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

```
% \interlinepenalty 0
```

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

```
% \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` `\headsep` 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.

```
\setlength\headheight{12\p@}
\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:

```
\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.

```
\if@compatibility
  \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).

```
  \setlength\textheight{505\p@}
\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`
`\evensidemargin`
`\marginparwidth`

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

```
\if@compatibility
  \setlength\oddsidemargin{53pt}
  \setlength\evensidemargin{53pt}
  \setlength\marginparwidth{90pt}
\else
  \setlength\@tempdima{\paperwidth}
  \addtolength\@tempdima{-2in}
  \addtolength\@tempdima{-\textwidth}
  \setlength\oddsidemargin   {.5\@tempdima}
  \setlength\evensidemargin  {\oddsidemargin}
  \setlength\marginparwidth  {90\p@}
\fi
```

`\marginparsep`
`\marginparpush`

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`.

```
\setlength\marginparsep {11\p@}
\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.

```
\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.

```
\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.

```
\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`
`\@oddhead`
`\@evenfoot`
`\@oddfoot`

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

LaTeX extends TeX'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 TeX's `\botmark` command.
`\rightmark`: Used in the `\@oddhead`, `\@oddfoot`, `\@evenhead` or `\@evenfoot` macros, it gets the current 'right' mark. `\rightmark` works like TeX'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 pagestyles *empty* and *plain* are defined in the LaTeX 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.

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

The running feet are empty in this page style.

```
      \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.

```
      \def\@oddhead{\slshape\headtoname{} \ignorespaces\toname
                    \hfil \@date
                    \hfil \pagename{} \thepage}%
      \let\@evenhead\@oddhead}
```

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

```
\else
  \def\ps@headings{%
      \let\@oddfoot\@empty
      \def\@oddhead{\slshape\headtoname{} \ignorespaces\toname
                    \hfil \@date
                    \hfil \pagename{} \thepage}}
\fi
```

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

```
\def\ps@empty{%
      \let\@oddfoot\@empty\let\@oddhead\@empty
      \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.

```
\def\ps@firstpage{%
     \let\@oddhead\@empty
     \def\@oddfoot{\raisebox{-45\p@}[\z@]{%
        \hb@xt@\textwidth{\hspace*{100\p@}%
          \ifcase \@ptsize\relax
             \normalsize
          \or
             \small
          \or
             \footnotesize
          \fi
        \fromlocation \hfill \telephonenum}}\hss}}
```

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

```
\def\ps@plain{%
      \let\@oddhead\@empty
      \def\@oddfoot{\normalfont\hfil\thepage\hfil}%
      \def\@evenfoot{\normalfont\hfil\thepage\hfil}}
```

5 Document Markup

5.1 Global Declarations

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

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

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

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

```
    \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.

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

- `\location{Room 374}` : Acts as modifier to the standard institutional address.

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

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

```
    \newcommand*\telephone[1]{\def\telephonenum{#1}}
```

`\fromname` `\fromsig` `\fromaddress` `\fromlocation` `\telephonenum` We make sure that the internal control sequences that are used to store the information exist and are empty.

```
\name{}
\signature{}
\address{}
\location{}
\telephone{}
```

`\makelabels` The `\makelabels` declaration causes mailing labels to be made.

```
\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.

```
  \AtBeginDocument{%
     \let\@startlabels\startlabels
     \let\@mlabel\mlabel
     \if@filesw
       \immediate\write\@mainaux{\string\@startlabels}\fi}%
```

At the end of the document we need to write `\clearpage` to the `.aux` file.

```
  \AtEndDocument{%
     \if@filesw\immediate\write\@mainaux{\string\clearpage}\fi}}
```

`\makelabels` is allowed only before the `\begin{document}` command.

```
\@onlypreamble\makelabels
```

5.2 The generic letter commands

`letter` (*env.*) 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}`.

```
\newenvironment{letter}[1]
  {\newpage
    \if@twoside \ifodd\c@page
                \else\thispagestyle{empty}\null\newpage\fi
    \fi
    \c@page \@ne
    \c@footnote \z@
    \interlinepenalty=200 % smaller than the TeXbook value
```

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

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

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

```
  {\stopletter\@@par\pagebreak\@@par
    \if@filesw
      \begingroup
        \let\\=\relax
        \let\protect\@unexpandable@protect
        \immediate\write\@auxout{%
          \string\@mlabel{\returnaddress}{\toname\\\toaddress}}%
      \endgroup
    \fi}
```

`\@processto`
`\@xproc`
`\@yproc`

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

```
\long\def\@processto#1{%
  \@xproc #1\\@@@%
  \ifx\toaddress\@empty
  \else
    \@yproc #1@@@%
  \fi}
```

```
\long\def\@xproc #1\\#2@@@{\def\toname{#1}\def\toaddress{#2}}
```

```
\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.

```
\newcommand*\stopbreaks{%
  \interlinepenalty\@M
  \def\par{\@@par\nobreak}%
  \let\\\@nobreakcr
  \let\vspace\@nobreakvspace}
```

`\nobreakvspace`
`\@nobreakvspacex`
`\@nobreakcr`

These are needed by `\stopbreaks`

```
\DeclareRobustCommand\@nobreakvspace
   {\@ifstar\@nobreakvspacex\@nobreakvspacex}
```

```
\def\@nobreakvspacex#1{%
  \ifvmode
```

```
    \nobreak\vskip #1\relax
  \else
    \@bsphack\vadjust{\nobreak\vskip #1}\@esphack
  \fi}
```

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

`\startbreaks` This cancels the effect of `\stopbreaks`.

```
\newcommand*\startbreaks{%
  \let\\\@normalcr
  \interlinepenalty 200%
  \def\par{\@@par\penalty 200\relax}}
```

`\longindentation` The size of the indent to use before the closing of the letter.

```
\newdimen\longindentation
\longindentation=.5\textwidth
```

`\indentedwidth` The width of the closing of the letter.

```
\newdimen\indentedwidth
\indentedwidth=\textwidth
\advance\indentedwidth -\longindentation
```

`\opening` Text is begun with the `\opening` command, whose argument generates the salutation, 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.
- `\telephonenum` : argument of current `\telephone` declaration–null if none.

```
\newcommand*\opening[1]{\ifx\@empty\fromaddress
  \thispagestyle{firstpage}%
    {\raggedleft\@date\par}%
  \else  % home address
    \thispagestyle{empty}%

    {\raggedleft\begin{tabular}{l@{}}\ignorespaces
      \fromaddress \\*[2\parskip]%
      \@date \end{tabular}\par}%
  \fi
  \vspace{2\parskip}%
```

If the address field used `\\[...]` then we have `\toaddress` starting with the bracket argument as the split was done simply at `\\`. So we add `\expandafter` here so that is will be used and not typeset. A better fix would be to do a proper parsing but ...

Other classes based on `letter.cls` could benefit from the same fix, e.g., `akletter.cls`, `bletter.cls`, `letter.cls`, `chletter.cls`, `extletter.cls`, `frletter.cls`, `hletter.cls`, `scrlttr2.cls` (change rejected), `lettre.cls`, `beletter.cls`,
`brief.cls` and perhaps others.

```
  {\raggedright \toname \expandafter\\\toaddress \par}%
  \vspace{2\parskip}%
  #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.

```
\newcommand\closing[1]{\par\nobreak\vspace{\parskip}%
  \stopbreaks
  \noindent
  \ifx\@empty\fromaddress\else
  \hspace*{\longindentation}\fi
  \parbox{\indentedwidth}{\raggedright
       \ignorespaces #1\\[6\medskipamount]%
       \ifx\@empty\fromsig
           \fromname
       \else \fromsig \fi\strut}%
   \par}
```

`\smallskipamount`
`\medskipamount`
`\bigskipamount`

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

```
%\smallskipamount=.5\parskip
\medskipamount=\parskip
%\bigskipamount=2\parskip
```

`\cc`
`\encl`
`\ps`

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

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

which produces:
cc: Tinker
Evers
Chance
Note the obvious use of `\parbox`.

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

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

which produces:

encl: Foo(2)
Bar

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

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

```
\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`.

```
\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.

```
\newcommand*\returnaddress{}
```

`\labelcount` A register to count the labels

```
\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.

```
\newcommand*\startlabels{\labelcount\z@
  \pagestyle{empty}%
  \let\@texttop\relax
  \topmargin -50\p@
  \headsep \z@
  \oddsidemargin -35\p@
  \evensidemargin -35\p@
  \textheight 10in
  \@colht\textheight  \@colroom\textheight \vsize\textheight
  \textwidth 550\p@
  \columnsep 26\p@
  \ifcase \@ptsize\relax
    \normalsize
  \or
    \small
  \or
    \footnotesize
  \fi
  \baselineskip \z@
  \lineskip \z@
  \boxmaxdepth \z@
  \parindent \z@
  \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.

```
\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 argument 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.

```
\newcommand*\mlabel[2]{%
  \parbox[b][2in][c]{262\p@}{\strut\ignorespaces #2}%
  }
```

`\@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.

```
\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 environment's parameters. See the LaTeX 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 `K`th 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`
`\leftmargini`
`\leftmarginii`
`\leftmarginiii`
`\leftmarginiv`
`\leftmarginv`
`\leftmarginvi`

For efficiency, level-one list's values are defined at top level, and `\@listi` is defined to set only `\leftmargin`.

```
\setlength\leftmargini   {2.5em}
```

The following three are calculated so that they are larger than the sum of `\labelsep` and the width of the default labels (which are '(m)', 'vii.' and 'M.').

```
\setlength\leftmarginii  {2.2em}
\setlength\leftmarginiii {1.87em}
\setlength\leftmarginiv  {1.7em}
\setlength\leftmarginv   {1em}
\setlength\leftmarginvi  {1em}
```

Here we set the top level leftmargin.

```
\setlength\leftmargin    {\leftmargini}
```

`\labelsep`
`\labelwidth`

`\labelsep` is the distance between the label and the text of an item; `\labelwidth` is the width of the label.

```
\setlength  \labelsep  {5\p@}
\setlength  \labelwidth{\leftmargini}
\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`.

```
\setlength\partopsep{0\p@}
```

`\@beginparpenalty`
`\@endparpenalty`

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

`\@itempenalty`

This penalty is inserted between list items.

```
\@beginparpenalty -\@lowpenalty
\@endparpenalty   -\@lowpenalty
\@itempenalty     -\@lowpenalty
```

`\@listI`
`\@listi`

`\@listI` defines top level and `\@listi` values of `\leftmargin`, `\parsep`, `\topsep`, and `\itemsep`

```
\def\@listI{\setlength\leftmargin{\leftmargini}%
            \setlength\parsep {0\p@}%
            \setlength\topsep {.4em}%
            \setlength\itemsep{.4em}}
\let\@listi\@listI
```

We have to initialize these parameters.

```
\@listi
```

`\@listii`
`\@listiii`
`\@listiv`
`\@listv`
`\@listvi`

Here are the same macros for the higher level lists.

```
\def\@listii {\setlength  \leftmargin{\leftmarginii}%
             \setlength  \labelwidth{\leftmarginii}%
             \addtolength\labelwidth{-\labelsep}}
\def\@listiii{\setlength  \leftmargin{\leftmarginiii}%
             \setlength  \labelwidth{\leftmarginiii}%
             \addtolength\labelwidth{-\labelsep}%
             \setlength  \topsep    {.2em}%
             \setlength  \itemsep   {\topsep}}
\def\@listiv {\setlength  \leftmargin{\leftmarginiv}%
             \setlength  \labelwidth{\leftmarginiv}%
             \addtolength\labelwidth{-\labelsep}}
\def\@listv  {\setlength  \leftmargin{\leftmarginv}%
             \setlength  \labelwidth{\leftmarginv}%
             \addtolength\labelwidth{-\labelsep}}
\def\@listvi {\setlength  \leftmargin{\leftmarginvi}%
             \setlength  \labelwidth{\leftmarginvi}%
             \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.

`\theenumi`
`\theenumii`
`\theenumiii`
`\theenumiv`

The counters are already defined in the LaTeX kernel (`ltlists.dtx`), but their representation is changed here.

```
\renewcommand\theenumi{\@arabic\c@enumi}
\renewcommand\theenumii{\@alph\c@enumii}
\renewcommand\theenumiii{\@roman\c@enumiii}
\renewcommand\theenumiv{\@Alph\c@enumiv}
```

`\labelenumi`
`\labelenumii`
`\labelenumiii`
`\labelenumiv`

The commands `\labelenumi` ... `\labelenumiv` generate the label for each item.

```
\newcommand\labelenumi{\theenumi.}
\newcommand\labelenumii{(\theenumii)}
\newcommand\labelenumiii{\theenumiii.}
\newcommand\labelenumiv{\theenumiv.}
```

`\p@enumii`
`\p@enumiii`
`\p@enumiv`

The expansion of `\p@enumN\theenumN` defines the output of a `\ref` command when referencing an item of the Nth level of an enumerated list.

```
\renewcommand\p@enumii{\theenumi}
\renewcommand\p@enumiii{\theenumi(\theenumii)}
\renewcommand\p@enumiv{\p@enumiii\theenumiii}
```

5.4.3 Itemize

`\labelitemi`
`\labelitemii`
`\labelitemiii`
`\labelitemiv`

Itemization is controlled by `\labelitemi`, `\labelitemii`, `\labelitemiii`, and `\labelitemiv`, which define the labels of the various itemization levels: the symbols used are bullet, bold en-dash, asterisk and centered dot.

```
\newcommand\labelitemi  {\labelitemfont \textbullet}
\newcommand\labelitemii {\labelitemfont \bfseries \textendash}
\newcommand\labelitemiii{\labelitemfont \textasteriskcentered}
\newcommand\labelitemiv {\labelitemfont \textperiodcentered}
```

`\labelitemfont` The default definition for `\labelitemfont` is to reset the font to `\normalfont` so that always the same symbol is produced regardless of surrounding conditions.

A possible alternative would be

```
\renewcommand\labelitemfont{%
   \fontseries\seriesdefault
   \fontshape\shapedefault\selectfont}
```

which resets series and shape doesn't touch the family.

```
\newcommand\labelitemfont{\normalfont}
```

5.4.4 Description

`description` (*env.*) The description environment is defined here – while the itemize and enumerate environments are defined in the LaTeX kernel (`ltlists.dtx`).

```
\newenvironment{description}
               {\list{}{\labelwidth\z@ \itemindent-\leftmargin
                        \let\makelabel\descriptionlabel}}
               {\endlist}
```

`\descriptionlabel` To change the formatting of the label, you must redefine `\descriptionlabel`.

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

5.5 Defining new environments

5.5.1 Verse

`verse` (*env.*) 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 `\let`'ing `\\` equal `\@centercr`.

```
\newenvironment{verse}
               {\let\\=\@centercr
                \list{}{\setlength\itemsep{\z@}%
                        \setlength\itemindent{-15\p@}%
                        \setlength\listparindent{\itemindent}%
                        \setlength\rightmargin{\leftmargin}%
                        \addtolength\leftmargin{15\p@}}%
                \item[]}
               {\endlist}
```

5.5.2 Quotation

`quotation` (*env.*) 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.

```
\newenvironment{quotation}
               {\list{}{\setlength\listparindent{1.5em}%
                        \setlength\itemindent{\listparindent}%
                        \setlength\rightmargin{\leftmargin}}%
                \item[]}
               {\endlist}
```

5.5.3 Quote

`quote` (*env.*) The quote environment is like the quotation environment except that paragraphs are not indented.

```
\newenvironment{quote}
               {\list{}{\setlength\rightmargin{\leftmargin}}%
                \item[]}
               {\endlist}
```

5.5.4 Theorem

This document class does not define it's own theorem environments, the defaults, supplied by the LaTeX 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`.

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

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

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

`\arrayrulewidth` The width of vertical rules in the array and tabular environments is given by `\arrayrulewidth`.

```
\setlength\arrayrulewidth{.4\p@}
```

`\doublerulesep` The space between adjacent rules in the array and tabular environments is given by `\doublerulesep`.

```
\setlength\doublerulesep{2\p@}
```

5.6.2 Tabbing

`\tabbingsep` This controls the space that the `\'` command puts in. (See LaTeX manual for an explanation.)

```
\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.

```
\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`.

```
\setlength\fboxsep{3\p@}
\setlength\fboxrule{.4\p@}
```

5.6.5 Equation and eqnarray

`\theequation` The equation counter will be typeset using arabic numbers.

```
\renewcommand\theequation{\@arabic\c@equation}
```

`\jot` `\jot` is the extra space added between lines of an eqnarray environment. The default value is used.

```
% \setlength\jot{3pt}
```

`\@eqnnum` The macro `\@eqnnum` defines how equation numbers are to appear in equations. Again the default is used.

```
% \def\@eqnnum{(\theequation)}
```

5.7 Font changing

Here we supply the declarative font changing commands that were common in LaTeX 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; LaTeX commands to execute in text mode and LaTeX commands to execute in math mode.

`\rm` `\tt` `\sf` The commands to change the family.

```
\DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\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.

```
\DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}
```

`\sl` `\it` `\sc` And the commands to change the shape of the font. The slanted and small caps shapes are not available by default as math alphabets, so those changes do nothing in math mode. One should use `\upshape` to explicitly change back to the upright shape.

```
\DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
\DeclareOldFontCommand{\sl}{\normalfont\slshape}{\relax}
\DeclareOldFontCommand{\sc}{\normalfont\scshape}{\relax}
```

`\cal` `\mit` The commands `\cal` and `\mit` should only be used in math mode, outside math 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'.

```
\DeclareRobustCommand*{\cal}{\@fontswitch{\relax}{\mathcal}}
\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.

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

`\c@footnote` A counter for footnotes.

```
% \newcounter{footnote}
```

`\@makefntext` The footnote mechanism of LaTeX 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 TeX code.

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

```
\newcommand\@makefntext[1]{%
    \noindent
    \hangindent 5\p@
    \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.

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

6 Initialization

6.1 Words

`\ccname` `\enclname` `\pagename` `\headtoname`

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

```
\newcommand*\ccname{cc}
\newcommand*\enclname{encl}
\newcommand*\pagename{Page}
\newcommand*\headtoname{To}
```

6.2 Date

`\today`

This macro uses the TEX primitives `\month`, `\day` and `\year` to provide the date of the LATEX-run.

```
\newcommand*\today{\ifcase\month\or
  January\or February\or March\or April\or May\or June\or
  July\or August\or September\or October\or November\or December\fi
  \space\number\day, \number\year}
```

6.3 Two column mode

`\columnsep`

This gives the distance between two columns in two column mode.

```
\setlength\columnsep{10\p@}
```

`\columnseprule`

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

```
\setlength\columnseprule{0\p@}
```

6.4 The page style

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

```
\pagestyle{plain}
\pagenumbering{arabic}
```

6.5 Single or double sided printing

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

```
\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`.

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

We always start in one column mode.

```
\onecolumn
⟨/letter⟩
```

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

\@@par 142, 161, 176
\@auxout 147
\@beginparpenalty 262
\@bsphack 170
\@colht 231
\@colroom 231
\@date 85, 92, 184, 189
\@eha 30
\@endparpenalty 262
\@eqnnum 339
\@esphack 170
\@evenfoot 81, 83, 97, 113
\@evenhead 81, 87, 97
\@fontswitch 347, 348
\@gobbletwo 250
\@hangfrom 212, 217
\@highpenalty 47
\@ifstar 165, 172
\@itempenalty 262
\@latexerr 29
\@listI 265
\@listi 265
\@listii 271
\@listiii 271
\@listiv 271
\@listv 271
\@listvi 271
\@lowpenalty 47, 262, 263, 264
\@mainaux 129, 131
\@makefnmark 357, 358
\@makefntext 354
\@medpenalty 47
\@minipagerestore 334
\@mlabel 127, 148, 250
\@mparswitchfalse 34
\@mparswitchtrue 32
\@mpfootins 334
\@nobreakcr 162, 164
\@nobreakvspace 163, 164
\@nobreakvspacex 164
\@normalcr 172, 174
\@oddfoot 81, 83, 90, 96, 100, 112
\@oddhead 81, 84, 87, 91, 96, 99, 111
\@onlypreamble 132
\@processto 141, 151
\@ptsize 1, 25, 26, 27, 41, 102, 234
\@roman 290
\@startlabels 126, 129, 246
\@tempdima 22, 24, 69, 70, 71, 72
\@texttop 225, 372
\@thefnmark 358
\@twosidefalse 34
\@twosidetrue 32
\@unexpandable@protect 146
\@width 351
\@xproc 151
\@yproc 151

A

\address 114, 121
\arraycolsep 329
\arrayrulewidth 331
\AtBeginDocument 125
\AtEndDocument 130

B

\baselineskip 241
\baselinestretch 44
\bf 343
\bigskipamount 206
\boxmaxdepth 243
\brokenpenalty 56

C

\c@enumi 288
\c@enumii 289
\c@enumiii 290
\c@enumiv 291
\c@equation 337
\c@footnote 139, 353
\c@page 135, 138, 372
\cal 347
\cc 209
\ccname 212, 359
\clearpage 131
\closing 195
\clubpenalty 50
\columnsep 233, 367
\columnseprule 368
\columnwidth 351

D

\DeclareOldFontCommand 340, 341, 342, 343, 344, 345, 346
\DeclareRobustCommand 164, 347, 348
description (env.) 304
\descriptionlabel 306, 308
\displaywidowpenalty 52
\doublerulesep 332

E

\encl 209
\enclname 217, 359
environments:
    description 304
    letter 133
    quotation 319
    quote 325
    verse 310
\evensidemargin 64, 229
\expandafter 192

F

\fboxrule 335
\fboxsep 335
\footins 80, 334
\footnoterule 349
\footnotesep 79
\footnotesize 107, 239
\footskip 59
\fromaddress 116, 119, 182, 188, 198
\fromlocation 109, 117, 119
\fromname 114, 119, 203
\fromsig 115, 119, 202, 204

H

\hb@xt@ 101, 357
\headheight 57
\headsep 57, 227
\headtoname 84, 91, 359

I

\if@compatibility 28, 60, 64
\if@filesw 128, 131, 143
\if@twoside 81, 135
\ifodd 135
\ignorespaces 84, 91, 141, 187, 201, 213, 218, 248
\immediate 129, 131, 147
\indentedwidth 179, 200
\interlinepenalty 55, 140, 160, 175
\it 344
\item 317, 323, 327
\itemindent 305, 313, 314, 321
\itemsep 268, 278, 312
\itshape 344

J

\jot 338

L

\labelcount 222, 223
\labelenumi 292
\labelenumii 292
\labelenumiii 292
\labelenumiv 292
\labelitemfont 299, 300, 301, 302, 303
\labelitemi 299
\labelitemii 299
\labelitemiii 299
\labelitemiv 299
\labelsep 258, 273, 276, 281, 284, 287, 308, 333
\labelwidth 258, 272, 273, 275, 276, 280, 281, 283, 284, 286, 287, 305
\leftmargin 251, 265, 271, 274, 279, 282, 285, 305, 315, 316, 322, 326
\leftmargini 251, 259, 265
\leftmarginii 251, 271, 272
\leftmarginiii 251, 274, 275
\leftmarginiv 251, 279, 280
\leftmarginv 251, 282, 283
\leftmarginvi 251, 285, 286
letter (env.) 133
\lineskip 42, 242
\listparindent 314, 320, 321
\location 114, 122
\longindentation 177, 181, 199

M

\makelabel 306
\makelabels 124
\marginparpush 76
\marginparsep 76
\marginparwidth 64
\mathbf 343
\mathcal 347
\mathit 344
\mathnormal 348
\mathrm 340
\mathsf 341
\mathtt 342
\medskipamount 201, 206
\mit 347
\mlabel 127, 247

N

\name 114, 119
\newcount 222
\nobreakvspace 164
\normalfont 112, 113, 212, 217, 303, 309, 340, 341, 342, 343, 344, 345, 346
\normallineskip 42
\normalsize 103, 235
\null 136

O

\oddsidemargin 64, 228
\onecolumn 373
\opening 182
\overfullrule 35, 36

P

\p@enumii 296
\p@enumiii 296
\p@enumiv 296
\pagebreak 142
\pagename 86, 93, 359
\pagenumbering 370
\paperheight 4, 7, 10, 13, 16, 19, 22, 23
\paperwidth 5, 8, 11, 14, 17, 20, 23, 24, 69
\parbox 200, 211, 216, 248
\parindent 45, 244
\parsep 266
\parskip 45, 188, 191, 193, 195, 206, 207, 208
\partopsep 261
\postdisplaypenalty 52
\predisplaypenalty 52
\ps 209
\ps@empty 95
\ps@firstpage 98
\ps@headings 81
\ps@plain 110

Q

quotation (env.) 319
quote (env.) 325

R

\raggedleft 184, 187
\raisebox 100
\returnaddress 148, 221
\rm 340

S

\sc 344
\scshape 346
\sf 340
\sffamily 341
\signature 114, 120
\sl 344
\small 105, 237
\smallskipamount 206
\startbreaks 173, 219
\startlabels 126, 223
\stopbreaks 159, 196
\stopletter 142, 220
\string 129, 131, 148
\strut 204, 213, 218, 248

T

\tabbingsep 333
\tabcolsep 330
\telephone 114, 123
\telephonenum 109, 118, 119
\textasteriskcentered 301
\textbullet 299

\textendash 300
\textheight 62, 230, 231
\textperiodcentered 302
\textwidth 60, 71, 101, 178, 180, 211, 216, 232
\theenumi 288, 292, 296, 297
\theenumii 288, 293, 297
\theenumiii 288, 294, 298
\theenumiv . . . 288, 295
\theequation . . 337, 339
\thepage 86, 93, 112, 113
\thispagestyle 136, 183, 186
\toaddress . . . 148, 153, 157, 158, 192
\today 363
\toname 84, 91, 148, 157, 192
\topmargin 78, 226
\topsep . . . 267, 277, 278
\tt 340
\ttfamily 342
\twocolumn 245

V

\vadjust 170
verse (env.) 310
\vsize 231

W

\widowpenalty 50
\write 129, 131, 147