Standard Letter Document Class for \LaTeX{} version 2e

Leslie Lamport and Frank Mittelbach and Rainer Schöpf

June 26, 2024

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
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 \langle∗\texttt{letter}\rangle
2 \newcommand\@ptsize{}

1.1 Setting Paper Sizes

The variables \texttt{\paperwidth} and \texttt{\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}\%

\section{Initial Code}

\section{Initialization}

\subsection{Words}

\subsection{Date}

\subsection{Two column mode}

\subsection{The page style}

\subsection{Single or double sided printing
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.

\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}{\twosidetruefalse \@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 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 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 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 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@}
\textwidth Now that we have computed the width of the text, we have to take care of the height. The \textwidth is the height of text (including footnotes and figures, excluding running head and foot).
\begin{verbatim}
62 \setlength\textwidth{505\p@}
63 \fi
\end{verbatim}

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
\textwidth The values of \oddsidemargin and \evensidemargin are computed from those of \paperwidth and \textwidth.
\begin{verbatim}
64 \if@compatibility
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
\end{verbatim}

\textwidth 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.
\begin{verbatim}
76 \setlength\marginparsep{11\p@}
77 \setlength\marginparpush{5\p@}
\end{verbatim}

\textwidth 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.

\textwidth It can now be computed from the values set above.
\begin{verbatim}
78 \setlength\topmargin{27pt}
\end{verbatim}

4.2.4 Footnotes
\textwidth \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.
\begin{verbatim}
79 \setlength\footnotesep{12\p@}
\end{verbatim}

\textwidth \skips\footins is the space between the last line of the main text and the top of the first footnote.
\begin{verbatim}
80 \setlength{\skips\footins}{10\p@ \@plus 2\p@ \@minus 4\p@}
\end{verbatim}
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 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.)

\@mkboth 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.

81 \if@twoside
82 \def\ps@headings{%
83 \let\@oddfoot\@empty\let\@evenfoot\@empty

The running feet are empty in this page style.
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\@oddhead\@empty\let\@oddfoot\@empty
\let\@evenhead\@empty\let\@evenfoot\@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@}{\[
\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 The following declarations, shown with examples, give information about the sender:
\signature \name{Dr. L. User}: to be used for the return address on the envelope.
\address
\location
\telephone
\begin{itemize}
\item \texttt{\signature{}\{Larry User\}}: goes after the closing.
\item \texttt{\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.
\item \texttt{\location{}\{Room 374\}}: Acts as modifier to the standard institutional address.
\item \texttt{\telephone{}\{(415)123-4567\}}: Just in case some style puts it on the letter.
\end{itemize}

\texttt{\makelabels} The \texttt{\makelabels} declaration causes mailing labels to be made.

\AtBeginDocument{\let\@startlabels\startlabels \let\@mlabel\mlabel \if@filesw \immediate\write\@mainaux{\string\@startlabels}\fi} %
\AtEndDocument{\if@filesw\immediate\write\@mainaux{\string\clearpage}\fi} %
\makelabels is allowed only before the \texttt{\begin{document}} command.

\subsection{The generic letter commands}
\texttt{\begin{letter}}\texttt{\{Sam Jones \}}\texttt{\{Institute for Retarded Study\}}\texttt{\{Princeton, N.J.\}}
Local declarations, such as `\address`, can follow the `\begin{letter}`.

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

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

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

```latex
\stopletter\@@par\pagebreak\@@par
\if@filesw
\begingroup
\let\\@nobreakcr
\let\vspace\@nobreakvspace
\interlinepenalty\@M
\def\par{\@@par\nobreak}
\if\toaddress\@empty
\else
\\yproc\#1000%
\fi
\long\def\@processto#1{\@xproc #1\@@@%
\ifx\toaddress\@empty
\else
\\yproc\#1000%
\fi
\long\def\@processto#1{\@xproc #1\@@@%
\ifx\toaddress\@empty
\else
\\yproc\#1000%
\fi
\long\def\@processto#1{\@xproc #1\@@@%
\ifx\toaddress\@empty
\else
\\yproc\#1000%
\fi
\long\def\toaddress\#1\#2000{\def\toaddress\#1\#2000}
```

### 5.2.1 Page breaking control

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

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

These are needed by `\stopbreaks`

```latex
\DeclareRobustCommand{\nobreakvspace}{
\ifstar\nobreakvspace\nobreakvspace}
```

```latex
\nobreakvspace
\nobreakvspace
\let\vspace\@nobreakvspace
```

---

133 \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
\leavevmode
\ignorespaces
\@processto\leavevmode\ignorespaces #1}

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

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

```latex
\stopletter\@@par\pagebreak\@@par
\if@filesw
\begingroup
\let\\@nobreakcr
\let\vspace\@nobreakvspace
\interlinepenalty\@M
\def\par{\@@par\nobreak}
\if\toaddress\@empty
\else
\\yproc\#1000%
\fi
\long\def\@processto#1{\@xproc #1\@@@%
\ifx\toaddress\@empty
\else
\\yproc\#1000%
\fi
\long\def\@processto#1{\@xproc #1\@@@%
\ifx\toaddress\@empty
\else
\\yproc\#1000%
\fi
\long\def\@processto#1{\@xproc #1\@@@%
\ifx\toaddress\@empty
\else
\\yproc\#1000%
\fi
\long\def\toaddress\#1\#2000{\def\toaddress\#1\#2000}
```

### 5.2.1 Page breaking control

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

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

These are needed by `\stopbreaks`

```latex
\DeclareRobustCommand{\nobreakvspace}{
\ifstar\nobreakvspace\nobreakvspace}

\nobreakvspace
\nobreakvspace
\let\vspace\@nobreakvspace
```

---

133 \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
\leavevmode
\ignorespaces
\@processto\leavevmode\ignorespaces #1}

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

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

```latex
\stopletter\@@par\pagebreak\@@par
\if@filesw
\begingroup
\let\\@nobreakcr
\let\vspace\@nobreakvspace
\interlinepenalty\@M
\def\par{\@@par\nobreak}
\if\toaddress\@empty
\else
\\yproc\#1000%
\fi
\long\def\@processto#1{\@xproc #1\@@@%
\ifx\toaddress\@empty
\else
\\yproc\#1000%
\fi
\long\def\@processto#1{\@xproc #1\@@@%
\ifx\toaddress\@empty
\else
\\yproc\#1000%
\fi
```

### 5.2.1 Page breaking control

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

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

These are needed by `\stopbreaks`

```latex
\DeclareRobustCommand{\nobreakvspace}{
\ifstar\nobreakvspace\nobreakvspace}

\nobreakvspace
\nobreakvspace
\let\vspace\@nobreakvspace
```

---

133 \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
\leavevmode
\ignorespaces
\@processto\leavevmode\ignorespaces #1}

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

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

```latex
\stopletter\@@par\pagebreak\@@par
\if@filesw
\begingroup
\let\\@nobreakcr
\let\vspace\@nobreakvspace
\interlinepenalty\@M
\def\par{\@@par\nobreak}
\if\toaddress\@empty
\else
\\yproc\#1000%
\fi
\long\def\@processto#1{\@xproc #1\@@@%
\ifx\toaddress\@empty
\else
\\yproc\#1000%
\fi
```

### 5.2.1 Page breaking control

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

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

These are needed by `\stopbreaks`

```latex
\DeclareRobustCommand{\nobreakvspace}{
\ifstar\nobreakvspace\nobreakvspace}

\nobreakvspace
\nobreakvspace
\let\vspace\@nobreakvspace
```

---

133 \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
\leavevmode
\ignorespaces
\@processto\leavevmode\ignorespaces #1}

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

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

```latex
\stopletter\@@par\pagebreak\@@par
\if@filesw
\begingroup
\let\\@nobreakcr
\let\vspace\@nobreakvspace
\interlinepenalty\@M
\def\par{\@@par\nobreak}
\if\toaddress\@empty
\else
\\yproc\#1000%
\fi
\long\def\@processto#1{\@xproc #1\@@@%
\ifx\toaddress\@empty
\else
```

### 5.2.1 Page breaking control

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

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

These are needed by `\stopbreaks`

```latex
\DeclareRobustCommand{\nobreakvspace}{
\ifstar\nobreakvspace\nobreakvspace}

\nobreakvspace
\nobreakvspace
\let\vspace\@nobreakvspace
```
\startbreaks This cancels the effect of \stopbreaks.
\newcommand*{\startbreaks}{\let\\@normalcr \interlinepenalty 200\relax \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:
\begin{itemize}
  \item \toname: name part of ‘to’ address. Will be one line long.
  \item \toaddress: address part of ‘to’ address. The lines separated by \.
  \item \fromname: name of sender.
  \item \fromaddress: argument of current \address declaration– null if none. Should use standard institutional address if null.
  \item \fromlocation: argument of current \location declaration– null if none.
  \item \telephonenum: argument of current \telephone declaration– null if none.
\end{itemize}
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, fletter.cls, hletter.cls, scrlttr2.cls (change rejected), lettre.cls, beletter.cls, brief.cls and perhaps others.

\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.

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

\cc 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:
\begin{verbatim}
encl: Foo(2)
Bar
\end{verbatim}

\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}
\labelcount 13
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 -50pt
\headsep z@ \oddsidemargin -35pt \evensidemargin -35pt
\textheight 10in
\@colht\textheight \@colroom\textheight \vsize\textheight
\textwidth 550pt
\columnsep 26pt
\ifcase\@ptsize\relax
\normalsize
\or
\small
\or
\footnotesize
\fi
\baselineskip z@ \lineskip z@ \boxmaxdepth z@ \parindent z@\twocolumn\relax}

\@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

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]{262pt}{\strut\ignorespaces #2}}

\@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 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.

\begin{verbatim}
\leftmargin \leftmargini \leftmarginii \leftmarginiii \leftmarginiv \leftmarginv \leftmarginvi
\end{verbatim}

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

\begin{verbatim}
251 \setlength{\leftmargini} {2.5em}
252 \setlength{\leftmarginii} {2.2em}
253 \setlength{\leftmarginiii} {1.87em}
254 \setlength{\leftmarginiv} {1.7em}
255 \setlength{\leftmarginv} {1em}
256 \setlength{\leftmarginvi} {1em}
\end{verbatim}

Here we set the top level leftmargin.

\begin{verbatim}
257 \setlength{\leftmargin} {\leftmargini}
\end{verbatim}

\begin{verbatim}
\labelsep \labelwidth
\end{verbatim}

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

\begin{verbatim}
258 \setlength{\labelsep} {5\p@}
259 \setlength{\labelwidth}{\leftmargini}
260 \addtolength{\labelwidth}{-\labelsep}
\end{verbatim}

\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.

\begin{verbatim}
261 \setlength{\partopsep} {0\p@}
\end{verbatim}

\@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.

\begin{verbatim}
262 \@beginparpenalty \-\@lowpenalty
263 \@endparpenalty \-\@lowpenalty
264 \@itempenalty \-\@lowpenalty
\end{verbatim}

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

\begin{verbatim}
265 \def\@listI{\setlength{\leftmargin}{\leftmargini}%%
266 \setlength{\parsep} {0\p@}%%
267 \setlength{\topsep} {.4em}%%
268 \setlength{\itemsep} {.4em}}
269 \let\@list\@listI
\end{verbatim}

We have to initialize these parameters.

\begin{verbatim}
270 \@listi
\end{verbatim}
5.4.2 Enumerate

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

\labelenumi \labelenumii \labelenumiii \labelenumiv

The commands \texttt{\labelenumi} ... \texttt{\labelenumiv} generate the label for each item.

\p@enumi \p@enumii \p@enumiii \p@enumiv

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

5.4.3 Itemize

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

\begin{verbatim}
\renewcommand\labelitemfont{\fontseries\seriesdefault \fontshape\shapedefault\selectfont}
\end{verbatim}

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\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}
\begin{list}{}
\setlength{rightmargin}{\leftmargin}
\item[]
\end{list}

5.5.4 Theorem

This document class does not define its 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.
\@minipagerestore
\@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 The commands to change the family.
\tt \DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\sf \DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\tt \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.
\it \DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}
\sl \DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
\sc \DeclareOldFontCommand{\sl}{\normalfont\slshape}{\relax}
\sc \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'.
\DeclareRobustCommand{\cal}{\@fontswitch{\relax}{\mathcal}}
\Decla\eRobustCommand{\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 \@tempdima-10pt \@tempdima} \par \parindent 1em\noindent \hb@xt@5\p@\{\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@\z@\{\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{$^\text{\@thefnmark}$}}

20
6 Initialization

6.1 Words

\ccname 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.

\enclname
\pagename
\headtoname

359 \newcommand*{\ccname}{cc}
360 \newcommand*{\enclname}{encl}
361 \newcommand*{\pagename}{Page}
362 \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.

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

6.3 Two column mode

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

367 \setlength{\columnsep}{10\p@}
\columnseprule This gives the width of the rule between two columns in two column mode. We have no visible rule.

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

6.4 The page style

We have \textit{plain} pages in this document class by default. We use arabic page numbers.

369 \pagestyle{plain}
370 \pagENUMBERing{arabic}

6.5 Single or double sided printing

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

371 \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 fill 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.

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

We always start in one column mode.

373 \onecolumn
374 \endletter
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

| @processto | 141, 151 |
| \columnseprule | 368 |
| \columnwidth | 351 |

D

| DeclareOldFontCommand | 340, 341, 342, 343, 344, 345, 346 |
| DeclareRobustCommand | 164, 347, 348 |

| description (env.) | 304 |
| descriptionlabel | |
| doublerulesep | 332 |

E

| encl | 209 |
| enclname | 217, 359 |
| environments: | |
| description | 304 |
| letter | 131 |
| quotation | 319 |
| quote | 325 |
| verse | 310 |
| even sided | |
| margin | 54, 229 |
| expandafter | 192 |

F

| fbboxrule | 335 |
| fboxsep | 335 |
| footins | 80, 334 |
| footnoterule | 349 |
| footnotesep | 79 |
| footnotegap | 107, 230 |
| footskip | 59 |
| fromaddress | 116, 119, 182, 188, 198 |
| fromlocation | 109, 117, 119 |
| fromname | 114, 119, 203 |
| fromnavigation | 115, 119, 202, 204 |

C

| @enumi | 288 |
| @enumii | 289 |
| @enumiii | 290 |
| @enumiv | 291 |
| @equation | 337 |
| @equation | 337 |
| @equation | 337 |
| @footnote | 139, 353 |
| @page | 135, 138, 372 |
| @cal | 347 |
| @cc | 209 |
| @cnum | 212, 359 |

H

| @bbxt@ | 101, 357 |
| @close | 195 |
| @headheight | 57 |
| @headsep | 57, 227 |
| @headtoname | 84, 91, 359 |