Standard Letter Document Class for \LaTeX version 2e

Leslie Lamport and Frank Mittelbach and Rainer Schöpf

December 6, 2021

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.

1 \texttt{@ptsize}
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 \texttt{\DeclareOption{a4paper}}
4 \texttt{\setlength{\paperheight}{297mm}\setlength{\paperwidth}{210mm}}
5 \texttt{\DeclareOption{a5paper}}
6 \texttt{\setlength{\paperheight}{210mm}\setlength{\paperwidth}{148mm}}
7 \texttt{\DeclareOption{b5paper}}
8 \texttt{\setlength{\paperheight}{148mm}\setlength{\paperwidth}{176mm}}
9 \texttt{\DeclareOption{letterpaper}}
10 \texttt{\setlength{\paperheight}{250mm}\setlength{\paperwidth}{11in}}
11 \texttt{\setlength{\paperheight}{11in}}
The option \texttt{landscape} switches the values of \texttt{paperheight} and \texttt{paperwidth}, assuming the dimensions were given for portrait paper.

\begin{verbatim}
\DeclareOption{landscape}
{\setlength\@tempdima {\paperheight}\
\setlength\paperheight {\paperwidth}\
\setlength\paperwidth {\@tempdima}}
\end{verbatim}

\subsection*{1.2 Choosing the type size}

The type size options are handled by defining \texttt{@ptsize} to contain the last digit of the size in question and branching on \texttt{ifcase} statements. This is done for historical reasons to stay compatible with other packages that use the \texttt{@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.

\begin{verbatim}
\DeclareOption{10pt}{\renewcommand{@ptsize{0}}}\
\DeclareOption{11pt}{\renewcommand{@ptsize{1}}}\
\DeclareOption{12pt}{\renewcommand{@ptsize{2}}}
\end{verbatim}

\subsection*{1.3 Two-side or one-side printing}

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

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

\subsection*{1.4 Draft option}

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

\begin{verbatim}
\DeclareOption{draft}{\setlength\overfullrule{5pt}}\
\DeclareOption{final}{\setlength\overfullrule{0pt}}
\end{verbatim}

\subsection*{1.5 Equation numbering on the left}

The option \texttt{leqno} can be used to get the equation numbers on the left side of the equation.

\begin{verbatim}
\DeclareOption{leqno}{\input{leqno.clo}}
\end{verbatim}
1.6 Flush left displays

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

\begin{verbatim}
38 \DeclareOption{fleqn}{\input{fleqn.clo}}
\end{verbatim}

2 Executing Options

Here we execute the default options to initialize certain variables.

\begin{verbatim}
39 \ExecuteOptions{letterpaper,10pt,oneside,onecolumn,final}
\end{verbatim}

The \texttt{\ProcessOptions} command causes the execution of the code for every option \texttt{foo} which is declared and for which the user typed the \texttt{foo} option in his \texttt{\documentclass} command. For every option \texttt{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 \texttt{\usepackage} command in the document preamble.

\begin{verbatim}
40 \ProcessOptions
Now that all the options have been executed we can load the chosen class option file that contains all size dependent code.
\end{verbatim}

\begin{verbatim}
41 \input{size1@ptsize.clo}
\end{verbatim}

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

\begin{verbatim}
\lineskip \normallineskip
These parameters control \TeX's behavior when two lines tend to come too close together.
\end{verbatim}

\begin{verbatim}
42 \setlength\lineskip{1\p@}
43 \setlength\normallineskip{1\p@}
\end{verbatim}

\begin{verbatim}
\baselinestretch
This is used as a multiplier for \texttt{\baselineskip}. The default is to \texttt{not} stretch the baselines.
\end{verbatim}

\begin{verbatim}
44 \renewcommand\baselinestretch{}
\end{verbatim}

\begin{verbatim}
\parskip \parindent
\parskip gives extra vertical space between paragraphs and \texttt{\parindent} is the width of the paragraph indentation. Letters are typeset without paragraph indentation.
\end{verbatim}

\begin{verbatim}
45 \setlength\parskip{0.7em}
46 \setlength\parindent{0\p@}
\end{verbatim}
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  \medpenalty  \highpenalty

\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@}
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).

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

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

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.

4.2.4 Footnotes

\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.

\skipline is the space between the last line of the main text and the top of the first footnote.
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).

`\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.)

`\ tex` 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 pagesyles `empty` and `plain` are defined in the `\TEX` kernel (`ltpage.dtx`), but these definitions are changed to a simpler version for this document class.

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

```
\ps@headings
\if@twoside
  \def\ps@headings{%
    \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\@oddhead\@empty
\def\@oddfoot{\normalfont\hfil\thepage\hfil}
\def\@evenfoot{\normalfont\hfil\thepage\hfil}}

\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{\textwidth{\hspace*{100\p@}
\ifcase \@ptsize\relax
\normalsize
\or
\small
\or
\footnotesize
\fi
\fromlocation \hfill \telephonenum}}

\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

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}}
\begin{itemize}
\item \texttt{\signature{Larry User}} \: goes after the closing.
\item \texttt{\newcommand*{\signature}[1]{\def\fromsig{#1}}} 
\item \texttt{\address{3245 Foo St.\,\textbackslash\,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{\newcommand*{\address}[1]{\def\fromaddress{#1}}} 
\item \texttt{\location{Room 374}} \: Acts as modifier to the standard institutional address.
\item \texttt{\newcommand*{\location}[1]{\def\fromlocation{#1}}} 
\item \texttt{\telephone{(415)123-4567}} \: Just in case some style puts it on the letter.
\end{itemize}

We make sure that the internal control sequences that are used to store the information exist and are empty.

\begin{verbatim}
\makelabels
\end{verbatim}

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

\begin{verbatim}
\AtBeginDocument{\let\@startlabels\startlabels \let\@mlabel\mlabel}
\if@filesw\immediate\write\@mainaux{\string\@startlabels}\fi}
\AtEndDocument{\if@filesw\immediate\write\@mainaux{\string\clearpage}\fi}
\end{verbatim}

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

\subsection{The generic letter commands}

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{verbatim}
\begin{letter}{Sam Jones \\ \\ \\ Institute for Retarded Study\ \ \ \ \ Princeton, N.J.}
\end{letter}
\end{verbatim}
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}
\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\make@relax
      \let\protect\@unexpandable@protect
      \immediate\write\@auxout{\string\@mlabel{\returnaddress}{\toname\toaddress}}
    \endgroup
  \fi}
```

\@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}
```

\@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}
```

5.2.1 Page breaking control

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

```
\newcommand*{\stopbreaks}{\let\par\@@par\nobreak}
\let\vspace\@nobreakvspace
```

\@nobreakvspace ex These are needed by \stopbreaks

```
\DeclareRobustCommand*{\nobreakvspace}{\makebox[0pt]{}}
\let\vspace\@nobreakvspace
```

\nobreakvspace ex These are needed by \stopbreaks

```
\DeclareRobustCommand*{\nobreakvspace}{\makebox[0pt]{}}
```

\nobreakvspace This is used by \stopbreaks

```
\DeclareRobustCommand*{\nobreakvspace}{\makebox[0pt]{}}
```
\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}%
\else % home address
\thispagestyle{empty}%
\fi
\vspace{2\parskip}%
If the address field used \[\ldots\] then we have \texttt{toaddress} starting with the bracket argument as the split was done simply at \. So we add \texttt{expandafter} here so that is will be used and not typeset. A better fix would be to do a proper parsing but \ldots

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.

\`\raggedright \toname \texttt{expandafter}"\texttt{toaddress} \par\`
\vspace{2\parskip}\`
\#1\par\nobreak\`
\texttt{closing} The body of the letter follows, ended by a \texttt{closing} command, as in
\`
\texttt{closing\{Yours truly,\}}
\`
This command generates the closing matter, and the signature. An obvious thing to do is to use a \texttt{parbox} for the closing and the signature. Should use the following:

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

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

\texttt{smallskipamount} Of these three, only \texttt{medskipamount} is actually used above.
\texttt{medskipamount} \`
\texttt{\bigskipamount} \%
\texttt{\smallskipamount=}=.5\parskip
\texttt{\medskipamount=}\parskip
\texttt{\bigskipamount=}2\parskip
\texttt{cc} After the \texttt{closing} you can put arbitrary stuff, which is typeset with zero \texttt{parindent} and no page breaking. Commands designed for use after the closing are:
\`
\texttt{cc}\{Tinker\ Evers\ Chance\}
\`
which produces:
\texttt{cc}: Tinker
Evers
Chance
Note the obvious use of \texttt{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 13
\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@
textwidth 550\p@
columnsep 26\p@
colht\textwidth \@colroom\textwidth \vsize\textwidth\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 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

\@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

14
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
For efficiency, level-one list’s values are defined at top level, and \@listi is defined to set only \leftmargin.
\end{verbatim}

\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}
Here we set the top level leftmargin.
257 \setlength\leftmargin {\leftmargini}
\end{verbatim}

\begin{verbatim}
\labelsep \labelwidth
\labelsep is the distance between the label and the text of an item; \labelwidth is the width of the label.
\end{verbatim}

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

\begin{verbatim}
\partopsep
\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@}
\end{verbatim}

\begin{verbatim}
\@beginparpenalty \@endparpenalty
These penalties are inserted before and after a list or paragraph environment.
\@beginparpenalty -@lowpenalty
\@endparpenalty -@lowpenalty
\@itempenalty -@lowpenalty
\@listI \@listi \@listI defines top level and \@listi values of \leftmargin, \parsep, \topsep, \itemsep
\end{verbatim}

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

\begin{verbatim}
265 \def\@listI{\setlength\leftmargin{\leftmargini}}%
266 \setlength\parsep {0\p@}%
267 \setlength\topsep {0.4em}%
268 \setlength\itemsep{0.4em}}
269 \let\@listi@\@listI
We have to initialize these parameters.
270 \@listi
\end{verbatim}
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: \texttt{enumi}, \texttt{enumii}, \texttt{enumiii} and \texttt{enumiv}, where \texttt{enumN} controls the numbering of the \texttt{N}th level enumeration.

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

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

\newcommand{\labelenumi}{\textbullet}
\newcommand{\labelenumii}{\bfseries \textendash}
\newcommand{\labelenumiii}{\textasteriskcentered}
\newcommand{\labelenumiv}{\textperiodcentered}

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.

\newcommand{\labelitemi}{\textbullet}
\newcommand{\labelitemii}{\bfseries \textendash}
\newcommand{\labelitemiii}{\textasteriskcentered}
\newcommand{\labelitemiv}{\textperiodcentered}
The default definition for \texttt{\labelitemfont} is to reset the font to \texttt{\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.

\subsection*{5.4.4 Description}

\texttt{description}  The description environment is defined here – while the itemize and enumerate environments are defined in the \LaTeX{} kernel (\texttt{ltlists.dtx}).
\begin{verbatim}
\newenvironment{description} {[}\list{}{\labelwidth\z@ \itemindent\leftmargin i\labelsep \let\makelabel\descriptionlabel}} {\endlist}
\end{verbatim}

\texttt{\descriptionlabel}  To change the formatting of the label, you must redefine \texttt{\descriptionlabel}.
\begin{verbatim}
\newcommand*{\descriptionlabel}[1]{\hspace{\labelsep}\normalfont\bfseries #1}
\end{verbatim}

\subsection*{5.5 Defining new environments}

\subsubsection*{5.5.1 Verse}

\texttt{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 \texttt{\let{\\@centercr}}
\begin{verbatim}
\newenvironment{verse} {[}\let\\@centercr \list{}{\setlength\itemsep{\z@} \setlength\itemindent{-15\p@} \setlength\rightmargin{\leftmargin} \addtolength\leftmargin{15\p@}}\item[]} {\endlist}
\end{verbatim}

\subsubsection*{5.5.2 Quotation}

\texttt{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 \texttt{\textwidth}. The first line of a paragraph inside this environment is indented.
\begin{verbatim}
\newenvironment{quotation} {[}\list{}{\setlength\listparindent{1.5em} \setlength\itemindent{\listparindent}\setlength\rightmargin{\leftmargin}}\item[]} {\endlist}
\end{verbatim}
5.5.3 Quote

\begin{quote}
The quote environment is like the quotation environment except that paragraphs are not indented.
\end{quote}

\begin{verbatim}
\newenvironment{quote}{\begin{list}{}\setlength\rightmargin{\leftmargin}}{%
\item[]}{\end{list}}
\end{verbatim}

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

\begin{itemize}
\item \texttt{arraycolsep} The columns in an array environment are separated by \texttt{2\arraycolsep}.
\item \texttt{\setlength\arraycolsep{5\p@}}
\item \texttt{tabcolsep} The columns in an tabular environment are separated by \texttt{2\tabcolsep}.
\item \texttt{\setlength\tabcolsep{6\p@}}
\item \texttt{\arrayrulewidth} The width of vertical rules in the array and tabular environments is given by \texttt{\arrayrulewidth}.
\item \texttt{\setlength\arrayrulewidth{.4\p@}}
\end{itemize}

5.6.2 Tabbing

\begin{itemize}
\item \texttt{\tabbingsep} This controls the space that the \texttt{\'} command puts in. (See \LaTeX manual for an explanation.)
\item \texttt{\setlength\tabbingsep{\labelsep}}
\end{itemize}

5.6.3 Minipage

\begin{itemize}
\item \texttt{\@minipagerestore} The macro \texttt{\@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.
\item \texttt{\@mpfootins} Minipages have their own footnotes; \texttt{\skip\@mpfootins} plays same rôle for footnotes in a minipage as \texttt{\skip\footins} does for ordinary footnotes.
\item \texttt{\setlength\@mpfootins = \skip\footins}
\end{itemize}
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.
335 \setlength\fboxsep{3\p@}
336 \setlength\fboxrule{.4\p@}

5.6.5 Equation and eqnarray
\theequation The equation counter will be typeset using arabic numbers.
337 \renewcommand\theequation{\@arabic\c@equation}
\jot \jot is the extra space added between lines of an eqnarray environment. The default value is used.
338 \% \setlength\jot{3pt}
\@eqnnum The macro \@eqnnum defines how equation numbers are to appear in equations. Again the default is used.
339 \% \def\@eqnnum{\theequation}

5.7 Font changing
Here we supply the declarative font changing commands that were common in \TeX 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; \TeX commands to execute in text mode and \TeX commands to execute in math mode.
\rm The commands to change the family.
\tt 340 \DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\sf 341 \DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
342 \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.
343 \DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}
\sl 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.
344 \DeclareOldFontCommand{\sl}{\normalfont\itshape}{\mathit}
345 \DeclareOldFontCommand{\sc}{\normalfont\ssshape}{\relax}
346 \ DeclareOldFontCommand{\sc}{\normalfont\ssshape}{\relax}
\cal 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’.
347 \DeclareRobustCommand*{\cal}{\@fontswitch{\relax}{\mathcal}}
348 \ 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 began 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 \@makefmark 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
  \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@5\p@{\hss\@makefnmark#1}}

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

%\renewcommand\@makefmark{$^\m@th\@thefnmark$}
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\columnseprule

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

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

\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
\endinput
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 \ldots 142, 161, 176
\@auxout \ldots 147
\begin{parpenalty} \ldots 262
\@mboxattach \ldots 170
\@colht \ldots 231
\@colroom \ldots 231
\@date \ldots 85, 92, 184, 189
\@eha \ldots 30
\@endparpenalty \ldots 262
\@eqnum \ldots 339
\@epsbox \ldots 170
\@evenfoot \ldots 81, 83, 97, 113
\@evenhead \ldots 81, 87, 97
\@fosswitch \ldots 343, 348
\@gobbletwo \ldots 250
\@hangfrom \ldots 212, 217
\@highpenalty \ldots 47
\@ifstar \ldots 165, 172
\@itempenalty \ldots 262
\@latexerr \ldots 29
\@listi \ldots 265
\@listii \ldots 271
\@listiii \ldots 271
\@listiv \ldots 271
\@listv \ldots 271
\@listvi \ldots 271
\@lowpenalty \ldots 47, 262, 263, 264
\@mainaux \ldots 129, 131
\@marking \ldots 357, 358
\@maketag \ldots 354
\@medpenalty \ldots 47
\@minipagerestore \ldots 474
\@minipagemark \ldots 434
\@minipage \ldots 127, 148, 250
\@parswitchfalse \ldots 34
\@parswitchtrue \ldots 32
\@pfootins \ldots 334
\@pfootnote \ldots 334
\@breakcr \ldots 162, 164
\@breakvspace \ldots 163, 164
\@breakvpace \ldots 164
\@normalcr \ldots 172, 174
\@oddfoot \ldots 81, 83, 90, 96, 100, 112
\@oddhead \ldots 81, 84, 87, 91, 96, 99, 111
\@onlypreamble \ldots 132
\atprocessto \ldots 141, 151
\atopsize \ldots L 25, 26, 27, 41, 102, 234
\atstartlabels \ldots 126, 129, 246
\atempspace \ldots 22
\atexttop \ldots 225, 372
\atempnum \ldots 358
\atwosidefalse \ldots 34
\atwosidetru \ldots 32
\atunexpandable@protect \ldots 146
\atwidth \ldots 351
\atproc \ldots 151
\atypc \ldots 151
\@address \ldots 114, 121
\@arraycolsep \ldots 320
\@arrayrulewidth \ldots 331
\@atbegindocument \ldots 125
\@atenddocument \ldots 130
\@baselinestretch \ldots 44
\@bf \ldots 343
\@bigskipamount \ldots 296
\@boxmaxdepth \ldots 243
\@brokenpenalty \ldots 56
\@c@enumi \ldots 288
\@c@enumii \ldots 289
\@c@enumiii \ldots 290
\@c@enumiv \ldots 291
\@c@enquote \ldots 337
\@c@footnote \ldots 139, 353
\@c@footnotemark \ldots 135, 138, 372
\@c@footnotetext \ldots 347
\@c@fbox \ldots 209
\@c@fboxsep \ldots 212, 259
\@c@clearpage \ldots 131
\@c@closing \ldots 195
\@c@clubpenalty \ldots 50
\@c@columnsep \ldots 233, 367
\@c@columnseprule \ldots 368
\@c@columnwidth \ldots 351
\@c@description \ldots 304
\@c@letter \ldots 134
\@c@quote \ldots 319
\@c@quote \ldots 325
\@c@verse \ldots 310
\@c@evensidemargin \ldots 84, 229
\@c@expandafter \ldots 192
\@c@fboxrule \ldots 335
\@c@fboxsep \ldots 335
\@c@footnotesize \ldots 70, 353
\@c@footnoterule \ldots 349
\@c@footnotesep \ldots 79
\@c@footnotesize \ldots 70, 123, 198
\@c@fromaddress \ldots 116, 119, 182, 188, 198
\@c@fromlocation \ldots 109, 117, 119
\@c@fromnames \ldots 114, 119, 203
\@c@fromsize \ldots 115, 119, 202, 204
\@c@hb@xt@ \ldots 101, 357
\@c@headheight \ldots 37
\@c@headsep \ldots 57, 227
\@c@headsep \ldots 57, 227