Standard Letter Document Class for \LaTeX version 2e

Leslie Lamport and Frank Mittelbach and Rainer Schöpf

February 20, 2020

This file is maintained by the \LaTeX Project team.
Bug reports can be opened (category \verb|latex|) at
\url{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 ................................................... 14
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 \(*\text{letter}\)
2 \texttt{\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}\%}
5 \texttt{\setlength\paperwidth {210mm}}
6 \texttt{\DeclareOption{a5paper}}
7 \texttt{\setlength\paperheight {210mm}\%}
8 \texttt{\setlength\paperwidth {148mm}}
9 \texttt{\DeclareOption{b5paper}}
10 \texttt{\setlength\paperheight {250mm}\%}
11 \texttt{\setlength\paperwidth {176mm}}
12 \texttt{\DeclareOption{letterpaper}}
13 \texttt{\setlength\paperheight {11in}\%}
14 \texttt{\setlength\paperwidth {8.5in}}
15 \texttt{\DeclareOption{legalpaper}}
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}{\twosidetrue \@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}{\overfullrule{5pt}}
\DeclareOption{final}{\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 \texttt{mathindent} from the prevailing left margin.

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

2 Executing Options

Here we execute the default options to initialize certain variables.

\begin{verbatim}
\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}
\ProcessOptions
\end{verbatim}

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

\begin{verbatim}
\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
\end{verbatim}

These parameters control \TeX{}’s behavior when two lines tend to come too close together.

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

\begin{verbatim}
\baselinestretch
\end{verbatim}

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

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

\begin{verbatim}
\parskip \parindent
\end{verbatim}

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

\begin{verbatim}
\setlength\parskip{0.7em}
\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.

\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

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).

\setlength{\textheight}{505\p@}
\fi

In native mode we use the dimensions as they were computed by one of the \textwidthoptions, 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.

\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

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@}
\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@}

4.2.4 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.

\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 \texttt{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, 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 \@markboth 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{%
\let\@oddfoot\@empty\let\@evenfoot\@empty
\endinput
\fi
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@}\[
\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
The following declarations, shown with examples, give information about the
sender:
\name\signature\address\location\telephone

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

\makelabels

\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 \cleardoublepage to the .aux file.

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

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

\onlypreamble\makelabels

5.2 The generic letter commands

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

\newenvironment{letter}{\newpage
\if@twoside \ifodd\c@page
\else\thispagestyle{empty}\null\newpage\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.

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

\@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}

These are needed by \stopbreaks

\Stopbreaks
When the command \Stopbreaks is issued no page breaks should occur until \Startbreaks is called.

\nobreakvspace
\DeclareRobustCommand\nobreakvspace
\nobreakvspace
\nobreakcr
These are needed by \Stopbreaks
This cancels the effect of \texttt{\stopbreaks}.

\texttt{\startbreaks} This cancels the effect of \texttt{\stopbreaks}.

\texttt{\longindentation} The size of the indent to use before the closing of the letter.

\texttt{\indentedwidth} The width of the closing of the letter.

\texttt{\opening} Text is begun with the \texttt{\opening} command, whose argument generates the salutation, as in

\texttt{\opening\{Dear Henry,\}}

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

- \texttt{\toname}: name part of ‘to’ address. Will be one line long.
- \texttt{\toaddress}: address part of ‘to’ address. The lines separated by \texttt{\\}.
- \texttt{\fromname}: name of sender.
- \texttt{\fromaddress}: argument of current \texttt{\address} declaration—null if none. Should use standard institutional address if null.
- \texttt{\fromlocation}: argument of current \texttt{\location} declaration—null if none.
- \texttt{\telephone}: argument of current \texttt{\telephone} declaration—null if none.
\textbf{The body of the letter follows, ended by a 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.

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

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

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

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

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:

cc: Tinker
Evers
Chance

Note the obvious use of \texttt{\parbox}.

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

which produces:

cc: Foo(2)
Bar

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

which produces:

encl: Foo(2)
Bar
The only thing \ps needs to do is call \startbreaks, which allows page breaking again.

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

\texttt{\stopletter} The \texttt{\stopletter} command is called by \texttt{\endletter} to do the following:

- Add any desired fill or other material at the end of the letter.
- Define \texttt{\returnaddress} to be the return address for the mailing label. More precisely, it is the first argument of the \texttt{\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 \texttt{\mlabel} command is actually executed must be preceded by \texttt{\protect}. Whenever possible, \texttt{\protect} commands in the definition of \texttt{\returnaddress}—it’s much more efficient that way. In particular, when the standard return address is used, you should define \texttt{\returnaddress} to something like \texttt{\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 \texttt{\end{document}} command. You have to define the following two commands:

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

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

\newcommand*{\returnaddress}{}

\texttt{\labelcount} A register to count the labels

\newcount\labelcount

\texttt{\startlabels} The following \texttt{\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
\normalsize
\fi}

\texttt{\@colht\textheight} \@colroom\textwidth \vsize\textwidth
\textwidth 550\p@
\columnsep 26\p@
\ifcase \@ptsize
\normalsize
\fi

13
\@startlabels \@startlabels is the command name that is written to the \input{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 \input{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 \textit{K}th level list, the command \@listK is called, where \texttt{‘K’} denotes ‘i’, ‘ii’, ..., ‘vi’. (I.e., \@listiii is called for a third-level list.) By convention, \@listK should set \leftmargin to \leftmarginK.

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

\leftmargini \leftmarginii \leftmarginiii \leftmarginiv \leftmarginv \leftmarginvi

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.’).
Here we set the top level left margin.

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

\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.

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.

This penalty is inserted between list items.

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

We have to initialize these parameters.

Here are the same macros for the higher level lists.
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.

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

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

\begin{verbatim}
\renewcommand\p@enumii{\theenumi}
\renewcommand\p@enumiii{\theenumi(\theenumii)}
\renewcommand\p@enumiv{\p@enumiii\theenumiii}
\end{verbatim}

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.

\begin{verbatim}
\newcommand{\labelitemi}{\textbullet}
\newcommand{\labelitemii}{\normalfont\bfseries \textendash}
\newcommand{\labelitemiii}{\textasteriskcentered}
\newcommand{\labelitemiv}{\textperiodcentered}
\end{verbatim}

5.4.4 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}
\let\makelabel\descriptionlabel}}{\endlist}
\end{verbatim}

\begin{verbatim}
\newcommand*{\descriptionlabel}[1]{\hspace*{\labelsep}\normalfont\bfseries #1}
\end{verbatim}

To change the formatting of the label, you must redefine \texttt{\descriptionlabel}.
5.5 Defining new environments

5.5.1 Verse
verse The verse environment is defined by making clever use of the list environment’s parameters. The user types `\` to end a line. This is implemented by `\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 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}\%
 \addtolength\leftmargin{15\p@}\%}
 \item[\]}
{\endlist}
```

5.5.3 Quote
quote 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 a 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.
330 \setlength{\arrayrulewidth}{.4\p@}

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

5.6.2 Tabbing
\tabbingsep This controls the space that the \' command puts in. (See \LaTeX{} manual for an explanation.)
332 \setlength{\tabbingsep}{\labelsep}

5.6.3 Minipage
\minipagerestore The macro \minipagerestore is called upon entry to a minipage environment to set up things that are to be handled differently inside a minipage environment. In the current styles, it does nothing.
\@mpfootins Minipages have their own footnotes; \skip\@mpfootins plays same rôle for footnotes in a minipage as \skip\footins does for ordinary footnotes.
333 \skip\@mpfootins = \skip\footins

5.6.4 Framed boxes
\fboxsep The space left by \fbox and \framebox between the box and the text in it.
\fboxrule The width of the rules in the box made by \fbox and \framebox.
334 \setlength{\fboxsep}{3\p@}
335 \setlength{\fboxrule}{.4\p@}

5.6.5 Equation and eqnarray
\theequation The equation counter will be typeset using arabic numbers.
336 \renewcommand{\theequation}{\@arabic{\c@equation}}
\jot \jot is the extra space added between lines of an eqnarray environment. The default value is used.
337 \% \setlength{\jot}{3pt}
\@eqnum The macro \@eqnum defines how equation numbers are to appear in equations. Again the default is used.
338 \% \def{\@eqnum}{{\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 \textit{and} in math mode. They are provided for compatibility, but one should start using the \texttt{\textbackslash text...} \texttt{and \textbackslash math...} commands instead. These commands are redefined using \texttt{\textbackslash @renewfontswitch}, a command with three arguments: the user command to be defined; \LaTeX commands to execute in text mode \texttt{and} \LaTeX commands to execute in math mode.

\texttt{\textbackslash rm} \ The commands to change the family.
\texttt{\textbackslash tt} \ DeclareOldFontCommand\{\textbackslash rm\}\{\texttt{\textbackslash normalfont\textbackslash rmfamily}\}\{\texttt{\textbackslash mathrm}\}
\texttt{\textbackslash sf} \ DeclareOldFontCommand\{\textbackslash sf\}\{\texttt{\textbackslash normalfont\textbackslash sffamily}\}\{\texttt{\textbackslash mathsf}\}
\texttt{\textbackslash tt} \ DeclareOldFontCommand\{\textbackslash tt\}\{\texttt{\textbackslash normalfont\textbackslash ttfamily}\}\{\texttt{\textbackslash mathtt}\}

\texttt{\textbackslash bf} \ The command to change to the bold series. One should use \texttt{\textbackslash mdseries} to explicitly switch back to medium series.
\texttt{\textbackslash sl} \ DeclareOldFontCommand\{\textbackslash bf\}\{\texttt{\textbackslash normalfont\textbackslash bseries}\}\{\texttt{\textbackslash mathbf}\}
\texttt{\textbackslash it} \ DeclareOldFontCommand\{\textbackslash it\}\{\texttt{\textbackslash normalfont\textbackslash itshape}\}\{\texttt{\textbackslash mathit}\}
\texttt{\textbackslash sc} \ DeclareOldFontCommand\{\textbackslash sl\}\{\texttt{\textbackslash normalfont\textbackslash slshape}\}\{\texttt{\relax}\}
\texttt{\textbackslash sc} \ DeclareOldFontCommand\{\textbackslash sc\}\{\texttt{\textbackslash normalfont\textbackslash scshape}\}\{\texttt{\relax}\}

\texttt{\textbackslash cal} \ The commands \texttt{\textbackslash cal} and \texttt{\textbackslash 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’.
\texttt{\textbackslash cal} \ DeclareRobustCommand\*\{\textbackslash cal\}\{\texttt{\textbackslash @fontswitch\{\textbackslash relax\}\}\{\texttt{\textbackslash mathcal}\}\}
\texttt{\textbackslash mit} \ DeclareRobustCommand\*\{\textbackslash mit\}\{\texttt{\textbackslash @fontswitch\{\textbackslash relax\}\}\{\texttt{\textbackslash mathnormal}\}\}

5.8 Footnotes

\texttt{\textbackslash footnoterule} \ Usually, footnotes are separated from the main body of the text by a small rule. This rule is drawn by the macro \texttt{\textbackslash footnoterule}. We have to make sure that the rule takes no vertical space (see \texttt{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.
\texttt{\textbackslash renenewcommand\textbackslash footnoterule\{%}
\texttt{\kern\textminus\textbackslash p@}\texttt{\textbackslash hrule \textbackslash @width 0.4\textbackslash columnwidth}\texttt{\kern 0.6\textbackslash p@\}}

\texttt{\textbackslash c@footnote} \ A counter for footnotes.
\texttt{\% \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.

\begin{verbatim}
\long\def\@makefntext#1{%
\setpar{\@@par
\@tempdima = \hsize
\advance\@tempdima-10pt
\parshape \@ne 10pt \@tempdima}%
\par
\parindent 1em\noindent
\hb@xt@5\p@{\hss\@makefnmark}#1}
\end{verbatim}

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.

\begin{verbatim}
\long\def\@makefntext#1{%
\noindent
\hangindent 5\p@\hb@xt@5\p@{\hss\@makefnmark}#1}
\end{verbatim}

\@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.

5.9 Words

\begin{verbatim}
\%\def\@makefnmark{\hbox{$^\{\thefnmark\m@th$}}
\end{verbatim}

5.10 Date

\begin{verbatim}
\today\thispagestyle{plain}
\end{verbatim}
5.11 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@}

5.12 The page style

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

5.13 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
\end(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.

<table>
<thead>
<tr>
<th>Symbols</th>
<th>$@$eqnum</th>
<th>$@$par</th>
<th>$@$auxout</th>
<th>$@$beginparpenalty</th>
<th>$@$bsphack</th>
<th>$@$colht</th>
<th>$@$coldroom</th>
<th>$@$date</th>
<th>$@$eha</th>
<th>$@$endparpenalty</th>
<th>$@$eqnnum</th>
<th>$@$latexerr</th>
<th>$@$listI</th>
<th>$@$listii</th>
<th>$@$listiii</th>
<th>$@$listiv</th>
<th>$@$listv</th>
<th>$@$listvi</th>
<th>$@$listvii</th>
<th>$@$listviii</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>338</td>
<td>142, 161, 176</td>
<td>170</td>
<td>81, 83, 97, 113</td>
<td>265</td>
<td>346, 347</td>
<td>250</td>
<td>212, 217</td>
<td>165, 172</td>
<td>262</td>
<td>165, 172</td>
<td>265</td>
<td>47</td>
<td>47</td>
<td>47, 262, 263, 264</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>