NTG Document Class `brief` for LaTeX version 2e

Victor Eijkhout Johannes Braams

January 10, 2023

Contents

1 Introduction

This file contains the document class `brief` that was made available by Working Group 13 of the NTG (Nederlandstalige TEX Gebruikersgroep). It defines more commands than the standard document class `letter`, but a letter made with the `letter` document class is still processable with this document class.

2 Initial Code

In this part we define a few comands 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.

```
\langle*brief\rangle
\newcommand*\@ptsize{}
```

`\if@typhulp` This switch is used to decide whether or not to put a small line on the paper that is used to align the paper in a typewriter.

```
\newif\if@typhulp
```

`\if@streepjes` A switch to indicate if the 'folding lines' should be printed

```
\newif\if@streepjes
```

`\if@adresrechts` This switch indicates if the addressing information is to be set on the left or on the right side of the letter.

```
\newif\if@adresrechts
```

`\if@elfinch` A switch to remember whether we are using A4 or letter paper. (possibly obsolete)

```
\newif\if@elfinch
```

2.1 Setting Paper Sizes

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

```
\DeclareOption{a4paper}
   {\setlength\paperheight {297mm}%
    \setlength\paperwidth  {210mm}\@elfinchfalse}
\DeclareOption{a5paper}
   {\ClassWarning{brief}{Paper size A5 not supported, using A4}%
    \setlength\paperheight {297mm}%
    \setlength\paperwidth  {210mm}\@elfinchfalse}
\DeclareOption{b5paper}
   {\ClassWarning{brief}{Paper size B5 not supported, using A4}%
    \setlength\paperheight {297mm}%
    \setlength\paperwidth  {210mm}\@elfinchfalse}
\DeclareOption{letterpaper}
   {\setlength\paperheight {11in}%
    \setlength\paperwidth  {8.5in}\@elfinchtrue}
\DeclareOption{USletter}
   {\setlength\paperheight {11in}%
    \setlength\paperwidth  {8.5in}\@elfinchtrue}
\DeclareOption{legalpaper}
   {\ClassWarning{brief}
                 {Paper size 'legal' not supported, using 'letter'}%
    \setlength\paperheight {14in}%
    \setlength\paperwidth  {8.5in}\@elfinchtrue}
\DeclareOption{executivepaper}
   {\ClassWarning{brief}
```

```
                {Paper size 'executive' not supported, using 'letter'}%
  \setlength\paperheight {10.5in}%
  \setlength\paperwidth  {7.25in}\@elfinchtrue}
```

2.2 Choosing the type size

The type size options are handled by defining `\@ptsize` to contain the last digit of the size in question and branching on `\ifcase` statements. This is done for historical reasons to stay compatible with other packages that use the `\@ptsize` variable to select special actions. It makes the declarations of size options less than 10pt difficult, although one can probably use `9` and `8` assuming that a class wont define both `8pt` and `18pt` options.

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

2.3 Two-side or one-side printing

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

```
\if@compatibility
  \DeclareOption{twoside}{\@latexerr{No 'twoside' layout for letters}%
                                    \@eha}
\else
  \DeclareOption{twoside}{\@twosidetrue  \@mparswitchtrue}
\fi
\DeclareOption{oneside}{\@twosidefalse \@mparswitchfalse}
```

2.4 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}}
```

2.5 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}}
```

2.6 Flush left displays

The option `fleqn` redefines the displayed math environmens 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}}
```

2.7 Typewriter alignment

```
\DeclareOption{typhulp}{\@typhulptrue}
\DeclareOption{geentyphulp}{\@typhulpfalse}
```

2.8 Folding lines

It is possible to print 'folding lines' on the far right side of the paper.

```
\DeclareOption{streepjes}{\@streepjestrue}
\DeclareOption{geenstreepjes}{\@streepjesfalse}
```

2.9 Address placement

The address information can be put either on the left or on the right side of the letter

```
\DeclareOption{adreslinks}{\@adresrechtsfalse}
\DeclareOption{adresrechts}{\@adresrechtstrue}
```

2.10 Support for different languages

In the original document style `brief` the options to support the various languages were all dutch words. To be compatible with both the old version of the document class and with the recommended set of language options we have at least two options for each language.

First Dutch,

```
\DeclareOption{nederlands}{\AtEndOfClass{\dutchbrief}}
\DeclareOption{dutch}     {\AtEndOfClass{\dutchbrief}}
```

then British English,

```
\DeclareOption{engels}    {\AtEndOfClass{\englishbrief}}
\DeclareOption{english}   {\AtEndOfClass{\englishbrief}}
```

American English,

```
\DeclareOption{USengels}  {\AtEndOfClass{\americanbrief}}
\DeclareOption{american}  {\AtEndOfClass{\americanbrief}}
```

German

```
\DeclareOption{duits}     {\AtEndOfClass{\germanbrief}}
\DeclareOption{german}    {\AtEndOfClass{\germanbrief}}
```

and finally french.

```
\DeclareOption{frans}     {\AtEndOfClass{\frenchbrief}}
\DeclareOption{french}    {\AtEndOfClass{\frenchbrief}}
\DeclareOption{francais}  {\AtEndOfClass{\frenchbrief}}
```

3 Executing Options

Here we execute the default options to initialize certain variables.

```
\ExecuteOptions{a4paper,11pt,oneside,onecolumn,final,%
```

```
                geentyphulp,geenstreepjes,adreslinks,%
                nederlands}
```

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\relax
```

Now that all the options have been executed we can define the user-level size changing commands. Their definition depends on which of the 10pt, 11pt or 12pt options was specified.

`\normalsize` `\@normalsize`

The user level command for the main size is `\normalsize`. Internally LaTeX uses `\@normalsize` when it refers to the main size. `\@normalsize` will be defined to work like `\normalsize` if the latter is redefined from its default definition (that just issues an error message). Otherwise `\@normalsize` simply selects a 10pt/12pt size.

The `\normalsize` macro also sets new values for `\abovedisplayskip`, `\abovedisplayshortskip` and

```
\ifcase\@ptsize
  \renewcommand*\normalsize{%
    \@setfontsize\normalsize\@xpt\@xiipt
    \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
    \abovedisplayshortskip \z@ \@plus3\p@
    \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
    \belowdisplayskip \abovedisplayskip
    \let\@listi\@listI}
\or
  \renewcommand*\normalsize{%
    \@setfontsize\normalsize\@xipt{13.6}%
    \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
    \abovedisplayshortskip \z@ \@plus3\p@
    \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
    \belowdisplayskip \abovedisplayskip
    \let\@listi\@listI}
\or
  \renewcommand*\normalsize{%
    \@setfontsize\normalsize\@xiipt{15}%
    \abovedisplayskip 12\p@ \@plus3\p@ \@minus7\p@
    \abovedisplayshortskip \z@ \@plus3\p@
    \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
    \belowdisplayskip \abovedisplayskip
    \let\@listi\@listI}
\fi
```

Make `\@normalsize` a synonymn for `\normalsize`.

```
\let\@normalsize\normalsize
```

We initially choose the normalsize font.

```
\normalsize
```

We use `\MakeRobust` instead of `\DeclareRobustCommand` above to avoid a log entry for the redefinition. But if we are running in a rollback situation (prior to 2015) we don't touch it.

```
\ifx\MakeRobust\@undefined \else
    \MakeRobust\normalsize
\fi
```

`\small` This is similar to `\normalsize`.

```
\ifcase\@ptsize
  \DeclareRobustCommand\small{%
    \@setfontsize\small\@ixpt{11}%
    \abovedisplayskip 8.5\p@ \@plus3\p@ \@minus4\p@
    \abovedisplayshortskip \z@ \@plus2\p@
    \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
    \belowdisplayskip \abovedisplayskip}
\or
  \DeclareRobustCommand\small{%
    \@setfontsize\small\@xpt\@xiipt
    \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
    \abovedisplayshortskip \z@ \@plus3\p@
    \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
    \belowdisplayskip \abovedisplayskip}
\or
  \DeclareRobustCommand\small{%
    \@setfontsize\small\@xipt{13.6}%
    \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
    \abovedisplayshortskip \z@ \@plus3\p@
    \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
    \belowdisplayskip \abovedisplayskip}
\fi
```

`\footnotesize` This is similar to `\normalsize`.

```
\ifcase\@ptsize
  \DeclareRobustCommand\footnotesize{%
    \@setfontsize\footnotesize\@viiipt{9.5}%
    \abovedisplayskip 6\p@ \@plus2\p@ \@minus4\p@
    \abovedisplayshortskip \z@ \@plus\p@
    \belowdisplayshortskip 3\p@ \@plus\p@ \@minus2\p@
    \belowdisplayskip \abovedisplayskip}
\or
  \DeclareRobustCommand\footnotesize{%
    \@setfontsize\footnotesize\@ixpt{11}%
    \abovedisplayskip 8\p@ \@plus2\p@ \@minus4\p@
    \abovedisplayshortskip \z@ \@plus\p@
    \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
    \belowdisplayskip \abovedisplayskip}
\or
```

```
  \DeclareRobustCommand\footnotesize{%
    \@setfontsize\footnotesize\@xpt\@xiipt
    \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
    \abovedisplayshortskip \z@ \@plus3\p@
    \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
    \belowdisplayskip \abovedisplayskip}
\fi
```

\scriptsize
\tiny
\large
\Large
\LARGE
\huge
\Huge

These are all much simpler than the previous macros, they just select a new fontsize, but leave the parameters for displays and lists alone.

```
\ifcase\@ptsize
  \DeclareRobustCommand\scriptsize{\@setfontsize\scriptsize\@viipt\@viiipt}
  \DeclareRobustCommand\tiny{\@setfontsize\tiny\@vpt\@vipt}
  \DeclareRobustCommand\large{\@setfontsize\large\@xiipt{14}}
  \DeclareRobustCommand\Large{\@setfontsize\Large\@xivpt{18}}
  \DeclareRobustCommand\LARGE{\@setfontsize\LARGE\@xviipt{22}}
  \DeclareRobustCommand\huge{\@setfontsize\huge\@xxpt{25}}
  \DeclareRobustCommand\Huge{\@setfontsize\Huge\@xxvpt{30}}
\or
  \DeclareRobustCommand\scriptsize{\@setfontsize\scriptsize\@viiipt{9.5}}
  \DeclareRobustCommand\tiny{\@setfontsize\tiny\@vipt\@viipt}
  \DeclareRobustCommand\large{\@setfontsize\large\@xiipt{14}}
  \DeclareRobustCommand\Large{\@setfontsize\Large\@xivpt{18}}
  \DeclareRobustCommand\LARGE{\@setfontsize\LARGE\@xviipt{22}}
  \DeclareRobustCommand\huge{\@setfontsize\huge\@xxpt{25}}
  \DeclareRobustCommand\Huge{\@setfontsize\Huge\@xxvpt{30}}
\or
  \DeclareRobustCommand\scriptsize{\@setfontsize\scriptsize\@viiipt{9.5}}
  \DeclareRobustCommand\tiny{\@setfontsize\tiny\@vipt\@viipt}
  \DeclareRobustCommand\large{\@setfontsize\large\@xivpt{18}}
  \DeclareRobustCommand\Large{\@setfontsize\Large\@xviipt{22}}
  \DeclareRobustCommand\LARGE{\@setfontsize\LARGE\@xxpt{25}}
  \DeclareRobustCommand\huge{\@setfontsize\huge\@xxvpt{30}}
  \let\Huge=\huge
\fi
```

4 Loading Packages

This class file does not load additional packages.

5 Document Layout

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

5.1 Fonts

We use two fixed fonts in these letters.

```
\newfont\refkopfont{cmssq8}
\DeclareFixedFont\kleinvet{\encodingdefault}%
                          {\rmdefault}%
                          {\bfdefault}%
                          {\shapedefault}%
                          {7}
```

5.2 Paragraphing

`\lineskip` `\normallineskip` These parameters control TEX's behaviour 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 \@plus .3em \@minus .2em}
\setlength\parindent{0\p@}
```

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

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

`\clubpenalty` `\widowpenalty` These penalties are use 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
```

5.3 Page Layout

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

5.3.1 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.

```
\setlength\headheight{37mm}
\setlength\headsep   {0mm}
```

\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@}
```

\maxdepth
\@maxdepth

The TEX primitive register \maxdepth has a function that is similar to that of \topskip. The register \@maxdepth should always contain a copy of \maxdepth. In both plain TEX and LATEX 2.09 \maxdepth had a fixed value of 4pt; in native LATEX2e mode we let the value depend on the typesize. We set it so that \maxdepth + \topskip = typesize ×1.5. As it happens, in these classes \topskip is equal to the typesize, therefor we set \maxdepth to half the value of \topskip.

```
\if@compatibility
  \setlength\maxdepth{4\p@}
\else
  \setlength\maxdepth{.5\topskip}
\fi
\setlength\@maxdepth\maxdepth
```

5.3.2 The dimension of text

\textwidth
\textheight

The dimensions of the text are fixed; they are defined in the NEN norm which this class implements.

```
\setlength\textwidth{144mm}
\setlength\textheight{197mm}
\if@elfinch \addtolength\textheight{-17.6mm} \fi
```

\rightskip
\@rightskip

```
\setlength\@rightskip{0cm \@plus 5cm}
\setlength\rightskip{\@rightskip}
```

5.3.3 Margins

\oddsidemargin
\evensidemargin
\marginparwidth

Again, these dimensions are based on the NEN norm.

```
\setlength\@tempdima{\paperwidth}
\addtolength\@tempdima{-2in}
\addtolength\@tempdima{-\textwidth}
```

```
\setlength\oddsidemargin   {7.6mm}
\setlength\evensidemargin  {\oddsidemargin}
\setlength\marginparwidth  {0\p@}
```

\marginparsep
\marginparpush

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

```
\setlength\marginparsep {0\p@}
\setlength\marginparpush{0\p@}
```

\topmargin

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

```
\setlength\topmargin{-12.4mm}
```

5.3.4 The address field

The address information has to be put on a specific place.

\vensterskip
\@vensterskip

```
\newdimen\vensterskip
\setlength\vensterskip{50mm}
\newdimen\@vensterskip
```

5.3.5 Changing head and text heights

This class has a much higher head on the first page of a letter than on subsequent pages.

\@firstheadheight
\@otherheadheight
\@othertextheight
\@otherheadsep
\@vervolgsep

```
\newdimen\@firstheadheight
\newdimen\@otherheadheight
\newdimen\@othertextheight
\newdimen\@otherheadsep
\newdimen\@vervolgsep
\setlength\@otherheadsep{2mm}
```

\@prepareerhoofden

```
\def\@prepareerhoofden{%
  \setlength\@vensterskip{\vensterskip}%
  \addtolength\@vensterskip{-50mm}%
  \setlength\@firstheadheight{\headheight}%
  \setlength\@otherheadheight{\headheight}%
  \setlength\@othertextheight{\textheight}%
  }
```

5.3.6 Information in the foot

We also reserve some space at the bottom of the paper to print some information about the sender of the letter.

`\footsep` The distance between the text and this foot information

```
\newdimen\footsep
\setlength\footsep{15mm}
```

5.3.7 Footnotes

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

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

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

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

5.4 Page Styles

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

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

5.4.1 Marking conventions

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

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

LaTeX extends TeX's `\mark` facility by producing two kinds of marks, a 'left' and a 'right' mark, using the following commands:

`\markboth{`⟨*LEFT*⟩`}{`⟨*RIGHT*⟩`}`: Adds both marks.
`\markright{`⟨*RIGHT*⟩`}`: Adds a 'right' mark.
`\leftmark`: Used in the `\@oddhead`, `\@oddfoot`, `\@evenhead` or `\@evenfoot` macros, it gets the current 'left' mark. `\leftmark` works like TeX's `\botmark` command.

`\rightmark`: Used in the `\@oddhead`, `\@oddfoot`, `\@evenhead` or `\@evenfoot` macros, it gets the current 'right' mark. `\rightmark` works like TEX's `\firstmark` command.

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

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

```
% %%%\mark{{}{}}   % Initializes TeX's marks   <--- can vanish
```

5.4.2 Defining the page styles

The pagestyles *empty* and *plain* are defined in the LATEX kernel (`ltpage.dtx`), but these definitions are changed to a simpler version for this document class.

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

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

The running feet contain some information about the sender of the letter. The feet aer the same for even and odd pages.

```
      \def\@oddfoot{\voetregel\hss}%
      \let\@evenfoot\@oddfoot
```

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

```
      \def\@oddhead{%
        \vbox to \@otherheadheight
          {\vervolghoofd\vfil
           \if@streepjes\streepjes{\@firstheadheight}\fi}\hss}
      \let\@evenhead\@oddhead}
```

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

```
\else
  \def\ps@headings{%
      \def\@oddfoot{\voetregel\hss}%
      \def\@oddhead{%
        \vbox to \@otherheadheight
          {\vervolghoofd\vfil
           \if@streepjes\streepjes{\@otherheadheight}\fi}\hss}}
\fi
```

`\ps@firstpage` On the first page the head contains much more than on other pages, therefore the height of the head and text need to be adapted.

```
\def\ps@firstpage{%
  \global\headheight=\@otherheadheight
  \global\textheight=\@othertextheight %?? werkt dit ??
  \global\headsep=\@otherheadsep
  \def\@oddhead{\vbox to \@firstheadheight
    {\briefhoofd\vfil
      \if@streepjes\streepjes{\@firstheadheight}\fi}%
    \hss}
  \def\@evenhead{}
  \def\@oddfoot{\voetregel\hss} \let\@evenfoot\@oddfoot}
```

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

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

6 Document Markup

6.1 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.
- `\signature{Larry User}` : goes after the closing.
- `\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.
- `\location{Room 374}` : Acts as modifier to the standard institutional address.
- `\telephone{(415)123-4567}` : Just in case some style puts it on the letter.

`\name`
`\fromname`

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

`\ondertekening` This macro stores the signature.
`\signature`
`\fromsig`

```
\newcommand*\ondertekening[1]{\def\fromsig{#1}}
\def\fromsig{}
\let\signature\ondertekening
```

\address

```
\newcommand*\address[1]{\maakbriefhoofd*{}{#1}}
```

\location
\fromlocation

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

\telephone
\telephonenum

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

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

```
\newcommand*\makelabels{%
```

At the beginning of the document, we need to activate the \@mlabel and \@startlabels commands, as well as write \@startlabels to the .aux file.

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

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

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

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

```
\@onlypreamble\makelabels
```

6.2 The generic letter commands

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

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

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

```
\newenvironment{brief}[1]
 {\newpage
   \if@twoside \ifodd\c@page
               \else\thispagestyle{empty} \hbox{}\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}%
```

Now we can start filling in the various fields in the references line. First the adressee.

```
    \@defrefveld{\@Ad}{\geadresseerdetekst}{\toname}
```

Then the date. When nothing was specified we use `\vandaag`.

```
    \ifdim\wd\@Dt=0cm \@defrefveld{\@Dt}{\datumtekst}{\vandaag}\fi
```

Now we can prepare the letterheads. It couldn't be done earlier because the user can specify that he uses a different kind of 'window envelope'.

```
    \@prepareerhoofden
```

We may need to adapt the height of the head and the text body on the following pages. Therefore we measure the height of the head on those pages.

```
    {\setbox\@tempboxa\vervolghoofd
      \@tempdima\ht\@tempboxa
      \advance\@tempdima by -\@otherheadheight
      \ifdim\@tempdima>0\p@
        \global\advance\@otherheadheight by \@tempdima
        \global\advance\@othertextheight by -\@tempdima
      \fi}
```

We have to do the same for the foot of the letter.

```
    {\setbox\@tempboxa=\vbox{\voetregel}
      \global\footskip=\ht\@tempboxa
      \global\advance\footskip by \footsep}%
    }
```

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

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

`letter` The letter environment is a synonime for the brief environment, to provide compatibility with the standard `letter` document class.

```
\let\letter\brief
\let\endletter\endbrief
```

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

```
\long\def\@processto#1{\@xproc #1\\@@@\ifx\toaddress\@empty
    \else \@yproc #1@@@\fi}
\long\def\@xproc #1\\#2@@@{\def\toname{#1}\def\toaddress{#2}}
\long\def\@yproc #1\\#2@@@{\def\toaddress{#2}}
```

\antwoordadres The command \antwoordadres takes the return address as an argument. The various parts of the address should be separated by \\, which will be turned into bullets.

```
\newif\if@antwoordadres
\newcommand*\antwoordadres[1]{%
  \@antwoordadrestrue\renewcommand*\@antwoordadres{#1}}
\newcommand*\@antwoordadres{}
\let\replyaddress\antwoordadres
```

6.2.1 The address window

The address for the letter will be placed in such a way that a 'window envelope' can be used to send the letter.

\adresveldbreedte The width of the address window.

```
\newdimen\adresveldbreedte
```

\adresveld This command formats the address window.

```
\newcommand*\adresveld{%
  \hbox{}\kern-\topskip
  \kern\@vensterskip
  \begingroup
```

Compute the width of the address window

```
    \if@adresrechts
      \setlength\adresveldbreedte{4\refveldbreedte}%
      \addtolength\adresveldbreedte{-76mm}%
      \def\@tempa{\moveright 76mm}%
    \else
      \let\@tempa\relax
      \setlength\adresveldbreedte{83mm}%
    \fi
```

Store the address in a box.

```
    \setbox\@tempboxa\vtop{%
      \hsize\adresveldbreedte
      \@normalsize
      \parindent\z@\parskip\z@
      \rightskip0\p@\@plus\adresveldbreedte
      \let\\\@nobreakcr \toname \\ \toaddress}
```

Format the return address if one was given.

```
    {\baselineskip\z@\lineskip\z@
      \if@antwoordadres
        \@tempa\vbox to \z@{%
          \hb@xt@\adresveldbreedte{%
            \kleinvet
            \def\\{\unskip\enspace{\textbullet}\enspace\ignorespaces}%
            \@antwoordadres\hfil}
          \kern2\p@\hrule \vss}
      \fi
```

Print a small rule as typing aid if required.

```
      \if@typhulp
        \@tempa\llap{\vbox to \z@{\vskip9mm\streepje\vss}}
      \fi
```

And finally print the address information. Note that this way of position the box which contains the address information has the advantage that no matter how high or deep the box is, the following information will always be printed in the same spot on the paper.

```
      \kern9mm \kern-\ht\@tempboxa \@tempdima=\dp\@tempboxa
      \@tempa\box\@tempboxa \kern-\@tempdima
      \vskip31mm}\endgroup}
```

6.2.2 The reference line

\refveldbreedte The width of the various fields in this line. It is determined in NEN 3516

```
\newdimen\refveldbreedte
\setlength\refveldbreedte{38mm}
```

\@defrefveld A macro to help in defining the various fields.

```
\def\@defrefveld#1#2#3{\setbox#1\@refveld{#2}{#3}}
```

\@refveld The macro \@refveld stores the formatted field in a box.

```
\def\@refveld#1#2{%
  \vtop{\hsize\refveldbreedte
    \parskip\z@\parindent\z@
    \everypar{}%
    \lineskiplimit\z@\baselineskip12\p@
    \lineskip\z@
    \rightskip0\p@ \@plus \refveldbreedte \@minus .5\refveldbreedte
    \vbox{\refkopfont\baselineskip10\p@#1\@@par}
    \kern2\p@
    \strut #2}}
```

\@UB We allocate four box registers to store the four fields in

\@UK

\@OK

\@Dt

```
\newbox\@UB \newbox\@UK \newbox\@OK \newbox\@Dt
```

\uwbriefvan The command\uwbriefvan can be used to show the date of the letter to which your letter is an answer

```
\newcommand*\uwbriefvan[1]{\@defrefveld{\@UB}{\uwbrieftekst}{#1}}
\let\yourletterof\uwbriefvan
```

\uwkenmerk The command \uwkenmerk can be used to show the reference of the letter to which your letter is an answer

```
\newcommand*\uwkenmerk[1]{\@defrefveld{\@UK}{\uwkenmerktekst}{#1}}
\let\yourreference\uwkenmerk
```

\onskenmerk Store our reference in a box register.

```
\newcommand*\onskenmerk[1]{\@defrefveld{\@OK}{\onskenmerktekst}{#1}}
```

\datum To store the date in a box register. When the user gives an empty argument no date will be printed. When he doesn't use \datum he will get today's date.

```
\newcommand*\datum[1]{\def\@tempa{}\def\@tempb{#1}%
  \ifx\@tempa\@tempb
    \setbox\@Dt\hbox{ }%
  \else
    \@defrefveld{\@Dt}{\datumtekst}{#1}%
  \fi}
\let\date\datum
```

\referentieregel This collects all the information for the reference line.

```
\def\referentieregel{\hbox
              {\hb@xt@\refveldbreedte{\copy\@UB\hfil}%
               \hb@xt@\refveldbreedte{\copy\@UK\hfil}%
               \hb@xt@\refveldbreedte{\copy\@OK\hfil}%
               \hb@xt@\refveldbreedte{\copy\@Dt\hfil}\hss}}
```

\vervolgreferentieregel On the second and following pages a simple reference line can be printed. It contains the address information, the date and the page number.

\@Ad For this purpose we need to allocate another box register.

```
\newbox\@Ad
```

```
\def\vervolgreferentieregel{%
  \hbox{%
    \hb@xt@\refveldbreedte{\copy\@Ad\hfil}%
    \hskip\refveldbreedte
    \hb@xt@\refveldbreedte{\copy\@Dt\hfil}%
    \@refveld{\bladnummertekst}{\thepage}\hss}}
```

6.2.3 The headers and footers

\briefhoofd The headings are empty by default.
\vervolghoofd

```
\newcommand*\briefhoofd{}
\newcommand*\vervolghoofd{\vbox{}}
```

\maakbriefhoofd The usage of this command creates non-empty headers.

```
\newcommand*\maakbriefhoofd
   {\@ifstar {\@kortvervolgbriefhoofd}{\@langvervolgbriefhoofd}}
\let\makeheader\maakbriefhoofd
```

\@kortvervolgbriefhoofd This creates a shortened heading for following pages

```
\newcommand*\@kortvervolgbriefhoofd[2]{%
  \@maakbriefhoofd{#1}{#2}
  \def\vervolghoofd{%
    \vbox{\hsize=4\refveldbreedte
      \hb@xt@\hsize{\Large \normalfont\sffamily #1\strut\hfil}
      \hrule \kern2mm \vervolgreferentieregel}}}
```

\@langvervolgbriefhoofd This creates a long heading for following pages by just using \briefhoofd.

```
\newcommand*\@langvervolgbriefhoofd[2]{
  \@maakbriefhoofd{#1}{#2}
  \def\vervolghoofd{%
    \vbox{\briefhoofd\vskip2mm
      \vervolgreferentieregel
      \vbox{}}}}
```

\@maakbriefhoofd This was used in the two preceding macros; it defines \briefhoofd.

```
\newcommand*\@maakbriefhoofd[2]{\def\briefhoofd{%
    \vbox{\hsize=4\refveldbreedte
      \hb@xt@\hsize{\Large \normalfont\sffamily #1\strut\hfil}
      \hrule
      \moveright 3\refveldbreedte\@refveld{\strut #2}{}
      \vbox{}}}}
```

\@voetruimte A box to store the footer in.

```
\newbox\@voetruimte
\setbox\@voetruimte=\hbox{}
```

\@voetteller We need to know how many items are placed in the footer.

```
\newcount\@voetteller
```

\voetregel \voetregel just copies the box \@voetruimte.

```
\newcommand*\voetregel{\copy\@voetruimte}
```

\voetitem A command to add an information field to the footer.

```
\newcommand*\voetitem[2]{%
  \advance\@voetteller by 1
  \setbox\@voetruimte\hb@xt@4\refveldbreedte{%
    \unhbox\@voetruimte
    \ifcase\@voetteller \relax \or \relax \or \hfil \else \hfill
    \fi
    \@refveld{#1}{#2}\hskip0\p@ \@plus 3\refveldbreedte}}
\let\footitem\voetitem
```

6.2.4 The little rules

\streepje A shorthand for one little rule.

```
\newcommand*\streepje{\hb@xt@2mm{\rule{2mm}{.1pt}}}
```

\streepjes This prints the folding rules

```
\newcommand*\streepjes[1]{%
  \vbox to \z@{%
```

We have to backup to a position 13mm below the edge of the paper.

```
    \kern-#1\relax
    \hb@xt@\textwidth{%
```

Then we can print a rule on the left side of the paper, half way down to align for a perforator.

```
      \llap{\perfstreepje\kern24mm}\hfill
```

The folding rules are printed on the right hand side of the paper.

```
      \rlap{\kern24mm\vouwstreepjes}}
  \vss}}
```

\perfstreepje Prints a \streepje halfway down the paper. A4 paper is 297 mm high; we start from a position 13mm below the edge of the paper. Hence the \kern 135mm.

```
\newcommand*\perfstreepje{\vtop{\kern\z@ \kern 135mm \streepje}}
```

\vouwstreepjes This prints two folding rules.

```
\newcommand*\vouwstreepjes{%
  \vtop{\kern\z@
        \kern 95mm %% 108-13
        \streepje  %% denk maar dat dit geen dikte heeft
        \kern 45mm %% 155-150
        \streepje}}
```

6.2.5 Page breaking control

\stopbreaks

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

\@nobreakvspace
\@nobreakvspacex
\@nobreakcr

```
\DeclareRobustCommand\@nobreakvspace
   {\@ifstar{\@nobreakvspacex}{\@nobreakvspacex}}
\def\@nobreakvspacex#1{%
  \ifvmode
    \nobreak\vskip #1\relax
  \else
    \@bsphack\vadjust{\nobreak\vskip #1}\@esphack
  \fi}

\def\@nobreakcr{%
  \let\reserved@e\relax
  \let\reserved@f\relax
  \vadjust{\nobreak}\@ifstar{\@xnewline}{\@xnewline}}
```

\startbreaks

```
\def\startbreaks{\let\\=\@normalcr
   \interlinepenalty 200\def\par{\@@par\penalty 200\relax}}
```

\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 command that follows. Since there's a `\vfil` at the bottom of every page, it can add vertical fil 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]{%
  \thispagestyle{firstpage}%
  \adresveld
  \prevdepth=-1000\p@ \vskip-2\p@ %% ????
  \referentieregel
  \@dosubject #1\par\nobreak}
```

`\@dosubject` This prints the subject of the letter if one was specified.

```
\def\@dosubject{%
  \ifx\@empty\@subject
  \else
    \par\noindent
    \parbox[t]{\textwidth}
              {\@hangfrom{\refkopfont \betrefttekst \enspace}%
                \normalfont\rmfamily\ignorespaces \@subject\strut}%
    \par
  \fi}
```

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

```
\afsluiting{Yours truly,}
```

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

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

```
\newcommand*\afsluiting[1]{\par\nobreak\vspace{\parskip}%
  \stopbreaks
  \ifx\@empty\fromsig
    \def\ondertekening##1{\def\fromsig{##1}\@afsluiting{#1}}%
```

```
  \else
    \@afsluiting{#1}%
  \fi}
\let\closing\afsluiting
\def\open@af{\vtop\bgroup\hsize.3\textwidth \raggedright}
```

The internal command `\@afsluiting` takes care of printing the closing text.

```
\newcommand*\@afsluiting[1]{%
  \def\en{\strut\egroup\open@af}%
  \let\and\en
  \noindent
  \parbox{.5\textwidth}{%
    \raggedright \ignorespaces #1\\[6\medskipamount]%
    \leavevmode\open@af \fromsig \strut\egroup}}
```

`\smallskipamount` `\mdeskipamount` `\bigskipamount`

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

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

`\betreft` `\re`

The command `\betreft` (`\re`) stores the subject of the letter.

```
\newcommand*\betreft[1]{\def\@subject{#1}}
\let\onderwerp\betreft
\let\subject\betreft
\def\@subject{}
\let\re\betreft
```

`\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}
```

`\bijlage` `\bijlagen` `\encl`

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

which produces:

bijlagen: Foo(2)
Bar

```
\newcommand*\bijlage[1]{%
  \par\noindent
  \parbox[t]{\textwidth}{\@hangfrom{\normalfont\bijlagetekst\ }%
```

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

`\ps` 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 fil 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{}
```

6.3 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 adress*⟩`}` : Command to generate a single label.

`\returnaddress`

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

`\labelcount`

```
\newcount\labelcount
```

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

```
\newcommand*\startlabels{\labelcount\z@
  \pagestyle{empty}%
```

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

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

```
\let\@startlabels=\relax
```

`\mlabel` This command prints an address label; it is used when the user specified `\makelabels` in the preamble of his document. The command `\mlabel` takes two arguments; the second argument is supposed to be the address; the first argument can be used to print a return address. In this document class we ignore the first argument. Also the labels are supposed to be 2 inch high and 3.6 inch wide. When your address labels have a different width you will have to defined your own `\mlabel` command.

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

`\@mlabel` `\@mlabel` is written to the `.aux` file in place of `\mlabel`. That allows to define it as a no-op per default, and activate it in the `\begin{document}` hook.

```
\let\@mlabel=\@gobbletwo
```

6.4 Lists

6.4.1 General List Parameters

The following commands are used to set the default values for the list environment's parameters. See the LaTeX manual for an explanation of the meanings

of the parameters. Defaults for the list environment are set as follows. First, `\rightmargin`, `\listparindent` and `\itemindent` are set to 0pt. Then, for a `K`th level list, the command `\@listK` is called, where '`K`' denotes '`i`', '`ii`', ... , '`vi`'. (I.e., `\@listiii` is called for a third-level list.) By convention, `\@listK` should set `\leftmargin` to `\leftmarginK`.

`\leftmargin` `\leftmargini` `\leftmarginii` `\leftmarginiii` `\leftmarginiv` `\leftmarginv` `\leftmarginvi`

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

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

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

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

Here we set the top level leftmargin.

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

`\labelsep` `\labelwidth`

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

```
\setlength  \labelsep  {5\p@}
\setlength  \labelwidth{\leftmargini}
\addtolength\labelwidth{-\labelsep}
```

`\partopsep`

When the user leaves a blank line before the environment an extra vertical space of `\partopsep` is inserted, in addition to `\parskip` and `\topsep`.

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

`\topsep`

Extra vertical space, in addition to `\parskip`, added above and below list and paragraphing environments.

```
\setlength\topsep{.4em}
```

`\@beginparpenalty` `\@endparpenalty`

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

`\@itempenalty`

This penalty is inserted between list items.

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

`\@listI` `\@listi`

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

These values have been taken from the ones in the document class `artikel3`.

```
\def\@listI{\leftmargin\leftmargini
            \labelsep.5em%
            \labelwidth\leftmargin
```

```
            \advance\labelwidth-\labelsep
            \topsep .5\parskip \@plus \p@
            \parsep \z@
            \itemsep\parsep}
\let\@listi\@listI
```

We have to initialise these parameters.

```
\@listi
```

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

Here are the same macros for the higher level lists.

```
\def\@listii {\leftmargin\leftmarginii
              \labelsep  .5em%
              \labelwidth\leftmarginii
              \advance\labelwidth-\labelsep
              \topsep -.5\parskip \@plus \p@
              \parsep \z@
              \itemsep\parsep}
\def\@listiii{\leftmargin\leftmarginiii
              \labelsep  .5em%
              \labelwidth\leftmarginiii
              \advance\labelwidth-\labelsep
              \topsep -.5\parskip \@plus \p@
              \parsep \z@
              \partopsep \z@
              \itemsep   \topsep}
\def\@listiv {\leftmargin\leftmarginiv
              \labelsep  .5em%
              \labelwidth\leftmarginiv
              \advance\labelwidth-\labelsep
              \topsep -.5\parskip \@plus \p@}
\def\@listv  {\leftmargin\leftmarginv
              \labelsep  .5em%
              \labelwidth\leftmarginv
              \advance\labelwidth-\labelsep
              \topsep -.5\parskip \@plus \p@}
\def\@listvi {\leftmargin\leftmarginvi
              \labelsep  .5em%
              \labelwidth\leftmarginvi
              \advance\labelwidth-\labelsep
              \topsep -.5\parskip \@plus \p@}
```

6.4.2 Enumerate

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

\theenumi
\theenumii
\theenumiii
\theenumiv

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

```
\renewcommand*\theenumi  {\@arabic\c@enumi}
\renewcommand*\theenumii {\@alph\c@enumii}
```

```
\renewcommand*\theenumiii{\@roman\c@enumiii}
\renewcommand*\theenumiv {\@Alph\c@enumiv}
```

\labelenumi
\labelenumii
\labelenumiii
\labelenumiv

The label for each item is generated by the commands `\labelenumi` ... `\labelenumiv`.

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

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

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

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

6.4.3 Itemize

\labelitemi
\labelitemii
\labelitemiii
\labelitemiv

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

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

\labelitemfont

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

A possible alternative would be

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

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

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

6.4.4 Description

description

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}
```

6.5 Defining new environments

6.5.1 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}
```

6.5.2 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}}%
                \item[]}
               {\endlist}
```

6.5.3 Quote

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

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

6.5.4 Theorem

This document class does not define it's own theorem environemts, the defaults, supplied by LaTeX kernel (`ltthm.dtx`) are available.

6.6 Setting parameters for existing environments

6.6.1 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@}
```

6.6.2 Tabbing

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

```
\setlength\tabbingsep{\labelsep}
```

6.6.3 Minipage

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

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

```
\skip\@mpfootins = \skip\footins
```

6.6.4 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@}
```

6.6.5 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)}
```

6.7 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 compatiblity, but one should start using the `\text...` and `\math...` commands instead. These commands are redefined using `\@renewfontswitch`, a command with three arguments: the user command to be defined; LaTeX commands to execute in text mode and LaTeX commands to execute in math mode.

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

```
\DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\DeclareOldFontCommand{\tt}{\normalfont\ttfamily}{\mathtt}
```

`\bf` The command to change to the bold series. One should use `\mdseries` to explicitly switch back to medium series.

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

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

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

`\cal` `\mit` The commands `\cal` and `\mit` should only be used in math mode, outside math mode they have no effect. Currently the New Font Selection Scheme defines these commands to generate warning messages. Therefore we have to define them 'by hand'.

```
\DeclareRobustCommand*{\cal}{\@fontswitch{\relax}{\mathcal}}
\DeclareRobustCommand*{\mit}{\@fontswitch{\relax}{\mathnormal}}
```

6.8 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 heigth 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 Footnotes are numbered within chapters in the report and book document styles.

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

\@makefntext The footnote mechanism of LaTeX calls the macro \@makefntext to produce the actual footnote. The macro gets the text of the footnote as its argument and should use \@makefnmark to produce the mark of the footnote. The macro \@makefntext is called when effectively inside a \parbox of width \columnwidth (i.e., with \hsize = \columnwidth).

An example of what can be achieved is given by the following piece of TeX code.

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

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

In these document classes 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.

```
\long\def\@makefntext#1{%
  \noindent\hb@xt@\leftmargini{\normalfont\@thefnmark.\hfil}#1}
```

\@makefnmark The footnote markers that are printed in the text to point to the footnotes should be produced by the macro \@makefnmark.

```
% \def\@makefnmark{\hb@xt@\leftmargini{\normalfont\@thefnmark.\hfil}}
```

6.9 Words

This document class supports a number of languages. All words that will be printed byu the class code are stored in commands which can be redefined if you want to use a different language.

\dutchbrief This stores dutch strings.

```
\newcommand*\dutchbrief{%
  \def\uwbrieftekst{Uw brief van}
  \def\uwkenmerktekst{Uw kenmerk}
  \def\onskenmerktekst{Ons kenmerk}
```

```
  \def\datumtekst{Datum}
  \def\geadresseerdetekst{Geadresseerde}
  \def\bladnummertekst{Bladnummer}
  \def\vandaag{\number\day~\ifcase\month\or
    januari\or februari\or maart\or april\or mei\or juni\or juli\or
    augustus\or september\or oktober\or november\or december\fi
    \space \number\year}
  \def\betrefttekst{Onderwerp:}
  \def\ccname{cc}
  \def\bijlagetekst{Bijlage:}
  \def\bijlagentekst{Bijlagen:}
  \def\telefoontekst{telefoon}}
```

`\englishbrief` This stores English strings.

```
\newcommand*\englishbrief{%
  \def\uwbrieftekst{Your letter of}
  \def\uwkenmerktekst{Your reference}
  \def\onskenmerktekst{Our reference}
  \def\datumtekst{Date}
  \def\geadresseerdetekst{To}
  \def\bladnummertekst{Page}
  \def\vandaag{\ifcase\day\or
    1st\or 2nd\or 3rd\or 4th\or 5th\or
    6th\or 7th\or 8th\or 9th\or 10th\or
    11th\or 12th\or 13th\or 14th\or 15th\or
    16th\or 17th\or 18th\or 19th\or 20th\or
    21st\or 22nd\or 23rd\or 24th\or 25th\or
    26th\or 27th\or 28th\or 29th\or 30th\or
    31st\fi~\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\year}
  \def\betrefttekst{Re:}
  \def\ccname{cc}
  \def\bijlagetekst{Enclosure:}
  \def\bijlagentekst{Enclosures:}
  \def\telefoontekst{telephone}}
```

`\americanbrief` This stores American english strings

```
\newcommand*\americanbrief{%
  \def\uwbrieftekst{Your letter of}
  \def\uwkenmerktekst{Your reference}
  \def\onskenmerktekst{Our reference}
  \def\datumtekst{Date}
  \def\geadresseerdetekst{To}
  \def\bladnummertekst{Page}
  \def\vandaag{\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}
```

```
  \def\betrefttekst{Re:}
  \def\ccname{cc}
  \def\bijlagetekst{Enclosure:}
  \def\bijlagentekst{Enclosures:}
  \def\telefoontekst{telephone}}
```

`\germanbrief` This stores the German versions of the strings.

```
\newcommand*\germanbrief{%
  \def\uwbrieftekst{Ihr Brief vom}
  \def\uwkenmerktekst{Ihr Zeichen}
  \def\onskenmerktekst{Unser Zeichen}
  \def\datumtekst{Datum}
  \def\geadresseerdetekst{An}
  \def\bladnummertekst{Seite}
  \def\vandaag{\number\day.~\ifcase\month\or
    Januar\or Februar\or M\"arz\or April\or Mai\or Juni\or
    Juli\or August\or September\or Oktober\or November\or Dezember\fi
    \space\number\year}
  \def\betrefttekst{Betrifft:}
  \def\ccname{Kopien an}
  \def\bijlagetekst{Anlage:}
  \def\bijlagentekst{Anlagen:}
  \def\telefoontekst{Telefon}}
```

`\frenchbrief` And finally to store the french strings

```
\newcommand*\frenchbrief{%
  \def\uwbrieftekst{Votre lettre du}
  \def\uwkenmerktekst{Vos r\'ef\'erences:}
  \def\onskenmerktekst{Nos r\'ef\'erences:}
  \def\datumtekst{Date:}
  \def\geadresseerdetekst{\`A l'attention de}
  \def\bladnummertekst{Page}
  \def\vandaag{\number\day\ifnum\day=1$^{er}$\fi
         ~\ifcase\month\or janvier\or
          f\'evrier\or mars\or avril\or mai\or juin\or
          juillet\or ao\^ut\or septembre\or octobre\or
          novembre\or d\'ecembre\fi \space \number\year}
  \def\betrefttekst{Objet:}
  \def\ccname{Copie \`a}
  \def\bijlagetekst{Pi\`ece jointe:}
  \def\bijlagentekst{Pi\`eces jointes:}
  \def\telefoontekst{T\'el\'ephone:}}
```

6.10 Two column mode

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

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

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

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

6.11 The page style

We have *headings* pages in this document class by default. We use arabic pagenumbers.

```
\pagestyle{headings}
\pagenumbering{arabic}
```

6.12 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 class however doesn't want the letter to be centered on the page.

```
\let\@texttop\relax
```

We always start in one column mode.

```
\onecolumn
⟨/brief⟩
```

Change History

Change History