

# NTG Document Class **brief** for L<sup>A</sup>T<sub>E</sub>X version 2e

Victor Eijkhout      Johannes Braams

April 16, 2004

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Initial Code</b>	<b>1</b>
2.1	Setting Paper Sizes . . . . .	2
2.2	Choosing the type size . . . . .	2
2.3	Two-side or one-side printing . . . . .	2
2.4	Draft option . . . . .	3
2.5	Equation numbering on the left . . . . .	3
2.6	Flush left displays . . . . .	3
2.7	Typewriter alignment . . . . .	3
2.8	Folding lines . . . . .	3
2.9	Address placement . . . . .	3
2.10	Support for different languages . . . . .	3
<b>3</b>	<b>Executing Options</b>	<b>4</b>
<b>4</b>	<b>Loading Packages</b>	<b>6</b>
<b>5</b>	<b>Document Layout</b>	<b>6</b>
5.1	Fonts . . . . .	7
5.2	Paragraphing . . . . .	7
5.3	Page Layout . . . . .	8
5.3.1	Vertical spacing . . . . .	8
5.3.2	The dimension of text . . . . .	8
5.3.3	Margins . . . . .	8
5.3.4	The address field . . . . .	9
5.3.5	Changing head and text heights . . . . .	9
5.3.6	Information in the foot . . . . .	9
5.3.7	Footnotes . . . . .	10
5.4	Page Styles . . . . .	10
5.4.1	Marking conventions . . . . .	10
5.4.2	Defining the page styles . . . . .	11

<b>6</b>	<b>Document Markup</b>	<b>12</b>
6.1	Global Declarations . . . . .	12
6.2	The generic letter commands . . . . .	13
6.2.1	The address window . . . . .	14
6.2.2	The reference line . . . . .	15
6.2.3	The headers and footers . . . . .	16
6.2.4	The little rules . . . . .	18
6.2.5	Page breaking control . . . . .	18
6.3	Customizing the labels . . . . .	21
6.4	Lists . . . . .	22
6.4.1	General List Parameters . . . . .	22
6.4.2	Enumerate . . . . .	24
6.4.3	Itemize . . . . .	24
6.4.4	Description . . . . .	25
6.5	Defining new environments . . . . .	25
6.5.1	Verse . . . . .	25
6.5.2	Quotation . . . . .	25
6.5.3	Quote . . . . .	25
6.5.4	Theorem . . . . .	26
6.6	Setting parameters for existing environments . . . . .	26
6.6.1	Array and tabular . . . . .	26
6.6.2	Tabbing . . . . .	26
6.6.3	Minipage . . . . .	26
6.6.4	Framed boxes . . . . .	26
6.6.5	Equation and eqnarray . . . . .	27
6.7	Font changing . . . . .	27
6.8	Footnotes . . . . .	28
6.9	Words . . . . .	29
6.10	Two column mode . . . . .	30
6.11	The page style . . . . .	31
6.12	Single or double sided printing . . . . .	31

## 1 Introduction

This file contains the document class `brief` that was made available by Working Group 13 of the NTG (Nederlandstalige T<sub>E</sub>X 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.

```
1 \newcommand*\@ptsize{}
2 \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.  
`3 \newif\if@typhulp`

`\if@streepjes` A switch to indicate if the ‘folding lines’ should be printed  
`4 \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.  
`5 \newif\if@adresrechts`

`\if@elfinch` A switch to remember whether we are using A4 or letter paper. (possibly obsolete)  
`6 \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.

```

7 \DeclareOption{a4paper}
8   {\setlength\paperheight {297mm}%
9     \setlength\paperwidth  {210mm}\@elfinchfalse}
10 \DeclareOption{a5paper}
11   {\ClassWarning{brief}{Paper size A5 not supported, using A4}%
12     \setlength\paperheight {297mm}%
13     \setlength\paperwidth  {210mm}\@elfinchfalse}
14 \DeclareOption{b5paper}
15   {\ClassWarning{brief}{Paper size B5 not supported, using A4}%
16     \setlength\paperheight {297mm}%
17     \setlength\paperwidth  {210mm}\@elfinchfalse}
18 \DeclareOption{letterpaper}
19   {\setlength\paperheight {11in}%
20     \setlength\paperwidth  {8.5in}\@elfinchtrue}
21 \DeclareOption{USletter}
22   {\setlength\paperheight {11in}%
23     \setlength\paperwidth  {8.5in}\@elfinchtrue}
24 \DeclareOption{legalpaper}
25   {\ClassWarning{brief}
26     {Paper size ‘legal’ not supported, using ‘letter’}%
27     \setlength\paperheight {14in}%
28     \setlength\paperwidth  {8.5in}\@elfinchtrue}
29 \DeclareOption{executivepaper}
30   {\ClassWarning{brief}
31     {Paper size ‘executive’ not supported, using ‘letter’}%
32     \setlength\paperheight {10.5in}%
33     \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.

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

## 2.3 Two-side or one-side printing

Two-sided printing was not supported in the L<sup>A</sup>T<sub>E</sub>X 2.09 version of this document-class.

```
37 \if@compatibility
38   \DeclareOption{twoside}{\@latexerr{No 'twoside' layout for letters}%
39                               \@eha}
40 \else
41   \DeclareOption{twoside}{\@twosidetrue \mparswitchtrue}
42 \fi
43 \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.

```
44 \DeclareOption{draft}{\setlength\overfullrule{5pt}}
45 \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.

```
46 \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.

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

## 2.7 Typewriter alignment

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

## 2.8 Folding lines

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

```
50 \DeclareOption{streepjes}{\@streepjestrue}
51 \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

```
52 \DeclareOption{adreslinks}{\@adresrechtsfalse}
53 \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,

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

then British English,

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

American English,

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

German

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

and finally french.

```
62 \DeclareOption{frans}     {\AtEndOfClass{\frenchbrief}}
63 \DeclareOption{french}    {\AtEndOfClass{\frenchbrief}}
64 \DeclareOption{français}  {\AtEndOfClass{\frenchbrief}}
```

## 3 Executing Options

Here we execute the default options to initialize certain variables.

```
65 \ExecuteOptions{a4paper,11pt,oneside,onecolumn,final,%
66                geentyphulp,geenstreepjes,adreslinks,%
67                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.

```
68 \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` The user level command for the main size is `\normalsize`. Internally L<sup>A</sup>T<sub>E</sub>X 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

```

69 \ifcase\@ptsize
70   \renewcommand*\normalsize{%
71     \@setfontsize\normalsize\@xpt\@xipt
72     \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
73     \abovedisplayshortskip \z@ \@plus3\p@
74     \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
75     \belowdisplayskip \abovedisplayskip
76     \let\@listi\@listI}
77 \or
78   \renewcommand*\normalsize{%
79     \@setfontsize\normalsize\@xipt{13.6}%
80     \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
81     \abovedisplayshortskip \z@ \@plus3\p@
82     \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
83     \belowdisplayskip \abovedisplayskip
84     \let\@listi\@listI}
85 \or
86   \renewcommand*\normalsize{%
87     \@setfontsize\normalsize\@xipt{15}%
88     \abovedisplayskip 12\p@ \@plus3\p@ \@minus7\p@
89     \abovedisplayshortskip \z@ \@plus3\p@
90     \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
91     \belowdisplayskip \abovedisplayskip
92     \let\@listi\@listI}
93 \fi

```

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

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

We initially choose the normalsize font.

```
95 \normalsize
```

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

```

96 \ifcase\@ptsize
97   \newcommand*\small{%
98     \@setfontsize\small\@ixpt{11}%
99     \abovedisplayskip 8.5\p@ \@plus3\p@ \@minus4\p@
100    \abovedisplayshortskip \z@ \@plus2\p@
101    \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
102    \belowdisplayskip \abovedisplayskip}
103 \or
104   \newcommand*\small{%
105     \@setfontsize\small\@xpt\@xipt
106     \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
107     \abovedisplayshortskip \z@ \@plus3\p@
108     \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
109     \belowdisplayskip \abovedisplayskip}

```

```

110 \or
111 \newcommand*\small{%
112   \@setfontsize\small\@xipt{13.6}%
113   \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
114   \abovedisplayshortskip \z@ \@plus3\p@
115   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
116   \belowdisplayskip \abovedisplayskip}
117 \fi

```

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

```

118 \ifcase\@ptsize
119 \newcommand*\footnotesize{%
120   \@setfontsize\footnotesize\@viiipt{9.5}%
121   \abovedisplayskip 6\p@ \@plus2\p@ \@minus4\p@
122   \abovedisplayshortskip \z@ \@plus\p@
123   \belowdisplayshortskip 3\p@ \@plus\p@ \@minus2\p@
124   \belowdisplayskip \abovedisplayskip}
125 \or
126 \newcommand*\footnotesize{%
127   \@setfontsize\footnotesize\@ixpt{11}%
128   \abovedisplayskip 8\p@ \@plus2\p@ \@minus4\p@
129   \abovedisplayshortskip \z@ \@plus\p@
130   \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
131   \belowdisplayskip \abovedisplayskip}
132 \or
133 \newcommand*\footnotesize{%
134   \@setfontsize\footnotesize\@xpt\@xiipt
135   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
136   \abovedisplayshortskip \z@ \@plus3\p@
137   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
138   \belowdisplayskip \abovedisplayskip}
139 \fi

```

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

```

\large 140 \ifcase\@ptsize
\Large 141 \newcommand*\scriptsize{\@setfontsize\scriptsize\@viipt\@viiipt}
\LARGE 142 \newcommand*\tiny{\@setfontsize\tiny\@vpt\@vipt}
\huge 143 \newcommand*\large{\@setfontsize\large\@xipt{14}}
\Huge 144 \newcommand*\Large{\@setfontsize\Large\@xivpt{18}}
145 \newcommand*\LARGE{\@setfontsize\LARGE\@xviipt{22}}
146 \newcommand*\huge{\@setfontsize\huge\@xxpt{25}}
147 \newcommand*\Huge{\@setfontsize\Huge\@xxvpt{30}}
148 \or
149 \newcommand*\scriptsize{\@setfontsize\scriptsize\@viiipt{9.5}}
150 \newcommand*\tiny{\@setfontsize\tiny\@vipt\@viipt}
151 \newcommand*\large{\@setfontsize\large\@xipt{14}}
152 \newcommand*\Large{\@setfontsize\Large\@xivpt{18}}
153 \newcommand*\LARGE{\@setfontsize\LARGE\@xviipt{22}}
154 \newcommand*\huge{\@setfontsize\huge\@xxpt{25}}
155 \newcommand*\Huge{\@setfontsize\Huge\@xxvpt{30}}
156 \or
157 \newcommand*\scriptsize{\@setfontsize\scriptsize\@viiipt{9.5}}
158 \newcommand*\tiny{\@setfontsize\tiny\@vipt\@viipt}

```

```

159 \newcommand*\large{\@setfontsize\large\@xivpt{18}}
160 \newcommand*\Large{\@setfontsize\Large\@xxviipt{22}}
161 \newcommand*\LARGE{\@setfontsize\LARGE\@xxxpt{25}}
162 \newcommand*\huge{\@setfontsize\huge\@xxvpt{30}}
163 \let\Huge=\huge
164 \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.

```

165 \newfont\refkopfont{cmssq8}
166 \DeclareFixedFont\kleinvet{\encodingdefault}%
167                               {\rmdefault}%
168                               {\bfdefault}%
169                               {\shapedefault}%
170                               {7}

```

### 5.2 Paragraphing

`\lineskip` `\normallineskip` These parameters control  $\TeX$ 's behaviour when two lines tend to come too close together.

```

171 \setlength\lineskip{1\p@}
172 \setlength\normallineskip{1\p@}

```

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

```

173 \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.

```

174 \setlength\parskip{0.7em \@plus .3em \@minus .2em}
175 \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.

```

176 \@lowpenalty 51
177 \@medpenalty 151
178 \@highpenalty 301

```

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

`\widowpenalty`

```
179 % \clubpenalty 150
180 % \widowpenalty 150
```

`\displaywidowpenalty` Discourage (but not so much) widows in front of a math display and forbid breaking directly in front of a display. Allow break after a display without a penalty.

`\predisplaypenalty`

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

```
181 % \displaywidowpenalty 50
182 % \predisplaypenalty 10000
183 % \postdisplaypenalty 0
```

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

```
184 % \interlinepenalty 0
```

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

```
185 % \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` The `\headheight` is the height of the box that will contain the running head. The

`\headsep` `\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.

```
186 \setlength\headheight{37mm}
187 \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:

```
188 \setlength\footskip{25\p@}
```

`\maxdepth` The `TEX` primitive register `\maxdepth` has a function that is similar to that of

`\@maxdepth` `\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 `LATEX 2e` 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`.

```
189 \if@compatibility
190   \setlength\maxdepth{4\p@}
191 \else
192   \setlength\maxdepth{.5\topskip}
193 \fi
194 \setlength\@maxdepth\maxdepth
```

### 5.3.2 The dimension of text

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

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

`\rightskip`  
`\@rightskip` 198 `\setlength\@rightskip{0cm \@plus 5cm}`  
199 `\setlength\rightskip{\@rightskip}`

### 5.3.3 Margins

`\oddsidemargin` Again, these dimensions are based on the NEN norm.

```
200 \setlength\@tempdima{\paperwidth}
\evensidemargin 201 \addtolength\@tempdima{-2in}
\marginparwidth 202 \addtolength\@tempdima{-\textwidth}
203 \setlength\oddsidemargin {7.6mm}
204 \setlength\evensidemargin {\oddsidemargin}
205 \setlength\marginparwidth {0\p@}
```

`\marginparsep` 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`.

```
206 \setlength\marginparsep {0\p@}
207 \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.

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

### 5.3.4 The address field

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

```
\vensterskip
\@vensterskip 209 \newdimen\vensterskip
210 \setlength\vensterskip{50mm}
211 \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 212 \newdimen\@firstheadheight
\@othertextheight 213 \newdimen\@otherheadheight
\@otherheadsep 214 \newdimen\@othertextheight
\@vervolgsep 215 \newdimen\@otherheadsep
216 \newdimen\@vervolgsep
217 \setlength\@otherheadsep{2mm}
```

`\@prepareerhoofden`

```
218 \def\@prepareerhoofden{%
219   \setlength\@vensterskip{\vensterskip}%
220   \addtolength\@vensterskip{-50mm}%
221   \setlength\@firstheadheight{\headheight}%
222   \setlength\@otherheadheight{\headheight}%
223   \setlength\@othertextheight{\textheight}%
224 }
```

### 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

```
225 \newdimen\footsep
226 \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.

```
227 \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.

```
228 \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` The `\ps@...` command defines the macros `\@oddhead`, `\@oddfoot`, `\@evenhead`, `\@oddhead` 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.)

L<sup>A</sup>T<sub>E</sub>X extends T<sub>E</sub>X'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.

```
229 %%%\mark{-}{-} % Initializes TeX's marks <--- can vanish
```

#### 5.4.2 Defining the page styles

The page styles *empty* and *plain* are defined in the L<sup>A</sup>T<sub>E</sub>X 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.

```
230 \if@twoside
```

```
231 \def\ps@headings{%
```

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

```
232 \def\@oddfoot{\voetregel\hss}%
```

```
233 \let\@evenfoot\@oddfoot
```

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

```
234 \def\@oddhead{%
```

```
235 \vbox to \@otherheadheight
```

```
236 {\vervolghoofd\vfil
```

```
237 \if@streepjes\streepjes{\@firstheadheight}\fi}\hss}
```

```
238 \let\@evenhead\@oddhead}
```

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

```
239 \else
```

```
240 \def\ps@headings{%
```

```
241 \def\@oddfoot{\voetregel\hss}%
```

```
242 \def\@oddhead{%
```

```
243 \vbox to \@otherheadheight
```

```
244 {\vervolghoofd\vfil
```

```
245 \if@streepjes\streepjes{\@otherheadheight}\fi}\hss}}
```

```
246 \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.

```
247 \def\ps@firstpage{%
248   \global\headheight=\@otherheadheight
249   \global\textheight=\@othertextheight %?? werkt dit ??
250   \global\headsep=\@otherheadsep
251   \def\@oddhead{\vbox to \@firstheadheight
252     {\briefhoofd\vfil
253       \if\@streepjes\streepjes{\@firstheadheight}\fi}%
254     \hss}
255   \def\@evenhead{}
256   \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.

```
257 \def\ps@empty{%
258   \let\@oddfoot\@empty\let\@oddhead\@empty
259   \let\@evenfoot\@empty\let\@evenhead\@empty}
```

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

```
260 \def\ps@plain{%
261   \let\@oddhead\@empty
262   \def\@oddfoot{\normalfont\hfil\thepage}%
263   \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 264 \def\name#1{\def\fromname{#1}}
          265 \def\fromname{}}
```

`\ondertekening` This macro stores the signature.

```
\signature 266 \newcommand*\ondertekening[1]{\def\fromsig{#1}}
\fromsig   267 \def\fromsig{}
          268 \let\signature\ondertekening
```

```

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

\location
\fromlocation 270 \newcommand*{\location}[1]{\def\fromlocation{#1}}
271 \def\fromlocation{}

\telephone
\telephonenumber 272 \newcommand*{\telephone}[1]{\def\telephonenumber{#1}}
273 \def\telephonenumber{}

\makelabels The \makelabels declaration causes mailing labels to be made.
274 \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.
275 \AtBeginDocument{%
276 \let\@startlabels\startlabels
277 \let\@mlabel\mlabel
278 \if@filesw
279 \immediate\write\@mainaux{\string\@startlabels}\fi}%
At the end of the document we need to write \clearpage to the .aux file.
280 \AtEndDocument{%
281 \if@filesw\immediate\write\@mainaux{\string\clearpage}\fi}}
\makelabels is allowed only before the \begin{document} command.
282 \@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 addressee 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}`.

```

283 \newenvironment{brief}[1]
284 {\newpage
285 \if@twoside \ifodd\c@page
286 \else\thispagestyle{empty} \hbox{} \newpage\fi
287 \fi
288 \c@page\@ne
289 \interlinepenalty=200 % smaller than the TeXbook value

```

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

```

290 \@processto{\leavevmode\ignorespaces #1}%

```

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

```

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

```

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

```
292 \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'.

```
293 \@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.

```
294 {\setbox\@tempboxa\vervolghoofd
295 \@tempdima\ht\@tempboxa
296 \advance\@tempdima by -\@otherheadheight
297 \ifdim\@tempdima>0\p@
298 \global\advance\@otherheadheight by \@tempdima
299 \global\advance\@othertextheight by -\@tempdima
300 \fi}
```

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

```
301 {\setbox\@tempboxa=\vbox{\voetregel}
302 \global\footskip=\ht\@tempboxa
303 \global\advance\footskip by \footsep}%
304 }
```

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

```
305 {\stopletter\@par\pagebreak\@par
306 \if@filesw
307 \begingroup
308 \let\=\relax
309 \let\protect\@unexpandable@protect
310 \immediate\write\@auxout
311 {\string\@mlabel{returnaddress}{\toname\\\toaddress}}%
312 \endgroup
313 \fi}
```

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

```
314 \let\letter\brief
315 \let\endletter\endbrief
```

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

```
\@yproc 316 \long\def\@processto#1{\@xproc #1\\\@@@\ifx\toaddress\@empty
317 \else \@yproc #1@@@\fi}
318 \long\def\@xproc #1\#2@@@{\def\toname{#1}\def\toaddress{#2}}
319 \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.

```
320 \newif\if@antwoordadres
321 \newcommand*\antwoordadres[1]{%
322 \@antwoordadrestrue\renewcommand*\@antwoordadres{#1}}
323 \newcommand*\@antwoordadres{}
324 \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.
325 \newdimen\adresveldbreedte

\adresveld This command formats the address window.
326 \newcommand*{\adresveld}{%
327   \hbox{} \kern-\topskip
328   \kern\@vensterskip
329   \beginingroup

  Compute the width of the address window
330   \if@adresrechts
331     \setlength\adresveldbreedte{4\refveldbreedte}%
332     \addtolength\adresveldbreedte{-76mm}%
333     \def\@tempa{\moveright 76mm}%
334   \else
335     \let\@tempa\relax
336     \setlength\adresveldbreedte{83mm}%
337   \fi

  Store the address in a box.
338   \setbox\@tempboxa\vtop{%
339     \hsize\adresveldbreedte
340     \@normalsize
341     \parindent\z@\parskip\z@
342     \rightskip0\p@\@plus\adresveldbreedte
343     \let\\\@nbreakcr \toname \\ \toaddress}

  Format the return address if one was given.
344   {\baselineskip\z@\lineskip\z@
345     \if@antwoordadres
346       \@tempa\vbox to \z{%
347         \hb@xt@\adresveldbreedte{%
348           \kleinvet
349           \def\{\unskip\enspace{\textbullet}\enspace\ignorespaces}%
350           \@antwoordadres\hfil}
351         \kern2\p@\hrule \vss}
352     \fi

  Print a small rule as typing aid if required.
353     \if@typhulp
354       \@tempa\llap{\vbox to \z{\vskip9mm\streepje\vss}}
355     \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.
356     \kern9mm \kern-\ht\@tempboxa \@tempdima=\dp\@tempboxa
357     \@tempa\box\@tempboxa \kern-\@tempdima
358     \vskip31mm\endgroup}
```

## 6.2.2 The reference line

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

```

359 \newdimen\refveldbreedte
360 \setlength\refveldbreedte{38mm}

```

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

```

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

```

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

```

362 \def\@refveld#1#2{%
363   \vtop{\hsize\refveldbreedte
364     \parskip\z@\parindent\z@
365     \everypar{}}%
366   \lineskiplimit\z@\baselineskip12\p@
367   \lineskip\z@
368   \rightskip0\p@ \@plus \refveldbreedte \@minus .5\refveldbreedte
369   \vbox{\refkopfont\baselineskip10\p@#1\@par}
370   \kern2\p@
371   \strut #2}}

```

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

```

\@UK 372 \newbox\@UB \newbox\@UK \newbox\@OK \newbox\@Dt
\@OK

```

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

```

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

```

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

```

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

```

`\onskenmerk` Store our reference in a box register.

```

377 \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.

```

378 \newcommand*\datum[1]{\def\@tempa{}\def\@tempb{#1}%
379   \ifx\@tempa\@tempb
380     \setbox\@Dt\hbox{ }%
381   \else
382     \@defrefveld{\@Dt}{\datumtekst}{#1}%
383   \fi}
384 \let\date\datum

```

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

```

385 \def\referentieregel{\hbox
386   {\hb@xt@\refveldbreedte{\copy\@UB\hfil}%
387   \hb@xt@\refveldbreedte{\copy\@UK\hfil}%
388   \hb@xt@\refveldbreedte{\copy\@OK\hfil}%
389   \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.

```
390 \newbox\@Ad
391 \def\vervolgreferentieregel{%
392   \hbox{%
393     \hb@xt@\refveldbreedte{\copy\@Ad\hfil}%
394     \hskip\refveldbreedte
395     \hb@xt@\refveldbreedte{\copy\@Dt\hfil}%
396     \@refveld{\bladnummertekst}{\thepage}\hss}}
```

### 6.2.3 The headers and footers

`\briefhoofd` The headings are empty by default.

```
\vervolghoofd 397 \newcommand*\briefhoofd{}
398 \newcommand*\vervolghoofd{\vbox{}}
```

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

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

`\@kortvervolgbriefhoofd` This creates a shortened heading for following pages

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

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

```
408 \newcommand*\@langvervolgbriefhoofd[2]{
409   \@maakbriefhoofd{#1}{#2}
410   \def\vervolghoofd{%
411     \vbox{\briefhoofd\vskip2mm
412           \vervolgreferentieregel
413           \vbox{}}}}
```

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

```
414 \newcommand*\@maakbriefhoofd[2]{\def\brefhoofd{%
415   \vbox{\hsize=4\refveldbreedte
416         \hb@xt@\hsize{\Large \normalfont\sffamily #1\strut\hfil}
417         \hrule
418         \moveright 3\refveldbreedte\@refveld{\strut #2}{}
419         \vbox{}}}}
```

`\@voetruimte` A box to store the footer in.

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

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

```
422 \newcount\@voetteller
```

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

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

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

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

#### 6.2.4 The little rules

`\streepje` A shorthand for one little rule.

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

`\streepjes` This prints the folding rules

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

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

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

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

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

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

```
438     \rlap{\kern24mm\vouwstreepjes}}
439   \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`.

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

`\vouwstreepjes` This prints two folding rules.

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

#### 6.2.5 Page breaking control

`\stopbreaks`

```
447 \def\stopbreaks{\interlinepenalty \@M
448   \def\par{\@par\nobreak}\let\@=\@nobreakcr
449   \let\vspace\@nobreakvspace}
```

```

\nobreakvspace
\nobreakvspacex 450 \def\nobreakvspace{\@ifstar{\@nobreakvspacex}{\@nobreakvspace}}
\nobreakcr      451
                452 \def\nobreakvspacex#1{\ifvmode\nobreak\vskip #1\relax\else
                453                 \bsphack\vadjust{\nobreak\vskip #1}\@esphack\fi}

                454 \def\nobreakcr{%
                455   \let\reserved@e\relax
                456   \let\reserved@f\relax
                457   \vadjust{\nobreak}\@ifstar{\@xnewline}{\@xnewline}}

```

`\startbreaks`

```

458 \def\startbreaks{\let\=\@normalcr
459   \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.
- `\telephonenumber` : argument of current `\telephone` declaration–null if none.

```

460 \newcommand*\opening[1]{%
461   \thispagestyle{firstpage}%
462   \adresveld
463   \prevdepth=-1000\p@ \vskip-2\p@ %% ???
464   \referentieregel
465   \@dosubject #1\par\nobreak}

```

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

```

466 \def\@dosubject{%
467   \ifx\@empty\@subject
468   \else
469     \par\noindent
470     \parbox[t]{\textwidth}
471       {\@hangfrom{\refkopfont \betrefttekst \enspace}%
472        \normalfont\rmfamily\ignorespaces \@subject\strut}%
473     \par
474   \fi}

```

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

```
475 \newcommand*\afsluiting[1]{\par\nobreak\vspace{\parskip}%
476   \stopbreaks
477   \ifx\@empty\fromsig
478     \def\ondertekening##1{\def\fromsig{##1}\@afsluiting{##1}}%
479   \else
480     \@afsluiting{##1}%
481   \fi}
482 \let\closing\afsluiting
483 \def\open@af{\vtop\bgroup\hsize.3\textwidth \raggedright}
```

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

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

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

```
\mdeskipamount 491 %\smallskipamount=.5\parskip
\bigskipamount 492 \medskipamount=\parskip
493 %\bigskipamount=2\parskip
```

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

```
\re 494 \newcommand*\betreft[1]{\def\@subject{##1}}
495 \let\onderwerp\betreft
496 \let\subject\betreft
497 \def\@subject{}
498 \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`.

```

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

\bijlage   \bijlagen{Foo(2)\Bar}
\bijlagen   which produces:
\encl      bijlagen:  Foo(2)
              Bar

502 \newcommand*\bijlage}[1]{%
503   \par\noindent
504   \parbox[t]{\textwidth}{\@hangfrom{\normalfont\bijlagetekst\ }%
505     \ignorespaces #1\strut}\par}
506 \newcommand*\bijlagen}[1]{%
507   \par\noindent
508   \parbox[t]{\textwidth}{\@hangfrom{\normalfont\bijlagetekst\ }%
509     \ignorespaces #1\strut}\par}
510 \let\encl\bijlagen

```

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

```
511 \def\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`.

```
512 \def\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
```

```
513 \def\returnaddress{}
```

```
\labelcount
```

```
514 \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.

```

515 \newcommand*\startlabels{\labelcount\z@
516 \pagestyle{empty}%
517 \let\@texttop\relax
518 \topmargin -50\p@
519 \headsep \z@
520 \oddsidemargin -35\p@
521 \evensidemargin -35\p@
522 \textheight 10in
523 \@colht\textheight \@colroom\textheight \vsize\textheight
524 \textwidth 550\p@
525 \columnsep 26\p@
526 \ifcase \@ptsize\relax
527 \normalsize
528 \or
529 \small
530 \or
531 \footnotesize
532 \fi
533 \baselineskip \z@
534 \lineskip \z@
535 \boxmaxdepth \z@
536 \parindent \z@
537 \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.

```
538 \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.

```

539 \newcommand*\mlabel}[2]{%
540 \parbox[b][2in][c]{262\p@}{\strut\ignorespaces #2}%
541 }

```

`\@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.

```
542 \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 L<sup>A</sup>T<sub>E</sub>X manual for an explanation of the meanings

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

```

\leftmargin For efficiency, level-one list's values are defined at top level, and \@listi is defined
\leftmargini to set only \leftmargin.
\leftmarginii 543 \setlength\leftmargini {2.5em}
\leftmarginiii The following three are calculated so that they are larger than the sum of
\leftmarginiv \labelsep and the width of the default labels (which are '(m)', 'vii.' and 'M.').
\leftmarginv 544 \setlength\leftmarginii {2.2em}
\leftmarginvi 545 \setlength\leftmarginiii {1.87em}
546 \setlength\leftmarginiv {1.7em}
547 \setlength\leftmarginv {1em}
548 \setlength\leftmarginvi {1em}

Here we set the top level leftmargin.
549 \setlength\leftmargin {\leftmargini}

\labelsep \labelsep is the distance between the label and the text of an item; \labelwidth
\labelwidth is the width of the label.
550 \setlength \labelsep {5\p@}
551 \setlength \labelwidth{\leftmargini}
552 \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.
553 \setlength\partopsep{0\p@}

\topsep Extra vertical space, in addition to \parskip, added above and below list and
paragraphing environments.
554 \setlength\topsep{.4em}

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

\@itempenalty This penalty is inserted between list items.
555 \@beginparpenalty -\@lowpenalty
556 \@endparpenalty -\@lowpenalty
557 \@itempenalty -\@lowpenalty

\@listI \@listI defines top level and \@listi values of \leftmargin, \parsep, \topsep,
\@listi and \itemsep
These values have been taken from the ones in the document class artikel3.
558 \def\@listI{\leftmargin\leftmargini
559 \labelsep.5em%
560 \labelwidth\leftmargin
561 \advance\labelwidth-\labelsep
562 \topsep .5\parskip \@plus \p@
563 \parsep \z@
564 \itemsep\parsep}
565 \let\@listi\@listI

```

We have to initialise these parameters.

```

566 \@listi

\@listii Here are the same macros for the higher level lists.
\@listiii 567 \def\@listii {\leftmargin\leftmarginii
\@listiv 568 \labelsep .5em%
\@listv 569 \labelwidth\leftmarginii
\@listvi 570 \advance\labelwidth-\labelsep
571 \topsep -.5\parskip \@plus \p@
572 \parsep \z@
573 \itemsep\parsep}
574 \def\@listiii{\leftmargin\leftmarginiii
575 \labelsep .5em%
576 \labelwidth\leftmarginiii
577 \advance\labelwidth-\labelsep
578 \topsep -.5\parskip \@plus \p@
579 \parsep \z@
580 \partopsep \z@
581 \itemsep \topsep}
582 \def\@listiv {\leftmargin\leftmarginiv
583 \labelsep .5em%
584 \labelwidth\leftmarginiv
585 \advance\labelwidth-\labelsep
586 \topsep -.5\parskip \@plus \p@}
587 \def\@listv {\leftmargin\leftmarginv
588 \labelsep .5em%
589 \labelwidth\leftmarginv
590 \advance\labelwidth-\labelsep
591 \topsep -.5\parskip \@plus \p@}
592 \def\@listvi {\leftmargin\leftmarginvi
593 \labelsep .5em%
594 \labelwidth\leftmarginvi
595 \advance\labelwidth-\labelsep
596 \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 The counters are already defined in in the LATEX kernel (ltlists.dtx), but their
\theenumii representation is changed here.
\theenumiii 597 \renewcommand*\theenumi{\@arabic\c@enumi}
\theenumiv 598 \renewcommand*\theenumii{\@alph\c@enumii}
599 \renewcommand*\theenumiii{\@roman\c@enumiii}
600 \renewcommand*\theenumiv{\@Alph\c@enumiv}

\labelenumi The label for each item is generated by the commands \labelenumi ... \labelenumiv.
\labelenumii 601 \newcommand*\labelenumi{\theenumi.}
\labelenumiii 602 \newcommand*\labelenumii{(\theenumii)}
\labelenumiv 603 \newcommand*\labelenumiii{\theenumiii.}
604 \newcommand*\labelenumiv{\theenumiv.}

```

`\p@enumii` The expansion of `\p@enumN\theenumN` defines the output of a `\ref` command  
`\p@enumiii` when referencing an item of the Nth level of an enumerated list.  
`\p@enumiv` 605 `\renewcommand*\p@enumii}{\theenumi}`  
606 `\renewcommand*\p@enumiii}{\theenumi(\theenumii)}`  
607 `\renewcommand*\p@enumiv}{\p@enumiii\theenumiii}`

### 6.4.3 Itemize

`\labelitemi` Itemization is controlled by `\labelitemi`, `\labelitemii`, `\labelitemiii`, and  
`\labelitemii` `\labelitemiv`, which define the labels of the various itemization levels: the sym-  
`\labelitemiii` bols used are bullet, bold en-dash, asterisk and centred dot.  
`\labelitemiv` 608 `\newcommand*\labelitemi}{\textbullet}`  
609 `\newcommand*\labelitemii}{\normalfont\bfseries \textendash}`  
610 `\newcommand*\labelitemiii}{\textasteriskcentered}`  
611 `\newcommand*\labelitemiv}{\textperiodcentered}`

### 6.4.4 Description

`description` The description environment is defined here – while the `itemize` and `enumerate`  
environments are defined in the L<sup>A</sup>T<sub>E</sub>X kernel (`ltxlists.dtx`).

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

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

```
616 \newcommand*\descriptionlabel[1]{\hspace\labelsep
617                               \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`.

```
618 \newenvironment{verse}
619     {\let\=\@centercr
620     \list{}{\setlength\itemsep{\z@}%
621             \setlength\itemindent{-15\p@}%
622             \setlength\listparindent{\itemindent}%
623             \setlength\rightmargin{\leftmargin}%
624             \addtolength\leftmargin{15\p@}}%
625     \item[]
626     {\endlist}
```

### 6.5.2 Quotation

`quotation` The quotation environment is also defined by making clever use of the list environ-  
ment's parameters. The lines in the environment are set smaller than `\textwidth`.  
The first line of a paragraph inside this environment is indented.

```

627 \newenvironment{quotation}
628     {\list{}{\setlength\listparindent{1.5em}%
629             \setlength\itemindent{\listparindent}%
630             \setlength\rightmargin{\leftmargin}}%
631     \item[]}
632     {\endlist}

```

### 6.5.3 Quote

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

```

633 \newenvironment{quote}
634     {\list{}{\setlength\rightmargin{\leftmargin}}%
635     \item[]}
636     {\endlist}

```

### 6.5.4 Theorem

This document class does not define it's own theorem environments, the defaults, supplied by L<sup>A</sup>T<sub>E</sub>X kernel (`lthm.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`.  
637 `\setlength\arraycolsep{5\p@}`

`\tabcolsep` The columns in an tabular environment are separated by `2\tabcolsep`.  
638 `\setlength\tabcolsep{6\p@}`

`\arrayrulewidth` The width of vertical rules in the array and tabular environments is given by `\arrayrulewidth`.  
639 `\setlength\arrayrulewidth{.4\p@}`

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

### 6.6.2 Tabbing

`\tabbingsep` This controls the space that the `\'` command puts in. (See L<sup>A</sup>T<sub>E</sub>X manual for an explanation.)  
641 `\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.

```
642 \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`.

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

#### 6.6.5 Equation and eqnarray

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

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

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

```
646 % \setlength\jot{3pt}
```

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

```
647 % \def\@eqnnum{(\theequation)}
```

### 6.7 Font changing

Here we supply the declarative font changing commands that were common in  $\text{\LaTeX}$  version 2.09 and earlier. These commands work in text mode *and* in math mode. They are provided for compatibility, but one should start using the `\text...` and `\math...` commands instead. These commands are redefined using `\@renewfontswitch`, a command with three arguments: the user command to be defined;  $\text{\LaTeX}$  commands to execute in text mode and  $\text{\LaTeX}$  commands to execute in math mode.

`\rm` The commands to change the family.

```
\tt 648 \DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\sff 649 \DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
650 \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.

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

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

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

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

```
655 \DeclareRobustCommand*\cal{\@fontswitch{\relax}{\mathcal}}
656 \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 height of the rule of 0.4pt by adding the right amount of vertical skip.

To prevent the rule from colliding with the footnote we first add a little negative vertical skip, then we put the rule and make sure we end up at the same point where we began this operation.

```
657 \renewcommand*\footnoterule{%
658   \kern-\p@
659   \hrule \@width .4\columnwidth
660   \kern .6\p@}
```

`\c@footnote` Footnotes are numbered within chapters in the report and book document styles.

```
661 % \newcounter{footnote}
```

`\@makefnmark` The footnote mechanism of L<sup>A</sup>T<sub>E</sub>X calls the macro `\@makefnmark` 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 `\@makefnmark` 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 T<sub>E</sub>X code.

```
\long\def\@makefnmark#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.



```

702 \def\bijlagetekst{Enclosures:}
703 \def\telefoontekst{telephone}}

```

\americanbrief This stores American english strings

```

704 \newcommand*{\americanbrief}{%
705 \def\uwbrieftekst{Your letter of}
706 \def\uwkenmerktekst{Your reference}
707 \def\onskenmerktekst{Our reference}
708 \def\datumtekst{Date}
709 \def\geadresseerdetekst{To}
710 \def\bladnummertekst{Page}
711 \def\vandaag{\ifcase\month\or
712 January\or February\or March\or April\or May\or June\or
713 July\or August\or September\or October\or November\or December\fi
714 \space\number\day, \number\year}
715 \def\betrefttekst{Re:}
716 \def\ccname{cc}
717 \def\bijlagetekst{Enclosure:}
718 \def\bijlagetekst{Enclosures:}
719 \def\telefoontekst{telephone}}

```

\germanbrief This stores the German versions of the strings.

```

720 \newcommand*{\germanbrief}{%
721 \def\uwbrieftekst{Ihr Brief vom}
722 \def\uwkenmerktekst{Ihr Zeichen}
723 \def\onskenmerktekst{Unser Zeichen}
724 \def\datumtekst{Datum}
725 \def\geadresseerdetekst{An}
726 \def\bladnummertekst{Seite}
727 \def\vandaag{\number\day.\~\ifcase\month\or
728 Januar\or Februar\or M"arz\or April\or Mai\or Juni\or
729 Juli\or August\or September\or Oktober\or November\or Dezember\fi
730 \space\number\year}
731 \def\betrefttekst{Betrifft:}
732 \def\ccname{Kopien an}
733 \def\bijlagetekst{Anlage:}
734 \def\bijlagetekst{Anlagen:}
735 \def\telefoontekst{Telefon}}

```

\frenchbrief And finally to store the french strings

```

736 \newcommand*{\frenchbrief}{%
737 \def\uwbrieftekst{Votre lettre du}
738 \def\uwkenmerktekst{Vos r'ef'erences:}
739 \def\onskenmerktekst{Nos r'ef'erences:}
740 \def\datumtekst{Date:}
741 \def\geadresseerdetekst{'A l'attention de}
742 \def\bladnummertekst{Page}
743 \def\vandaag{\number\day\ifnum\day=1$\^er}$\fi
744 \~\ifcase\month\or janvier\or
745 f'evrier\or mars\or avril\or mai\or juin\or
746 juillet\or ao\^ut\or septembre\or octobre\or
747 novembre\or d'ecembre\fi \space \number\year}
748 \def\betrefttekst{Objet:}
749 \def\ccname{Copie \a}

```

```

750 \def\bijlagetekst{Pi\‘ece jointe:}
751 \def\bijlagentekst{Pi\‘eces jointes:}
752 \def\telefoontekst{T\‘el\‘ephone:}}

```

## 6.10 Two column mode

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

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

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

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

## 6.11 The page style

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

```

755 \pagestyle{headings}
756 \pagenumbering{arabic}

```

## 6.12 Single or double sided printing

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

```
757 \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.

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

We always start in one column mode.

```

759 \onecolumn
760 </brief>

```