

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

Copyright (C) 1992 by Leslie Lamport  
Copyright (C) 1994,1999 by Victor Eijkhout      Johannes Braams

2004/06/07

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>The DOCSTRIP modules</b>	<b>3</b>
<b>3</b>	<b>Initial Code</b>	<b>3</b>
<b>4</b>	<b>Declaration of Options</b>	<b>4</b>
4.1	Setting Paper Sizes . . . . .	4
4.2	Choosing the type size . . . . .	5
4.3	Two-side or one-side printing . . . . .	5
4.4	Draft option . . . . .	5
4.5	Titlepage option . . . . .	5
4.6	openright option . . . . .	5
4.7	Table of contents formatting . . . . .	6
4.8	Formatting of the title . . . . .	6
4.9	Twocolumn printing . . . . .	6
4.10	Equation numbering on the left . . . . .	6
4.11	Flush left displays . . . . .	6
4.12	Open bibliography . . . . .	7
<b>5</b>	<b>Executing Options</b>	<b>7</b>
<b>6</b>	<b>Loading Packages</b>	<b>7</b>
<b>7</b>	<b>Document Layout</b>	<b>7</b>
7.1	Fonts . . . . .	8
7.2	Paragraphing . . . . .	10
7.3	Page Layout . . . . .	12
7.3.1	Vertical spacing . . . . .	12
7.3.2	The dimension of text . . . . .	13
7.3.3	Margins . . . . .	15
7.3.4	Footnotes . . . . .	17
7.3.5	Float placement parameters . . . . .	17

---

\*This file has version number v2.1a, last revised 2004/06/07.

7.4	Page Styles . . . . .	20
7.4.1	Marking conventions . . . . .	20
7.4.2	Defining the page styles . . . . .	21
<b>8</b>	<b>Document Markup</b>	<b>24</b>
8.1	The title . . . . .	24
8.2	Chapters and Sections . . . . .	28
8.2.1	Building blocks . . . . .	28
8.2.2	Mark commands . . . . .	31
8.2.3	Define Counters . . . . .	31
8.2.4	Front Matter, Main Matter, and Back Matter . . . . .	32
8.2.5	Parts . . . . .	33
8.2.6	Chapters . . . . .	36
8.2.7	Lower level headings . . . . .	38
8.3	Lists . . . . .	40
8.3.1	General List Parameters . . . . .	40
8.3.2	Enumerate . . . . .	42
8.3.3	Itemize . . . . .	43
8.3.4	Description . . . . .	44
8.4	Adapting existing environments . . . . .	44
8.5	Defining new environments . . . . .	45
8.5.1	Abstract . . . . .	45
8.5.2	Verse . . . . .	46
8.5.3	Quotation . . . . .	46
8.5.4	Quote . . . . .	47
8.5.5	Theorem . . . . .	47
8.5.6	Titlepage . . . . .	47
8.5.7	Appendix . . . . .	48
8.6	Setting parameters for existing environments . . . . .	48
8.6.1	Array and tabular . . . . .	48
8.6.2	Tabbing . . . . .	49
8.6.3	Minipage . . . . .	49
8.6.4	Framed boxes . . . . .	49
8.6.5	Equation and eqnarray . . . . .	49
8.7	Floating objects . . . . .	50
8.7.1	Figure . . . . .	50
8.7.2	Table . . . . .	51
8.7.3	Captions . . . . .	51
8.8	Font changing . . . . .	52
<b>9</b>	<b>Cross Referencing</b>	<b>53</b>
9.1	Table of Contents, etc. . . . .	53
9.1.1	Table of Contents . . . . .	54
9.1.2	List of figures . . . . .	59
9.1.3	List of tables . . . . .	59
9.2	Bibliography . . . . .	59
9.3	The index . . . . .	61
9.4	Footnotes . . . . .	61

<b>10 Initialization</b>	<b>63</b>
10.1 Words . . . . .	63
10.2 Date . . . . .	64
10.3 Two column mode . . . . .	64
10.4 The page style . . . . .	64
10.5 Single or double sided printing . . . . .	64

## 1 Introduction

This file contains the set of document classes that were made available by Working Group 13 of the NTG (Nederlandstalige  $\TeX$  Gebruikersgroep). They are compatible with the standard  $\LaTeX$ 2e document classes, but implement different layouts.

## 2 The DOCSTRIP modules

The following modules are used in the implementation to direct DOCSTRIP in generating the external files:

artikel	produce the documentclasses artikel?
rapport	produce the documentclasses rapport?
10pt	produce the class option for 10pt
11pt	produce the class option for 11pt
12pt	produce the class option for 12pt
boek	produce the documentclasses boek?
type1	produce the ‘1’ variants of the classes
type2	produce the ‘2’ variants of the classes
type3	produce the ‘3’ variants of the classes
driver	produce a documentation driver file

## 3 Initial Code

In this part we define a few commands that are used later on.

<code>\@ptsize</code>	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. <pre> 1 (*artikel   rapport   boek) 2 \newcommand*\@ptsize{} 3 </pre>
<code>\if@restonecol</code>	When the document has to printed in two columns, we sometimes have to temporarily switch to one column. This switch is used to remember to switch back. <pre> 4 \newif\if@restonecol </pre>
<code>\if@titlepage</code>	A switch to indicate if a titlepage has to be produced. For the artikel document class the default is not to make a seperate titlepage. <pre> 5 \newif\if@titlepage 6 (artikel)\@titlepagefalse 7 (!artikel)\@titlepagetrue </pre>

<code>\if@openright</code>	A switch to indicate if chapters must start on a right-hand page. The default for the report class is no; for the book class it's yes. 8 <code>!\artikel\newif\if@openright</code>
<code>\if@mainmatter</code>	The switch <code>\if@mainmatter</code> , only available in the document class book, indicates whether we are processing the main material in the book. 9 <code>\book\newif\if@mainmatter \@mainmattertrue</code>
<code>\if@oldtoc</code>	A switch to indicate if 'old' layout of the table of contents should be produced. These document classes normally produce a table of contents that looks quite different from what the standard classes produce. 10 <code>\newif\if@oldtoc</code> 11 <code>\@oldtocfalse</code>
<code>\if@allcaps</code>	By default the text on the titlepage is set in capital letters. This can be disabled by the option <code>mctitle</code> , which sets the switch <code>\if@allcaps</code> to false. 12 <code>\newif\if@allcaps</code>
<code>\if@titlecentered</code>	In the document classes <code>artikel3</code> and <code>rapport3</code> the default placement of the title that is produced by <code>\maketitle</code> is <code>flushleft</code> . This can be changed by the switch <code>\if@titlecentered</code> . 13 <code>\type3\newif\if@titlecentered</code> 14 <code>\type3\@titlecenteredfalse</code>
<code>\if@revlabel</code>	These document classes need to be able to change the positioning of the label in labeled lists. This switch is used for that purpose. 15 <code>\newif\if@revlabel</code>

## 4 Declaration of Options

### 4.1 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. Classes for real book production will probably add other paper sizes and additionally the production of crop marks for trimming.

```

16 \DeclareOption{a4paper}
17   {\setlength\paperheight {297mm}%
18    \setlength\paperwidth  {210mm}}
19 \DeclareOption{a5paper}
20   {\setlength\paperheight {210mm}%
21    \setlength\paperwidth  {148mm}}
22 \DeclareOption{b5paper}
23   {\setlength\paperheight {250mm}%
24    \setlength\paperwidth  {176mm}}
25 \DeclareOption{letterpaper}
26   {\setlength\paperheight {11in}%
27    \setlength\paperwidth  {8.5in}}
28 \DeclareOption{legalpaper}
29   {\setlength\paperheight {14in}%
30    \setlength\paperwidth  {8.5in}}

```

```

31 \DeclareOption{executivepaper}
32   {\setlength\paperheight {10.5in}%
33    \setlength\paperwidth  {7.25in}}

```

The option `landscape` switches the values of `\paperheight` and `\paperwidth`, assuming the dimensions were given for portrait paper.

```

34 \DeclareOption{landscape}
35   {\setlength\@tempdima  {\paperheight}%
36    \setlength\paperheight {\paperwidth}%
37    \setlength\paperwidth  {\@tempdima}}

```

## 4.2 Choosing the type size

The type size options are handled by defining `\@ptsize` to contain the last digit of the size in question and branching on `\ifcase` statements. This is done for historical reasons to stay compatible with other packages that use the `\@ptsize` variable to select special actions. It makes the declarations of size options less than 10pt difficult, although one can probably use 9 and 8 assuming that a class won't define both 8pt and 18pt options.

```

38 \DeclareOption{10pt}{\renewcommand\@ptsize{0}}
39 \DeclareOption{11pt}{\renewcommand\@ptsize{1}}
40 \DeclareOption{12pt}{\renewcommand\@ptsize{2}}

```

## 4.3 Two-side or one-side printing

For two-sided printing we use the switch `\if@twoside`. In addition we have to set the `\if@mparswitch` to get any margin paragraphs into the outside margin.

```

41 \DeclareOption{oneside}{\@twosidefalse \mparswitchfalse}
42 \DeclareOption{twoside}{\@twosidetrue  \mparswitchtrue}

```

## 4.4 Draft option

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

```

43 \DeclareOption{draft}{\setlength\overfullrule{5pt}}
44 \DeclareOption{final}{\setlength\overfullrule{0pt}}

```

## 4.5 Titlepage option

An article usually has no separate titlepage, but the user can request one.

```

45 \DeclareOption{titlepage}{\@titlepagetrue}
46 \DeclareOption{notitlepage}{\@titlepagefalse}

```

## 4.6 openright option

This option determines whether or not a chapter must start on a right-hand page request one.

```

47 \!artikel\DeclareOption{openright}{\@openrighttrue}
48 \!artikel\DeclareOption{openany}{\@openrightfalse}

```

For these document classes there used to be a file `voorwerk.sty` which was a replacement for `titlepag.sty`. Therefore we also have the option `voorwerk`.

```
49 \DeclareOption{voorwerk}{\@titlepagetrue}
50 \DeclareOption{geenvoorwerk}{\@titlepagefalse}
```

## 4.7 Table of contents formatting

This document class uses a new layout for the table of contents, but in order to maintain compatibility with the standard L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> document classes we supply an extra option: `oldtoc`. If this option is specified the switch `\ifoldtoc` will be set true.

```
51 \DeclareOption{oldtoc}{\@oldtoctrue}
```

## 4.8 Formatting of the title

The option `titlecentered` changes the behaviour of the `\maketitle` command. It then produces a title like it does for the `artikel1` document class.

```
52 \type3\DeclareOption{titlecentered}{\@titlecenteredtrue}
```

In the `rapport` and `boek` document styles the titlepage uses all capital letters. The option `mctitle` (for ‘mixed case’) prevents this.

```
53 \langle rapport | boek \rangle \DeclareOption{mctitle}{\@allcapsfalse}
54 \langle rapport | boek \rangle \DeclareOption{uctitle}{\@allcapstrue}
```

## 4.9 Twocolumn printing

Two-column and one-column printing is again realized via a switch.

```
55 \DeclareOption{onecolumn}{\@twocolumnfalse}
56 \DeclareOption{twocolumn}{\@twocolumntrue}
```

## 4.10 Equation numbering on the left

The option `leqno` can be used to get the equation numbers on the left side of the equation. It loads code which is generated automatically from the kernel files when the format is built. If the equation number does get a special formatting then instead of using the kernel file the class would need to provide the code explicitly.

```
57 \DeclareOption{leqno}{\input{leqno.clo}}
```

## 4.11 Flush left displays

The option `fleqn` redefines the displayed math environments in such a way that they come out flush left, with an indentation of `\mathindent` from the prevailing left margin. It loads code which is generated automatically from the kernel files when the format is built.

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

## 4.12 Open bibliography

The option `openbib` produces the “open” bibliography style, in which each block starts on a new line, and succeeding lines in a block are indented by `\bibindent`.

```
59 \DeclareOption{openbib}{%
```

First some hook into the bibliography environment is filled.

```
60 \AtEndOfPackage{%
61   \renewcommand\@openbib@code{%
62     \advance\leftmargin\bibindent
63     \itemindent -\bibindent
64     \listparindent \itemindent
65     \parsep \z@
66   }%
```

In addition the definition of `\newblock` is overwritten.

```
67   \renewcommand\newblock{\par}}%
68 }
```

## 5 Executing Options

Here we execute the default options to initialize certain variables. Note that the document class ‘boek’ always uses two sided printing.

```
69 (*artikel)
70 \ExecuteOptions{a4paper,10pt,oneside,onecolumn,final,uctitle}
71 (/artikel)
72 (*rapport)
73 \ExecuteOptions{a4paper,10pt,oneside,onecolumn,final,uctitle,openany}
74 (/rapport)
75 (*boek)
76 \ExecuteOptions{a4paper,10pt,twoside,onecolumn,final,uctitle,openright}
77 (/boek)
```

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.

```
78 \ProcessOptions
```

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

```
79 \input{ntg1\@ptsize.clo}
80 (/artikel | rapport | boek)
```

## 6 Loading Packages

These class files do not load additional packages.

## 7 Document Layout

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

## 7.1 Fonts

L<sup>A</sup>T<sub>E</sub>X offers the user commands to change the size of the font, relative to the ‘main’ size. Each relative size changing command `\size` executes the command `\setfontsize\size<font-size><baselineskip>` where:

`<font-size>` The absolute size of the font to use from now on.

`<baselineskip>` The normal value of `\baselineskip` for the size of the font selected. (The actual value will be `\baselinestretch * <baselineskip>`.)

A number of commands, defined in the L<sup>A</sup>T<sub>E</sub>X kernel, shorten the following definitions and are used throughout. They are:

<code>\@vpt</code>	5	<code>\@vipt</code>	6	<code>\@viipt</code>	7
<code>\@viiipt</code>	8	<code>\@ixpt</code>	9	<code>\@xpt</code>	10
<code>\@xipt</code>	10.95	<code>\@xiipt</code>	12	<code>\@xivpt</code>	14.4
...					

`\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

```

81 (*10pt | 11pt | 12pt)
82 \renewcommand\normalsize{%
83 (*10pt)
84   \@setfontsize\normalsize\@xpt\@xipt
85   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
86   \abovedisplayshortskip \z@ \@plus3\p@
87   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
88 </10pt)
89 (*11pt)
90   \@setfontsize\normalsize\@xipt{13.6}%
91   \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
92   \abovedisplayshortskip \z@ \@plus3\p@
93   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
94 </11pt)
95 (*12pt)
96   \@setfontsize\normalsize\@xiipt{14.5}%
97   \abovedisplayskip 12\p@ \@plus3\p@ \@minus7\p@
98   \abovedisplayshortskip \z@ \@plus3\p@
99   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
100 </12pt)

```

The `\belowdisplayskip` is always equal to the `\abovedisplayskip`. The parameters of the first level list are always given by `\@listI`.

```

101 \belowdisplayskip \abovedisplayskip
102 \let\@listi\@listI}

```

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

```

103 \let\@normalsize\normalsize

```

We initially choose the normalsize font.

```
104 \normalsize

\small This is similar to \normalsize.
105 \newcommand*\small{%
106 (*10pt)
107   \@setfontsize\small\@ixpt{11}%
108   \abovedisplayskip 8.5\p@ \@plus3\p@ \@minus4\p@
109   \abovedisplayshortskip \z@ \@plus2\p@
110   \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
111 /10pt)
112 (*11pt)
113   \@setfontsize\small\@xpt\@xipt
114   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
115   \abovedisplayshortskip \z@ \@plus3\p@
116   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
117 /11pt)
118 (*12pt)
119   \@setfontsize\small\@xipt{13.6}%
120   \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
121   \abovedisplayshortskip \z@ \@plus3\p@
122   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
123 /12pt)
124   \belowdisplayskip \abovedisplayskip
125 }

\footnotesize This is similar to \normalsize.
126 \newcommand*\footnotesize{%
127 (*10pt)
128   \@setfontsize\footnotesize\@viiipt{9.5}%
129   \abovedisplayskip 6\p@ \@plus2\p@ \@minus4\p@
130   \abovedisplayshortskip \z@ \@plus\p@
131   \belowdisplayshortskip 3\p@ \@plus\p@ \@minus2\p@
132 /10pt)
133 (*11pt)
134   \@setfontsize\footnotesize\@ixpt{11}%
135   \abovedisplayskip 8\p@ \@plus2\p@ \@minus4\p@
136   \abovedisplayshortskip \z@ \@plus\p@
137   \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
138 /11pt)
139 (*12pt)
140   \@setfontsize\footnotesize\@xpt\@xipt
141   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
142   \abovedisplayshortskip \z@ \@plus3\p@
143   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
144 /12pt)
145   \belowdisplayskip \abovedisplayskip
146 }

\scriptsize These are all much simpler than the previous macros, they just select a new
\tiny fontsize, but leave the parameters for displays and lists alone.
\large 147 (*10pt)
\Large 148 \newcommand*\scriptsize{\@setfontsize\scriptsize\@viipt\@viiipt}
\LARGE
\huge
\Huge
```

```

149 \newcommand* \tiny{\@setfontsize \tiny \@vpt \@vipt}
150 \newcommand* \large{\@setfontsize \large \@xipt{14}}
151 \newcommand* \Large{\@setfontsize \Large \@xivpt{18}}
152 \newcommand* \LARGE{\@setfontsize \LARGE \@xvipt{22}}
153 \newcommand* \huge{\@setfontsize \huge \@xxpt{25}}
154 \newcommand* \Huge{\@setfontsize \Huge \@xxvpt{30}}
155 </10pt>
156 (*11pt)
157 \newcommand* \scriptsize{\@setfontsize \scriptsize \@viipt{9.5}}
158 \newcommand* \tiny{\@setfontsize \tiny \@vipt \@vipt}
159 \newcommand* \large{\@setfontsize \large \@xipt{14}}
160 \newcommand* \Large{\@setfontsize \Large \@xivpt{18}}
161 \newcommand* \LARGE{\@setfontsize \LARGE \@xvipt{22}}
162 \newcommand* \huge{\@setfontsize \huge \@xxpt{25}}
163 \newcommand* \Huge{\@setfontsize \Huge \@xxvpt{30}}
164 </11pt>
165 (*12pt)
166 \newcommand* \scriptsize{\@setfontsize \scriptsize \@viipt{9.5}}
167 \newcommand* \tiny{\@setfontsize \tiny \@vipt \@vipt}
168 \newcommand* \large{\@setfontsize \large \@xivpt{18}}
169 \newcommand* \Large{\@setfontsize \Large \@xvipt{22}}
170 \newcommand* \LARGE{\@setfontsize \LARGE \@xxpt{25}}
171 \newcommand* \huge{\@setfontsize \huge \@xxvpt{30}}
172 \let \Huge = \huge
173 </12pt>
174 </10pt | 11pt | 12pt>

```

## 7.2 Paragraphing

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

```

175 (*artikel | rapport | boek)
176 \setlength \lineskip {1\p@}
177 \setlength \normallineskip {1\p@}

```

`\baselinestretch` This is used as a multiplier for `\baselineskip`. The default is to *not* stretch the baselines. Note that if this command doesn't resolve to "empty" any plus or minus part in the specification of `\baselineskip` is ignored.

```

178 \renewcommand \baselinestretch {}

```

`\unitindent` These document classes all use a single dimension for a number of layout parameters:

- the label width in section heading,
- the `\parindent`
- the footnote label indent (= half `\unitindent`)
- listindent on the first level

```

179 \newdimen \unitindent

```

The default setting accomodates three levels of single digit section numbering.

```
180 <*type1 | type3>
181 {\setbox0\hbox{\normalsize\rmfamily 2.2.2\hskip.5em}}
182 \global\unitindent=\wd0}
183 </type1 | type3>
```

`\othermargin` Other indentations are maximal label width plus white space.

```
184 \newdimen\othermargin
185 {\setbox0\hbox{\normalsize (m)\hskip.6em}\global\othermargin=\wd0}
```

`if@needwriteindent` If this is not enough, a new width is calculated, set, and the file.aux file contains an instruction that will set `\unitindent` on the next run.

For this we need a switch

```
186 <*type1 | type3>
187 \newif\if@needwriteindent
```

`\@indentset` And a command that sets the various parameters.

```
188 \newcommand*\@indentset{%
189 <!type3> \global\parindent=\unitindent
190 \global\leftmargini=\unitindent
191 \global\@needwriteindenttrue}
```

`\@writeindent` The `\end{document}` command will call `\@writeindent` to write the final width of `\unitindent` on the .aux file. Also a command is written to set `\unitindent`. To be compatible with other document classes a check is written to the .aux file for the existence of `\unitindent`. This prevents nasty errors when another document class is used.

```
192 \newcommand*\@writeindent[1]{\immediate\write\@mainaux
193   {\string@ifundefined{unitindent}{\string\newdimen\string\unitindent
194     \let\string\@indentset\relax}{}}
195   \immediate\write\@mainaux{\global\string\unitindent=#1\string\relax
196     \string\@indentset \string\relax}}
```

We need to use the hook into `\end{document}` to write the final value of `\unitindent` om the file.aux file for the next run.

```
197 \AtEndDocument{%
198   \if@filesw
199     \if@needwriteindent
200       \@writeindent{the\unitindent}
201     \fi
202   \fi}
203 </type1 | type3>
```

In the document class `artikel2` the width of `\unitindent` is fixed and related to `\othermargin`.

```
204 <type2>\unitindent=2\othermargin
```

`\parskip` `\parskip` gives extra vertical space between paragraphs and `\parindent` is the width of the paragraph indentation. The value of `\parindent` depends on whether we are in two column mode.

```
205 <*type1>
206 \setlength\parskip{0\p@}
```

```

207 \setlength\parindent{\unitindent}
208 \type1}
209 \type3}
210 \setlength\parskip{.5\baselineskip \@plus .1\baselineskip
211 \@minus .1\baselineskip}
212 \setlength\parindent{\z@}
213 \type3}

```

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

```

214 \@lowpenalty 51
215 \@medpenalty 151
216 \@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.

```

217 % \clubpenalty 150
218 % \widowpenalty 150

```

`\displaywidowpenalty` Discourage (but not so much) widows in front of a math display and forbid breaking directly in front of a display. Allow break after a display without a penalty. Again the default values are used, therefore we only show them here.

```

219 % \displaywidowpenalty 50
220 % \predisplaypenalty 10000
221 % \postdisplaypenalty 0

```

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

```

222 % \interlinepenalty 0

```

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

```

223 % \brokenpenalty 0
224 \</artikel | rapport | boek>

```

## 7.3 Page Layout

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

### 7.3.1 Vertical spacing

`\headheight` The `\headheight` is the height of the box that will contain the running head. The `\headsep` is the distance between the bottom of the running head and the top of the text. `\topskip` is the `\baselineskip` for the first line on a page.

```

225 \<10pt | 11pt | 12pt>
226 \setlength\headheight{12\p@}
227 \setlength\headsep {25\p@}
228 \<10pt>\setlength\topskip {10\p@}
229 \<11pt>\setlength\topskip {11\p@}
230 \<12pt>\setlength\topskip {12\p@}

```

`\footskip` The distance from the baseline of the box which contains the running footer to the baseline of last line of text is controlled by the `\footskip`. Bottom of page:

```
231 \setlength\footskip{30\p@} %
```

`\maxdepth` The T<sub>E</sub>X 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 T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X 2.09 `\maxdepth` had a fixed value of 4pt; in native L<sup>A</sup>T<sub>E</sub>X2e 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, therefore we set `\maxdepth` to half the value of `\topskip`.

```
232 \if@compatibility
233   \setlength\maxdepth{4\p@}
234 \else
235   \setlength\maxdepth{.5\topskip}
236 \fi
```

### 7.3.2 The dimension of text

`\textwidth` When we are in compatibility mode we have to make sure that the dimensions of the printed area are not different from what the user was used to see.

```
237 \if@compatibility
238   \if@twocolumn
239     \setlength\textwidth{410\p@}
240   \else
241     <10pt> \setlength\textwidth{345\p@}
242     <11pt> \setlength\textwidth{360\p@}
243     <12pt> \setlength\textwidth{390\p@}
244   \fi
```

When we are not in compatibility mode we can set some of the dimensions differently, taking into account the paper size for instance.

```
245 \else
```

First, we calculate the maximum `\textwidth`, which will we will allow on the selected paper and store it in `\@tempdima`. Then we store the length of a line with approximately 60 – 70 characters in `\@tempdimb`. The values given are taken from the file `a4.sty` by Johannes Braams and Nico Poppelier and are more or less suitable when Computer Modern fonts are used.

```
246   \setlength\@tempdima{\paperwidth}
247   \addtolength\@tempdima{-2in}
248   <10pt> \setlength\@tempdimb{361\p@}
249   <11pt> \setlength\@tempdimb{376\p@}
250   <12pt> \setlength\@tempdimb{412\p@}
```

Now we can set the `\textwidth`, depending on whether we will be setting one or two columns.

In two column mode each *column* shouldn't be wider than `\@tempdimb` (which could happen on A3 paper for instance).

```
251   \if@twocolumn
252     \ifdim\@tempdima>2\@tempdimb\relax
253       \setlength\textwidth{2\@tempdimb}
254     \else
255       \setlength\textwidth{\@tempdima}
256   \fi
```

In one column mode the text should not be wider than the minimum of the paperwidth (minus 2 inches for the margins) and the maximum length of a line as defined by the number of characters.

```

257 \else
258 \ifdim\@tempdima>\@tempdimb\relax
259 \setlength\textwidth{\@tempdimb}
260 \else
261 \setlength\textwidth{\@tempdima}
262 \fi
263 \fi
264 \fi

```

Here we modify the width of the text a little to be a whole number of points.

```

265 \if@compatibility
266 \else
267 \@settopoint\textwidth
268 \fi

```

`\textheight` Now that we have computed the width of the text, we have to take care of the height. The `\textheight` is the height of text (including footnotes and figures, excluding running head and foot).

First make sure that the compatibility mode gets the same dimensions as we had with L<sup>A</sup>T<sub>E</sub>X2.09. The number of lines was calculated as the floor of the old `\textheight` minus `\topskip`, divided by `\baselineskip` for `\normalsize`. The old value of `\textheight` was 528pt.

```

269 \if@compatibility
270 <10pt> \setlength\textheight{43\baselineskip}
271 <11pt> \setlength\textheight{38\baselineskip}
272 <12pt> \setlength\textheight{36\baselineskip}

```

Again we compute this, depending on the papersize and depending on the `\baselineskip` that is used, in order to have a whole number of lines on the page.

```

273 \else
274 \setlength\@tempdima{\paperheight}

```

We leave at least a 1 inch margin on the top and the bottom of the page.

```

275 \addtolength\@tempdima{-2in}

```

We also have to leave room for the running headers and footers.

```

276 \addtolength\@tempdima{-1.5in}

```

Then we divide the result by the current `\baselineskip` and store this in the count register `\@tempcnta`, which then contains the number of lines that fit on this page.

```

277 \divide\@tempdima\baselineskip
278 \@tempcnta=\@tempdima

```

From this we can calculate the height of the text.

```

279 \setlength\textheight{\@tempcnta\baselineskip}
280 \fi

```

The first line on the page has a height of `\topskip`.

```

281 \advance\textheight by \topskip

```

### 7.3.3 Margins

Most of the values of these parameters are now calculated, based on the papersize in use. In the calculations the `\marginparsep` needs to be taken into account so we give it its value first.

`\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`.

```
282 \if@twocolumn
283 \setlength\marginparsep {10\p@}
284 \else
285 <10pt> \setlength\marginparsep{11\p@}
286 <11pt> \setlength\marginparsep{10\p@}
287 <12pt> \setlength\marginparsep{10\p@}
288 \fi
289 <10pt | 11pt>\setlength\marginparpush{5\p@}
290 <12pt>\setlength\marginparpush{7\p@}
```

Now we can give the values for the other margin parameters. For native L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>, these are calculated.

`\oddsidemargin` First we give the values for the compatibility mode.  
`\evensidemargin` Values for two-sided printing:

```
\marginparwidth 291 \if@compatibility
292 \if@twoside
293 <10pt> \setlength\oddsidemargin {44\p@}
294 <11pt> \setlength\oddsidemargin {36\p@}
295 <12pt> \setlength\oddsidemargin {21\p@}
296 <10pt> \setlength\evensidemargin {82\p@}
297 <11pt> \setlength\evensidemargin {74\p@}
298 <12pt> \setlength\evensidemargin {59\p@}
299 <10pt> \setlength\marginparwidth {107\p@}
300 <11pt> \setlength\marginparwidth {100\p@}
301 <12pt> \setlength\marginparwidth {85\p@}
```

Values for one-sided printing:

```
302 \else
303 <10pt> \setlength\oddsidemargin {63\p@}
304 <11pt> \setlength\oddsidemargin {54\p@}
305 <12pt> \setlength\oddsidemargin {39.5\p@}
306 <10pt> \setlength\evensidemargin {63\p@}
307 <11pt> \setlength\evensidemargin {54\p@}
308 <12pt> \setlength\evensidemargin {39.5\p@}
309 <10pt> \setlength\marginparwidth {90\p@}
310 <11pt> \setlength\marginparwidth {83\p@}
311 <12pt> \setlength\marginparwidth {68\p@}
312 \fi
```

And values for two column mode:

```
313 \if@twocolumn
314 \setlength\oddsidemargin {30\p@}
315 \setlength\evensidemargin {30\p@}
316 \setlength\marginparwidth {48\p@}
317 \fi
```

When we are not in compatibility mode we can take the dimensions of the selected paper into account.

The values for `\oddsidemargin` and `\marginparwidth` will be set depending on the status of the `\if@twoside`.

If `@twoside` is true (which is always the case for boek) we make the inner margin smaller than the outer one.

```
318 \else
319   \if@twoside
320     \setlength\@tempdima      {\paperwidth}
321     \addtolength\@tempdima   {-\textwidth}
322     \setlength\oddsidemargin {.4\@tempdima}
323     \addtolength\oddsidemargin {-1in}
```

The width of the margin for text is set to the remainder of the width except for a ‘real margin’ of white space of width 0.4in. A check should perhaps be built in to ensure that the (text) margin width does not get too small!

```
324     \setlength\marginparwidth  {.6\@tempdima}
325     \addtolength\marginparwidth {-\marginparsep}
326     \addtolength\marginparwidth {-0.4in}
```

For one-sided printing we center the text on the page, by calculating the difference between `textwidth` and `\paperwidth`. Half of that difference is then used for the margin (thus `\oddsidemargin` is 1in less).

```
327   \else
328     \setlength\@tempdima      {\paperwidth}
329     \addtolength\@tempdima   {-\textwidth}
330     \setlength\oddsidemargin {.5\@tempdima}
331     \addtolength\oddsidemargin {-1in}
332     \setlength\marginparwidth {.5\@tempdima}
333     \addtolength\marginparwidth {-\marginparsep}
334     \addtolength\marginparwidth {-.4in}
335   \fi
```

With the above algorithm the `\marginparwidth` can come out quite large which we may not want.

```
336   \ifdim \marginparwidth >2in
337     \setlength\marginparwidth{2in}
338   \fi
```

Having done these calculations we make them pt values.

```
339   \@settopoint\oddsidemargin
340   \@settopoint\marginparwidth
```

The `\evensidemargin` can now be computed from the values set above.

```
341   \setlength\evensidemargin {\paperwidth}
342   \addtolength\evensidemargin{-2in}
343   \addtolength\evensidemargin{-\textwidth}
344   \addtolength\evensidemargin{-\oddsidemargin}
```

Setting `\evensidemargin` to a full point value may produce a small error. However it will lie within the error range a doublesided printer of today's technology can accurately print.

```
345   \@settopoint\evensidemargin
346 \fi
```

`\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.

It can now be computed from the values set above.

```

347 \if@compatibility
348   \setlength\topmargin{27pt}
349 \else
350   \setlength\topmargin{\paperheight}
351   \addtolength\topmargin{-2in}
352   \addtolength\topmargin{-\headheight}
353   \addtolength\topmargin{-\headsep}
354   \addtolength\topmargin{-\textheight}
355   \addtolength\topmargin{-\footskip}      % this might be wrong!

```

By changing the factor in the next line the complete page can be shifted vertically.

```

356   \addtolength\topmargin{-.5\topmargin}
357   \@settopoint\topmargin
358 \fi

```

### 7.3.4 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.

```

359 <10pt>\setlength\footnotesep{6.65\p@}
360 <11pt>\setlength\footnotesep{7.7\p@}
361 <12pt>\setlength\footnotesep{8.4\p@}

```

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

```

362 <10pt>\setlength\skip\footins}{9\p@ \@plus 4\p@ \@minus 2\p@}
363 <11pt>\setlength\skip\footins}{10\p@ \@plus 4\p@ \@minus 2\p@}
364 <12pt>\setlength\skip\footins}{10.8\p@ \@plus 4\p@ \@minus 2\p@}
365 </10pt | 11pt | 12pt>

```

### 7.3.5 Float placement parameters

All float parameters are given default values in the L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> kernel. For this reason parameters that are not counters need to be set with `\renewcommand`.

#### Limits for the placement of floating objects

`\c@topnumber` The `topnumber` counter holds the maximum number of floats that can appear on the top of a text page.

```

366 (*artikel | rapport | boek)
367 \setcounter{topnumber}{2}

```

`\topfraction` This indicates the maximum part of a text page that can be occupied by floats at the top.

```

368 \renewcommand\topfraction{.7}

```

`\c@bottomnumber` The *bottomnumber* counter holds the maximum number of floats that can appear on the bottom of a text page.  
369 `\setcounter{bottomnumber}{1}`

`\bottomfraction` This indicates the maximum part of a text page that can be occupied by floats at the bottom.  
370 `\renewcommand\bottomfraction{.3}`

`\c@totalnumber` This indicates the maximum number of floats that can appear on any text page.  
371 `\setcounter{totalnumber}{3}`

`\textfraction` This indicates the minimum part of a text page that has to be occupied by text.  
372 `\renewcommand\textfraction{.2}`

`\floatpagefraction` This indicates the minimum part of a page that has to be occupied by floating objects before a ‘float page’ is produced.  
373 `\renewcommand\floatpagefraction{.5}`

`\c@dbltopnumber` The *dbltopnumber* counter holds the maximum number of two column floats that can appear on the top of a two column text page.  
374 `\setcounter{dbltopnumber}{2}`

`\dbltopfraction` This indicates the maximum part of a two column text page that can be occupied by two column floats at the top.  
375 `\renewcommand\dbltopfraction{.7}`

`\dblfloatpagefraction` This indicates the minimum part of a page that has to be occupied by two column wide floating objects before a ‘float page’ is produced.  
376 `\renewcommand\dblfloatpagefraction{.5}`  
377 `</artikel | rapport | boek`

### Floats on a text page

`\floatsep` When a floating object is placed on a page with text, these parameters control the separation between the float and the other objects on the page. These parameters are used for both one-column mode and single-column floats in two-column mode.

`\textfloatsep` `\floatsep` is the space between adjacent floats that are moved to the top or bottom of the text page.

`\intextsep` `\textfloatsep` is the space between the main text and floats at the top or bottom of the page.

`\intextsep` is the space between in-text floats and the text.

```

378 (*10pt)
379 \setlength\floatsep    {12\p@ \@plus 2\p@ \@minus 2\p@}
380 \setlength\textfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
381 \setlength\intextsep   {12\p@ \@plus 2\p@ \@minus 2\p@}
382 </10pt)
383 (*11pt)
384 \setlength\floatsep    {12\p@ \@plus 2\p@ \@minus 2\p@}
385 \setlength\textfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
386 \setlength\intextsep   {12\p@ \@plus 2\p@ \@minus 2\p@}
387 </11pt)

```

```

388 (*12pt)
389 \setlength\floatsep {12\p@ \@plus 2\p@ \@minus 4\p@}
390 \setlength\textfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
391 \setlength\intextsep {14\p@ \@plus 4\p@ \@minus 4\p@}
392 (/12pt)

```

`\dblfloatsep` `\dbltextfloatsep` When floating objects that span the whole `\textwidth` are placed on a text page when we are in twocolumn mode the separation between the float and the text is controlled by `\dblfloatsep` and `\dbltextfloatsep`.

`\dblfloatsep` is the space between adjacent floats that are moved to the top or bottom of the text page.

`\dbltextfloatsep` is the space between the main text and floats at the top or bottom of the page.

```

393 (*10pt)
394 \setlength\dblfloatsep {12\p@ \@plus 2\p@ \@minus 2\p@}
395 \setlength\dbltextfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
396 (/10pt)
397 (*11pt)
398 \setlength\dblfloatsep {12\p@ \@plus 2\p@ \@minus 2\p@}
399 \setlength\dbltextfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
400 (/11pt)
401 (*12pt)
402 \setlength\dblfloatsep {14\p@ \@plus 2\p@ \@minus 4\p@}
403 \setlength\dbltextfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
404 (/12pt)

```

### Floats on their own page or column

`\@fptop` `\@fpsep` `\@fpbot` When floating objects are placed on separate pages the layout of such pages is controlled by these parameters. At the top of the page `\@fptop` amount of stretchable whitespace is inserted, at the bottom of the page we get an `\@fpbot` amount of stretchable whitespace. Between adjacent floats the `\@fpsep` is inserted.

These parameters are used for the placement of floating objects in one column mode, or in single column floats in two column mode.

Note that at least one of the two parameters `\@fptop` and `\@fpbot` should contain a `plus ...fil` to allow filling the remaining empty space.

```

405 (*10pt)
406 \setlength\@fptop{0\p@ \@plus 1fil}
407 \setlength\@fpsep{8\p@ \@plus 2fil}
408 \setlength\@fpbot{0\p@ \@plus 1fil}
409 (/10pt)
410 (*11pt)
411 \setlength\@fptop{0\p@ \@plus 1fil}
412 \setlength\@fpsep{8\p@ \@plus 2fil}
413 \setlength\@fpbot{0\p@ \@plus 1fil}
414 (/11pt)
415 (*12pt)
416 \setlength\@fptop{0\p@ \@plus 1fil}
417 \setlength\@fpsep{10\p@ \@plus 2fil}
418 \setlength\@fpbot{0\p@ \@plus 1fil}
419 (/12pt)

```

`\@dblftop` Double column floats in two column mode are handled with similar parameters.

```

\@dblfpsep 420 (*10pt)
\@dblfpbot 421 \setlength\@dblftop{0\p@ \@plus 1fil}
            422 \setlength\@dblfpsep{8\p@ \@plus 2fil}
            423 \setlength\@dblfpbot{0\p@ \@plus 1fil}
            424 </10pt>
            425 (*11pt)
            426 \setlength\@dblftop{0\p@ \@plus 1fil}
            427 \setlength\@dblfpsep{8\p@ \@plus 2fil}
            428 \setlength\@dblfpbot{0\p@ \@plus 1fil}
            429 </11pt>
            430 (*12pt)
            431 \setlength\@dblftop{0\p@ \@plus 1fil}
            432 \setlength\@dblfpsep{10\p@ \@plus 2fil}
            433 \setlength\@dblfpbot{0\p@ \@plus 1fil}
            434 </12pt>
            435 (*artikel | rapport | boek)

```

## 7.4 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`.

`\thispagestyle` Several commands (`\index`, `\maketitle`) give a `\thispagestyle{plain}` command, which will overrule a `\pagestyle{empty}` command. This situation is almost always unwanted. Therefore we provide a more careful definition.

First save the original definition.

```
436 \let\Thispagestyle\thispagestyle
```

Then we provide the new definition, for which we must also adapt `\pagestyle` a little.

```

437 \newcommand*\@emptypagestyle{empty}
438 \renewcommand*\pagestyle[1]{\@nameuse{ps@#1}\def\@currentpagestyle{#1}}
439 \renewcommand*\thispagestyle[1]{%
440   \ifx\@currentpagestyle\@emptypagestyle
441     \else
442       \global\@specialpagetrue
443       \gdef\@specialstyle{#1}%
444     \fi}

```

### 7.4.1 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.

#### 7.4.2 Defining the page styles

The `pagestyle empty` is defined in `latex.dtx`, but the `pagestyle plain` is slightly altered here. The difference is that the page numbers are set flush right in `onesided` and flush left and right in the `twosided` style.

`\ps@plain`

```
445 \renewcommand*\ps@plain{%
```

The running head are empty in this `pagestyle`, the page number appears in the running foot.

```

446   \let\@oddhead\@empty\let\@evenhead\@empty
447   \def\@oddfoot{\hfil\PageFont\thepage}%
448   \if@twoside
449     \def\@evenfoot{\PageFont\thepage\hfil}%
450   \else
451     \let\@evenfoot\@oddfoot
452   \fi

```

Because the running heads should be empty we `let \mkboth` to `\gobbletwo`, thus disabling the mark commands.

```
453   \let\mkboth\gobbletwo}
```

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

```

454 \if@twoside
455   \def\ps@headings{%

```

The running feet are empty in this page style, the running head contains the page number and one of the marks.

```
456 \let\@oddfoot\@empty\let\@evenfoot\@empty
457 \def\@evenhead{\PageFont\thepage}\hfil\MarkFont\leftmark}%
458 \def\@oddhead{\MarkFont\rightmark}\hfil\PageFont\thepage}%
```

When using this page style, the contents of the running head is determined by the chapter and section titles. So we \let \mkboth to \markboth.

```
459 \let\mkboth\markboth
```

For the artikel document classes we define \sectionmark to clear the right mark and put the number of the section (when it is numbered) and its title in the left mark. The rightmark is set by \subsectionmark to contain the subsection titles.

Note the use of ##1 for the parameter of the \sectionmark command, which will be defined when \ps@headings is executed.

```
460 (*artikel)
461 \def\sectionmark##1{%
462 \markboth {\MakeUppercase{%
463 \ifnum \c@secnumdepth >\z@
464 \thesection\quad
465 \fi
466 ##1}}{}}%
467 \def\subsectionmark##1{%
468 \markright {%
469 \ifnum \c@secnumdepth >\@ne
470 \thesubsection\quad
471 \fi
472 ##1}}
473 (/artikel)
```

In the rapport and boek document classes we use the \chaptermark and \sectionmark macros to fill the running heads.

Note the use of ##1 for the parameter of the \chaptermark command, which will be defined when \ps@headings is executed.

```
474 (*rapport | boek)
475 \def\chaptermark##1{%
476 \markboth {\MakeUppercase{\ifnum \c@secnumdepth >\m@ne
477 (boek) \if@mainmatter
478 \@chapapp\ thechapter. \ %
479 (boek) \fi
480 \fi
481 ##1}}{}}%
482 \def\sectionmark##1{%
483 \markright {\MakeUppercase{\ifnum \c@secnumdepth >\z@
484 \thesection. \ \fi
485 ##1}}}}
486 (/rapport | boek)
```

The definition of \ps@headings for one sided printing can be much simpler, because we treat even and odd pages the same. Therefore we don't need to define \@even....

```
487 \else
488 \def\ps@headings{%
```

```

489 \let\@oddfoot\@empty
490 \def\@oddhead{\MarkFont\rightmark}\hfil\PageFont\thepage}%
491 \let\@mkboth\markboth

```

We use `\markright` now instead of `\markboth` as we did for two sided printing.

```

492 (*artikel)
493 \def\sectionmark##1{%
494 \markright {\MakeUppercase{%
495 \ifnum \c@secnumdepth >\m@ne
496 \thesection\quad
497 \fi
498 ##1}}}}
499 (/artikel)

500 (*rapport | boek)
501 \def\chaptermark##1{%
502 \markright {\MakeUppercase{%
503 \ifnum \c@secnumdepth >\m@ne
504 (boek) \if@mainmatter
505 \@chapapp\ thechapter. \ %
506 (boek) \fi
507 \fi
508 ##1}}}}
509 (/rapport | boek)
510 \fi

```

`\ps@myheadings` The definition of the page style *myheadings* is fairly simple because the user determines the contents of the running head himself by using the `\markboth` and `\markright` commands.

```

511 \def\ps@myheadings{%
512 \let\@oddfoot\@empty\let\@evenfoot\@empty
513 \def\@evenhead{\PageFont\thepage}\hfil\MarkFont\leftmark}%
514 \def\@oddhead{\MarkFont\rightmark}\hfil\PageFont\thepage}%

```

We have to make sure that the marking commands that are used by the chapter and section headings are disabled. We do this `\let`ting them to a macro that gobbles its argument(s).

```

515 \let\@mkboth\@gobbletwo
516 (!artikel) \let\chaptermark@gobble
517 \let\sectionmark@gobble
518 (artikel) \let\subsectionmark@gobble
519 }

```

`\PageFont` These macros are use to store the fonts that are used to typeset the pagenumber  
`\MarkFont` (`\PageFont`) and the marks (`\MarkFont`) in the running head and feet.

```

520 \newcommand*\PageFont{\rmfamily}
521 \newcommand*\MarkFont{\slshape}

```

`\RunningFonts` Use this macro to change the fonts that are used in the running heads.

```

522 \newcommand*\RunningFonts[2]{%
523 \renewcommand*\PageFont{#1}\renewcommand*\MarkFont{#2}}

```

## 8 Document Markup

### 8.1 The title

`\title` These three macros are provided by `latex.dtx` to provide information about the title, author(s) and date of the document. The information is stored away in internal control sequences. It is the task of the `\maketitle` command to use the information provided. The definitions of these macros are shown here for information.

```
524 % \newcommand*\title[1]{\gdef\@title{#1}}
525 % \newcommand*\author[1]{\gdef\@author{#1}}
526 % \newcommand*\date[1]{\gdef\@date{#1}}
```

The `\date` macro gets today's date by default.

```
527 % \gdef\@date{\today}
```

`\TitleFont` This selects the font to use in the title of the document.

```
528 \newcommand*\TitleFont{\bfseries}
```

`\maketitle` The definition of `\maketitle` depends on whether a separate title page is made. This is the default for the `rapport` and `boek` document classes, but for the `artikel` classes it is optional. Note that the title, author and date information is printed in capital letters by default. This can be changed by the option `mctitle`.

When we are making a title page, we locally redefine `\footnotesize` and `\footnoterule` to change the appearance of the footnotes that are produced by the `\thanks` command.

```
529 (!boek)\if@titlepage
530 \renewcommand*\TitleFont{\rmfamily}
531 \newcommand*\maketitle{%
532   \begin{titlepage}%
533     \let\footnotesize\small
534     \let\footnoterule\relax
535     \let \footnote \thanks
```

Footnotes on the titlepage, generated by the use of `\thanks`, use symbols in these document classes.

```
536   \long\def\@makefnmark##1{\parindent\z@
537     \def\labelitemi{\textendash}\@revlabeltrue
538     \leavevmode\@textsuperscript{\@thefnmark}\kern1em\relax ##1}
539   \renewcommand*\thefootnote{\@fnsymbol\c@footnote}%
```

We center the entire title vertically; the centering is set off a little by adding a `\vskip`. In compatibility mode the `pagenumber` is set to 0 to keep the behaviour of L<sup>A</sup>T<sub>E</sub>X 2.09 style files

```
540   \if@compatibility\setcounter{page}{0}\fi
541   \null\vfil
542   \vskip 60\p@
```

Then we set the title, in a `\LARGE` font; leave a little space and set the author(s) in a `\large` font. We do this inside a tabular environment to get them in a single column. Before the date we leave a little whitespace again.

```
543   \begin{center}%
544     \TitleFont
545     {\LARGE \def\\{\penalty -\@M}
```

```

546     \if@allcaps
547       \expandafter\uc@nothanks\@title\thanks\relax
548     \else
549       \@title
550     \fi\par}%
551 \vskip 3em%
552 {\large
553  \lineskip .75em \parindent\z@
554  \begin{tabular}[t]{c}%
555    \if@allcaps
556      \expandafter\uc@authornothanks\@author\and\relax
557    \else
558      \@author
559    \fi
560  \end{tabular}\par}%
561 \vskip 1.5em%
562 {\large
563  \if@allcaps
564    \uppercase\expandafter{\@date}%
565  \else
566    \@date
567  \fi\par}%
568 \end{center}\par

```

Then we call `\@thanks` to print the information that goes into the footnote and finish the page.

```

569   \@thanks
570   \vfil\null
571 \end{titlepage}%

```

We reset the `footnote` counter, disable `\thanks` and `\maketitle` and save some storage space by emptying the internal information macros.

```

572 \setcounter{footnote}{0}%
573 \global\let\thanks\relax
574 \global\let\maketitle\relax
575 \global\let\@thanks\@empty
576 \global\let\@author\@empty
577 \global\let\@title\@empty
578 \global\let\@date\@empty

```

After the title is set the declaration commands `\title`, etc. can vanish. The definition of `\and` makes only sense within the argument of `\author` so this can go as well.

```

579 \global\let\title\relax
580 \global\let\author\relax
581 \global\let\date\relax
582 \global\let\and\relax
583 }

```

We want to have the title, author and date information in uppercase, but we have to be very careful not to put too much text in uppercase. The macros that perform the filtering of texts that shouldn't be in uppercase were developed with the help of Howard Trickey.

`\uc@nothanks` This macro takes all the text up to the first use of `\thanks` and passes it to

\uppercase. The use of \futurelet will store the token *after* the \thanks in \@tempa. The macro \u@tx uses that information to determine what to do next.

```
584 \def\uc@nothanks#1\thanks{\uppercase{#1}\futurelet\@tempa\u@tx}
```

**\uc@authornothanks** A document can have more than one author. Usually they are separated with \and. For each author a footnote –using \thanks can be present. Therefore this macro takes all the text up to the first use of \and, thus picking up all the information for one author. This is then passed to \uc@nothanks, which checks for the presence of \thanks. For this to work the argument of \uc@nothanks has to be delimited by \thanks\relax.

```
585 \def\uc@authornothanks#1\and{\uc@nothanks#1\thanks\relax
```

Then we have to check whether the \and we found earlier was put in by the user, in which case information for another user will follow, or by the call from another macro, in which case the \and will be followed by a \relax token. The \futurelet construct stores the first token *after* the \and in \@tempa to be inspected by \u@ax.

```
586 \futurelet\@tempa\u@ax}
```

**\uc@ax** When \@tempa contains a \relax token nothing needs to be done, when it doesn't we put in a linebreak \\ the word 'and' (stored in \andname so that this control sequence can be redefined for other languages), another linebreak and we call \uc@authornothanks to continue processing. The \expandafter lets T<sub>E</sub>X see the \fi first.

```
587 \def\uc@ax{%
588 \ifx\@tempa\relax
589 \else
590 \\ \andname \\ \expandafter\uc@authornothanks
591 \fi}
```

**\uc@tx** This macro simply checks whether \@tempa contains a \relax token. When it doesn't further processing is performed by \u@ty.

```
592 \def\uc@tx{\ifx\@tempa\relax
593 \else \expandafter\uc@ty \fi}
```

**\uc@ty** The macro \uc@ty gets executed when the \thanks that delimited text earlier on in the processing had a real argument. In that case it was a \thanks put in by the user, *not* by these macros. Therefore the argument is now passed to \thanks and processing continues by calling \uc@nothanks.

```
594 \def\uc@ty#1{\thanks{#1}\uc@nothanks}
```

When the title is not on a page of its own, the layout of the title is a little different. We use symbols to mark the footnotes and we have to deal with two column documents.

Therefore we first start a new group to keep changes local. Then we redefine \thefootnote to use \fnsymbol; and change \@makefnmark so that footnotemarks have zero width (to make the centering of the author names look better). We also want raised footnotemarkers in the footnotes here.

```
595 (*!boek)
596 \else
597 \newcommand*\maketitle{\par
```

```

598 \begingroup
599 \renewcommand*{\thefootnote}{\@fnsymbol\c@footnote}%
600 \!type2) \def\@makefnmark{\rlap{%
601 \!type2) \@textsuperscript{\normalfont\@thefnmark}}}%
602 \!type2) \long\def\@makefntext{\@xmakefntext{%
603 \!type2) \@textsuperscript{\normalfont\@thefnmark}}}%
604 \*type2)

605 \long\def\@makefntext##1{\parindent\z@
606 \def\labelitemi{\textendash}%
607 \leavevmode\hb@xt@.5\unitindent{%
608 \@textsuperscript{\normalfont\@thefnmark}\hfil}##1}
609 \!type2)

```

If this is a twocolumn document we start a new page in twocolumn mode, with the title set to the full width of the text. The actual printing of the title information is left to `\@maketitle`.

```

610 \if@twocolumn
611 \ifnum \col@number=\@ne
612 \maketitle
613 \else
614 \twocolumn[\@maketitle]%
615 \fi
616 \else

```

When this is not a twocolumn document we just start a new page, prevent floating objects from appearing on the top of this page and print the title information.

```

617 \newpage
618 \global\@topnum\z@
619 \maketitle
620 \fi

```

This page gets a *plain* layout. We call `\@thanks` to produce the footnotes.

```

621 \thispagestyle{plain}\@thanks

```

Now we can close the group, reset the *footnote* counter, disable `\thanks`, `\maketitle` and `\@maketitle` and save some storage space by emptying the internal information macros.

```

622 \endgroup
623 \setcounter{footnote}{0}%
624 \global\let\thanks\relax
625 \global\let\maketitle\relax
626 \global\let\@maketitle\relax
627 \global\let\@thanks\@empty
628 \global\let\@author\@empty
629 \global\let\@title\@empty
630 \global\let\@date\@empty
631 \global\let\title\relax
632 \global\let\author\relax
633 \global\let\date\relax
634 \global\let\and\relax
635 }

```

`\@maketitle` This macro takes care of formatting the title information when we have no separate title page.

We always start a new page, leave some white space and center the information. The title is set in a `\LARGE` font, the author names and the in a `\large` font.

```

636 \def\@maketitle{%
637   \newpage
638   \null
639   \vskip 2em%
640 (type3)\if@titlecentered
641   \begin{center}%
642     \let \footnote \thanks
643     {\LARGE \TitleFont \@title \par}%
644     \vskip 1.5em%
645     {\large \TitleFont
646       \lineskip .5em%
647       \begin{tabular}[t]{c}%
648         \@author
649         \end{tabular}\par}%
650     \vskip 1em%
651     {\large \TitleFont \@date}%
652   \end{center}%
653 (*type3)
654   \else
655     {\LARGE \TitleFont \head@style \@title \par} \vskip 1.5em
656     {\large \TitleFont \lineskip .5em \tabcolsep\z@
657       \def\and{%% \begin{tabular} has already started
658         \end{tabular}\hskip 1em plus .17fil
659         \begin{tabular}[t]{l}%% \end{tabular} will come
660         \begin{tabular}[t]{l}\@author\end{tabular}\par}
661     \vskip 1em {\large \TitleFont \@date}
662   \fi
663 </type3>
664   \par
665   \vskip 1.5em}
666 \fi
667 </!boek>

```

## 8.2 Chapters and Sections

### 8.2.1 Building blocks

The definitions in this part of the class file make use of two macros, `\@startsection` and `\secdef`, which are defined by `latex.dtx`. To understand what is going on here, we describe their syntax.

The macro `\@startsection` has 6 required arguments, optionally followed by a `*`, an optional argument and a required argument:

```

\@startsection<name><level><indent><beforekip><afterskip><style> optional *
    [<altheading>]<heading>

```

It is a generic command to start a section, the arguments have the following meaning:

**<name>** The name of the user level command, e.g., ‘section’.

**<level>** A number, denoting the depth of the section – e.g., chapter=1, section = 2, etc. A section number will be printed if and only if *<level>* ≤ the value of the *secnumdepth* counter.

- ⟨*indent*⟩ The indentation of the heading from the left margin
- ⟨*beforeskip*⟩ The absolute value of this argument gives the skip to leave above the heading. If it is negative, then the paragraph indent of the text following the heading is suppressed.
- ⟨*afterskip*⟩ If positive, this gives the skip to leave below the heading, else it gives the skip to leave to the right of a run-in heading.
- ⟨*style*⟩ Commands to set the style of the heading. Since the June 1996 release of L<sup>A</sup>T<sub>E</sub>X the *last* command in this argument may be a command such as `\MakeUppercase` or `\fbox` that takes an argument. The section heading will be supplied as the argument to this command. So setting #6 to, say, `\bfseries\MakeUppercase` would produce bold, uppercase headings.
- \* When this is missing the heading is numbered and the corresponding counter is incremented.
- ⟨*altheading*⟩ Gives an alternative heading to use in the table of contents and in the running heads. This should be not present when the \* form is used.
- ⟨*heading*⟩ The heading of the new section.

A sectioning command is normally defined to `\@startsection` and its first six arguments.

The macro `\secdef` can be used when a sectioning command is defined without using `\@startsection`. It has two arguments:

```
\secdef⟨unstarcmds⟩⟨starcmds⟩
```

⟨*unstarcmds*⟩ Used for the normal form of the sectioning command.

⟨*starcmds*⟩ Used for the \*-form of the sectioning command.

You can use `\secdef` as follows:

```
\def\chapter { ... \secdef \CMDA \CMDB }
\def\CMDA    [#1]#2{ ... } % Command to define
                % \chapter[...]{...}
\def\CMDB    #1{ ... }   % Command to define
                % \chapter*{...}
```

`\head@style` In the definition of chapter and section commands a number of settings frequently occur. Therefore we store them in a control sequence.

Section headings are to be set extremely raggedright, with no hyphenations, not even at explicit hyphens.

```
668 \newcommand*\head@style{%
669   \interlinepenalty \@M
670   \hyphenpenalty=\@M \exhyphenpenalty=\@M
671   \rightskip=0cm plus .7\hsize\relax}
```

`\@sect` The definition of this macro from `latex.dtx` needs to be repeated here because we want to modify its behaviour with respect to:

1. the width of the number, which is fixed;

2. checking the value of `\unitindent`;
3. formatting the section title ragged right;
4. changing the argument of `\contentsline`.

```

672 \def\@sect#1#2#3#4#5#6[#7]#8{%
673   \ifnum #2>\c@secnumdepth
674     \let\@svsec\@empty
675   \else
676     \refstepcounter{#1}%

```

The following code (within the group) checks the value of `\unitindent`. If the sectionnumber is wider than `\unitindent` its value is adapted and a flag is set to remember to store the new value in the `.aux`-file.

```

677 (*type1 | type3)
678   \begingroup
679     \setbox\@tempboxa=\hbox{#6\relax
680                               \csname the#1\endcsname
681                               \hskip.5em}
682   \ifdim\wd\@tempboxa>\unitindent
683     \global\unitindent=\wd\@tempboxa
684     \@indentset
685   \fi
686   \endgroup
687 </type1 | type3)

```

Since `\@secntformat` might end with an improper `\hskip` which is scanning forward for plus or minus we end the definition of `\@svsec` with `\relax` as a precaution.

```

688   \protected@edef\@svsec{\@secntformat{#1}\relax}%
689   \fi
690   \@tempkipa #5\relax
691   \ifdim \@tempkipa>\z@
692     \begingroup

```

This `{` used to be after the argument to `\@hangfrom` but was moved here to allow commands such as `\MakeUppercase` to be used at the end of `#6`.

```

693   #6{%
694 (*type1 | type3)
695   \@hangfrom{\hskip #3\relax\@svsec}\head@style #8\endgraf}%
696 </type1 | type3)
697 (*type2)
698   \@hangfrom{\hskip #3}
699           \head@style\@svsec \hskip.3em\relax #8\endgraf}
700 </type2)
701   \endgroup
702   \csname #1mark\endcsname{#7}%
703   \addcontentsline{toc}{#1}{%
704     \ifnum #2>\c@secnumdepth
705     \else
706       \protect\numberline{\csname the#1\endcsname}%
707     \fi
708     \toc@font#2 #7}%
709   \else

```

```

710     \def\@svsechd{#6\hskip #3\relax
711         \@svsec #8\csname #1mark\endcsname{#7}%
712         \addcontentsline{toc}{#1}{%
713             \ifnum #2>\c@secnumdepth
714             \else
715                 \protect\numberline{\csname the#1\endcsname}%
716             \fi
717             \toc@font#2 #7}}%
718     \fi
719     \@xsect{#5}}

```

This macro was introduced in L<sup>A</sup>T<sub>E</sub>X<sub>2</sub> $\epsilon$ , its definition is changed here to get the fixed width of the section number.

```

720 \def\@seccntformat#1{%
721 \!type2) \hb@xt@\unitindent{\csname the#1\endcsname \hfil}%
722 \!type2) \csname the#1\endcsname\hskip.3em\relax
723 }

```

`\@ssect` Similar changes need to be made to the definition of `\@ssect`, which is used in ‘starred’ sections.

```

724 \def\@ssect#1#2#3#4#5{\@tempskipa #3\relax
725 \ifdim \@tempskipa>\z@
726     \begingroup

```

This { used to be after the argument to `\@hangfrom` but was moved here to allow commands such as `\MakeUppercase` to be used at the end of #6.

```

727     #4{%
728         \@hangfrom{\hskip #1}\head@style #5\endgraf}%
729     \endgroup
730 \else
731     \def\@svsechd{#4\hskip #1\relax #5}%
732 \fi
733 \@xsect{#3}}

```

## 8.2.2 Mark commands

`\chaptermark` Default initializations of `\...mark` commands. These commands are used in the definition of the page styles (see section 7.4.2) Most of them are already defined by `latex.tex`, so they are only shown here.

```

\subsubsectionmark 734 \!artikel) \newcommand*\chaptermark[1]{}
\paragraphmark     735 % \newcommand*\sectionmark[1]{}
\subparagraphmark  736 % \newcommand*\subsectionmark[1]{}
                   737 % \newcommand*\subsubsectionmark[1]{}
                   738 % \newcommand*\paragraphmark[1]{}
                   739 % \newcommand*\subparagraphmark[1]{}

```

## 8.2.3 Define Counters

`\c@secnumdepth` The value of the counter `secnumdepth` gives the depth of the highest-level sectioning command that is to produce section numbers.

```

740 \!artikel) \setcounter{secnumdepth}{3}
741 \!artikel) \setcounter{secnumdepth}{2}

```

`\c@part` These counters are used for the section numbers. The macro  
`\c@chapter` `\newcounter{<newctr>}[<oldctr>]`  
`\c@section` defines `<newctr>` to be a counter, which is reset to zero when counter `<oldctr>` is  
`\c@subsection` stepped. Counter `<oldctr>` must already be defined.  
`\c@subsubsection` 742 `\newcounter {part}`  
`\c@paragraph` 743 `<artikel>\newcounter {section}`  
`\c@subparagraph` 744 `<*rapport | boek>`  
745 `\newcounter {chapter}`  
746 `\newcounter {section}[chapter]`  
747 `</rapport | boek>`  
748 `\newcounter {subsection}[section]`  
749 `\newcounter {subsubsection}[subsection]`  
750 `\newcounter {paragraph}[subsubsection]`  
751 `\newcounter {subparagraph}[paragraph]`

`\thepart` For any counter `CTR`, `\theCTR` is a macro that defines the printed version of  
`\thechapter` counter `CTR`. It is defined in terms of the following macros:  
`\thesection` `\arabic{COUNTER}` prints the value of `COUNTER` as an arabic numeral.  
`\thesubsection` `\roman{COUNTER}` prints the value of `COUNTER` as a lowercase roman num-  
`\thesubsubsection` beral.  
`\theparagraph` `\Roman{COUNTER}` prints the value of `COUNTER` as an uppercase roman  
`\thesubparagraph` numeral.  
`\alph{COUNTER}` prints the value of `COUNTER` as a lowercase letter: 1 = a,  
2 = b, etc.  
`\Alph{COUNTER}` prints the value of `COUNTER` as an uppercase letter:  
1 = A, 2 = B, etc.  
Actually to save space the internal counter representations and the commands  
operating on those are used.  
752 `\renewcommand*\thepart{\@Roman\c@part}`  
753 `<artikel>\renewcommand\thesection{\@arabic\c@section}`  
754 `<*rapport | boek>`  
755 `\renewcommand*\thechapter{\@arabic\c@chapter}`  
756 `\renewcommand*\thesection{\thechapter.\@arabic\c@section}`  
757 `</rapport | boek>`  
758 `\renewcommand*\thesubsection{\thesection.\@arabic\c@subsection}`  
759 `\renewcommand*\thesubsubsection{\thesubsection.\@arabic\c@subsubsection}`  
760 `\renewcommand*\theparagraph{\thesubsubsection.\@arabic\c@paragraph}`  
761 `\renewcommand*\thesubparagraph{\theparagraph.\@arabic\c@subparagraph}`

`\@chapapp` `\@chapapp` is initially defined to be `'\chaptername'`. The `\appendix` command  
redefines it to be `'\appendixname'`.  
762 `<rapport | boek>\newcommand*\@chapapp{\chaptername}`

## 8.2.4 Front Matter, Main Matter, and Back Matter

A boek contains these three sections. First, we define the switch `\@mainmatter` that is true iff we are processing Main Matter. When this switch is false, the `\chapter` command does not print chapter numbers.

Here we define the commands that start these sections.

`\frontmatter` This command starts Roman page numbering and turns off chapter numbering.

```

763 (*boek)
764 \newcommand*\frontmatter{%
765   \cleardoublepage
766   \@mainmatterfalse
767   \pagenumbering{roman}}

```

`\mainmatter` This command clears the page, starts arabic page numbering and turns on chapter numbering.

```

768 \newcommand*\mainmatter{%
769   \cleardoublepage
770   \@mainmattertrue
771   \pagenumbering{arabic}}

```

`\backmatter` This clears the page, turns off chapter numbering and leaves page numbering unchanged.

```

772 \newcommand*\backmatter{%
773   \if@openright\cleardoublepage\else\clearpage\fi
774   \@mainmatterfalse}
775 </boek>

```

## 8.2.5 Parts

`\part` The command to start a new part of our document.

In the `artikel` classes the definition of `\part` is rather simple; we start a new paragraph, add a little white space, suppress the indentation of the first paragraph (not for the `artikel2` document class) and make use of `\@secdef`.

```

776 (*artikel)
777 \newcommand*\part{%
778   \if@noskipsec \leavevmode \fi
779   \par
780   \addvspace{4ex}%
781   (!type2)   \@afterindentfalse
782   (type2)   \@afterindenttrue
783   \secdef\@part\@spart}
784 </artikel>

```

For the `rapport` and `boek` classes we things a bit different.

We start a new (righthand) page and use the `empty` pagestyle.

```

785 (*rapport | boek)
786 \newcommand*\part{%
787   \cleardoublepage
788   \thispagestyle{empty}}

```

When we are making a two column document, this will be a one column page. We use `@tempswa` to remember to switch back to two columns.

```

789   \if@twocolumn
790     \onecolumn
791     \@tempswatrue
792   \else
793     \@tempswafalse
794   \fi

```

We need an empty box to prevent the fil glue from disappearing.

```

795   \null\vfil

```

Here we use `\secdef` to indicate which commands to use to make the actual heading.

```
796 \secdef\@part\@spart}
797 </rapport | boek>
```

`\@part` This macro does the actual formatting of the title of the part. Again the macro is differently defined for the artikel document classes than for the document classes `rapport` and `boek`.

`\PartFont` The font used to typeset the part is stored in this maro.

```
798 \newcommand*\PartFont{\bfseries}
```

When `secnumdepth` is larger than  $-1$  for the artikel document classes, we have a numbered part, otherwise it is unnumbered.

```
799 (*artikel)
800 \def\@part[#1]#2{%
801     \ifnum \c@secnumdepth >\m@ne
802         \refstepcounter{part}%
803         \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
804     \else
805         \addcontentsline{toc}{part}{#1}%
806     \fi
```

We print the title flush left in the artikel classes. Also we prevent breaking between lines and reset the font.

```
807     {\head@style
808     \parindent\unitindent
809     \normalfont
```

When this is a numbered part we have to print the number and the title. The `\nobreak` should prevent a page break here.

```
810     \ifnum \c@secnumdepth >\m@ne
811 <!type2>         \Large\PartFont\noindent \partname\nobreakspace\thepart
812 <type2>         \Large\PartFont\indent \partname\nobreakspace\thepart
813         \par\nobreak
814     \fi
815 <!type2>         \Large \PartFont \noindent #2%
816 <type2>         \Large \PartFont #2%
```

Then we empty the mark registers, leave some white space and call `\@afterheading` to takes care of suppressing the indentation.

```
817     \markboth{}{\par}%
818     \nobreak
819     \vskip 3ex
820     \@afterheading}
821 </artikel>
```

When `secnumdepth` is larger than  $-2$  for the document class `rapport` and `boek`, we have a numbered part, otherwise it is unnumbered.

```
822 (*rapport | boek)
823 \def\@part[#1]#2{%
824     \ifnum \c@secnumdepth >-2\relax
825         \refstepcounter{part}%
826         \addcontentsline{toc}{part}{\thepart\hspace{1em}\toc@case{#1}}%
827     \else
```

```
828     \addcontentsline{toc}{part}{\toc@case{#1}}%
829     \fi
```

We empty the mark registers and center the title on the page in the rapport and boek document classes. Also we prevent breaking between lines and reset the font.

```
830     \markboth{}{}%
831     {\centering
832     \interlinepenalty \@M
833     \normalfont
```

When this is a numbered part we have to print the number. We have to expand `\partname` before `\uppercase` is called, therefore we use a temporary control sequence that, when called will execute `\uppercase` on the contents of `\partname`.

```
834     \ifnum \c@secnumdepth >-2\relax
835         \Large\PartFont
836         \edef\@tempa{\noexpand\uppercase{\partname}}\@tempa
837         \nobreakspace\thepart
838         \par
```

We leave some space before we print the title and leave the finishing up to `\@endpart`.

```
839         \vskip 20\p@
840         \fi
841         \Large \PartFont \uppercase{#2}\par}%
842     \@endpart}
843 </rapport | boek>
```

`\@spart` This macro does the actual formatting of the title of the part when the star form of the user command was used. In this case we *never* print a number. Otherwise the formatting is the same.

The differences between the definition of this macro in the artikel document classes and in the rapport and boek document classes are similar as they were for `\@part`.

```
844 <*artikel>
845 \def\@spart#1{%
846     {\parindent \z@
847     \head@style
848     \normalfont
849 <!type2>     \Large \PartFont \noindent #1\par}%
850 <type2>     \Large \PartFont \indent #1\par}%
851     \nobreak
852     \vskip 3ex
853     \@afterheading}
854 </artikel>
855 <*rapport | boek>
856 \def\@spart#1{%
857     {\centering
858     \interlinepenalty \@M
859     \normalfont
860     \Large \PartFont #1\par}%
861     \@endpart}
```

`\@endpart` This macro finishes the part page, for both `\@part` and `\@spart`. First we fill the current page.

```
862 \def\@endpart{\vfil\newpage
```

Then, when we are in twosided mode and chapters are supposed to be on right hand sides, we produce a completely blank page.

```
863 \!boek)          \if@twoside
864                 \if@openright
865                 \null
866                 \thispagestyle{empty}%
867                 \newpage
868                 \fi
869 \!boek)          \fi
```

When this was a two column document we have to switch back to two column mode.

```
870                 \if@tempswa
871                 \twocolumn
872                 \fi}
873 \!rapport | boek)
```

## 8.2.6 Chapters

`\chapter` A chapter should always start on a new page therefore we start by calling `\clearpage` and setting the pagestyle for this page to *plain*.

```
874 \!rapport | boek)
875 \newcommand*\chapter{\if@openright\cleardoublepage\else\clearpage\fi
876                 \thispagestyle{plain}%
```

Then we prevent floats from appearing at the top of this page because it looks weird to see a floating object above a chapter title.

```
877                 \global\@topnum\z@
```

Then we suppress the indentation of the first paragraph by setting the switch `\@afterindent` to *false*. We use `\secdef` to specify the macros to use for actually setting the chapter title.

```
878                 \@afterindentfalse
879                 \secdef\@chapter\@schapter}
```

`\@chapter` This macro is called when we have a numbered chapter. When *secnumdepth* is larger than  $-1$  and, in the *boek* class, `\@mainmatter` is true, we display the chapter number. We also inform the user that a new chapter is about to be typeset by writing a message to the terminal.

```
880 \def\@chapter[#1]#2{%
881     \ifnum \c@secnumdepth >\m@ne
882 \!boek)         \if@mainmatter
883                 \refstepcounter{chapter}%
884                 \typeout{\@chapapp\space\thechapter.}%
885                 \addcontentsline{toc}{chapter}%
886                 {\protect\numberline{\thechapter}\toc@font0 #1}%
887 \!boek)
888     \else
889                 \addcontentsline{toc}{chapter}{\toc@font0 #1}%
890                 \fi
891 \!boek)
892     \else
```

```

893     \addcontentsline{toc}{chapter}{\toc@font0 #1}%
894     \fi

```

After having written an entry to the table of contents we store the (alternative) title of this chapter with `\chaptermark` and add some white space to the lists of figures and tables.

```

895     \chaptermark{#1}%
896     \addtocontents{lof}{\protect\addvspace{10\p@}}%
897     \addtocontents{lot}{\protect\addvspace{10\p@}}%

```

Then we call upon `\@makechapterhead` to format the actual chapter title. We have to do this in a special way when we are in twocolumn mode in order to have the chapter title use the entire `\textwidth`. In one column mode we call `\@afterheading` which takes care of suppressing the indentation.

```

898     \if@twocolumn
899         \topnewpage[\@makechapterhead{#2}]%
900     \else
901         \@makechapterhead{#2}%
902         \@afterheading
903     \fi}

```

`\ChapFont` The font used to typeset the chapters is stored in this macro.

```

904 \newcommand*\ChapFont{\bfseries}

```

`\@makechapterhead` The macro above uses `\@makechapterhead<text>` to format the heading of the chapter.

We begin by leaving some white space. Then we open a group in which we have a paragraph indent of 0pt, and in which we have the text set ragged right. We also reset the font.

```

905 \def\@makechapterhead#1{%
906 (!boek) \vspace*{50\p@ \@plus 5\p@}%
907 (boek) \vspace*{50\p@ \@plus 20\p@}%
908 {\setlength\parindent{\z@}%
909 \setlength\parskip {\z@}%
910 \head@style \normalfont

```

Then we check whether the number of the chapter has to be printed. If so we leave some whitespace between the chapter number and its title.

```

911     \ifnum \c@secnumdepth >\m@ne
912 (boek)     \if@mainmatter
913             \Large\ChapFont \@chapapp{} \thechapter
914             \par\nobreak
915             \vskip 20\p@
916 (boek)     \fi
917     \fi

```

Now we set the title in a large bold font. We prevent a pagebreak at this point and leave some whitespace before the text begins.

```

918     \Large \ChapFont #1\par
919     \nobreak
920     \vskip 40\p@
921 }

```

`\@schapter` This macro is called when we have an unnumbered chapter. It is much simpler than `\@chapter` because it only needs to typeset the chapter title.

```

922 \def\@schapter#1{\if@twocolumn
923     \topnewpage[\@makeschapterhead{#1}]%
924     \else
925     \@makeschapterhead{#1}%
926     \@afterheading
927     \fi}

```

`\@makeschapterhead` The macro above uses `\@makeschapterhead(text)` to format the heading of the chapter. It is similar to `\@makechapterhead` except that it never has to print a chapter number.

```

928 \def\@makeschapterhead#1{%
929 \!boek) \vspace*{50\p@\@plus 5\p@}%
930 \boek) \vspace*{50\p@\@plus 20\p@}%
931 {\setlength\parindent{\z@}%
932  \setlength\parskip{\z@}%
933  \head@style
934  \normalfont
935  \Large \ChapFont #1\par
936  \nobreak
937  \vskip 40\p@
938  }}
939 \langle rapport | boek)

```

## 8.2.7 Lower level headings

These commands all make use of `\@startsection`.

`\section` This gives a normal heading with white space above the heading (the whitespace below the heading will be generated by the `\parskip` that is inserted at the start of the first paragraph), the title set in `\large\bfseries`, and no indentation on the first paragraph.

```

940 \newcommand*\section{%
941 \langle *type1 | type3)
942  \@startsection {section}{1}{\z@}%
943  {-2\baselineskip\@plus -1\baselineskip \@minus -.5\baselineskip}%
944 \langle /type1 | type3)
945 \langle *type2)
946  \@startsection {section}{1}{\unitindent}%
947  {2\baselineskip\@plus \baselineskip \@minus .5\baselineskip}%
948 \langle /type2)
949 \langle type1)  {.5\baselineskip}%
950 \langle type2 | type3)  {.01\baselineskip}%
951  {\normalfont\large\SectFont}}

```

`\SectFont` The font used to typeset the sections is stored in this macro.

```

952 \newcommand*\SectFont{\bfseries}

```

`\subsection` This gives a normal heading with white space above the heading, the title set in `\normalsize\bfseries`, and no indentation on the first paragraph.

```

953 \newcommand*\subsection{%
954 \langle *type1 | type3)

```

```

955 \startsection{subsection}{2}{\z@}%
956 {-1\baselineskip\@plus -.5\baselineskip \@minus -.25\baselineskip}%
957 </type1 | type3>
958 (*type2)
959 \startsection{subsection}{2}{\unitindent}%
960 {1\baselineskip\@plus .5\baselineskip \@minus .25\baselineskip}%
961 </type2>
962 (type1) {.25\baselineskip}%
963 (type2 | type3) {.01\baselineskip}%
964 {\normalfont\normalsize\SSectFont}}

```

`\SSectFont` The font used to typeset the subsections is stored in this maro.

```
965 \newcommand*\SSectFont{\bfseries}
```

`\subsubsection` This gives a normal heading with white space above the heading, the title set in `\normalsize\tm`, and no indentation on the first paragraph.

```

966 \newcommand*\subsubsection{%
967 (*type1 | type3)
968 \startsection{subsubsection}{3}{\z@}%
969 {-1\baselineskip plus -.5\baselineskip minus -.25\baselineskip}%
970 </type1 | type3>
971 (*type2)
972 \startsection{subsubsection}{3}{\unitindent}%
973 {1\baselineskip plus .5\baselineskip minus .25\baselineskip}%
974 </type2>
975 (type1) {.25\baselineskip}%
976 (type2 | type3) {.01\baselineskip}%
977 {\normalfont\normalsize\SSSectFont}}

```

`\SSSectFont` The font used to typeset the subsubsections is stored in this maro.

```

978 (artikel & (type1 | type3)) \newcommand*\SSSectFont{\rmfamily}
979 (type2) \newcommand*\SSSectFont{\scshape}
980 (rapport | boek) \newcommand*\SSSectFont{\slshape}

```

`\paragraph` This gives a run-in heading with white space above and to the right of the heading, the title set in `\normalsize\slshape`.

```

981 \newcommand*\paragraph{%
982 (!type2) \startsection{paragraph}{4}{\z@}%
983 (type2) \startsection{paragraph}{4}{\unitindent}%
984 {3.25ex \@plus1ex \@minus.2ex}%
985 {-1em}%
986 {\normalfont\normalsize\ParaFont}}

```

`\ParaFont` The font used to typeset the paragraphs is stored in this maro.

```

987 (!type2) \newcommand*\ParaFont{\slshape}
988 (type2) \newcommand*\ParaFont{\scshape}

```

`\subparagraph` This gives an indented run-in heading with white space above and to the right of the heading, the title set in `\normalsize\slshape`.

```

989 \newcommand*\subparagraph{%
990 (!type2) \startsection{subparagraph}{5}{\parindent}%
991 (type2) \startsection{subparagraph}{5}{\unitindent}%
992 {3.25ex \@plus1ex \@minus .2ex}%

```

```

993 {-1em}%
994 {\normalfont\normalsize\SParaFont}}

```

`\SParaFont` The font used to typeset the subparagraphs is stored in this macro.

```

995 \newcommand*\SParaFont{\slshape}

```

`\Headingfonts` To change the fonts that are used to typeset the title, part, chapter and section headings this macro can be used.

```

996 (*artikel)
997 \newcommand*\HeadingFonts[7]{%
998   \renewcommand*\TitleFont{#1}%
999   \renewcommand*\PartFont{#2}%
1000  \renewcommand*\SectFont{#3}%
1001  \renewcommand*\SSectFont{#4}%
1002  \renewcommand*\SSSectFont{#5}%
1003  \renewcommand*\ParaFont{#6}%
1004  \renewcommand*\SParaFont{#7}}
1005 (/artikel)
1006 (*rapport | boek)
1007 \newcommand*\HeadingFonts[8]{%
1008   \renewcommand*\TitleFont{#1}%
1009   \renewcommand*\PartFont{#2}%
1010   \renewcommand*\ChapFont{#3}%
1011   \renewcommand*\SectFont{#4}%
1012   \renewcommand*\SSectFont{#5}%
1013   \renewcommand*\SSSectFont{#6}%
1014   \renewcommand*\ParaFont{#7}%
1015   \renewcommand*\SParaFont{#8}}
1016 (/rapport | boek)

```

## 8.3 Lists

### 8.3.1 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', 'i', ... , '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.
\leftmarginiii 1017 (!type2)\setlength\leftmargini {\unitindent}
\leftmarginiiii 1018 (type2)\setlength\leftmargini {\othermargin}
\leftmarginiv 1019 \setlength\leftmarginiii {\othermargin}
\leftmarginv 1020 \setlength\leftmarginiiii{\othermargin}
\leftmarginvi 1021 \setlength\leftmarginiv {\othermargin}
1022 \setlength\leftmarginv {\othermargin}
1023 \setlength\leftmarginvi {1em}

```

Here we set the top level `\leftmargin`.

```

1024 \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.

```
1025 \setlength \labelsep {5\p@}
1026 \setlength \labelwidth{\leftmargini}
1027 \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`.

```
1028 \setlength\partopsep{\z@}
```

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

```
1029 \setlength\topsep{\z@}
```

`\@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.

```
1030 \@beginparpenalty -\@lowpenalty
1031 \@endparpenalty -\@lowpenalty
1032 \@itempenalty -\@lowpenalty
```

`\@listi` `\@listi` defines values of `\leftmargin`, `\parsep`, `\topsep`, and `\itemsep`, etc.

`\@listI` for the lists that appear on top-level. Its definition is modified by the font-size  
commands (eg within `\small` the list parameters get “smaller” values).

For this reason `listI` is defined to hold a saved copy of `listi` so that `\normalsize`  
can switch all parameters back.

```
1033 \def\@listi{%
1034 \!type2) \leftmargin\unitindent
1035 \!type2) \leftmargin\leftmargini
1036 \!type2) \labelsep.5em%
1037 \!type2) \labelsep.45em%
1038 \labelwidth\leftmargin
1039 \advance\labelwidth-\labelsep
1040 \parsep \z@
1041 \!type3) \topsep 0\p@ \@plus\p@
1042 \!type3) \topsep -.5\parskip \@plus\p@
1043 \itemsep 0\p@ \@plus1\p@}
1044 \let\@listI\@listi
```

We initialise these parameters although strictly speaking that is not necessary.

```
1045 \@listi
```

`\@listii` Here are the same macros for the higher level lists. Note that they don't have

`\@listiii` saved versions and are not modified by the font size commands. In other words

`\@listiv` this class assumes that nested lists only appear in `\normalsize`, i.e. the main

`\@listv` document size.

```
\@listvi 1046 \def\@listii {\leftmargin\leftmarginii
1047 \!type2) \labelsep .5em%
1048 \!type2) \labelsep .3em%
1049 \labelwidth\leftmarginii
1050 \advance\labelwidth-\labelsep
```

```

1051 (!type3)          \topsep    0\p@ \@plus\p@
1052 (type3)          \topsep    -.5\parskip\@plus\p@
1053                  \parsep    \z@
1054                  \itemsep   \z@ \@plus\p@}
1055 \def\@listiii{\leftmargin\leftmarginiii
1056 (!type2)          \labelsep   .5em%
1057 (type2)          \labelsep   .3em%
1058                  \labelwidth\leftmarginiii
1059                  \advance\labelwidth-\labelsep
1060 (!type3)          \topsep    0\p@ \@plus\p@
1061 (type3)          \topsep    -.5\parskip\@plus\p@
1062                  \parsep    \z@
1063                  \partopsep \z@ \@plus\p@
1064                  \itemsep   \z@ \@plus\p@}
1065 \def\@listiv {\leftmargin\leftmarginiv
1066 (!type2)          \labelsep   .5em%
1067 (type2)          \labelsep   .3em%
1068                  \labelwidth\leftmarginiv%
1069                  \advance\labelwidth-\labelsep
1070 (!type3)          \topsep    0\p@ \@plus\p@
1071 (type3)          \topsep    -.5\parskip\@plus\p@
1072                  \parsep    \z@
1073                  \itemsep   \z@ \@plus\p@}
1074 \def\@listv  {\leftmargin\leftmarginv
1075 (!type2)          \labelsep   .5em%
1076 (type2)          \labelsep   .3em%
1077                  \labelwidth\leftmarginv
1078                  \advance\labelwidth-\labelsep%
1079 (!type3)          \topsep    0\p@ \@plus\p@
1080 (type3)          \topsep    -.5\parskip\@plus\p@
1081                  \parsep    \z@
1082                  \itemsep   \z@ \@plus\p@}
1083 \def\@listvi {\leftmargin\leftmarginvi
1084 (!type2)          \labelsep   .5em
1085 (type2)          \labelsep   .3em
1086                  \labelwidth\leftmarginvi
1087                  \advance\labelwidth{-\labelsep}%
1088 (!type3)          \topsep    0\p@ \@plus\p@
1089 (type3)          \topsep    -.5\parskip\@plus\p@
1090                  \parsep    \z@
1091                  \itemsep   \z@ \@plus\p@}

```

### 8.3.2 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 latex.dtx, but their representation is changed
\theenumii here.
\theenumiii
\theenumiv 1092 \renewcommand*\theenumi{\@arabic\c@enumi}
1093 \renewcommand*\theenumii{\@alph\c@enumii}
1094 \renewcommand*\theenumiii{\@roman\c@enumiii}
1095 \renewcommand*\theenumiv{\@Alph\c@enumiv}

```

`\labelenumi` The label for each item is generated by the commands  
`\labelenumii` `\labelenumi` ... `\labelenumiv`.  
`\labelenumiii` 1096 `\newcommand*\labelenumi{\theenumi.}`  
`\labelenumiv` 1097 `\newcommand*\labelenumii{(\theenumii)}`  
1098 `\newcommand*\labelenumiii{\theenumiii.}`  
1099 `\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` 1100 `\renewcommand*\p@enumii{\theenumi}`  
1101 `\renewcommand*\p@enumiii{\theenumi(\theenumii)}`  
1102 `\renewcommand*\p@enumiv{\p@enumiii\theenumiii}`

`enumerate` We want to have different label positioning on different levels of list. To achieve this we have to redefine the `enumerate` environment.

```

1103 \renewenvironment{enumerate}{%
1104   \ifnum \@enumdepth >3
1105     \@toodeep
1106   \else
1107     \advance\@enumdepth \@ne
1108     \edef\@enumctr{enum\romannumeral\the\@enumdepth}%
1109     \list{\csname label\@enumctr\endcsname}
1110           {\usecounter{\@enumctr}%
1111 <type1>           \ifnum \@listdepth=1
1112 <*type1 | type3> \if@revlabel
1113                 \def\makelabel##1{\hskip .5\unitindent{##1\hfil}}%
1114                 \else
1115 <!type3>           \def\makelabel##1{\hfil##1}
1116 <type3>           \def\makelabel##1{##1\hfil}
1117 <type3>           \fi
1118                 \fi
1119 </type1 | type3> \else
1120 <type1>           \def\makelabel##1{##1\hfil}%
1121 <type1 | type2> \fi
1122 <type1>           \fi
1123           }%
1124   \fi}

```

We try to suppress spaces after these list constructs.

```
1125 {\global\@ignoretrue \endlist}
```

### 8.3.3 Itemize

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

```

1126 \newcommand*\labelitemi{\textbullet}
1127 \newcommand*\labelitemii{\normalfont\bfseries \textendash}
1128 \newcommand*\labelitemiii{\textasteriskcentered}
1129 \newcommand*\labelitemiv{\textperiodcentered}

```

`itemize` We want to have differen label positioning on different levels of list. To acheive this we have to redefine the `itemize` environment.

```

1130 \renewenvironment{itemize}{%
1131   \ifnum \@itemdepth >3
1132     \@toodeep
1133   \else
1134     \advance\@itemdepth \@one
1135     \edef\@itemitem{labelitem\romannumeral\the\@itemdepth}%
1136     \list{\csname\@itemitem\endcsname}%
1137     {%
1138 <type1>           \ifnum \@listdepth=1\relax
1139 <*type1 | type3>
1140                 \if@revlabel
1141                 \def\makelabel##1{\hskip .5\unitindent{##1\hfil}}\else
1142 <type1>           \def\makelabel##1{\hfil##1}
1143 <type3>           \def\makelabel##1{##1\hfil}
1144                 \fi
1145 </type1 | type3>
1146 <type1>           \else
1147 <type1 | type2>   \def\makelabel##1{##1\hfil}
1148 <type1>           \fi
1149                 }%
1150   \fi}

```

We try to suppress spaces after these list constructs.

```

1151 {\global\@ignoretrue \endlist}

```

### 8.3.4 Description

`description` The `description` environment is defined here – while the `itemize` and `enumerate` environments are defined in `latex.dtx`.

```

1152 \newenvironment{description}
1153         {\list{}{\labelwidth\z@ \itemindent-\leftmargin
1154                 \let\makelabel\descriptionlabel}}
1155         {\endlist}

```

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

```

1156 \newcommand*\descriptionlabel[1]{\hspace\labelsep \bfseries #1}

```

## 8.4 Adapting existing environments

Because we globally set `\topsep` to zero, we need to modify the definitions of a number of environments slightly to get a litle whitespace around them in the document classes `artikel1` and `rapport1`.

`center` Add a litle surrounding whitespace.

```

1157 <*type1>
1158 \def\center
1159   {\topsep=.25\baselineskip \@plus .1\baselineskip
1160     \@minus .1\baselineskip
1161     \trivlist \centering\item[]}
1162 \let\endcenter\endtrivlist

```

**flushleft** Add a little surrounding whitespace.

```
1163 \def\flushleft
1164   {\topsep=.25\baselineskip \@plus .1\baselineskip
1165    \@minus .1\baselineskip
1166    \trivlist \raggedright\item[]}}
1167 \let\endflushleft=\endtrivlist
```

**flushright** Add a little surrounding whitespace.

```
1168 \def\flushright
1169   {\topsep=.25\baselineskip \@plus .1\baselineskip
1170    \@minus .1\baselineskip
1171    \trivlist \raggedleft\item[]}}
1172 \let\endflushright=\endtrivlist
1173 </type1>
```

**verbatim** In verbatim we add a little surrounding whitespace, –which for `artikel3` and `rapport3` is negative to compensate for the positive `\parskip`– but also an indent for the `artikel1` and `rapport1` document classes.

```
1174 \def\verbatim{%
1175 <*type1 | type2>
1176   \topsep=.25\baselineskip \@plus .1\baselineskip
1177   \@minus .1\baselineskip
1178   \@verbatim
1179 </type1 | type2>
1180 <type1>   \leftskip\unitindent
1181 <type2>   \leftskip\z@
1182 <*type3>
1183   \topsep=-.5\parskip
1184   \@verbatim
1185 </type3>
1186   \frenchspacing\@vobeyspaces \@xverbatim}
1187 <type1>\def\endverbatim{\if@newlist \leavevmode\fi\endtrivlist}
```

## 8.5 Defining new environments

### 8.5.1 Abstract

**abstract** When we are producing a separate titlepage we also put the abstract on a page of its own. It will be centred vertically on the page.

Note that this environment is not defined for books.

```
1188 <!boek>\if@titlepage
1189   \newenvironment{abstract}{%
1190     \titlepage
1191     \null\vfil
1192     \hbox{\SectFont \abstractname}
1193     \noindent\ignorespaces}
1194     {\par\vfil\null\endtitlepage}
```

When we are not making a separate titlepage –the default for the artikel document classes– we have to check if we are in twocolumn mode. In that case the abstract is set as a `\section*`, otherwise the abstract is typeset flushleft, an amount `\unitindent` smaller as the normal text.

```
1195 <*artikel | rapport>
```

```

1196 \else
1197   \newenvironment{abstract}{%
1198     \if@twocolumn
1199       \section*{\abstractname}%
1200     \else
1201       \small
1202 <*type1 | type3>
1203       \bgroup\rightskip=\unitindent
1204       \hbox{\SectFont \abstractname}%
1205       \noindent\ignorespaces
1206 </type1 | type3>

```

As always, the `artikel2` document class has a different implementation.

```

1207 <*type2>
1208   \hbox{\hskip\unitindent\SectFont \abstractname}%
1209   \list{}{\setlength\listparindent{\unitindent}%
1210     \setlength\parindent   {\z@}%
1211     \setlength\leftmargin  {\unitindent}%
1212     \setlength\rightmargin {\unitindent}%
1213     \setlength\parsep     {\z@}}%
1214   \item[]%
1215 </type2>
1216   \fi}

```

Which implies that the definition of `\end{abstract}` is also different.

```

1217 <!type2>   {\if@twocolumn\else\par\egroup\fi}
1218 <type2>    {\if@twocolumn\else\par\endlist\fi}
1219 \fi
1220 </artikel | rapport>

```

### 8.5.2 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`.

```

1221 \newenvironment{verse}
1222   {\let\\ \@centercr
1223   \list{}{\itemsep\z@
1224     \itemindent-1.5em%
1225     \listparindent\itemindent
1226     \rightmargin\leftmargin
1227     \advance\leftmargin1.5em}%
1228   \item\relax}
1229   {\endlist}

```

### 8.5.3 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.

```

1230 \newenvironment{quotation}
1231   {\list{}{%
1232 <!type2>   \listparindent\z@
1233 <type2>    \listparindent\unitindent

```

```

1234 <boek> \listparindent1.5em%
1235 \itemindent\listparindent
1236 \rightmargin\leftmargin
1237 \parsep\z@ \@plus\p@}%
1238 \item\relax}
1239 {\endlist}

```

#### 8.5.4 Quote

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

```

1240 \newenvironment{quote}
1241 {\list{}{\rightmargin\leftmargin}%
1242 \item\relax}
1243 {\endlist}

```

#### 8.5.5 Theorem

`\@begintheorem` These document classes have a slightly modified theorem environment style. Surrounding whitespace is added and an initialisation of `\labelsep`. Finally a slanted font instead of an italic font is used.

```

1244 \def\@begintheorem#1#2{%
1245 \vskip\baselineskip \labelsep=.5em%
1246 \trivlist
1247 \item[\hskip \labelsep{\bfseries #1\ #2}]\slshape}
1248 \def\@opargbegintheorem#1#2#3{%
1249 \vskip\baselineskip \labelsep=.5em%
1250 \trivlist
1251 \item[\hskip \labelsep{\bfseries #1\ #2\ (#3)}]\slshape}
1252 \def\@endtheorem{\endtrivlist \vskip\baselineskip}

```

#### 8.5.6 Titlepage

`titlepage` In the normal environments, the titlepage environment does nothing but start and end a page, and inhibit page numbers. It also resets the page number to zero. This is incorrect since it results in using the page parameters for a right-hand page but it is the way it was. In two-column style, it still makes a one-column page.

```

1253 \newenvironment{titlepage}
1254 {
1255 <boek> \cleardoublepage
1256 \if@twocolumn
1257 \@restonecoltrue\onecolumn
1258 \else
1259 \@restonecolfalse\newpage
1260 \fi
1261 \thispagestyle{empty}%
1262 \if@compatibility
1263 \setcounter{page}\z@
1264 <*artikel | rapport>
1265 \else
1266 \setcounter{page}\@ne
1267 </artikel | rapport>

```

```

1268     \fi}
1269     {\if@restonecol\twocolumn \else \newpage \fi
1270 (artikel | rapport)     \setcounter{page}\@ne
1271     }

```

### 8.5.7 Appendix

`\appendix` The `\appendix` command is not really an environment, it is a macro that makes some changes in the way things are done.

In the artikel document classes the `\appendix` command must do the following:

- reset the section and subsection counters to zero,
- redefine `\thesection` to produce alphabetic appendix numbers.

```

1272 (*artikel)
1273 \newcommand*\appendix{\par
1274   \setcounter{section}{0}%
1275   \setcounter{subsection}{0}%
1276   \gdef\thesection{\@Alph@c@section}}
1277 (/artikel)

```

In the rapport and boek document classes the `\appendix` command must do the following:

- reset the chapter and section counters to zero,
- set `\@chapapp` to `\appendixname` (for messages),
- redefine the chapter counter to produce appendix numbers,
- possibly redefine the `\chapter` command if appendix titles and headings are to look different from chapter titles and headings.

```

1278 (*rapport | boek)
1279 \newcommand*\appendix{\par
1280   \setcounter{chapter}{0}%
1281   \setcounter{section}{0}%
1282   \gdef\@chapapp{\appendixname}%
1283   \gdef\thechapter{\@Alph@c@chapter}}
1284 (/rapport | boek)

```

## 8.6 Setting parameters for existing environments

### 8.6.1 Array and tabular

`\arraycolsep` The columns in an array environment are separated by `2\arraycolsep`.

```
1285 \setlength\arraycolsep{5\p@}
```

`\tabcolsep` The columns in an tabular environment are separated by `2\tabcolsep`.

```
1286 \setlength\tabcolsep{6\p@}
```

`\arrayrulewidth` The width of rules in the array and tabular environments is given by `\arrayrulewidth`.

```
1287 \setlength\arrayrulewidth{.4\p@}
```

`\doublerulesep` The space between adjacent rules in the array and tabular environments is given by `\doublerulesep`.

```
1288 \setlength\doublerulesep{2\p@}
```

### 8.6.2 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.)  
1289 `\setlength\tabbingsep{\labelsep}`

### 8.6.3 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.

```
1290 <type1>\def\@minipagerestore{\parindent\unitindent}
1291 <*type3>
1292 \def\@minipagerestore{%
1293     \parskip=.5\baselineskip \@plus .1\baselineskip
1294     \@minus .1\baselineskip}
1295 </type3>
```

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

```
1296 \skip\@mpfootins = \skip\footins
```

### 8.6.4 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`.

```
1297 \setlength\fboxsep{3\p@}
1298 \setlength\fboxrule{.4\p@}
```

### 8.6.5 Equation and eqnarray

`\theequation` When within chapters, the equation counter will be reset at beginning of a new chapter and the equation number will be prefixed by the chapter number.

This code must follow the `\chapter` definition, or more exactly the definition of the chapter counter.

```
1299 <artikel>\renewcommand*\theequation{\@arabic\c@equation}
1300 <*rapport | boek>
1301 \@addtoreset{equation}{chapter}
1302 \renewcommand*\theequation{%
1303     \ifnum \c@chapter>\z@ \thechapter.\fi\@arabic\c@equation}
1304 </rapport | boek>
```

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

```
1305 % \setlength\jot{3pt}
```

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

```
1306 % \def\@eqnnum{(\theequation)}
```

## 8.7 Floating objects

The file `latex.dtx` only defines a number of tools with which floating objects can be defined. This is done in the document class. It needs to define the following macros for each floating object of type `TYPE` (e.g., `TYPE = figure`).

`\fps@TYPE` The default placement specifier for floats of type `TYPE`.

`\ftype@TYPE` The type number for floats of type `TYPE`. Each `TYPE` has associated a unique positive `TYPE` number, which is a power of two. E.g., figures might have type number 1, tables type number 2, programs type number 4, etc.

`\ext@TYPE` The file extension indicating the file on which the contents list for float type `TYPE` is stored. For example, `\ext@figure = 'lof'`.

`\fnum@TYPE` A macro to generate the figure number for a caption. For example, `\fnum@TYPE == 'Figure \thefigure'`.

`\@makecaption<num><text>` A macro to make a caption, with `<num>` the value produced by `\fnum@...` and `<text>` the text of the caption. It can assume it's in a `\parbox` of the appropriate width. This will be used for *all* floating objects.

The actual environment that implements a floating object such as a figure is defined using the macros `\@float` and `\end@float`, which are defined in `latex.dtx`.

An environment that implements a single column floating object is started with `\@float{TYPE}[\langle placement \rangle]` of type `TYPE` with `<placement>` as the placement specifier. The default value of `<PLACEMENT>` is defined by `\fps@TYPE`.

The environment is ended by `\end@float`. E.g., `\figure == \@floatfigure`, `\endfigure == \end@float`.

### 8.7.1 Figure

Here is the implementation of the figure environment.

`\c@figure` First we have to allocate a counter to number the figures. In the `rapport` and `boek` document classes the figures are numbered per chapter.

```
1307 \*artikel)
1308 \newcounter{figure}
1309 \renewcommand*\thefigure{\@arabic\c@figure}
1310 \*artikel)
1311 \*rapport | boek)

1312 \newcounter{figure}[chapter]
1313 \renewcommand*\thefigure{%
1314   \ifnum\c@chapter>\z@\thechapter.\fi\@arabic\c@figure}
1315 \*rapport | boek)
```

`\fps@figure` Here are the parameters for the floating objects of type 'figure'.

```
\ftype@figure 1316 \def\fps@figure{tbp}
\ext@figure 1317 \def\ftype@figure{1}
\num@figure 1318 \def\ext@figure{lof}
1319 \def\fnum@figure{\figurename\nobreakspace\thefigure}
```

`figure` And the definition of the actual environment. The form with the `*` is used for  
`figure*` double column figures.

```
1320 \newenvironment{figure}
1321         {\@float{figure}}
1322         {\end@float}
1323 \newenvironment{figure*}
1324         {\@dblfloat{figure}}
1325         {\end@dblfloat}
```

### 8.7.2 Table

Here is the implementation of the table environment. It is very much the same as the figure environment.

`\c@table` First we have to allocate a counter to number the tables. In the rapport and boek document classes the tables are numbered per chapter.

```
1326 (*artikel)
1327 \newcounter{table}
1328 \renewcommand*\thetable{\@arabic\c@table}
1329 </artikel>
1330 (*rapport | boek)

1331 \newcounter{table}[chapter]
1332 \renewcommand*\thetable{%
1333   \ifnum\c@chapter>\z@\thechapter.\fi\@arabic\c@table}
1334 </rapport | boek>
```

`\fps@table` Here are the parameters for the floating objects of type ‘table’.

```
\ftype@table 1335 \def\fps@table{tbp}
\ext@table 1336 \def\ftype@table{2}
\num@table 1337 \def\ext@table{lot}
1338 \def\fnm@table{\tablename\nobreakspace\thetable}
```

`table` And the definition of the actual environment. The form with the `*` is used for  
`table*` double column tables.

```
1339 \newenvironment{table}
1340         {\@float{table}}
1341         {\end@float}
1342 \newenvironment{table*}
1343         {\@dblfloat{table}}
1344         {\end@dblfloat}
```

### 8.7.3 Captions

`\@makecaption` The `\caption` command calls `\@makecaption` to format the caption of floating objects. It gets two arguments, `<number>`, the number of the floating object and `<text>`, the text of the caption. Usually `<number>` contains a string such as ‘Figure 3.2’. The macro can assume it is called inside a `\parbox` of right width, with `\normalsize`.

`\abovecaptionskip` These lengths contain the amount of white space to leave above and below the  
`\belowcaptionskip` caption.

```
1345 \newlength\abovecaptionskip
```

```

1346 \newlength\belowcaptionskip
1347 \setlength\abovecaptionskip{10\p@}
1348 \setlength\belowcaptionskip{0\p@}

```

The definition of this macro is `\long` in order to allow more than one paragraph in a caption.

```

1349 \long\def\@makecaption#1#2{%
1350   \vskip\abovecaptionskip

```

We want to see if the caption fits on one line on the page, therefore we first typeset it in a temporary box.

```

1351   \sbox\@tempboxa{\CaptionLabelFont#1:} \CaptionTextFont#2}%

```

We can then measure its width. If that is larger than the current `\hsize` we typeset the caption as an ordinary paragraph.

```

1352   \ifdim \wd\@tempboxa >\hsize
1353     {\CaptionLabelFont#1:} \CaptionTextFont#2\par

```

If the caption fits, we center it. Because this uses an `\hbox` directly in vertical mode, it does not execute the `\everypar` tokens; the only thing that could be needed here is resetting the ‘minipage flag’ so we do this explicitly.

```

1354   \else
1355     \global \@minipagefalse
1356     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
1357   \fi
1358   \vskip\belowcaptionskip}

```

`\CaptionLabelFont` These macros can contain the fonts used for typesetting captions. By default they do nothing.

```

1359 \newcommand*\CaptionLabelFont{\relax}
1360 \newcommand*\CaptionTextFont{\relax}

```

`\CaptionFonts` To change the fonts that are used to typeset captions this macro can be used.

```

1361 \newcommand*\CaptionFonts[2]{%
1362   \renewcommand*\CaptionLabelFont{#1}%
1363   \renewcommand*\CaptionTextFont{#2}%
1364 }

```

## 8.8 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 defined using `\DeclareOldFontCommand`, 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. When in compatibility mode we select the `\tt` ‘default’ font first, to get  $\text{\LaTeX}$  2.09 behaviour.

```

\sf 1365 \DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
1366 \DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
1367 \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.

```
1368 \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  
`\it` in math mode. One should use `\upshape` to explicitly change back to the upright shape.

```
1369 \DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
1370 \DeclareOldFontCommand{\sl}{\normalfont\slshape}{\relax}
1371 \DeclareOldFontCommand{\sc}{\normalfont\scshape}{\relax}
```

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

```
1372 \DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}
1373 \DeclareRobustCommand*\mit{\@fontswitch\relax\mathnormal}
```

`\em` The definition of `\em` is changed here to have slanted instead of italic fonts.

```
1374 \DeclareRobustCommand*\em{%
1375   \@nomath\em
1376   \ifdim\fontdimen\@ne\font>\z@
1377     \upshape
1378   \else
1379     \slshape
1380   \fi}
```

## 9 Cross Referencing

### 9.1 Table of Contents, etc.

A `\section` command writes a `\contentsline{section}{<title>}{<page>}` command on the `.toc` file, where `<title>` contains the contents of the entry and `<page>` is the page number. If sections are being numbered, then `<title>` will be of the form `\numberline{<num>}{<heading>}` where `<num>` is the number produced by `\thesection`. Other sectioning commands work similarly.

A `\caption` command in a ‘figure’ environment writes

```
\contentsline{figure}{\numberline{<num>}{<caption>}}{<page>}
```

on the `.lof` file, where `<num>` is the number produced by `\thefigure` and `<caption>` is the figure caption. It works similarly for a ‘table’ environment.

The command `\contentsline{<name>}` expands to `\l@<name>`. So, to specify the table of contents, we must define `\l@chapter`, `\l@section`, `\l@subsection`, ... ; to specify the list of figures, we must define `\l@figure`; and so on. Most of these can be defined with either the `\@dottedtocline` or the `\@regtocline` command, which work as follows.

```
\@dottedtocline{<level>}{<indent>}{<numwidth>}{<title>}{<page>}
```

```
\@regtocline{<level>}{<title>}{<page>}
```

`<level>` An entry is produced only if `<level>`  $\leq$  value of the `tocdepth` counter.

Note, `\chapter` is level 0, `\section` is level 1, etc.

`<indent>` The indentation from the outer left margin of the start of the contents line.

`<numwidth>` The width of a box in which the section number is to go, if `<title>` includes a `\numberline` command.

`\@pnumwidth` This command uses the following three parameters, which are set with a `\newcommand` (so em's can be used to make them depend upon the font).

`\@tocrmarg`

`\@dotsep`

`\@pnumwidth` The width of a box in which the page number is put.

`\@tocrmarg` The right margin for multiple line entries. One wants `\@tocrmarg`  $\geq$  `\@pnumwidth`

`\@dotsep` Separation between dots, in mu units. Should be defined as a number like 2 or 1.7

```
1381 \newcommand*\@pnumwidth{1.55em}
```

```
1382 \newcommand*\@tocrmarg {2.55em}
```

```
1383 \newcommand*\@dotsep{4.5}
```

```
1384 <artikel> \setcounter{tocdepth}{3}
```

```
1385 !<artikel> \setcounter{tocdepth}{2}
```

### 9.1.1 Table of Contents

`\tableofcontents` This macro is used to request that L<sup>A</sup>T<sub>E</sub>X produces a table of contents. In the `rapport` and `boek` document classes the tables of contents, figures etc. are always set in single-column style.

```
1386 \newcommand*\tableofcontents{%
```

```
1387 <*rapport | boek>
```

```
1388     \if@twocolumn
```

```
1389         \@restonecoltrue\onecolumn
```

```
1390     \else
```

```
1391         \@restonecolfalse
```

```
1392     \fi
```

The title is set using the `\chapter*` command, making sure that the running head –if one is required– contains the right information.

```
1393     \chapter*{\contentsname}%
```

```
1394 </rapport | boek>
```

```
1395 <artikel>     \section*{\contentsname}%
```

```
1396     \mkboth{\MakeUppercase{\contentsname}}%
```

```
1397             {\MakeUppercase{\contentsname}}%
```

The the actual table of contents is made by calling `\@starttoc{toc}`. After that we restore `twocolumn` mode if necessary.

```
1398     \@starttoc{toc}%
```

```
1399 <!artikel>     \if@restonecol\twocolumn\fi
```

```
1400 }
```

`\@starttoc` The internal L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> macro `\@starttoc` needs to be adapted for the `artikel3` and `rapport3` document classes, in order to deal with the fact that for these document classes the `\parskip` is normally non-zero. We don't want that in the table of contents.

```

1401 (*type3)
1402 \def\starttoc#1{\begingroup
1403   \makeatletter
1404   \parskip\z@
1405   \@input{\jobname.#1}%
1406   \if@filesw
1407     \expandafter\newwrite\csname tf@#1\endcsname
1408     \immediate\openout \csname tf@#1\endcsname \jobname.#1\relax
1409     \fi \global\@nobreakfalse \endgroup}
1410 \end{type3}

```

`\@regtocline` These document classes use a different format for the table of contents than the standard classes from which they were developed. In order to achieve this different format we defined the macro `\@regtocline`.

```

1411 \newcommand*\@regtocline[3]{%
1412   \ifnum #1>\c@tocdepth
1413     \else
1414       \vskip\z@\@plus.2\p@
1415       {\hangindent\z@ \@afterindenttrue \interlinepenalty\@M
1416        \leftskip\unitindent
1417        \rightskip\unitindent\@plus 1fil
1418        \parfillskip\z@
1419        \@tempdima\unitindent
1420 (type2) \advance\@tempdima by \othermargin
1421        \parindent\z@
1422        \leavevmode
1423        \hbox{\hskip -\leftskip\relax#2\nobreak
1424         \hskip 1em \nobreak{\slshape #3}\par
1425         }%
1426   \fi}

```

`\numberline` This internal macro is redefined for the `artikel2` document class.

```

1427 (type2)\def\numberline#1{\hb@xt@\@tempdima{\hfil#1\hskip.3em}}

```

`\toc@font` The changed definition of `\@sect` that we use, selects a different font for the table of contents for the various header levels. It does this using `\toc@font`.

```

1428 \if@oldtoc
1429   \newcommand*\toc@font[1]{\relax}
1430 \else
1431   \newcommand*\toc@font[1]{%
1432 (*artikel)
1433   \ifcase#1\relax
1434 (type2) \Large\bfseries
1435   \or\bfseries
1436   \or\slshape
1437   \or\rmfamily
1438 \end{artikel}
1439 (*rapport | boek)
1440   \ifcase#1\relax
1441   \bfseries
1442   \or\slshape
1443   \or\rmfamily
1444 \end{rapport | boek}
1445   \fi}

```

1446 \fi

`\toc@case` In the `rapport` and `boek` document classes, the entries for parts are typeset in capital letters in the new style of the table of contents. In the old style this isn't done. The macro `\toc@case` is used to switch this.

```
1447 \if@oldtoc
1448   \newcommand*\toc@case{\relax}
1449 \else
1450   \newcommand*\toc@case{\MakeUppercase}
1451 \fi
```

`\l@part` Each sectioning command needs an additional macro to format its entry in the table of contents, as described above. The macro for the entry for parts is defined in a special way.

First we make sure that if a pagebreak should occur, it occurs *before* this entry. Also a little whitespace is added and a group begun to keep changes local.

First we have the definition from the standard classes.

```
1452 \if@oldtoc
1453 \newcommand*\l@part[2]{%
1454   \ifnum \c@tocdepth >-2\relax
1455   <artikel>   \addpenalty\@secpenalty
1456   <!artikel>  \addpenalty{-\@highpenalty}%
1457   \advspace{2.25em \@plus\p@}%
1458   \begingroup
```

The macro `\numberline` requires that the width of the box that holds the part number is stored in L<sup>A</sup>T<sub>E</sub>X's scratch register `\@tempdima`. Therefore we put it there.

```
1459   \setlength\@tempdima{3em}%
```

Then we set `\parindent` to 0pt and use `\rightskip` to leave enough room for the pagenumbers. To prevent overflow box messages the `\parfillskip` is set to a negative value.

```
1460   \parindent \z@ \rightskip \@pnumwidth
1461   \parfillskip -\@pnumwidth
```

Now we can set the entry, in a large bold font. We make sure to leave vertical mode, set the part title and add the pagenumber, set flush right.

```
1462   {\leavevmode
1463     \large \bfseries #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
```

Prevent a pagebreak immediately after this entry, but use `\everypar` to reset the `\if@nobreak` switch. Finally we close the group.

```
1464     \nobreak
1465 <artikel>   \if@compatibility
1466     \global\@nobreaktrue
1467     \everypar{\global\@nobreakfalse\everypar{}}%
1468 <artikel>   \fi
1469   \endgroup
1470 \fi}
```

Then we can introduce our new definition.

```
1471 \else
1472   \newcommand*\l@part{%
1473     \ifnum \c@tocdepth >-2\relax
```

```

1474 <artikel> \addpenalty\@secpenalty
1475 <!artikel> \addpenalty{-\@highpenalty}%
1476 \addvspace{2.25em \@plus \p}%
1477 \@regtocline{0}%
1478 \fi}
1479 \fi

```

`\l@chapter` This macro formats the entries in the table of contents for chapters. It is very similar to `\l@part`

First we make sure that if a pagebreak should occur, it occurs *before* this entry.

Also a little whitespace is added and a group begun to keep changes local.

Again we first present the ‘standard’ definition

```

1480 <*rapport | boek>
1481 \if@oldtoc
1482 \newcommand*\l@chapter[2]{%
1483 \addpenalty{-\@highpenalty}%
1484 \vskip 1.0em \@plus \p@

```

The macro `\numberline` requires that the width of the box that holds the part number is stored in L<sup>A</sup>T<sub>E</sub>X’s scratch register `\@tempdima`. Therefore we put it there. We begin a group, and change some of the paragraph parameters.

```

1485 \setlength\@tempdima{1.5em}%
1486 \begingroup
1487 \parindent \z@ \rightskip \@pnumwidth
1488 \parfillskip -\@pnumwidth

```

Then we leave vertical mode and switch to a bold font.

```

1489 \leavevmode \bfseries

```

Because we do not use `\numberline` here, we have to do some fine tuning ‘by hand’, before we can set the entry. We discourage but not disallow a pagebreak immediately after a chapter entry.

```

1490 \advance\leftskip\@tempdima
1491 \hskip -\leftskip
1492 #1\nobreak\hfil \nobreak\hb@xt@\@pnumwidth{\hss #2}\par
1493 \penalty\@highpenalty
1494 \endgroup}

```

Then we present our new definition.

```

1495 \else
1496 \newcommand*\l@chapter{\@regtocline{0}}
1497 \fi
1498 </rapport | boek>

```

`\l@section` In the `artikel` document classes the entry in the table of contents for sections looks much like the chapter entries for the `rapport` and `boek` document classes.

First we make sure that if a pagebreak should occur, it occurs *before* this entry.

Also a little whitespace is added and a group begun to keep changes local.

```

1499 <*artikel>
1500 \if@oldtoc
1501 \newcommand*\l@section[2]{%
1502 \addpenalty\@secpenalty
1503 \addvspace{1.0em \@plus \p}%

```

The macro `\numberline` requires that the width of the box that holds the part number is stored in L<sup>A</sup>T<sub>E</sub>X's scratch register `\@tempdima`. Therefore we put it there. We begin a group, and change some of the paragraph parameters.

```
1504 \setlength\@tempdima{1.5em}%
1505 \begingroup
1506 \parindent \z@ \rightskip \@pnumwidth
1507 \parfillskip -\@pnumwidth
```

Then we leave vertical mode and switch to a bold font.

```
1508 \leavevmode \bfseries
```

Because we do not use `\numberline` here, we have to do some fine tuning ‘by hand’, before we can set the entry. We discourage but not disallow a pagebreak immediately after a chapter entry.

```
1509 \advance\leftskip\@tempdima
1510 \hskip -\leftskip
1511 #1\nobreak\hfil \nobreak\hbext@\@pnumwidth{\hss #2}\par
1512 \endgroup}
```

The new definition:

```
1513 \else
1514 \newcommand*\l@section{\@regtocline{1}}
1515 \fi
1516 \</artikel>
```

In the `rapport` and `boek` document classes the definition for `\l@section` is much simpler.

```
1517 \<{*rapport | boek}>
1518 \if@oldtoc
1519 \newcommand*\l@section {\@dottedtocline{1}{1.5em}{2.3em}}
1520 \else
1521 \newcommand*\l@section {\@regtocline{1}}
1522 \fi
1523 \</rapport | boek>
```

`\l@subsection` All lower level entries are defined using the macro `\@dottedtocline` or `\@regtocline`  
`\l@subsubsection` (see above).

`\l@paragraph` 1524 `\if@oldtoc`

`\l@subparagraph` 1525 `\<{*artikel}>`

```
1526 \newcommand*\l@subsection {\@dottedtocline{2}{1.5em}{2.3em}}
1527 \newcommand*\l@subsubsection{\@dottedtocline{3}{3.8em}{3.2em}}
1528 \newcommand*\l@paragraph {\@dottedtocline{4}{7.0em}{4.1em}}
1529 \newcommand*\l@subparagraph {\@dottedtocline{5}{10em}{5em}}
```

1530 `\</artikel>`

1531 `\<{*rapport | boek}>`

```
1532 \newcommand*\l@subsection {\@dottedtocline{2}{3.8em}{3.2em}}
1533 \newcommand*\l@subsubsection{\@dottedtocline{3}{7.0em}{4.1em}}
1534 \newcommand*\l@paragraph {\@dottedtocline{4}{10em}{5em}}
1535 \newcommand*\l@subparagraph {\@dottedtocline{5}{12em}{6em}}
```

1536 `\</rapport | boek>`

1537 `\else`

```
1538 \newcommand*\l@subsection {\@regtocline{2}}
```

```
1539 \newcommand*\l@subsubsection{\@regtocline{3}}
```

```
1540 \newcommand*\l@paragraph {\@regtocline{4}}
```

```
1541 \newcommand*\l@subparagraph {\@regtocline{5}}
```

1542 `\fi`

### 9.1.2 List of figures

`\listoffigures` This macro is used to request that L<sup>A</sup>T<sub>E</sub>X produces a list of figures. It is very similar to `\tableofcontents`.

```
1543 \newcommand*\listoffigures{%
1544 (*rapport | boek)
1545   \if@twocolumn
1546     \@restonecoltrue\onecolumn
1547   \else
1548     \@restonecolfalse
1549   \fi
1550   \chapter*{\listfigurename}%
1551 \!/rapport | boek)
1552 (artikel)   \section*{\listfigurename}%
1553   \@mkboth{\MakeUppercase{\listfigurename}}%
1554             {\MakeUppercase{\listfigurename}}%
1555   \@starttoc{lof}%
1556 (rapport | boek)   \if@restonecol\twocolumn\fi
1557 }
```

`\l@figure` This macro produces an entry in the list of figures.

```
1558 \if@oldtoc
1559   \newcommand*\l@figure{\@dottedtocline{1}{1.5em}{2.3em}}
1560 \else
1561   \newcommand*\l@figure{\@regtocline{1}}
1562 \fi
```

### 9.1.3 List of tables

`\listoftables` This macro is used to request that L<sup>A</sup>T<sub>E</sub>X produces a list of tables. It is very similar to `\tableofcontents`.

```
1563 \newcommand*\listoftables{%
1564 (*rapport | boek)
1565   \if@twocolumn
1566     \@restonecoltrue\onecolumn
1567   \else
1568     \@restonecolfalse
1569   \fi
1570   \chapter*{\listtablename}%
1571 \!/rapport | boek)
1572 (artikel)   \section*{\listtablename}%
1573   \@mkboth{\MakeUppercase{\listtablename}}%
1574             {\MakeUppercase{\listtablename}}%
1575   \@starttoc{lot}%
1576 (rapport | boek)   \if@restonecol\twocolumn\fi
1577 }
```

`\l@table` This macro produces an entry in the list of tables.

```
1578 \let\l@table\l@figure
```

## 9.2 Bibliography

`\bibindent` The “open” bibliography format uses an indentation of `\bibindent`.

```

1579 \newdimen\bibindent
1580 \setlength\bibindent{1.5em}

\newblock This is a dummy definition for this macro which is used in the thebibliography
environment.
1581 \newcommand*\newblock{}

thebibliography The ‘thebibliography’ environment executes the following commands:
    \renewcommand\newblock{\hskip .11em \@plus .33em \@minus .07em} –
Defines the “closed” format, where the blocks (major units of information) of an
entry run together.
    \sloppy – Used because it’s rather hard to do line breaks in bibliographies,
    \sfcode‘\.=1000\relax – Causes a ‘.’ (period) not to produce an end-of-
sentence space.
    The implementation of this environment is based on the generic list environ-
ment. It uses the enumiv counter internally to generate the labels of the list.
    When an empty ‘thebibliography’ environment is found, a warning is issued.

1582 \newenvironment{thebibliography}[1]
1583 (*artikel)
1584     {\section*\refname}%
1585     \@mkboth{\MakeUppercase\refname}{\MakeUppercase\refname}%
1586 \!/artikel)
1587 (*!artikel)
1588     {\chapter*\bibname}%
1589     \@mkboth{\MakeUppercase\bibname}{\MakeUppercase\bibname}%
1590 \!/artikel)
1591     \list{\@biblabel{\@arabic\c@enumiv}}%
1592         {\settowidth\labelwidth{\@biblabel{#1}}%
1593         \leftmargin\labelwidth
1594         \advance\leftmargin\labelsep
1595         \@openbib@code
1596         \usecounter{enumiv}%
1597         \let\p@enumiv\@empty
1598         \renewcommand*\theenumiv{\@arabic\c@enumiv}}%
1599     \sloppy\clubpenalty4000\widowpenalty4000%
1600     \sfcode‘\.\@m}
1601     {\def\@noitemerr
1602         {\@latex@warning{Empty ‘thebibliography’ environment}}%
1603     \endlist}

\newblock The default definition for \newblock is to produce a small space.
1604 % \changes{v2.0t}{1996/04/01}{use \cs{renewcommand} instead of
1605 %     \cs{newcommand}}
1606 \renewcommand\newblock{\hskip.11em\@plus.33em\@minus.07em}

\@openbib@code The default definition for \@openbib@code is to do nothing. It will be changed by
the openbib option.
1607 \let\@openbib@code\@empty

\@biblabel The label for a \bibitem[...] command is produced by this macro. The default
from latex.dtx is used.
1608 % \renewcommand*\@biblabel[1]{[#1]\hfill}

```

`\@cite` The output of the `\cite` command is produced by this macro. The default from `latex.dtx` is used.

```
1609 % \renewcommand*\@cite[1]{[#1]}
```

### 9.3 The index

`theindex` The environment ‘theindex’ can be used for indices. It makes an index with two columns, with each entry a separate paragraph. At the user level the commands `\item`, `\subitem` and `\subsubitem` are used to produce index entries of various levels. When a new letter of the alphabet is encountered an amount of `\indexspace` white space can be added.

```
1610 \newenvironment{theindex}{%
1611   \if@twocolumn
1612     \@restonecolfalse
1613   \else
1614     \@restonecoltrue
1615   \fi
1616 (artikel) \twocolumn[\section*{\indexname}]%
1617 (!artikel) \twocolumn[\@makeschapterhead{\indexname}]%
1618   \@mkboth{\MakeUppercase{\indexname}}{\MakeUppercase{\indexname}}%
1619   \thispagestyle{plain}\parindent\z@
```

Parameter changes to `\columnseprule` and `\columnsep` have to be done after `\twocolumn` has acted. Otherwise they can affect the last page before the index.

```
1620   \columnseprule \z@
1621   \columnsep 35\p@
1622   \parskip\z@ \@plus .3\p@\relax
1623   \let\item\@idxitem
1624 }{%
```

When the document continues after the index and it was a one column document we have to switch back to one column after the index.

```
1625   \if@restonecol\onecolumn\else\clearpage\fi}
```

`\@idxitem` These macros are used to format the entries in the index.

```
\subitem 1626 \newcommand*\@idxitem {\par \hangindent 40\p@}
\subsubitem 1627 \newcommand*\subitem {\@idxitem \hspace*{20\p@}}
1628 \newcommand*\subsubitem{\@idxitem \hspace*{30\p@}}
```

`\indexspace` The amount of white space that is inserted between ‘letter blocks’ in the index.

```
1629 \newcommand*\indexspace{\par \vskip10\p@\@plus5\p@\@minus3\p@\relax}
```

### 9.4 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`. The standard `LATEX` document classes make sure that the rule takes no vertical space (see `plain.tex`) and compensate for the natural height of the rule of 0.4pt by adding the right amount of vertical skip. For the `artikel2` document class this is still true, but for the others the amount of whitespace between the last line of the text and the start of the footnotes is increased by giving `\footnoterule` a positive height<sup>1</sup>.

<sup>1</sup>This should perhaps have been done by increasing the value of `\skip\footins`, but changing that now would mean changing the formatting of existing documents. (JLB, 08/09/1997)

To prevent the rule from colliding with the footnote we first add a little negative vertical skip, then we put the rule and add some positive vertical skip.

```

1630 \renewcommand*\footnoterule{%
1631   \kern-3\p@
1632 <*type1 | type3>
1633   \kern.5\baselineskip
1634   \hrule\@width\unitindent
1635   \kern.4\baselineskip
1636 </type1 | type3>
1637 <*type2>
1638   \hrule\@width 3\unitindent
1639   \kern 2.6\p@
1640 </type2>
1641 }

```

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

```

1642 % \newcounter{footnote}
1643 \!artikel\@addtoreset{footnote}{chapter}

```

`\@makefn` The footnote mechanism of L<sup>A</sup>T<sub>E</sub>X calls the macro `\@makefn` to produce the actual footnote. The macro gets the text of the footnote as its argument and should use `\@thefnmark` as the mark of the footnote. The macro `\@makefn` 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\@xmakefn#1#2{%
%<!type3> \parindent=.5\unitindent
%<type3> \parindent=\z@\parskip=.5\baselineskip
\def\labelitemi{--}\@revelabeltrue
\setbox0=\hbox {#1\hskip.5em plus 1fil}%
\dimen0=2\wd0
\ifdim\dimen0>\unitindent
\global\unitindent=\dimen0
\@indentset
\fi}%
\@setpar{\@par
\@tempdima \hsize
\advance\@tempdima-.5\unitindent
\parshape \@ne .5\unitindent \@tempdima}%
\par
\noindent\llap{\hb@xt@.5\unitindent{#1\hfil}}#2}

```

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 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.

We change the label of itemized lists inside footnotes and need to check that the `\unitindent` is large enough for our purposes.

For most of the document classes produced from this file we need a slightly modified `\@makefntext` on the title page, so we introduce an extra macro, `\xmakefntext`.

```

1644 (*type1 | type3)
1645 \newcommand*\@makefntext{\xmakefntext{\normalfont\@thefnmark.}}
1646 \newcommand*\xmakefntext[1]{%
1647     \parindent\z@
1648     \def\labelitemi{\textendash}\@revlabeltrue
1649     {\setbox0\hbox {#1\hskip.5em plus 1fil}
1650     \dimen0=2\wd0\relax
1651     \ifdim\dimen0>\unitindent
1652     \global\unitindent\dimen0\relax
1653     \@indentset
1654     \fi}
1655     \leavevmode\hb@xt@.5\unitindent{#1\hfil}}
1656 </type1 | type3>

```

For the `artikel2` document class we have a simpler definition of `\@makefntext`.

```

1657 (*type2)
1658 \newcommand\@makefntext[1]{%
1659     \parindent\othermargin
1660     \noindent\hb@xt@\othermargin{\normalfont\@thefnmark\hfil\relax}#1}
1661 </type2>

```

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

```

1662 %\renewcommand\@makefnmark{\hbox{\@textsuperscript{\normalfont\@thefnmark}}}

```

## 10 Initialization

### 10.1 Words

`\contentsname` This document class is for documents prepared in the English language. To prepare  
`\listfigurename` a version for another language, various English words must be replaced. All the  
`\listtablename` English words that require replacement are defined below in command names.

```

1663 \newcommand*\contentsname{Contents}
1664 \newcommand*\listfigurename{List of Figures}
1665 \newcommand*\listtablename{List of Tables}

\refname
\bibname 1666 (artikel) \newcommand*\refname{References}
\indexname 1667 (rapport | boek) \newcommand*\bibname{Bibliography}
1668 \newcommand*\indexname{Index}

\figurename
\tablename 1669 \newcommand*\figurename{Figure}
1670 \newcommand*\tablename{Table}

```

```

\partname
\chaptername 1671 \newcommand*\partname{Part}
\appendixname
\abstractname
\seename
\andname

```

```

1672 <rapport | boek> \newcommand* \chaptername{Chapter}
1673 \newcommand* \appendixname{Appendix}
1674 <!boek> \newcommand* \abstractname{Abstract}
1675 \newcommand* \seename{see}
1676 \newcommand* \andname{and}

```

## 10.2 Date

`\today` This macro uses the  $\TeX$  primitives `\month`, `\day` and `\year` to provide the date of the  $\LaTeX$ -run.

```
1677 \newcommand* \today{}
```

To save space we define `\today` in a way that it is expanded when the class file is read in. This means that low-level changes to the internal  $\TeX$  registers that are happening later on (e.g. if some packages goes `\month=5`) are not reflected in `\today`.

```

1678 \def\today{\ifcase\month\or
1679   January\or February\or March\or April\or May\or June\or
1680   July\or August\or September\or October\or November\or December\fi
1681   \space\number\day, \number\year}

```

## 10.3 Two column mode

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

```
1682 \setlength\columnsep{10\p@}
```

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

```
1683 \setlength\columnseprule{0\p@}
```

## 10.4 The page style

We have *plain* pages in the document classes `artikel` and `rapport` unless the user specified otherwise. In the `boek` document class we use the page style *headings* by default. We use arabic pagenumbers.

```

1684 <!boek> \pagestyle{plain}
1685 <boek> \pagestyle{headings}
1686 \pagenumbering{arabic}      % Arabic page numbers

```

## 10.5 Single or double sided printing

When the `twoside` option wasn't specified, we don't try to make each page as long as all the others.

```

1687 <*artikel>
1688 \if@twoside
1689 \else
1690   \raggedbottom
1691 \fi
1692 </artikel>

```

When the `twocolumn` option was specified we call `\twocolumn` to activate this mode. We try to make each column as long as the others, but call `\sloppy` to make our life easier.

```
1693 \if@twocolumn
1694   \twocolumn
1695   \sloppy
1696   \flushbottom
```

Normally we call `\onecolumn` to initiate typesetting in one column.

```
1697 \else
1698   \onecolumn
1699 \fi
```

`\frenchspacing` Controls the amount of space after a punctuation mark.

```
1700 \frenchspacing
1701 (</artikel | rapport | boek)
```

## Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

<b>Symbols</b>	<code>\dotsep</code> . . . . . <u>1381</u>	<code>\indentset</code> . . . . . <u>188</u> ,
<code>\@Roman</code> . . . . . 752	<code>\dottedtocline</code> 1519,	194, 196, 684, 1653
<code>\@afterheading</code> . . . . .	1526–1529,	<code>\itemdepth</code> . . . . .
. 820, 853, 902, 926	1532–1535, 1559	. . 1131, 1134, 1135
<code>\@afterindentfalse</code> . . . . .	<code>\emptypagestyle</code> . . . . .	<code>\itemitem</code> . 1135, 1136
. . . . . 781, 878	. . . . . 437, 440	<code>\itempenalty</code> . . . . . <u>1030</u>
<code>\@afterindenttrue</code> . . . . .	<code>\endparpenalty</code> . . . . . <u>1030</u>	<code>\latex@warning</code> . . 1602
. . . . . 782, 1415	<code>\endpart</code> . 842, 861, <u>862</u>	<code>\listI</code> . . . . . 102, <u>1033</u>
<code>\@allcapsfalse</code> . . . . . 53	<code>\endtheorem</code> . . . . . <u>1244</u>	<code>\listdepth</code> . 1111, 1138
<code>\@allcapstrue</code> . . . . . 54	<code>\enumctr</code> . . . 1108–1110	<code>\listi</code> . . . . . 102, <u>1033</u>
<code>\@author</code> 525, 556, 558,	<code>\enumdepth</code> . . . . .	<code>\listiii</code> . . . . . <u>1046</u>
576, 628, 648, 660	. . 1104, 1107, 1108	<code>\listiv</code> . . . . . <u>1046</u>
<code>\@beginparpenalty</code> <u>1030</u>	<code>\eqnnum</code> . . . . . <u>1306</u>	<code>\listv</code> . . . . . <u>1046</u>
<code>\@begintheorem</code> . . . . . <u>1244</u>	<code>\evenfoot</code> . . . . . <u>436</u> ,	<code>\listvi</code> . . . . . <u>1046</u>
<code>\@biblabel</code> . . . . .	449, 451, 456, 512	<code>\lowpenalty</code> . . . . .
. . 1591, 1592, <u>1608</u>	<code>\evenhead</code> . . . . .	. . . . . <u>214</u> , 1030–1032
<code>\@chapapp</code> . 478, 505,	. . . . . <u>436</u> , 446, 457, 513	<code>\mainaux</code> . . . . . 192, 195
<u>762</u> , 884, 913, 1282	<code>\fnsymbol</code> . . . 539, 599	<code>\mainmatterfalse</code> . . . . .
<code>\@chapter</code> . . . . . 879, <u>880</u>	<code>\fontswitch</code> 1372, 1373	. . . . . 766, 774
<code>\@cite</code> . . . . . <u>1609</u>	<code>\fpbot</code> . . . . . <u>405</u>	<code>\mainmattertrue</code> 9, 770
<code>\@currentpagestyle</code> . . . . .	<code>\fpsep</code> . . . . . <u>405</u>	<code>\makecaption</code> . . . . . <u>1345</u>
. . . . . 438, 440	<code>\fptop</code> . . . . . <u>405</u>	<code>\makechapterhead</code> . . . . .
<code>\@date</code> . . . . . 526,	<code>\hangfrom</code> 695, 698, 728	. . . . . 899, 901, <u>905</u>
527, 564, 566,	<code>\highpenalty</code> . . . . .	<code>\makefnmark</code> . 600, <u>1662</u>
578, 630, 651, 661	. . . . . <u>214</u> , 1456,	<code>\makefnmark</code> . . . . .
<code>\@dblfloat</code> . 1324, 1343	1475, 1483, 1493	536, 602, 605, <u>1644</u>
<code>\@dblfpbot</code> . . . . . <u>420</u>	<code>\idxitem</code> . . 1623, <u>1626</u>	<code>\makechapterhead</code> . . . . .
<code>\@dblfpsep</code> . . . . . <u>420</u>	<code>\ifundefined</code> . . . . . 193	923, 925, <u>928</u> , 1617
<code>\@dblfpsep</code> . . . . . <u>420</u>		

<code>\@maketitle</code> ... 612,	119, 128, 134,	<code>\_</code> ..... 478,
614, 619, 626, <a href="#">636</a>	140, 148–154,	484, 505, 1247, 1251
<code>\@medpenalty</code> ..... <a href="#">214</a>	157–163, 166–171	
<code>\@minipagefalse</code> .. 1355	<code>\@settopoint</code> .. 267,	
<code>\@minipagerestore</code> <a href="#">1290</a>	339, 340, 345, 357	
<code>\@mparswitchfalse</code> . 41	<code>\@spart</code> ... 783, 796, <a href="#">844</a>	<b>A</b>
<code>\@mparswitchtrue</code> .. 42	<code>\@specialpagetrue</code> . 442	<code>\abovecaptionskip</code> .
<code>\@mpfootins</code> ..... <a href="#">1296</a>	<code>\@specialstyle</code> .... 443	..... <a href="#">1345</a> , <a href="#">1350</a>
<code>\@nameuse</code> ..... 438	<code>\@ssect</code> ..... <a href="#">724</a>	<code>\abovedisplayshortskip</code>
<code>\@needwriteindenttrue</code>	<code>\@startsection</code> ....	..... 86,
..... 191	. 942, 946, 955,	92, 98, 109, 115,
<code>\@nobreakfalse</code> ....	959, 968, 972,	121, 130, 136, 142
..... 1409, 1467	982, 983, 990, 991	<code>\abovedisplayskip</code> 85,
<code>\@nobreaktrue</code> .... 1466	<code>\@starttoc</code> ... 1398,	91, 97, 101, 108,
<code>\@noitemerr</code> ..... 1601	<a href="#">1401</a> , 1555, 1575	114, 120, 124,
<code>\@normalsize</code> ..... <a href="#">81</a>	<code>\@svsec</code> ..... 674,	129, 135, 141, 145
<code>\@oddfont</code> . <a href="#">436</a> , 447,	688, 695, 699, 711	abstract (environ-
451, 456, 489, 512	<code>\@svsechd</code> .... 710, 731	ment) ..... <a href="#">1188</a>
<code>\@oddfont</code> .... <a href="#">436</a> ,	<code>\@tempa</code> ..... 584,	<code>\abstractname</code> ....
446, 458, 490, 514	586, 588, 592, 836	... 1192, 1199,
<code>\@oldtocfalse</code> .... 11	<code>\@tempskipa</code> .....	1204, 1208, <a href="#">1671</a>
<code>\@oldtoctrue</code> ..... 51	. 690, 691, 724, 725	<code>\addcontentsline</code> ..
<code>\@opargbegintheorem</code>	<code>\@textsuperscript</code> .	.... 703, 712,
..... <a href="#">1244</a>	..... 538,	803, 805, 826,
<code>\@openbib@code</code> ....	601, 603, 608, 1662	828, 885, 889, 893
... 61, 1595, <a href="#">1607</a>	<code>\@thanks</code> .....	<code>\addtocontents</code> 896, 897
<code>\@openrightfalse</code> .. 48	. 569, 575, 621, 627	<code>\and</code> ..... 556,
<code>\@openrighttrue</code> ... 47	<code>\@thefnmark</code> ... 538,	582, 585, 634, 657
<code>\@part</code> .... 783, 796, <a href="#">798</a>	601, 603, 608,	<code>\andname</code> .... 590, <a href="#">1671</a>
<code>\@pnumwidth</code> ... <a href="#">1381</a> ,	1645, 1660, 1662	<code>\appendix</code> ..... <a href="#">1272</a>
1460,	<code>\@title</code> 524, 547, 549,	<code>\appendixname</code> 1282, <a href="#">1671</a>
1463, 1487,	577, 629, 643, 655	<code>\arraycolsep</code> ..... <a href="#">1285</a>
1488, 1492,	<code>\@titlecenteredfalse</code> 14	<code>\arrayrulewidth</code> .. <a href="#">1287</a>
1506, 1507, 1511	<code>\@titlecenteredtrue</code> 52	<code>\AtEndDocument</code> .... 197
<code>\@ptsize</code> .. <a href="#">1</a> , 38–40, 79	<code>\@titlepagefalse</code> ..	<code>\AtEndOfPackage</code> ... 60
<code>\@regtocline</code> .. <a href="#">1411</a> ,	..... 6, 46, 50	<code>\author</code> ... <a href="#">524</a> , 580, 632
1477, 1496,	<code>\@titlepagetrue</code> 7, 45, 49	
1514, 1521,	<code>\@tocrmarg</code> ..... <a href="#">1381</a>	<b>B</b>
1538–1541, 1561	<code>\@toodeep</code> .. 1105, 1132	<code>\backmatter</code> ..... <a href="#">772</a>
<code>\@restonecolfalse</code> .	<code>\@topnewpage</code> .. 899, 923	<code>\baselineskip</code> .....
... 1259, 1391,	<code>\@topnum</code> ..... 618, 877	.... 210, 211,
1548, 1568, 1612	<code>\@twocolumnfalse</code> .. 55	270–272, 277,
<code>\@restonecoltrue</code> ..	<code>\@twocolumntrue</code> ... 56	279, 943, 947,
... 1257, 1389,	<code>\@twosidefalse</code> .... 41	949, 950, 956,
1546, 1566, 1614	<code>\@twosidetrue</code> ..... 42	960, 962, 963,
<code>\@revlabeltrue</code> 537, 1648	<code>\@verbatim</code> . 1178, 1184	969, 973, 975,
<code>\@roman</code> ..... 1094	<code>\@vobeyspaces</code> .... 1186	976, 1159, 1160,
<code>\@schapter</code> ... 879, <a href="#">922</a>	<code>\@width</code> .... 1634, 1638	1164, 1165,
<code>\@seccntformat</code> 688, 720	<code>\@writeindent</code> . <a href="#">192</a> , 200	1169, 1170,
<code>\@secpenalty</code> .....	<code>\@xmakefntext</code> .....	1176, 1177,
... 1455, 1474, 1502	... 602, 1645, 1646	1245, 1249,
<code>\@sect</code> ..... <a href="#">672</a>	<code>\@xsect</code> ..... 719, 733	1252, 1293,
<code>\@setfontsize</code> .. 84,	<code>\@xverbatim</code> ..... 1186	1294, 1633, 1635
90, 96, 107, 113,		<code>\baselinestretch</code> .. <a href="#">178</a>
		<code>\belowcaptionskip</code> .
		..... <a href="#">1345</a> , <a href="#">1358</a>

<code>\belowdisplaysshortskip</code>	<code>\center</code> . . . . .	1158	721, 722, 1109,
. . . . . 87,	<code>center</code> (environment) <u>1157</u>		1136, 1407, 1408
93, 99, 110, 116,	<code>\centering</code> 831, 857, 1161		<code>\endflushleft</code> . . . . . 1167
122, 131, 137, 143	<code>\changes</code> . . . . . 1604		<code>\endflushright</code> . . . 1172
<code>\belowdisplayskip</code> .	<code>\ChapFont</code> . . . . . <u>904</u> ,		<code>\endgraf</code> . . 695, 699, 728
. . . . . 101, 124, 145	913, 918, 935, 1010		<code>\endtitlepage</code> . . . . . 1194
<code>\bf</code> . . . . . <u>1368</u>	<code>\chapter</code> . <u>874</u> , 1393,		<code>\endverbatim</code> . . . . . 1187
<code>\bibindent</code> 62, 63, <u>1579</u>	1550, <u>1570</u> , 1588		<code>enumerate</code> (environ-
<code>\bibname</code> 1588, 1589, <u>1666</u>	<code>\chaptermark</code> . . 475,		ment) . . . . . <u>1103</u>
<code>\bottomfraction</code> . . . <u>370</u>	501, 516, <u>734</u> , 895		environments:
<code>\brokenpenalty</code> . . . . <u>223</u>	<code>\chaptername</code> . 762, <u>1671</u>		<code>abstract</code> . . . . . <u>1188</u>
	<code>\cleardoublepage</code> .		<code>center</code> . . . . . <u>1157</u>
<b>C</b>	. . . . . 765, 769,		<code>description</code> . . . <u>1152</u>
<code>\c@bottomnumber</code> . . . <u>369</u>	773, 787, 875, 1255		<code>enumerate</code> . . . . . <u>1103</u>
<code>\c@chapter</code> . . . . .	<code>\clearpage</code> 773, 875, 1625		<code>figure</code> . . . . . <u>1320</u>
<u>742</u> , 755, 1283,	<code>\clubpenalty</code> . <u>217</u> , 1599		<code>figure*</code> . . . . . <u>1320</u>
1303, 1314, 1333	<code>\col@number</code> . . . . . 611		<code>flushleft</code> . . . . . <u>1163</u>
<code>\c@dbltopnumber</code> . . . <u>374</u>	<code>\columnsep</code> . 1621, <u>1682</u>		<code>flushright</code> . . . . . <u>1168</u>
<code>\c@enumi</code> . . . . . 1092	<code>\columnseprule</code> . . . .		<code>itemize</code> . . . . . <u>1130</u>
<code>\c@enumii</code> . . . . . 1093	. . . . . 1620, <u>1683</u>		<code>quotation</code> . . . . . <u>1230</u>
<code>\c@enumiii</code> . . . . . 1094	<code>\contentsname</code> . 1393,		<code>quote</code> . . . . . <u>1240</u>
<code>\c@enumiv</code> . . . . .	1395–1397, <u>1663</u>		<code>table</code> . . . . . <u>1339</u>
. . 1095, 1591, 1598	<code>\cs</code> . . . . . 1604, 1605		<code>table*</code> . . . . . <u>1339</u>
<code>\c@equation</code> . 1299, 1303	<code>\csname</code> . . . 680, 702,		<code>thebibliography</code> <u>1582</u>
<code>\c@figure</code> . . . . . <u>1307</u>	706, 711, 715,		<code>theindex</code> . . . . . <u>1610</u>
<code>\c@footnote</code> . . . . .	721, 722, 1109,		<code>titlepage</code> . . . . . <u>1253</u>
. . . 539, 599, <u>1642</u>	1136, 1407, 1408		<code>verbatim</code> . . . . . <u>1174</u>
<code>\c@paragraph</code> . . <u>742</u> , 760			<code>verse</code> . . . . . <u>1221</u>
<code>\c@part</code> . . . . . <u>742</u> , 752	<b>D</b>		<code>\evensidemargin</code> . . . <u>291</u>
<code>\c@secnumdepth</code> . . . .	<code>\date</code> . . . . . <u>524</u> , 581, 633		<code>\exhyphenpenalty</code> . . 670
. 463, 469, 476,	<code>\dblfloatpagefraction</code>		<code>\ext@figure</code> . . . . . <u>1316</u>
483, 495, 503,	. . . . . <u>376</u>		<code>\ext@table</code> . . . . . <u>1335</u>
673, 704, 713,	<code>\dblfloatsep</code> . . . . . <u>393</u>		
<u>740</u> , 801, 810,	<code>\dbltextfloatsep</code> . . <u>393</u>		<b>F</b>
824, 834, 881, 911	<code>\dbltopfraction</code> . . . <u>375</u>		<code>\fboxrule</code> . . . . . <u>1297</u>
<code>\c@section</code> . . . . .	<code>\DeclareOldFontCommand</code>		<code>\fboxsep</code> . . . . . <u>1297</u>
<u>742</u> , 753, 756, 1276	. . . . . 1365–1371		<code>figure</code> (environment) <u>1320</u>
<code>\c@subparagraph</code> <u>742</u> , 761	<code>\DeclareRobustCommand</code>		<code>figure*</code> (environment)
<code>\c@subsection</code> . <u>742</u> , 758	. . . . . 1372–1374		. . . . . <u>1320</u>
<code>\c@subsubsection</code> . .	<code>description</code> (environ-		<code>\figurename</code> . 1319, <u>1669</u>
. . . . . <u>742</u> , 759	ment) . . . . . <u>1152</u>		<code>\floatpagefraction</code> . <u>373</u>
<code>\c@table</code> . . . . . <u>1326</u>	<code>\descriptionlabel</code> .		<code>\floatsep</code> . . . . . <u>378</u>
<code>\c@tocdepth</code> . . . . .	. . . . . 1154, <u>1156</u>		<code>\flushbottom</code> . . . . . 1696
. . 1412, 1454, 1473	<code>\dimen</code> . . . . . 1650–1652		<code>\flushleft</code> . . . . . 1163
<code>\c@topnumber</code> . . . . . <u>366</u>	<code>\displaywidowpenalty</code>		<code>flushleft</code> (environ-
<code>\c@totalnumber</code> . . . . <u>371</u>	. . . . . <u>219</u>		ment) . . . . . <u>1163</u>
<code>\cal</code> . . . . . <u>1372</u>	<code>\doublerulesep</code> . . . <u>1288</u>		<code>\flushright</code> . . . . . 1168
<code>\CaptionFonts</code> . . . . <u>1361</u>			<code>flushright</code> (environ-
<code>\CaptionLabelFont</code> .	<b>E</b>		ment) . . . . . <u>1168</u>
. . . . . 1351,	<code>\edef</code> . . . 836, 1108, 1135		<code>\fnum@figure</code> . . . . . 1319
1353, <u>1359</u> , 1362	<code>\em</code> . . . . . <u>1374</u>		<code>\fnum@table</code> . . . . . 1338
<code>\CaptionTextFont</code> . .	<code>\endcenter</code> . . . . . 1162		<code>\font</code> . . . . . 1376
. . . . . 1351,	<code>\endcsname</code> 680, 702,		<code>\fontdimen</code> . . . . . 1376
1353, <u>1359</u> , 1363	706, 711, 715,		<code>\footins</code> . . . . . <u>362</u> , 1296

<code>\footnote</code> . . . . .	535, 642	<code>\if@revlabel</code> . . . . .		<code>\labelenumii</code> . . . . .	<a href="#">1096</a>
<code>\footnoterule</code> . . . . .	534, <a href="#">1630</a>	. . . . .	<a href="#">15</a> , 1113, 1140	<code>\labelenumiii</code> . . . . .	<a href="#">1096</a>
<code>\footnotesep</code> . . . . .	<a href="#">359</a>	<code>\if@titlecentered</code> . . . . .		<code>\labelenumiv</code> . . . . .	<a href="#">1096</a>
<code>\footnotesize</code> . . . . .	<a href="#">126</a> , 533	. . . . .	<a href="#">13</a> , 640	<code>\labelitemi</code> . . . . .	
<code>\footskip</code> . . . . .	<a href="#">231</a> , 355	<code>\if@titlepage</code> . . . . .		. . . . .	537, 606, <a href="#">1126</a> , 1648
<code>\fps@figure</code> . . . . .	<a href="#">1316</a>	. . . . .	<a href="#">5</a> , 529, 1188	<code>\labelitemii</code> . . . . .	<a href="#">1126</a>
<code>\fps@table</code> . . . . .	<a href="#">1335</a>	<code>\if@twocolumn</code> . . . . .	238,	<code>\labelitemiii</code> . . . . .	<a href="#">1126</a>
<code>\frenchspacing</code> . . . . .		. . . . .	251, 282, 313,	<code>\labelitemiv</code> . . . . .	<a href="#">1126</a>
. . . . .	1186, <a href="#">1700</a>	. . . . .	610, 789, 898,	<code>\labelsep</code> . . . . .	<a href="#">1025</a> ,
<code>\frontmatter</code> . . . . .	<a href="#">763</a>	. . . . .	922, 1198, 1217,	. . . . .	1036, 1037,
<code>\ftype@figure</code> . . . . .	<a href="#">1316</a>	. . . . .	1218, 1256,	. . . . .	1039, 1047,
<code>\ftype@table</code> . . . . .	<a href="#">1335</a>	. . . . .	1388, 1545,	. . . . .	1048, 1050,
<code>\futurelet</code> . . . . .	584, 586	. . . . .	1565, 1611, 1693	. . . . .	1056, 1057,
		<code>\if@twoside</code> 292, 319,		. . . . .	1059, 1066,
		. . . . .		. . . . .	1067, 1069,
		<code>\ignorespaces</code> 1193, 1205		. . . . .	1075, 1076,
		<code>\immediate</code> 192, 195, 1408		. . . . .	1078, 1084,
		<code>\indent</code> . . . . .	812, 850	. . . . .	1085, 1087,
		<code>\indexname</code> . . . . .		. . . . .	1156, 1245,
		. . . . .	1616–1618, <a href="#">1666</a>	. . . . .	1247, 1249,
		<code>\indexspace</code> . . . . .	<a href="#">1629</a>	. . . . .	1251, 1289, 1594
		<code>\interlinepenalty</code> . . . . .	<a href="#">222</a> ,	<code>\labelwidth</code> . . . . .	<a href="#">1025</a> ,
		. . . . .	669, 832, 858, 1415	. . . . .	1038, 1039,
		<code>\intertextsep</code> . . . . .	<a href="#">378</a>	. . . . .	1049, 1050,
		<code>\it</code> . . . . .	<a href="#">1369</a>	. . . . .	1058, 1059,
		<code>\item</code> 1161, 1166, 1171,		. . . . .	1068, 1069,
		. . . . .	1214, 1228,	. . . . .	1077, 1078,
		. . . . .	1238, 1242,	. . . . .	1086, 1087,
		. . . . .	1247, 1251, 1623	. . . . .	1153, 1592, 1593
		<code>\itemindent</code> . . . . .		<code>\LARGE</code> <a href="#">147</a> , 545, 643, 655	
		. . . . .	63, 64, 1153,	<code>\Large</code> . . . . .	<a href="#">147</a> ,
		. . . . .	1224, 1225, 1235	. . . . .	811, 812, 815,
		<code>itemize</code> (environment)		. . . . .	816, 835, 841,
		. . . . .	<a href="#">1130</a>	. . . . .	849, 850, 860,
		<code>\itemsep</code> . . . . .	1043, 1054,	. . . . .	913, 918, 935, 1434
		. . . . .	1064, 1073,	<code>\large</code> . . . . .	<a href="#">147</a> , 552,
		. . . . .	1082, 1091, 1223	. . . . .	562, 645, 651,
		<code>\itshape</code> . . . . .	1369	. . . . .	656, 661, 951, 1463
				<code>\leftmargin</code> . . . . .	62,
				. . . . .	<a href="#">1017</a> , 1034,
				. . . . .	1035, 1038,
				. . . . .	1046, 1055,
				. . . . .	1065, 1074,
				. . . . .	1083, 1153,
				. . . . .	1211, 1226,
				. . . . .	1227, 1236,
				. . . . .	1241, 1593, 1594
				<code>\leftmargini</code> . . . . .	190,
				. . . . .	<a href="#">1017</a> , 1026, 1035
				<code>\leftmarginii</code> . . . . .	
				. . . . .	1017, 1046, 1049
				<code>\leftmarginiii</code> . . . . .	
				. . . . .	1017, 1055, 1058

<code>\leftmarginiv</code> .....		<code>\paragraphmark</code> .... <a href="#">734</a>
.. <a href="#">1017</a> , 1065, 1068		<code>\parindent</code> ... 189,
<code>\leftmarginv</code> .....		<a href="#">205</a> , 536, 553,
.. <a href="#">1017</a> , 1074, 1077		605, 808, 846,
<code>\leftmarginvi</code> .....		908, 931, 990,
.. <a href="#">1017</a> , 1083, 1086		1210, 1290,
<code>\leftskip</code> .... 1180,		1421, 1460,
1181, 1416,		1487, 1506,
1423, 1490,		1619, 1647, 1659
1491, 1509, 1510		<code>\parsep</code> ... 65, 1040,
<code>\lineskip</code> .....		1053, 1062,
. <a href="#">175</a> , 553, 646, 656		1072, 1081,
<code>\listfigurename</code> 1550,		1090, 1213, 1237
1552–1554, <a href="#">1663</a>		<code>\parskip</code> .....
<code>\listoffigures</code> ... <a href="#">1543</a>		<a href="#">205</a> ,
<code>\listoftables</code> .... <a href="#">1563</a>		909, 932, 1042,
<code>\listparindent</code> ....		1052, 1061,
.... 64, 1209,		1071, 1080,
1225, 1232–1235		1089, 1183,
<code>\listtablename</code> 1570,		1293, 1404, 1622
1572–1574, <a href="#">1663</a>		<code>\part</code> .....
		<a href="#">776</a>
		<code>\PartFont</code> . <a href="#">798</a> , 811,
		812, 815, 816,
		835, 841, 849,
		850, 860, 999, 1009
		<code>\partname</code> .....
		811, 812, 836, <a href="#">1671</a>
		<code>\partopsep</code> . <a href="#">1028</a> , 1063
		<code>\postdisplaypenalty</code> <a href="#">219</a>
		<code>\predisplaypenalty</code> . <a href="#">219</a>
		<code>\protected@edef</code> ... 688
		<code>\ps@headings</code> .....
		<a href="#">454</a>
		<code>\ps@myheadings</code> .... <a href="#">511</a>
		<code>\ps@plain</code> .....
		<a href="#">445</a>
		<b>Q</b>
		<code>\quad</code> .... 464, 470, 496
		quotation (environ-
		ment) .....
		<a href="#">1230</a>
		quote (environment) <a href="#">1240</a>
		<b>R</b>
		<code>\raggedleft</code> .....
		1171
		<code>\refname</code> 1584, 1585, <a href="#">1666</a>
		<code>\renewenvironment</code> .
		..... 1103, 1130
		<code>\rm</code> .....
		<a href="#">1365</a>
		<code>\romannumeral</code> 1108, 1135
		<code>\RunningFonts</code> .... <a href="#">522</a>
		<b>S</b>
		<code>\sbox</code> .....
		<a href="#">1351</a>
		<code>\sc</code> .....
		<a href="#">1369</a>
		<code>\scriptsize</code> .....
		<a href="#">147</a>
		<code>\scshape</code> . 979, 988, 1371
<code>\leftmarginiv</code> .....	<b>N</b>	
.. <a href="#">1017</a> , 1065, 1068	<code>\newblock</code> 67, <a href="#">1581</a> , <a href="#">1604</a>	
<code>\leftmarginv</code> .....	<code>\newif</code> .....	4, 5,
.. <a href="#">1017</a> , 1074, 1077	8–10, 12, 13, 15, 187	
<code>\leftmarginvi</code> .....	<code>\newwrite</code> .....	1407
.. <a href="#">1017</a> , 1083, 1086	<code>\nobreakspace</code> . 811,	
<code>\leftskip</code> .... 1180,	812, 837, 1319, 1338	
1181, 1416,	<code>\noexpand</code> .....	836
1423, 1490,	<code>\normalfont</code> .....	
1491, 1509, 1510	. 601, 603, 608,	
<code>\lineskip</code> .....	809, 833, 848,	
. <a href="#">175</a> , 553, 646, 656	859, 910, 934,	
<code>\listfigurename</code> 1550,	951, 964, 977,	
1552–1554, <a href="#">1663</a>	986, 994, 1127,	
<code>\listoffigures</code> ... <a href="#">1543</a>	1365–1371,	
<code>\listoftables</code> .... <a href="#">1563</a>	1645, 1660, 1662	
<code>\listparindent</code> ....	<code>\normallineskip</code> ... <a href="#">175</a>	
.... 64, 1209,	<code>\normalsize</code> .....	
1225, 1232–1235	.. <a href="#">81</a> , 181, 185,	
<code>\listtablename</code> 1570,	964, 977, 986, 994	
1572–1574, <a href="#">1663</a>	<code>\num@figure</code> .....	<a href="#">1316</a>
	<code>\num@table</code> .....	<a href="#">1335</a>
	<code>\numberline</code> .....	<a href="#">1427</a>
	<b>O</b>	
<code>\mainmatter</code> .....	<code>\oddsidemargin</code> .... <a href="#">291</a>	
<a href="#">768</a>	<code>\onecolumn</code> 790, 1257,	
<code>\makeatletter</code> .... 1403	1389, 1546,	
<code>\makelabel</code> ... 1114,	1566, 1625, 1698	
1116, 1117,	<code>\openout</code> .....	1408
1121, 1141–	<code>\othermargin</code> .. <a href="#">184</a> ,	
1143, 1147, 1154	204, 1018–1022,	
<code>\maketitle</code> .....	1420, 1659, 1660	
<a href="#">529</a>	<code>\overfullrule</code> ... 43, 44	
<code>\MakeUppercase</code> ....		
. 462, 476, 483,		
494, 502, 1396,		
1397, 1450,		
1553, 1554,		
1573, 1574,		
1585, 1589, 1618		
<code>\marginparpush</code> .... <a href="#">282</a>		
<code>\marginparsep</code> .....		
.... <a href="#">282</a> , 325, 333		
<code>\marginparwidth</code> ... <a href="#">291</a>		
<code>\MarkFont</code> .....		
. 457, 458, 490,		
513, 514, <a href="#">520</a> , 523		
<code>\mathbf</code> .....		
1368		
<code>\mathcal</code> .....		
1372		
<code>\mathit</code> .....		
1369		
<code>\mathnormal</code> .....		
1373		
<code>\mathrm</code> .....		
1365		
<code>\mathsf</code> .....		
1366		
<code>\mathhtt</code> .....		
1367		
<code>\maxdepth</code> .....		
<a href="#">232</a>		
<code>\mit</code> .....		
<a href="#">1372</a>		
	<b>P</b>	
	<code>\p@enumii</code> .....	<a href="#">1100</a>
	<code>\p@enumiii</code> .....	<a href="#">1100</a>
	<code>\p@enumiv</code> .. <a href="#">1100</a> , 1597	
	<code>\PageFont</code> . 447, 449,	
	457, 458, 490,	
	513, 514, <a href="#">520</a> , 523	
	<code>\pagenumbering</code> ....	
	... 767, 771, 1686	
	<code>\paperheight</code> ... 17,	
	20, 23, 26, 29,	
	32, 35, 36, 274, 350	
	<code>\paperwidth</code> .....	
	. 18, 21, 24, 27,	
	30, 33, 36, 37,	
	246, 320, 328, 341	
	<code>\ParaFont</code> .....	
	986, <a href="#">987</a> , 1003, 1014	
	<code>\paragraph</code> .....	<a href="#">981</a>

