

NTG Document Classes for L^AT_EX version 2e*

Copyright (C) 1992 by Leslie Lamport
Copyright (C) 1994-2020 by Victor Eijkhout Johannes Braams

2020-11-17

Contents

1	Introduction	3
2	The DOCSTRIP modules	3
3	Initial Code	3
4	Declaration of Options	4
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	6
4.5	Titlepage option	6
4.6	openright option	6
4.7	Table of contents formatting	6
4.8	Formatting of the title	6
4.9	Twocolumn printing	7
4.10	Equation numbering on the left	7
4.11	Flush left displays	7
4.12	Open bibliography	7
5	Executing Options	7
6	Loading Packages	8
7	Document Layout	8
7.1	Fonts	8
7.2	Paragraphing	11
7.3	Page Layout	13
7.3.1	Vertical spacing	14
7.3.2	The dimension of text	14

*This file has version number v2.1e, last revised 2020-11-17.

7.3.3	Margins	16
7.3.4	Footnotes	19
7.3.5	Float placement parameters	19
7.4	Page Styles	22
7.4.1	Marking conventions	22
7.4.2	Defining the page styles	23
8	Document Markup	26
8.1	The title	26
8.2	Chapters and Sections	31
8.2.1	Building blocks	31
8.2.2	Mark commands	34
8.2.3	Define Counters	34
8.2.4	Front Matter, Main Matter, and Back Matter	35
8.2.5	Parts	36
8.2.6	Chapters	39
8.2.7	Lower level headings	41
8.3	Lists	44
8.3.1	General List Parameters	44
8.3.2	Enumerate	46
8.3.3	Itemize	47
8.3.4	Description	48
8.4	Adapting existing environments	49
8.5	Defining new environments	50
8.5.1	Abstract	50
8.5.2	Verse	51
8.5.3	Quotation	51
8.5.4	Quote	51
8.5.5	Theorem	51
8.5.6	Titlepage	52
8.5.7	Appendix	52
8.6	Setting parameters for existing environments	53
8.6.1	Array and tabular	53
8.6.2	Tabbing	53
8.6.3	Minipage	53
8.6.4	Framed boxes	54
8.6.5	Equation and eqnarray	54
8.7	Floating objects	54
8.7.1	Figure	55
8.7.2	Table	56
8.7.3	Captions	56
8.8	Font changing	57

9	Cross Referencing	58
9.1	Table of Contents, etc.	58
9.1.1	Table of Contents	59
9.1.2	List of figures	65
9.1.3	List of tables	66
9.2	Bibliography	66
9.3	The index	67
9.4	Footnotes	68
10	Initialization	70
10.1	Words	70
10.2	Date	71
10.3	Two column mode	71
10.4	The page style	71
10.5	Single or double sided printing	71

1 Introduction

This file contains the set of document classes that were made available by Working Group 13 of the NTG (Nederlandstalige T_EX Gebruikersgroep). They are compatible with the standard L^AT_EX2e 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.

`\@ptsize` This control sequence is used to store the second digit of the pointsize we are typesetting in. So, normally, it’s value is one of 0, 1 or 2.

	<pre> 1 <*artikel rapport boek> 2 \newcommand*\@ptsize{} 3 </pre>
<code>\if@restonecol</code>	<p>When the document has to be printed in two columns, we sometimes have to temporarily switch to one column. This switch is used to remember to switch back.</p> <pre> 4 \newif\if@restonecol </pre>
<code>\if@titlepage</code>	<p>A switch to indicate if a titlepage has to be produced. For the <code>artikel</code> document class the default is not to make a separate titlepage.</p> <pre> 5 \newif\if@titlepage 6 <artikel>\@titlepagefalse 7 <!artikel>\@titlepagetrue </pre>
<code>\if@openright</code>	<p>A switch to indicate if chapters must start on a right-hand page. The default for the <code>report</code> class is no; for the <code>book</code> class it's yes.</p> <pre> 8 <!artikel>\newif\if@openright </pre>
<code>\if@mainmatter</code>	<p>The switch <code>\if@mainmatter</code>, only available in the document class <code>book</code>, indicates whether we are processing the main material in the book.</p> <pre> 9 <boek>\newif\if@mainmatter \@mainmattertrue </pre>
<code>\if@oldtoc</code>	<p>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.</p> <pre> 10 \newif\if@oldtoc 11 \@oldtocfalse </pre>
<code>\if@allcaps</code>	<p>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.</p> <pre> 12 \newif\if@allcaps </pre>
<code>\if@titlecentered</code>	<p>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 flushleft. This can be changed by the switch <code>\if@titlecentered</code>.</p> <pre> 13 <type3>\newif\if@titlecentered 14 <type3>\@titlecenteredfalse </pre>
<code>\if@revlabel</code>	<p>These document classes need to be able to change the positioning of the label in labeled lists. This switch is used for that purpose.</p> <pre> 15 \newif\if@revlabel </pre>

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^AT_EX 2_ε document classes we supply an extra option: `oldtoc`. If this option is specified the switch `\if@oldtoc` 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 \!rapport | boek\DeclareOption{mctitle}{\@allcapsfalse}
54 \!rapport | boek\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 ‘book’ 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^AT_EX 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^AT_EX 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^AT_EX 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\@xipt{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

```

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

```

105 \ifx\MakeRobust\undefined \else
106   \MakeRobust\normalsize
107 \fi

```

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

```

108 \DeclareRobustCommand\small{%
109 <*10pt>
110   \@setfontsize\small\@ixpt{11}%

```

```

111 \abovedisplayskip 8.5\p@ \@plus3\p@ \@minus4\p@
112 \abovedisplayshortskip \z@ \@plus2\p@
113 \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
114 </10pt>
115 <*11pt>
116 \@setfontsize\small\@xpt\@xipt
117 \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
118 \abovedisplayshortskip \z@ \@plus3\p@
119 \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
120 </11pt>
121 <*12pt>
122 \@setfontsize\small\@xipt{13.6}%
123 \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
124 \abovedisplayshortskip \z@ \@plus3\p@
125 \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
126 </12pt>
127 \belowdisplayskip \abovedisplayskip
128 }

```

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

```

129 \DeclareRobustCommand\footnotesize{%
130 <*10pt>
131 \@setfontsize\footnotesize\@viipt{9.5}%
132 \abovedisplayskip 6\p@ \@plus2\p@ \@minus4\p@
133 \abovedisplayshortskip \z@ \@plus\p@
134 \belowdisplayshortskip 3\p@ \@plus\p@ \@minus2\p@
135 </10pt>
136 <*11pt>
137 \@setfontsize\footnotesize\@ixpt{11}%
138 \abovedisplayskip 8\p@ \@plus2\p@ \@minus4\p@
139 \abovedisplayshortskip \z@ \@plus\p@
140 \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
141 </11pt>
142 <*12pt>
143 \@setfontsize\footnotesize\@xpt\@xipt
144 \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
145 \abovedisplayshortskip \z@ \@plus3\p@
146 \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
147 </12pt>
148 \belowdisplayskip \abovedisplayskip
149 }

```

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

```

\large 150 <*10pt>
\Large 151 \DeclareRobustCommand\scriptsize{\@setfontsize\scriptsize\@viipt\@viipt}
\LARGE 152 \DeclareRobustCommand\tiny{\@setfontsize\tiny\@vpt\@vipt}
\huge 153 \DeclareRobustCommand\large{\@setfontsize\large\@xipt{14}}
\Huge 154 \DeclareRobustCommand\Large{\@setfontsize\Large\@xivpt{18}}
155 \DeclareRobustCommand\LARGE{\@setfontsize\LARGE\@xviipt{22}}

```

```

156 \DeclareRobustCommand\huge{\@setfontsize\huge\@xxpt{25}}
157 \DeclareRobustCommand\Huge{\@setfontsize\Huge\@xxvpt{30}}
158 \</10pt>
159 \*11pt>
160 \DeclareRobustCommand\scriptsize{\@setfontsize\scriptsize\@viiipt{9.5}}
161 \DeclareRobustCommand\tiny{\@setfontsize\tiny\@vipt\@viipt}
162 \DeclareRobustCommand\large{\@setfontsize\large\@xiipt{14}}
163 \DeclareRobustCommand\Large{\@setfontsize\Large\@xivpt{18}}
164 \DeclareRobustCommand\LARGE{\@setfontsize\LARGE\@xxviipt{22}}
165 \DeclareRobustCommand\huge{\@setfontsize\huge\@xxpt{25}}
166 \DeclareRobustCommand\Huge{\@setfontsize\Huge\@xxvpt{30}}
167 \</11pt>
168 \*12pt>
169 \DeclareRobustCommand\scriptsize{\@setfontsize\scriptsize\@viiipt{9.5}}
170 \DeclareRobustCommand\tiny{\@setfontsize\tiny\@vipt\@viipt}
171 \DeclareRobustCommand\large{\@setfontsize\large\@xivpt{18}}
172 \DeclareRobustCommand\Large{\@setfontsize\Large\@xxviipt{22}}
173 \DeclareRobustCommand\LARGE{\@setfontsize\LARGE\@xxpt{25}}
174 \DeclareRobustCommand\huge{\@setfontsize\huge\@xxvpt{30}}
175 \let\Huge=\huge
176 \</12pt>
177 \</10pt | 11pt | 12pt>

```

7.2 Paragraphing

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

```

178 \*artikel | rapport | boek>
179 \setlength\lineskip{1\p@}
180 \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.

```

181 \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

```

182 \newdimen\unitindent

```

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

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

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

```
187 \newdimen\othermargin
188 {\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

```
189 <*type1 | type3>
190 \newif\if@needwriteindent
```

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

```
191 \newcommand*\@indentset{%
192 <!type3> \global\parindent=\unitindent
193 \global\leftmargini=\unitindent
194 \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.

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

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

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

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

```
207 <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.

```
208 <*type1>
209 \setlength\parskip{0\p@}
210 \setlength\parindent{\unitindent}
211 </type1>
212 <*type3>
213 \setlength\parskip{.5\baselineskip \@plus .1\baselineskip
214 \@minus .1\baselineskip}
215 \setlength\parindent{\z@}
216 </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.

```
217 \@lowpenalty 51
218 \@medpenalty 151
219 \@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.

```
220 % \clubpenalty 150
221 % \widowpenalty 150
```

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

```
222 % \displaywidowpenalty 50
223 % \predisplaypenalty 10000
224 % \postdisplaypenalty 0
```

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

```
225 % \interlinepenalty 0
```

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

```
226 % \brokenpenalty 0
227 </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` `\headsep` is the distance between the bottom of the running head and the top of
`\topskip` the text. `\topskip` is the `\baselineskip` for the first line on a page.

```
228 <*10pt | 11pt | 12pt>
229 \setlength\headheight{12\p@}
230 \setlength\headsep    {25\p@}
231 <10pt>\setlength\topskip    {10\p@}
232 <11pt>\setlength\topskip    {11\p@}
233 <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:

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

`\maxdepth` The \TeX primitive register `\maxdepth` has a function that is similar to that of `\topskip`. The register `\@maxdepth` should always contain a copy of `\maxdepth`. In both plain \TeX and \LaTeX 2.09 `\maxdepth` had a fixed value of `4pt`; in native \LaTeX 2e mode we let the value depend on the typesize. We set it so that `\maxdepth + \topskip = typesize \times 1.5`. As it happens, in these classes `\topskip` is equal to the typesize, therefore we set `\maxdepth` to half the value of `\topskip`.

```
235 \if@compatibility
236   \setlength\maxdepth{4\p@}
237 \else
238   \setlength\maxdepth{.5\topskip}
239 \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.

```
240 \if@compatibility
241   \if@twocolumn
242     \setlength\textwidth{410\p@}
243   \else
244 <10pt>   \setlength\textwidth{345\p@}
245 <11pt>   \setlength\textwidth{360\p@}
246 <12pt>   \setlength\textwidth{390\p@}
247 \fi
```

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

```
248 \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.

```
249 \setlength\@tempdima{\paperwidth}
250 \addtolength\@tempdima{-2in}
251 <10pt> \setlength\@tempdimb{361\p@}
252 <11pt> \setlength\@tempdimb{376\p@}
253 <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).

```
254 \if@twocolumn
255   \ifdim\@tempdima>2\@tempdimb\relax
256     \setlength\textwidth{2\@tempdimb}
257   \else
258     \setlength\textwidth{\@tempdima}
259   \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.

```
260 \else
261   \ifdim\@tempdima>\@tempdimb\relax
262     \setlength\textwidth{\@tempdimb}
263   \else
264     \setlength\textwidth{\@tempdima}
265   \fi
266 \fi
267 \fi
```

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

```
268 \if@compatibility
269 \else
270 \settopoint\textwidth
271 \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^AT_EX 2.09. 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.

```
272 \if@compatibility
273 <10pt> \setlength\textheight{43\baselineskip}
274 <11pt> \setlength\textheight{38\baselineskip}
275 <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.

```

276 \else
277 \setlength\@tempdima{\paperheight}
    We leave at least a 1 inch margin on the top and the bottom of the page.
278 \addtolength\@tempdima{-2in}
    We also have to leave room for the running headers and footers.
279 \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.
280 \divide\@tempdima\baselineskip
281 \@tempcnta=\@tempdima
    From this we can calculate the height of the text.
282 \setlength\textheight{\@tempcnta\baselineskip}
283 \fi
    The first line on the page has a height of \topskip.
284 \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
`\marginparpush` `\marginparsep`, the minimum vertical separation between two marginal notes is controlled by `\marginparpush`.

```

285 \iftwocolumn
286 \setlength\marginparsep {10\p@}
287 \else
288 <10pt> \setlength\marginparsep{11\p@}
289 <11pt> \setlength\marginparsep{10\p@}
290 <12pt> \setlength\marginparsep{10\p@}
291 \fi
292 <10pt | 11pt> \setlength\marginparpush{5\p@}
293 <12pt> \setlength\marginparpush{7\p@}

```

Now we can give the values for the other margin parameters. For native L^AT_EX 2_ε, these are calculated.

`\oddsidemargin` First we give the values for the compatibility mode.
`\evensidemargin` Values for two-sided printing:
`\marginparwidth` 294 `\ifcompatibility`
 295 `\iftwoside`
 296 `<10pt> \setlength\oddsidemargin {44\p@}`


```

297 <11pt> \setlength\oddsidemargin {36\p@}
298 <12pt> \setlength\oddsidemargin {21\p@}
299 <10pt> \setlength\evensidemargin {82\p@}
300 <11pt> \setlength\evensidemargin {74\p@}
301 <12pt> \setlength\evensidemargin {59\p@}
302 <10pt> \setlength\marginparwidth {107\p@}
303 <11pt> \setlength\marginparwidth {100\p@}
304 <12pt> \setlength\marginparwidth {85\p@}

```

Values for one-sided printing:

```

305 \else
306 <10pt> \setlength\oddsidemargin {63\p@}
307 <11pt> \setlength\oddsidemargin {54\p@}
308 <12pt> \setlength\oddsidemargin {39.5\p@}
309 <10pt> \setlength\evensidemargin {63\p@}
310 <11pt> \setlength\evensidemargin {54\p@}
311 <12pt> \setlength\evensidemargin {39.5\p@}
312 <10pt> \setlength\marginparwidth {90\p@}
313 <11pt> \setlength\marginparwidth {83\p@}
314 <12pt> \setlength\marginparwidth {68\p@}
315 \fi

```

And values for two column mode:

```

316 \if@twocolumn
317 \setlength\oddsidemargin {30\p@}
318 \setlength\evensidemargin {30\p@}
319 \setlength\marginparwidth {48\p@}
320 \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.

```

321 \else
322 \if@twoside
323 \setlength@tempdima {\paperwidth}
324 \addtolength@tempdima {-\textwidth}
325 \setlength\oddsidemargin {.4@tempdima}
326 \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!

```

327 \setlength\marginparwidth {.6@tempdima}
328 \addtolength\marginparwidth {-\marginparsep}
329 \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 than used for the

margin (thus `\oddsidemargin` is 1in less).

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

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

```
339 \ifdim \marginparwidth >2in
340   \setlength\marginparwidth{2in}
341 \fi
```

Having done these calculations we make them pt values.

```
342 \@settopoint\oddsidemargin
343 \@settopoint\marginparwidth
```

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

```
344 \setlength\evensidemargin {\paperwidth}
345 \addtolength\evensidemargin{-2in}
346 \addtolength\evensidemargin{-\textwidth}
347 \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 todays technology can accurately print.

```
348 \@settopoint\evensidemargin
349 \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.

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

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

```
359 \addtolength\topmargin{-.5\topmargin}
360 \@settopoint\topmargin
361 \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.

```
362 <10pt>\setlength\footnotesep{6.65\p@}
363 <11pt>\setlength\footnotesep{7.7\p@}
364 <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.

```
365 <10pt>\setlength\skip\footins{9\p@ \@plus 4\p@ \@minus 2\p@}
366 <11pt>\setlength\skip\footins{10\p@ \@plus 4\p@ \@minus 2\p@}
367 <12pt>\setlength\skip\footins{10.8\p@ \@plus 4\p@ \@minus 2\p@}
368 </10pt | 11pt | 12pt>
```

7.3.5 Float placement parameters

All float parameters are given default values in the L^AT_EX 2_ε 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.

```
369 <*artikel | rapport | boek>
370 \setcounter{topnumber}{2}
```

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

```
371 \renewcommand\topfraction{.7}
```

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

```
372 \setcounter{bottomnumber}{1}
```

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

```
373 \renewcommand\bottomfraction{.3}
```

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

```
374 \setcounter{totalnumber}{3}
```

`\textfraction` This indicates the minimum part of a text page that has to be occupied by text.

```
375 \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.

```
376 \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.
377 `\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.
378 `\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.
379 `\renewcommand\dblfloatpagefraction{.5}`
380 `\(/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
`\textfloatsep` are used for both one-column mode and single-column floats in two-column mode.
`\intextsep`

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

`\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.

```

381 <*10pt>
382 \setlength\floatsep    {12\p@ \@plus 2\p@ \@minus 2\p@}
383 \setlength\textfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
384 \setlength\intextsep   {12\p@ \@plus 2\p@ \@minus 2\p@}
385 </10pt>
386 <*11pt>
387 \setlength\floatsep    {12\p@ \@plus 2\p@ \@minus 2\p@}
388 \setlength\textfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
389 \setlength\intextsep   {12\p@ \@plus 2\p@ \@minus 2\p@}
390 </11pt>
391 <*12pt>
392 \setlength\floatsep    {12\p@ \@plus 2\p@ \@minus 4\p@}
393 \setlength\textfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
394 \setlength\intextsep   {14\p@ \@plus 4\p@ \@minus 4\p@}
395 </12pt>

```

`\dblfloatsep` When floating objects that span the whole `\textwidth` are placed on a text page
`\dbltextfloatsep` 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.

```

396 <*10pt>

```

```

397 \setlength\dblfloatsep {12\p@ \@plus 2\p@ \@minus 2\p@}
398 \setlength\dbltextfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
399 </10pt>
400 <*11pt>
401 \setlength\dblfloatsep {12\p@ \@plus 2\p@ \@minus 2\p@}
402 \setlength\dbltextfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
403 </11pt>
404 <*12pt>
405 \setlength\dblfloatsep {14\p@ \@plus 2\p@ \@minus 4\p@}
406 \setlength\dbltextfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
407 </12pt>

```

Floats on their own page or column

`\@fptop` 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.

```

408 <*10pt>
409 \setlength\@fptop{0\p@ \@plus 1fil}
410 \setlength\@fpsep{8\p@ \@plus 2fil}
411 \setlength\@fpbot{0\p@ \@plus 1fil}
412 </10pt>
413 <*11pt>
414 \setlength\@fptop{0\p@ \@plus 1fil}
415 \setlength\@fpsep{8\p@ \@plus 2fil}
416 \setlength\@fpbot{0\p@ \@plus 1fil}
417 </11pt>
418 <*12pt>
419 \setlength\@fptop{0\p@ \@plus 1fil}
420 \setlength\@fpsep{10\p@ \@plus 2fil}
421 \setlength\@fpbot{0\p@ \@plus 1fil}
422 </12pt>

```

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

```

\@dblfpsep 423 <*10pt>
\@dblfpbot 424 \setlength\@dblftop{0\p@ \@plus 1fil}
425 \setlength\@dblfpsep{8\p@ \@plus 2fil}
426 \setlength\@dblfpbot{0\p@ \@plus 1fil}
427 </10pt>
428 <*11pt>
429 \setlength\@dblftop{0\p@ \@plus 1fil}
430 \setlength\@dblfpsep{8\p@ \@plus 2fil}
431 \setlength\@dblfpbot{0\p@ \@plus 1fil}
432 </11pt>

```

```

433 <*12pt>
434 \setlength\@dblftop{0\p@ \@plus 1fil}
435 \setlength\@dblfpsep{10\p@ \@plus 2fil}
436 \setlength\@dblfpbot{0\p@ \@plus 1fil}
437 </12pt>
438 <*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.

```
439 \let\Thispagestyle\thispagestyle
```

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

```

440 \newcommand*\@emptypagestyle{empty}
441 \renewcommand*\pagestyle[1]{\@nameuse{ps@#1}\def\@currentpagestyle{#1}}
442 \renewcommand*\thispagestyle[1]{%
443   \ifx\@currentpagestyle\@emptypagestyle
444     \else
445       \global\@specialpagetrue
446       \gdef\@specialstyle{#1}%
447   \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^AT_EX extends T_EX’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 T_EX’s `\botmark` command.
`\rightmark`: Used in the `\@oddhead`, `\@oddfoot`, `\@evenhead` or `\@evenfoot` macros, it gets the current ‘right’ mark. `\rightmark` works like T_EX’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`

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

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

```
449   \let\@oddhead\@empty\let\@evenhead\@empty
450   \def\@oddfoot{\hfil\PageFont\thepage}%
451   \if@twoside
452     \def\@evenfoot{\PageFont\thepage\hfil}%
453   \else
454     \let\@evenfoot\@oddfoot
455   \fi
```

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

```
456   \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.

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

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

459     \let\@oddfoot\@empty\let\@evenfoot\@empty
460     \def\@evenhead{\PageFont\thepage}\hfil\MarkFont\leftmark}%
461     \def\@oddhead{\PageFont\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`.

```
462 \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.

```
463 <*artikel>
464 \def\sectionmark##1{%
465 \markboth {\MakeUppercase{%
466 \ifnum \c@secnumdepth >\z@
467 \thesection\quad
468 \fi
469 ##1}}{}}%
470 \def\subsectionmark##1{%
471 \markright {%
472 \ifnum \c@secnumdepth >\@ne
473 \thesubsection\quad
474 \fi
475 ##1}}
476 </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.

```
477 <*rapport | boek>
478 \def\chaptermark##1{%
479 \markboth {\MakeUppercase{\ifnum \c@secnumdepth >\m@ne
480 <boek> \if@mainmatter
481 \@chapapp\ \thechapter. \ %
482 <boek> \fi
483 \fi
484 ##1}}{}}%
485 \def\sectionmark##1{%
486 \markright {\MakeUppercase{\ifnum \c@secnumdepth >\z@
487 \thesection. \ \fi
488 ##1}}}}
489 </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...`

```
490 \else
491 \def\ps@headings{%
492 \let\@oddfoot\@empty
```



```

493 \def\@oddhead{{\MarkFont\rightmark}\hfil\PageFont\thepage}%
494 \let\@mkboth\markboth

```

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

```

495 <*artikel>
496 \def\sectionmark##1{%
497 \markright {\MakeUppercase{%
498 \ifnum \c@secnumdepth >\m@ne
499 \thesection\quad
500 \fi
501 ##1}}}}
502 </artikel>

503 <*rapport | boek>
504 \def\chaptermark##1{%
505 \markright {\MakeUppercase{%
506 \ifnum \c@secnumdepth >\m@ne
507 <boek> \if@mainmatter
508 \@chapapp\ \thechapter. \ %
509 <boek> \fi
510 \fi
511 ##1}}}}
512 </rapport | boek>
513 \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.

```

514 \def\ps@myheadings{%
515 \let\@oddfoot\@empty\let\@evenfoot\@empty
516 \def\@evenhead{{\PageFont\thepage}\hfil\MarkFont\leftmark}%
517 \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 `\letting` them to a macro that gobbles its argument(s).

```

518 \let\@mkboth\@gobbletwo
519 <!artikel> \let\chaptermark@gobble
520 \let\sectionmark@gobble
521 <artikel> \let\subsectionmark@gobble
522 }

```

`\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.

```

523 \newcommand*\PageFont{\rmfamily}
524 \newcommand*\MarkFont{\slshape}

```

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

```

525 \newcommand*\RunningFonts[2]{%
526 \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
`\author` title, author(s) and date of the document. The information is stored away in
`\date` 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.

```
527 % \newcommand*\title[1]{\gdef\@title{#1}}
528 % \newcommand*\author[1]{\gdef\@author{#1}}
529 % \newcommand*\date[1]{\gdef\@date{#1}}
```

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

```
530 % \gdef\@date{\today}
```

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

```
531 \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.

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

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

```
539   \long\def\@makefnmark##1{\parindent\z@
540     \def\labelitemi{\textendash}\@revlabeltrue
541     \leavevmode\@textsuperscript{\@thefnmark}\kern1em\relax ##1}
542   \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^AT_EX 2.09 style files

```
543   \if@compatibility\setcounter{page}{0}\fi
544   \null\vfil
545   \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.

```

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

```

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

```

572 \@thanks
573 \vfil\null
574 \end{titlepage}%

```

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

```

575 \setcounter{footnote}{0}%
576 \global\let\thanks\relax
577 \global\let\maketitle\relax
578 \global\let\@thanks\@empty
579 \global\let\@author\@empty
580 \global\let\@title\@empty
581 \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.

```

582 \global\let\title\relax

```

```

583 \global\let\author\relax
584 \global\let\date\relax
585 \global\let\and\relax
586 }

```

We want to have the title, author and date information in uppercase, but we have to be very carefull 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 th 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.

```

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

```

\uc@authornothanks A document can have more than one author. Usually they are seperated 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 than 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`.

```

588 \def\uc@authornothanks#1\and{\uc@nothanks#1\thanks\relax}

```

Then we have to check whether the `\and` we ound 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` contstruct stores the first token *after* the `\and` in `\@tempa` to be inspected by `\u@ax`.

```

589 \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_EX see the `\fi` first.

```

590 \def\uc@ax{%
591 \ifx\@tempa\relax
592 \else
593 \\ \andname \\ \expandafter\uc@authornothanks
594 \fi}

```

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

```

595 \def\uc@tx{\ifx\@tempa\relax
596 \else \expandafter\u@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`.

```
597 \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.

```
598 <!*boek>
599 \else
600 \newcommand*\maketitle{\par
601   \begingroup
602     \renewcommand*\thefootnote{\@fnsymbol\c@footnote}%
603 <!type2>   \def\@makefnmark{\rlap{%
604 <!type2>     \@textsuperscript{\normalfont\@thefnmark}}}%
605 <!type2>   \long\def\@makefntext{\@xmakefntext{%
606 <!type2>     \@textsuperscript{\normalfont\@thefnmark}}}%
607 <*type2>

608   \long\def\@makefntext##1{\parindent\z@
609     \def\labelitemi{\textendash}%
610     \leavevmode\hb@xt@.5\unitindent{%
611       \@textsuperscript{\normalfont\@thefnmark}\hfil}##1}
612 </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`.

```
613   \iftwocolumn
614     \ifnum \col@number=\@ne
615       \@maketitle
616     \else
617       \twocolumn[\@maketitle]%
618     \fi
619   \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.

```
620     \newpage
621     \global\@topnum\z@
622     \@maketitle
623     \fi
```

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

```
624   \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.

```

625 \endgroup
626 \setcounter{footnote}{0}%
627 \global\let\thanks\relax
628 \global\let\maketitle\relax
629 \global\let\@maketitle\relax
630 \global\let\@thanks\@empty
631 \global\let\@author\@empty
632 \global\let\@title\@empty
633 \global\let\@date\@empty
634 \global\let\title\relax
635 \global\let\author\relax
636 \global\let\date\relax
637 \global\let\and\relax
638 }

```

`\@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.

```

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

```

```

666 </type3>
667 \par
668 \vskip 1.5em}
669 \fi
670 </!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><beforeskip><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>` \leq 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^AT_EX 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.

```
671 \newcommand*\head@style{%
672   \interlinepenalty \@M
673   \hyphenpenalty=\@M \exhyphenpenalty=\@M
674   \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`.

```
675 \def\@sect#1#2#3#4#5#6[#7]#8{%
676   \ifnum #2>\c@secnumdepth
677     \let\@svsec\@empty
678   \else
679     \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.

```
680 ⟨*type1 | type3⟩
681   \begingroup
682     \setbox\@tempboxa=\hbox{#6\relax
```



```

683             \csname the#1\endcsname
684             \hskip.5em}
685     \ifdim\wd\@tempboxa>\unitindent
686         \global\unitindent=\wd\@tempboxa
687         \@indentset
688     \fi
689     \endgroup
690 \</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.

```

691     \protected@edef\@svsec{\@secntformat{#1}\relax}%
692     \fi
693     \@tempskipa #5\relax
694     \ifdim \@tempskipa>\z@
695         \beginingroup

```

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.

```

696     #6{%
697 <*type1 | type3>
698     \@hangfrom{\hskip #3\relax\@svsec}\head@style #8\endgraf}%
699 </type1 | type3>
700 <*type2>
701     \@hangfrom{\hskip #3}
702         \head@style\@svsec \hskip.3em\relax #8\endgraf}
703 </type2>
704     \endgroup
705     \csname #1mark\endcsname{#7}%
706     \addcontentsline{toc}{#1}{%
707         \ifnum #2>\c@secnumdepth
708         \else
709             \protect\numberline{\csname the#1\endcsname}%
710         \fi
711         #7}%
712     \else
713         \def\@svsechd{#6\hskip #3\relax
714             \@svsec #8\csname #1mark\endcsname{#7}%
715             \addcontentsline{toc}{#1}{%
716                 \ifnum #2>\c@secnumdepth
717                 \else
718                     \protect\numberline{\csname the#1\endcsname}%
719                 \fi
720                 #7}}%
721     \fi
722     \@xsect{#5}}

```

This macro was introduced in L^AT_EX 2_ε, its definition is changed here to get the fixed with of the section number.

```

723 \def\@secntformat#1{%

```

```

724 <!type2> \hb@xt@<unitindent>{\csname the#1\endcsname \hfil}%
725 <type2> \csname the#1\endcsname\hskip.3em\relax
726 }

```

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

```

727 \def\@ssect#1#2#3#4#5{\@tempskipa #3\relax
728 \ifdim \@tempskipa>\z@
729 \begingroup

```

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

```

730 #4{%
731 \@changfrom{\hskip #1}\head@style #5\endgraf}%
732 \endgroup
733 \else
734 \def\@svsechd{#4\hskip #1\relax #5}%
735 \fi
736 \@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 737 <!artikel>\newcommand*\chaptermark[1]{ }
\paragraphmark     738 % \newcommand*\sectionmark[1]{ }
\subparagraphmark  739 % \newcommand*\subsectionmark[1]{ }
                    740 % \newcommand*\subsubsectionmark[1]{ }
                    741 % \newcommand*\paragraphmark[1]{ }
                    742 % \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.

```

743 <artikel>\setcounter{secnumdepth}{3}
744 <!artikel>\setcounter{secnumdepth}{2}

```

`\c@part` These counters are used for the section numbers. The macro `\newcounter{<newctr>}[<oldctr>]` defines `<newctr>` to be a counter, which is reset to zero when counter `<oldctr>` is stepped. Counter `<oldctr>` must already be defined.

```

\c@subsection 745 \newcounter {part}
\c@paragraph  746 <artikel>\newcounter {section}
\c@subparagraph 747 <*rapport | boek>
                    748 \newcounter {chapter}
                    749 \newcounter {section}[chapter]

```

```

750 </rapport | boek>
751 \newcounter {subsection}[section]
752 \newcounter {subsubsection}[subsection]
753 \newcounter {paragraph}[subsubsection]
754 \newcounter {subparagraph}[paragraph]

```

`\thepart` For any counter *CTR*, `\theCTR` is a macro that defines the printed version of counter *CTR*. It is defined in terms of the following macros:

`\thechapter` `\arabic{COUNTER}` prints the value of *COUNTER* as an arabic numeral.

`\thesection` `\roman{COUNTER}` prints the value of *COUNTER* as a lowercase roman numeral.

`\thesubsection` `\Roman{COUNTER}` prints the value of *COUNTER* as an uppercase roman numeral.

`\thesubsubsubsection` `\alph{COUNTER}` prints the value of *COUNTER* as a lowercase letter: 1 = a, 2 = b, etc.

`\theparagraph` `\Alph{COUNTER}` prints the value of *COUNTER* as an uppercase letter: 1 = A, 2 = B, etc.

`\thesubparagraph` Actually to save space the internal counter representations and the commands operating on those are used.

```

755 \renewcommand*\thepart{\@Roman\c@part}
756 <artikel>\renewcommand\thesection{\@arabic\c@section}
757 <*rapport | boek>
758 \renewcommand*\thechapter{\@arabic\c@chapter}
759 \renewcommand*\thesection{\thechapter.\@arabic\c@section}
760 </rapport | boek>
761 \renewcommand*\thesubsection{\thesection.\@arabic\c@subsection}
762 \renewcommand*\thesubsubsubsection{\thesubsection.\@arabic\c@subsubsection}
763 \renewcommand*\theparagraph{\thesubsubsubsection.\@arabic\c@paragraph}
764 \renewcommand*\thesubparagraph{\theparagraph.\@arabic\c@subparagraph}

```

`\@chapapp` `\@chapapp` is initially defined to be `\chaptername`. The `\appendix` command redefines it to be `\appendixname`.

```

765 <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.

```

766 <*boek>
767 \newcommand*\frontmatter{%
768   \cleardoublepage
769   \@mainmatterfalse
770   \pagenumbering{roman}}

```

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

```
771 \newcommand*\mainmatter{%
772   \cleardoublepage
773   \@mainmattertrue
774   \pagenumbering{arabic}}
```

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

```
775 \newcommand*\backmatter{%
776   \if@openright\cleardoublepage\else\clearpage\fi
777   \@mainmatterfalse}
778 \</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`.

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

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

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

```
788 <*rapport | boek>
789 \newcommand*\part{%
790   \cleardoublepage
791   \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.

```
792 \if@twocolumn
793   \onecolumn
794   \@tempswatrue
795 \else
796   \@tempswafalse
797 \fi
```

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

```
798 \null\vfil
```

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

```
799 \secdef\@part\@spart}
800 </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.

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

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

```
802 <*artikel>
803 \def\@part[#1]#2{%
804     \ifnum \c@secnumdepth >\m@ne
805         \refstepcounter{part}%
806         \addcontentsline{toc}{part}{\protect\numberline{thepart}#1}%
807     \else
808         \addcontentsline{toc}{part}{#1}%
809     \fi
```

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

```
810     {\head@style
811      \parindent\unitindent
812      \normalfont
```

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

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

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

```
820     \markboth{}{\par}%
821     \nobreak
822     \vskip 3ex
823     \@afterheading}
824 </artikel>
```

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

```
825 <*rapport | boek>
```

```

826 \def\@part[#1]#2{%
827     \ifnum \c@secnumdepth >-2\relax
828         \refstepcounter{part}%
829         \addcontentsline{toc}{part}{\protect\numberline{\thepart}{#1}}%
830     \else
831         \addcontentsline{toc}{part}{\toc@case{#1}}%
832     \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.

```

833     \markboth{}{}%
834     {\centering
835     \interlinepenalty \@M
836     \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 `\MakeUppercase` on the contents of `\partname`.

```

837     \ifnum \c@secnumdepth >-2\relax
838         \Large\PartFont
839         \edef\@tempa{\noexpand\MakeUppercase{\partname}}\@tempa
840         \nobreakspace\thepart
841     \par

```

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

```

842     \vskip 20\p@
843     \fi
844     \Large \PartFont \MakeUppercase{#2}\par}%
845     \@endpart}
846 </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`.

```

847 <*artikel>
848 \def\@spart#1{%
849     {\parindent \z@
850     \head@style
851     \normalfont
852 <!type2>     \Large \PartFont \noindent #1\par}%
853 <type2>     \Large \PartFont \indent #1\par}%
854     \nobreak
855     \vskip 3ex
856     \@afterheading}
857 </artikel>

```

```

858 <*rapport | boek>
859 \def\@spart#1{%
860     {\centering
861       \interlinepenalty \@M
862       \normalfont
863       \Large \PartFont #1\par}%
864     \@endpart}

```

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

```

865 \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.
866 (!boek)           \if@twoside
867                   \if@openright
868                   \null
869                   \thispagestyle{empty}%
870                   \newpage
871                   \fi
872 (!boek)           \fi
  When this was a two column document we have to switch back to two column
  mode.
873                   \if@tempwa
874                   \twocolumn
875                   \fi}
876 </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*.

```

877 <*rapport | boek>
878 \newcommand*\chapter{\if@openright\cleardoublepage\else\clearpage\fi
879                   \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.

```

880                   \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.

```

881                   \@afterindentfalse
882                   \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.

```

883 \def\@chapter[#1]#2{%
884     \ifnum \c@secnumdepth >\m@ne
885 <boek>     \if@mainmatter
886             \refstepcounter{chapter}%
887             \typeout{\@chapapp\space\thechapter.}%
888             \addcontentsline{toc}{chapter}%
889                 {\protect\numberline{\thechapter}#1}%
890 <*boek>
891     \else
892         \addcontentsline{toc}{chapter}{#1}%
893     \fi
894 </boek>
895     \else
896         \addcontentsline{toc}{chapter}{#1}%
897     \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.

```

898     \chaptermark{#1}%
899     \addtocontents{lof}{\protect\addvspace{10\p@}}%
900     \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.

```

901     \if@twocolumn
902         \@topnewpage[\@makechapterhead{#2}]%
903     \else
904         \@makechapterhead{#2}%
905         \@afterheading
906     \fi}

```

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

```

907 \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.

```

908 \def\@makechapterhead#1{%
909 <!boek> \vspace*{50\p@ \@plus 5\p@}%
910 <boek> \vspace*{50\p@ \@plus 20\p@}%
911 {\setlength\parindent{\z@}%
912 \setlength\parskip {\z@}%
913 \head@style \normalfont

```


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

```

914     \ifnum \c@secnumdepth >\m@ne
915 <boek>     \if@mainmatter
916           \Large\ChapFont \@chapapp{} \thechapter
917           \par\nobreak
918           \vskip 20\p@
919 <boek>     \fi
920     \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.

```

921     \Large \ChapFont #1\par
922     \nobreak
923     \vskip 40\p@
924   }}

```

`\@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.

```

925 \def\@schapter#1{\if@twocolumn
926           \@topnewpage[\@makeschapterhead{#1}]%
927           \else
928           \@makeschapterhead{#1}%
929           \@afterheading
930           \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.

```

931 \def\@makeschapterhead#1{%
932 <!boek> \vspace*{50\p@\@plus 5\p@}%
933 <boek> \vspace*{50\p@\@plus 20\p@}%
934 {\setlength\parindent{\z@}%
935  \setlength\parskip{\z@}%
936  \head@style
937  \normalfont
938  \Large \ChapFont #1\par
939  \nobreak
940  \vskip 40\p@
941  }}
942 </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.

```

943 \newcommand*\section{%
944 <*type1 | type3>
945   \@startsection {section}{1}{\z@}%
946     {-2\baselineskip\@plus -1\baselineskip \@minus -.5\baselineskip}%
947 </type1 | type3>
948 <*type2>
949   \@startsection {section}{1}{\unitindent}%
950     {2\baselineskip\@plus \baselineskip \@minus .5\baselineskip}%
951 </type2>
952 <type1>   {.5\baselineskip}%
953 <type2 | type3>   {.01\baselineskip}%
954   {\normalfont\large\SectFont}}

```

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

```

955 \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.

```

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

```

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

```

968 \newcommand*\SSectFont{\bfseries}

```

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

```

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

```

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

```

981 <artikel & (type1 | type3)>\newcommand*\SSSectFont{\rmfamily}
982 <type2>\newcommand*\SSSectFont{\scshape}
983 <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`.

```

984 \newcommand*\paragraph{%
985 <!type2> \@startsection{paragraph}{4}{\z@}%
986 <type2> \@startsection{paragraph}{4}{\unitindent}%
987 {3.25ex \@plus1ex \@minus.2ex}%
988 {-1em}%
989 {\normalfont\normalsize\ParaFont}}

```

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

```

990 <!type2>\newcommand*\ParaFont{\slshape}
991 <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`.

```

992 \newcommand*\subparagraph{%
993 <!type2> \@startsection{subparagraph}{5}{\parindent}%
994 <type2> \@startsection{subparagraph}{5}{\unitindent}%
995 {3.25ex \@plus1ex \@minus .2ex}%
996 {-1em}%
997 {\normalfont\normalsize\SParaFont}}

```

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

```

998 \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.

```

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

```

```

1016 \renewcommand*\SSsectFont{#6}%
1017 \renewcommand*\ParaFont{#7}%
1018 \renewcommand*\SParaFont{#8}
1019 </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^AT_EX manual for an explanation of the meanings of the parameters. Defaults for the list environment are set as follows. First, `\rightmargin`, `\listparindent` and `\itemindent` are set to 0pt. Then, for a Kth level list, the command `\@listK` is called, where 'K' denotes 'i', 'ii', ... , 'vi'. (I.e., `\@listiii` is called for a third-level list.) By convention, `\@listK` should set `\leftmargin` to `\leftmarginK`.

```

\leftmargin For efficiency, level-one list's values are defined at top level, and \@listi is defined
\leftmargini to set only \leftmargin.
\leftmarginii 1020 <!type2>\setlength\leftmargini {\unitindent}
\leftmarginiii 1021 <type2>\setlength\leftmargini {\othermargin}
\leftmarginiv 1022 \setlength\leftmarginii {\othermargin}
\leftmarginv 1023 \setlength\leftmarginiii{\othermargin}
\leftmarginvi 1024 \setlength\leftmarginiv {\othermargin}
1025 \setlength\leftmarginv {\othermargin}
1026 \setlength\leftmarginvi {1em}

```

Here we set the top level leftmargin.

```

1027 \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.

```

```

1028 \setlength \labelsep {5\p@}
1029 \setlength \labelwidth{\leftmargini}
1030 \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.

```

```

1031 \setlength\partopsep{\z@}

```

```

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

```

```

1032 \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.

```
1033 \@beginparpenalty -\@lowpenalty
1034 \@endparpenalty -\@lowpenalty
1035 \@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.

```
1036 \def\@listi{%
1037 <!type2> \leftmargin\unitindent
1038 <type2> \leftmargin\leftmargini
1039 <!type2> \labelsep.5em%
1040 <type2> \labelsep.45em%
1041 \labelwidth\leftmargin
1042 \advance\labelwidth-\labelsep
1043 \parsep \z@
1044 <!type3> \topsep 0\p@ \@plus\p@
1045 <type3> \topsep -.5\parskip \@plus\p@
1046 \itemsep 0\p@ \@plus1\p@}
1047 \let\@listI\@listi
```

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

```
1048 \@listi
```

`\@listii` Here are the same macros for the higher level lists. Note that they don't have saved versions and are not modified by the font size commands. In other words
`\@listiii` this class assumes that nested lists only appear in `\normalsize`, i.e. the main
`\@listiv` document size.
`\@listv`

```
\@listvi 1049 \def\@listii {\leftmargin\leftmarginii
1050 <!type2> \labelsep .5em%
1051 <type2> \labelsep .3em%
1052 \labelwidth\leftmarginii
1053 \advance\labelwidth-\labelsep
1054 <!type3> \topsep 0\p@ \@plus\p@
1055 <type3> \topsep -.5\parskip \@plus\p@
1056 \parsep \z@
1057 \itemsep \z@ \@plus\p@}
1058 \def\@listiii{\leftmargin\leftmarginiii
1059 <!type2> \labelsep .5em%
1060 <type2> \labelsep .3em%
1061 \labelwidth\leftmarginiii
1062 \advance\labelwidth-\labelsep
1063 <!type3> \topsep 0\p@ \@plus\p@
1064 <type3> \topsep -.5\parskip \@plus\p@
1065 \parsep \z@
1066 \partopsep \z@ \@plus\p@
1067 \itemsep \z@ \@plus\p@}
```

```

1068 \def\@listiv {\leftmargin\leftmarginiv
1069 <!type2>          \labelsep .5em%
1070 <type2>          \labelsep .3em%
1071                  \labelwidth\leftmarginiv%
1072                  \advance\labelwidth-\labelsep
1073 <!type3>          \topsep 0\p@ \@plus\p@
1074 <type3>          \topsep -.5\parskip\@plus\p@
1075                  \parsep \z@
1076                  \itemsep \z@ \@plus\p@}
1077 \def\@listv {\leftmargin\leftmarginv
1078 <!type2>          \labelsep .5em%
1079 <type2>          \labelsep .3em%
1080                  \labelwidth\leftmarginv
1081                  \advance\labelwidth-\labelsep%
1082 <!type3>          \topsep 0\p@ \@plus\p@
1083 <type3>          \topsep -.5\parskip\@plus\p@
1084                  \parsep \z@
1085                  \itemsep \z@ \@plus\p@}
1086 \def\@listvi {\leftmargin\leftmarginvi
1087 <!type2>          \labelsep .5em
1088 <type2>          \labelsep .3em
1089                  \labelwidth\leftmarginvi
1090                  \advance\labelwidth{-\labelsep}%
1091 <!type3>          \topsep 0\p@ \@plus\p@
1092 <type3>          \topsep -.5\parskip\@plus\p@
1093                  \parsep \z@
1094                  \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 1095 \renewcommand*\theenumi{\@arabic\c@enumi}
\theenumiv 1096 \renewcommand*\theenumii{\@alph\c@enumii}
1097 \renewcommand*\theenumiii{\@roman\c@enumiii}
1098 \renewcommand*\theenumiv{\@Alph\c@enumiv}

\labelenumi The label for each item is generated by the commands
\labelenumii \labelenumi ... \labelenumiv.
\labelenumiii 1099 \newcommand*\labelenumi{\theenumi.}
\labelenumiv 1100 \newcommand*\labelenumii{(\theenumii)}
1101 \newcommand*\labelenumiii{\theenumiii.}
1102 \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

```

```

1103 \renewcommand*\p@enumii{\theenumi}
1104 \renewcommand*\p@enumiii{\theenumi(\theenumii)}
1105 \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.

```

1106 \renewenvironment{enumerate}{%
1107   \ifnum \@enumdepth >\thr@@
1108     \@toodeep
1109   \else
1110     \advance\@enumdepth \@ne
1111     \edef\@enumctr{enum\romannumeral\the\@enumdepth}%
1112     \expandafter
1113     \list
1114       \csname label\@enumctr\endcsname
1115         {\usecounter{\@enumctr}}%
1116   <type1>           \ifnum \@listdepth=1
1117   <*type1 | type3>
1118     \if@revlabel
1119       \def\makelabel##1{\hskip .5\unitindent{##1\hfil}}%
1120     \else
1121   <!type3>           \def\makelabel##1{\hfil##1}
1122   <type3>           \def\makelabel##1{##1\hfil}
1123     \fi
1124   </type1 | type3>
1125   <type1>           \else
1126   <type1 | type2>   \def\makelabel##1{##1\hfil}%
1127   <type1>           \fi
1128   }%
1129   \fi}

```

We try to suppress spaces after these list constructs.

```

1130 {\global\@ignoretrue \endlist}

```

8.3.3 Itemize

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

```

1131 \newcommand\labelitemi {\labelitemfont \textbullet}
1132 \newcommand\labelitemii {\labelitemfont \bfseries \textendash}
1133 \newcommand\labelitemiii {\labelitemfont \textasteriskcentered}
1134 \newcommand\labelitemiv {\labelitemfont \textperiodcentered}

```

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

A possible alternative would be

```

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

```

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

```
1135 \newcommand\labelitemfont{\normalfont}
```

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

```

1136 \renewenvironment{itemize}{%
1137   \ifnum \@itemdepth >\thr@@
1138     \@toodeep
1139   \else
1140     \advance\@itemdepth \@ne
1141     \edef\@itemitem{labelitem\romannumeral\the\@itemdepth}%
1142     \expandafter
1143     \list
1144       \csname\@itemitem\endcsname
1145       {%
1146 <type1>           \ifnum \@listdepth=1\relax
1147 <*type1 | type3>
1148               \if@revlabel
1149                 \def\makelabel##1{\hskip .5\unitindent{##1\hfil}}\else
1150 <type1>           \def\makelabel##1{\hfil##1}
1151 <type3>           \def\makelabel##1{##1\hfil}
1152               \fi
1153 /<type1 | type3>
1154 <type1>           \else
1155 <type1 | type2>   \def\makelabel##1{##1\hfil}
1156 <type1>           \fi
1157               }%
1158 \fi}

```

We try to suppress spaces after these list constructs.

```
1159 {\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`.

```

1160 \newenvironment{description}
1161   {\list{}{\labelwidth\z@ \itemindent-\leftmargin
1162           \let\makelabel\descriptionlabel}}
1163   {\endlist}

```

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

```

1164 \newcommand*\descriptionlabel[1]{\hspace\labelsep
1165                               \normalfont\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 little whitespace around them in the document classes `artikel1` and `rapport1`.

center Add a little surrounding whitespace.

```
1166 <*type1>
1167 \def\center
1168   {\topsep=.25\baselineskip \@plus .1\baselineskip
1169     \@minus .1\baselineskip
1170     \trivlist \centering\item[]\}
1171 \let\endcenter\endtrivlist
```

flushleft Add a little surrounding whitespace.

```
1172 \def\flushleft
1173   {\topsep=.25\baselineskip \@plus .1\baselineskip
1174     \@minus .1\baselineskip
1175     \trivlist \raggedright\item[]\}
1176 \let\endflushleft=\endtrivlist
```

flushright Add a little surrounding whitespace.

```
1177 \def\flushright
1178   {\topsep=.25\baselineskip \@plus .1\baselineskip
1179     \@minus .1\baselineskip
1180     \trivlist \raggedleft\item[]\}
1181 \let\endflushright=\endtrivlist
1182 </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.

```
1183 \def\verbatim{%
1184 <*type1 | type2>
1185   \topsep=.25\baselineskip \@plus .1\baselineskip
1186     \@minus .1\baselineskip
1187   \@verbatim
1188 </type1 | type2>
1189 <type1>   \leftskip\unitindent
1190 <type2>   \leftskip\z@
1191 <*type3>
1192   \topsep=-.5\parskip
1193   \@verbatim
1194 </type3>
1195   \frenchspacing\@vobeyspaces \@xverbatim}
1196 <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.

```
1197 <!boek>\if@titlepage
1198   \newenvironment{abstract}{%
1199     \titlepage
1200     \null\vfil
1201     \@beginparpenalty\@lowpenalty
1202     \hbox{\SectFont \abstractname}%
1203     \@endparpenalty\@M
1204     \noindent\ignorespaces}
1205   {\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.

```
1206 <*artikel | rapport>
1207 \else
1208   \newenvironment{abstract}{%
1209     \if@twocolumn
1210       \section*{\abstractname}%
1211     \else
1212       \small
1213 <*type1 | type3>
1214       \bgroup\rightskip=\unitindent
1215       \hbox{\SectFont \abstractname}%
1216       \noindent\ignorespaces
1217 </type1 | type3>
```

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

```
1218 <*type2>
1219   \hbox{\hskip\unitindent\SectFont \abstractname}%
1220   \list{}{\setlength\listparindent{\unitindent}%
1221     \setlength\parindent {\z@}%
1222     \setlength\leftmargin {\unitindent}%
1223     \setlength\rightmargin {\unitindent}%
1224     \setlength\parsep {\z@}}%
1225   \item[]%
1226 </type2>
1227   \fi}
```

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

```
1228 <!type2>   {\if@twocolumn\else\par\egroup\fi}
1229 <type2>    {\if@twocolumn\else\par\endlist\fi}
1230 \fi
1231 </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`.

```
1232 \newenvironment{verse}
1233     {\let\\ \@centercr
1234      \list{}{\itemsep\z@
1235              \itemindent-1.5em%
1236              \listparindent\itemindent
1237              \rightmargin\leftmargin
1238              \advance\leftmargin1.5em}%
1239      \item\relax}
1240 {\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.

```
1241 \newenvironment{quotation}
1242     {\list{}{%
1243         \!type2)          \listparindent\z@
1244         \type2)          \listparindent\unitindent
1245         \boek)          \listparindent1.5em%
1246         \itemindent\listparindent
1247         \rightmargin\leftmargin
1248         \parsep\z@ \@plus\p@}%
1249     \item\relax}
1250 {\endlist}
```

8.5.4 Quote

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

```
1251 \newenvironment{quote}
1252     {\list{}{\rightmargin\leftmargin}%
1253      \item\relax}
1254 {\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.

```
1255 \def\@begintheorem#1#2{%
1256     \vskip\baselineskip \labelsep=.5em%
1257     \trivlist
```

```

1258 \item[\hskip \labelsep{\bfseries #1\ #2}]\slshape}
1259 \def\@opargbegintheorem#1#2#3{%
1260 \vskip\baselineskip \labelsep=.5em%
1261 \trivlist
1262 \item[\hskip \labelsep{\bfseries #1\ #2\ (#3)}]\slshape}
1263 \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.

```

1264 \newenvironment{titlepage}
1265 {
1266 \langle boek \rangle \cleardoublepage
1267 \if@twocolumn
1268 \@restonecoltrue\onecolumn
1269 \else
1270 \@restonecolfalse\newpage
1271 \fi
1272 \thispagestyle{empty}%
1273 \if@compatibility
1274 \setcounter{page}\z@
1275 \langle *artikel | rapport \rangle
1276 \else
1277 \setcounter{page}\@ne
1278 \langle /artikel | rapport \rangle
1279 \fi}
1280 {\if@restonecol\twocolumn \else \newpage \fi
1281 \langle artikel | rapport \rangle \setcounter{page}\@ne
1282 }

```

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.

```

1283 \langle *artikel \rangle
1284 \newcommand*\appendix{\par
1285 \setcounter{section}{0}%
1286 \setcounter{subsection}{0}%
1287 \gdef\thesection{\@Alph@c@section}}
1288 \langle /artikel \rangle

```

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.

```
1289 <*rapport | boek>
1290 \newcommand*{\appendix}{\par
1291   \setcounter{chapter}{0}%
1292   \setcounter{section}{0}%
1293   \gdef\@chapapp{\appendixname}%
1294   \gdef\thechapter{\@Alph\c@chapter}}
1295 </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 $2\arraycolsep$.

```
1296 \setlength\arraycolsep{5\p@}
```

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

```
1297 \setlength\tabcolsep{6\p@}
```

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

```
1298 \setlength\arrayrulewidth{.4\p@}
```

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

```
1299 \setlength\doublerulesep{2\p@}
```

8.6.2 Tabbing

`\tabbingsep` This controls the space that the `\'` command puts in. (See L^AT_EX manual for an explanation.)

```
1300 \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.

```
1301 <type1> \def \@minipagerestore{\parindent\unitindent}
```

```
1302 <*type3>
```

```

1303 \def\@minipagerestore{%
1304         \parskip=.5\baselineskip \@plus .1\baselineskip
1305         \@minus .1\baselineskip}
1306 (/type3)

```

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

```
1307 \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`.

```

1308 \setlength\fboxsep{3\p@}
1309 \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.

```

1310 <artikel>\renewcommand*\theequation{\@arabic\c@equation}
1311 <*rapport | boek>
1312 \@addtoreset{equation}{chapter}
1313 \renewcommand*\theequation{%
1314     \ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@equation}
1315 </rapport | boek>

```

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

```
1316 % \setlength\jot{3pt}
```

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

```
1317 % \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}[⟨placement⟩]` 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.

```
1318 ⟨*artikel⟩
1319 \newcounter{figure}
1320 \renewcommand*\thefigure{\@arabic\c@figure}
1321 ⟨/artikel⟩
1322 ⟨*rapport | boek⟩
1323 \newcounter{figure}[chapter]
1324 \renewcommand*\thefigure{%
1325   \ifnum\c@chapter>\z@\thechapter.\fi\@arabic\c@figure}
1326 ⟨/rapport | boek⟩
```

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

```
\ftype@figure 1327 \def\fps@figure{tbp}
\ext@figure 1328 \def\ftype@figure{1}
\num@figure 1329 \def\ext@figure{lof}
1330 \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.

```
1331 \newenvironment{figure}
1332     {\@float{figure}}
1333     {\end@float}
1334 \newenvironment{figure*}
1335     {\@dblfloat{figure}}
1336     {\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.

```
1337 <*artikel>
1338 \newcounter{table}
1339 \renewcommand*\thetable{\@arabic\c@table}
1340 </artikel>
1341 <*rapport | boek>
1342 \newcounter{table}[chapter]
1343 \renewcommand*\thetable{%
1344   \ifnum\c@chapter>\z@\thechapter.\fi\@arabic\c@table}
1345 </rapport | boek>
```

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

```
\ftype@table 1346 \def\fps@table{tbp}
\ext@table 1347 \def\ftype@table{2}
\num@table 1348 \def\ext@table{lot}
1349 \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.

```
1350 \newenvironment{table}
1351     {\@float{table}}
1352     {\end@float}
1353 \newenvironment{table*}
1354     {\@dblfloat{table}}
1355     {\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.

```
1356 \newlength\abovecaptionskip
1357 \newlength\belowcaptionskip
1358 \setlength\abovecaptionskip{10\p@}
1359 \setlength\belowcaptionskip{0\p@}
```


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

```

1360 \long\def\makecaption#1#2{%
1361   \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.
1362   \sbox\@tempboxa{\CaptionLabelFont#1:} \CaptionTextFont#2}%
      We can the measure its width. It that is larger than the current \hsize we typeset
      the caption as an ordinary paragraph.
1363   \ifdim \wd\@tempboxa >\hsize
1364     {\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.
1365   \else
1366     \global \@minipagefalse
1367     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
1368   \fi
1369   \vskip\belowcaptionskip}

```

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

```

1370 \newcommand*\CaptionLabelFont{\relax}
1371 \newcommand*\CaptionTextFont{\relax}

```

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

```

1372 \newcommand*\CaptionFonts[2]{%
1373   \renewcommand*\CaptionLabelFont{#1}%
1374   \renewcommand*\CaptionTextFont{#2}%
1375 }

```

8.8 Font changing

Here we supply the declarative font changing commands that were common in \LaTeX version 2.09 and earlier. These commands work in text mode *and* in math mode. They are provided for 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; \LaTeX commands to execute in text mode and \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 \LaTeX 2.09 behaviour.

```

\sff 1376 \DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
      1377 \DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
      1378 \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.

```
1379 \DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}
```

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

```
1380 \DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
```

```
1381 \DeclareOldFontCommand{\sl}{\normalfont\slshape}{\relax}
```

```
1382 \DeclareOldFontCommand{\sc}{\normalfont\scshape}{\relax}
```

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

```
1383 \DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}
```

```
1384 \DeclareRobustCommand*\mit{\@fontswitch\relax\mathnormal}
```

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

```
1385 \DeclareRobustCommand*\em{%
```

```
1386   \@nomath\em
```

```
1387   \ifdim\fontdimen\@ne\font>\z@
```

```
1388     \upshape
```

```
1389   \else
```

```
1390     \slshape
```

```
1391   \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

```
1392 \newcommand*\@pnumwidth{1.55em}
1393 \newcommand*\@tocrmarg {2.55em}
1394 \newcommand*\@dotsep{4.5}
1395 <artikel>\setcounter{tocdepth}{3}
1396 <!artikel>\setcounter{tocdepth}{2}
```

9.1.1 Table of Contents

`\tableofcontents` This macro is used to request that L^AT_EX 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.

```
1397 \newcommand*\tableofcontents{%
1398 <*rapport | boek>
1399   \if@twocolumn
1400     \@restonecoltrue\onecolumn
1401   \else
1402     \@restonecolfalse
1403   \fi
```

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

```
1404   \chapter*{\contentsname}%
1405 </rapport | boek>
1406 <artikel>   \section*{\contentsname}%
```

The code for `\mkboth` is placed inside the heading to avoid any influence on vertical spacing after the heading (in some cases). For other commands, such as `\listoffigures` below this has been changed from the L^AT_EX 2.09 version as it will produce a serious bug if used in two-column mode (see, L^AT_EX pr/3285). However `\tableofcontents` is always typeset in one-column mode in these classes, therefore the somewhat inconsistent setting has been retained for compatibility reasons.

```
1407 \mkboth{\MakeUppercase{\contentsname}}%
1408      {\MakeUppercase{\contentsname}}%
```

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

```
1409 \starttoc{toc}%
1410 \!artikel) \if@restonecol\twocolumn\fi
1411 }
```

`\starttoc` The internal L^AT_EX 2_ε macro `\starttoc` needs to be adapted for the `artike13` and `rapport3` document classes, in order to deal with a the fact that for these document classes the `\parskip` is normally non-zero. We don't want that in the table of contents.

```
1412 <*type3>
1413 \def\starttoc#1{\begingroup
1414   \makeatletter
1415   \parskip\z@
1416   \@input{\jobname.#1}%
1417   \if@filesw
1418     \expandafter\newwrite\csname tf@#1\endcsname
1419     \immediate\openout \csname tf@#1\endcsname \jobname.#1\relax
1420   \fi \@nobreakfalse \endgroup}
1421 </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 acheive this different format we defined the macro `\regtocline`.

```
1422 \newcommand*\regtocline[3]{%
1423   \ifnum #1>\c@tocdepth
1424   \else
1425     \vskip\z@\@plus.2\p@
1426     {\hangindent\z@ \@afterindenttrue \interlinepenalty\@M
1427      \leftskip\unitindent
1428      \rightskip\unitindent\@plus 1fil
1429      \parfillskip\z@
1430      \@tempdima\unitindent
1431 <type2> \advance\@tempdima by \othermargin
1432      \parindent\z@
1433      \leavevmode
1434      \hbox{\hskip -\leftskip\relax
1435      \ifnum#1<0\toc@case{#2}\else
```

```

1436     \toc@font#1 #2\fi\nobreak
1437     \hskip 1em \nobreak{\slshape #3}\par
1438     }%
1439 \fi}

```

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

```

1440 <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`.

`\toc@fontsel`

```

1441 \if@oldtoc
1442 \newcommand*\toc@font[1]{\relax}
1443 \else

```

A line of the table of contents contains `\numberline` and the section number as its first two elements. We don't want to set the section number using `\toc@font`, therefore we give it two additional arguments and pass them on first, *before* changing the font. Note that we need to re-insert the braces around the second argument.

```

1444 \newcommand*\toc@font[4]{%
1445 #2{#3}#4\toc@fontsel#1}
1446 \newcommand*\toc@fontsel[1]{%
1447 <*artikel>
1448 \ifcase#1\relax
1449 <type2> \Large\bfseries
1450 \or\bfseries
1451 \or\slshape
1452 \or\rmfamily
1453 </artikel>
1454 <*rapport | boek>
1455 \ifcase#1\relax
1456 \bfseries
1457 \or\slshape
1458 \or\rmfamily
1459 </rapport | boek>
1460 \fi}

```

When the user wants to produce a hyper-document using `hyperref` we need to take special precautions to make it work for the table of contents. We check for the existence of `\hyper@linkstart` to detect this situation at `\begin{document}`.

`Hyperref` injects extra tokens (`\hyper@linkstart{link}{Hy@tocdestname}`) into the stream in front of the real contentsline. The command `\hyper@linkstart` and its arguments need to be protected from expanding too early or being “uppercased” themselves.

```

1461 \AtBeginDocument{%
1462 \ifx\hyper@linkstart\undefined
1463 \else

```

In the contentslines for chapters, sections etc., the command selection of the appropriate font needs to come after the code that `hyperref` injects. we do this with some argument shuffling.

```

1464 \let\ORG@hyper@linkstart\hyper@linkstart
1465 \protected\def\hyper@linkstart#1#2{%
1466 \lowercase{\ORG@hyper@linkstart{#1}{#2}}}
1467 \fi}
1468 \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.

```

1469 \if@oldtoc
1470 \newcommand*\toc@case{\relax}
1471 \else
1472 \newcommand*\toc@case{\MakeUppercase}
1473 \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.

```

1474 \if@oldtoc
1475 \newcommand*\l@part[2]{%
1476 \ifnum \c@tocdepth >-2\relax
1477 <artikel> \addpenalty\@secpenalty
1478 <!artikel> \addpenalty{-\@highpenalty}%
1479 \addvspace{2.25em \@plus\p@}%
1480 \begingroup

```

The macro `\numberline` requires that the width of the box that holds the part number is stored in L^AT_EX's scratch register `\@tempdima`. Therefore we put it there.

```

1481 \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.

```

1482 \parindent \z@ \rightskip \@pnumwidth
1483 \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 page number, set flush right.

```

1484 {\leavevmode
1485 \large \bfseries #1\hfil \hb@xt@\@pnumwidth{\hss #2%
1486 \kern-\p@\kern\p@}}\par

```

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

```

1487 \nobreak

```

```

1488 <artikel>      \ifcompatibility
1489               \global\@nobreaktrue
1490               \everypar{\global\@nobreakfalse\everypar{}}%
1491 <artikel>      \fi
1492       \endgroup
1493 \fi}

```

Then we can introduce our new definition.

```

1494 \else
1495   \newcommand*\l@part{%
1496     \ifnum \c@tocdepth >-2\let\l@part\relax
1497 <artikel>     \addpenalty\@secpenalty
1498 <!artikel>    \addpenalty{-\@highpenalty}%
1499     \addvspace{2.25em \@plus \p@}%
1500     \def\l@part{\@regtocline{-1}}%
1501     \fi\l@part}
1502 \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

```

1503 <*rapport | boek>
1504 \ifoldtoc
1505 \newcommand*\l@chapter[2]{%
1506   \addpenalty{-\@highpenalty}%
1507   \vskip 1.0em \@plus\p@

```

The macro `\numberline` requires that the width of the box that holds the part number is stored in L^AT_EX’s scratch register `\@tempdima`. Therefore we put it there. We begin a group, and change some of the paragraph parameters.

```

1508   \setlength\@tempdima{1.5em}%
1509   \begingroup
1510   \parindent \z@ \rightskip \@pnumwidth
1511   \parfillskip -\@pnumwidth

```

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

```

1512   \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.

```

1513   \advance\leftskip\@tempdima
1514   \hskip -\leftskip
1515   #1\nobreak\hfil \nobreak\hb@xt@\@pnumwidth{\hss #2%
1516                                     \kern-\p@\kern\p@}\par
1517   \penalty\@highpenalty
1518 \endgroup}

```

Then we present our new definition.

```
1519 \else
1520 \newcommand*\l@chapter{\@regtocline{0}}
1521 \fi
1522 \</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.

```
1523 <*artikel>
1524 \if@oldtoc
1525 \newcommand*\l@section[2]{%
1526   \addpenalty\@secpenalty
1527   \addvspace{1.0em \@plus\p@}}%
```

The macro `\numberline` requires that the width of the box that holds the part number is stored in L^AT_EX's scratch register `\@tempdima`. Therefore we put it there. We begin a group, and change some of the paragraph parameters.

```
1528   \setlength\@tempdima{1.5em}%
1529   \begingroup
1530   \parindent \z@ \rightskip \@pnumwidth
1531   \parfillskip -\@pnumwidth
```

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

```
1532   \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.

```
1533   \advance\leftskip\@tempdima
1534   \hskip -\leftskip
1535   #1\nobreak\hfil \nobreak\hb@xt@\@pnumwidth{\hss #2%
1536   \kern-\p@\kern\p@}\par
1537 \endgroup}
```

The new definition:

```
1538 \else
1539 \newcommand*\l@section{\@regtocline{1}}
1540 \fi
1541 \</artikel>
```

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

```
1542 <*rapport | boek>
1543 \if@oldtoc
1544 \newcommand*\l@section {\@dottedtocline{1}{1.5em}{2.3em}}
1545 \else
1546 \newcommand*\l@section {\@regtocline{1}}
1547 \fi
1548 \</rapport | boek>
```


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

```

\l@paragraph 1549 \if@oldtoc
\l@subparagraph 1550 <*artikel>
1551 \newcommand*\l@section {\@dottedtocline{2}{1.5em}{2.3em}}
1552 \newcommand*\l@subsubsection{\@dottedtocline{3}{3.8em}{3.2em}}
1553 \newcommand*\l@paragraph {\@dottedtocline{4}{7.0em}{4.1em}}
1554 \newcommand*\l@subparagraph {\@dottedtocline{5}{10em}{5em}}
1555 </artikel>
1556 <*rapport | boek>
1557 \newcommand*\l@section {\@dottedtocline{2}{3.8em}{3.2em}}
1558 \newcommand*\l@subsubsection{\@dottedtocline{3}{7.0em}{4.1em}}
1559 \newcommand*\l@paragraph {\@dottedtocline{4}{10em}{5em}}
1560 \newcommand*\l@subparagraph {\@dottedtocline{5}{12em}{6em}}
1561 </rapport | boek>
1562 \else
1563 \newcommand*\l@section {\@regtocline{2}}
1564 \newcommand*\l@subsubsection{\@regtocline{3}}
1565 \newcommand*\l@paragraph {\@regtocline{4}}
1566 \newcommand*\l@subparagraph {\@regtocline{5}}
1567 \fi

```

9.1.2 List of figures

`\listoffigures` This macro is used to request that L^AT_EX produces a list of figures. It is very similar to `\tableofcontents`.

```

1568 \newcommand*\listoffigures{%
1569 <*rapport | boek>
1570 \if@twocolumn
1571 \@restonecoltrue\onecolumn
1572 \else
1573 \@restonecolfalse
1574 \fi
1575 \chapter*{\listfigurename}%
1576 </rapport | boek>
1577 <artikel> \section*{\listfigurename}%
1578 \mkboth{\MakeUppercase{\listfigurename}}%
1579 {\MakeUppercase{\listfigurename}}%
1580 \@starttoc{lof}%
1581 <rapport | boek> \if@restonecol\twocolumn\fi
1582 }

```

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

```

1583 \if@oldtoc
1584 \newcommand*\l@figure{\@dottedtocline{1}{1.5em}{2.3em}}
1585 \else
1586 \newcommand*\l@figure{\@regtocline{1}}
1587 \fi

```

9.1.3 List of tables

`\listoftables` This macro is used to request that L^AT_EX produces a list of tables. It is very similar to `\tableofcontents`.

```
1588 \newcommand*\listoftables{%
1589   <*rapport | boek>
1590     \if@twocolumn
1591       \@restonecoltrue\onecolumn
1592     \else
1593       \@restonecolfalse
1594     \fi
1595     \chapter*{\listtablename}%
1596 </rapport | boek>
1597 <artikel>   \section*{\listtablename}%
1598     \@mkboth{\MakeUppercase{\listtablename}}%
1599             {\MakeUppercase{\listtablename}}%
1600     \@starttoc{lot}%
1601 <rapport | boek>   \if@restonecol\twocolumn\fi
1602   }
```

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

```
1603 \let\l@table\l@figure
```

9.2 Bibliography

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

```
1604 \newdimen\bibindent
1605 \setlength\bibindent{1.5em}
```

`\newblock` This is a dummy definition for this macro which is used in the `thebibliography` environment.

```
1606 \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 environment. It uses the `enumiv` counter internally to generate the labels of the list.

When an empty ‘`thebibliography`’ environment is found, a warning is issued.

```
1607 \newenvironment{thebibliography}[1]
1608 <*artikel>
1609     {\section*{\refname}%
1610     \@mkboth{\MakeUppercase\refname}{\MakeUppercase\refname}%
1611 </artikel>
```

```

1612 <!*artikel)
1613     {\chapter*{\bibname}%
1614      \mkboth{\MakeUppercase\bibname}{\MakeUppercase\bibname}%
1615 </!*artikel)
1616     \list{\@biblabel{\@arabic\c@enumiv}}%
1617         {\settowidth\labelwidth{\@biblabel{#1}}%
1618          \leftmargin\labelwidth
1619          \advance\leftmargin\labelsep
1620          \@openbib@code
1621          \usecounter{enumiv}%
1622          \let\p@enumiv\@empty
1623          \renewcommand*\theenumiv{\@arabic\c@enumiv}}%
1624     \sloppy\clubpenalty4000\widowpenalty4000%
1625     \sfcode'\.\@m}
1626     {\def\@noitemerr
1627      {\@latex@warning{Empty ‘thebibliography’ environment}}}%
1628     \endlist}

```

`\newblock` The default definition for `\newblock` is to produce a small space.

```

1629 % \changes{v2.0t}{1996/04/01}{use \cs{renewcommand} instead of
1630 %   \cs{newcommand}}
1631 \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.

```

1632 \let\@openbib@code\@empty

```

`\@biblabel` The label for a `\bibitem[...]` command is produced by this macro. The default from `latex.dtx` is used.

```

1633 % \renewcommand*\@biblabel[1]{\#1\hfill}

```

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

```

1634 % \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.

```

1635 \newenvironment{theindex}{%
1636   \if@twocolumn
1637     \@restonecolfalse
1638   \else
1639     \@restonecoltrue
1640   \fi

```

```

1641 <artikel> \twocolumn[\section*{\indexname}]%
1642 <!artikel> \twocolumn[\@makeschapterhead{\indexname}]%
1643 \mkboth{\MakeUppercase{\indexname}}{\MakeUppercase{\indexname}}%
1644 \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.

```

1645 \columnseprule \z@
1646 \columnsep 35\p@
1647 \parskip\z@ \@plus .3\p@\relax
1648 \let\item\@idxitem
1649 }{%

```

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.

```

1650 \if@restonecol\onecolumn\else\clearpage\fi}

```

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

```

\subitem 1651 \newcommand*\@idxitem {\par\hangindent 40\p@}
\subsubitem 1652 \newcommand*\subitem {\@idxitem\hspace*{20\p@}}
1653 \newcommand*\subsubitem{\@idxitem\hspace*{30\p@}}

```

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

```

1654 \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 L^AT_EX 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 height¹.

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.

```

1655 \renewcommand*\footnoterule{%
1656 \kern-3\p@
1657 <*type1 | type3>
1658 \kern.5\baselineskip
1659 \hrule\@width\unitindent
1660 \kern.4\baselineskip
1661 </type1 | type3>
1662 <*type2>
1663 \hrule\@width 3\unitindent
1664 \kern 2.6\p@

```

¹This 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)

```
1665 </type2>
1666 }
```

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

```
1667 % \newcounter{footnote}
1668 <!artikel>\@addtoreset{footnote}{chapter}
```

`\@makefntext` The footnote mechanism of L^AT_EX calls the macro `\@makefntext` 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 `\@makefntextis` 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_EX code.

```
\long\def\xmakefntext#1#2{%
%<!type3> \parindent=.5\unitindent
%<type3> \parindent=\z@\parskip=.5\baselineskip
\def\labelitemi{--}\@revlabeltrue
{\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`.

```
1669 <*type1 | type3>
1670 \newcommand*\@makefntext{\@makefntext{\normalfont\@thefnmark.}}
1671 \newcommand*\xmakefntext[1]{%
```

```

1672 \parindent\z@
1673 \def\labelitemi{\textendash}\@revlabeltrue
1674 {\setbox0\hbox {#1\hskip.5em plus 1fil}
1675 \dimen0=2\wd0\relax
1676 \ifdim\dimen0>\unitindent
1677 \global\unitindent\dimen0\relax
1678 \indentset
1679 \fi}
1680 \leavevmode\hb@xt@.5\unitindent{#1\hfil}}
1681 </type1 | type3>

```

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

```

1682 <*type2>
1683 \newcommand\@makefnmark[1]{%
1684 \parindent\othermargin
1685 \noindent\hb@xt@\othermargin{\normalfont\@thefnmark\hfil\relax}#1}
1686 </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.

```

1687 %\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.

```

1688 \newcommand*\contentsname{Contents}
1689 \newcommand*\listfigurename{List of Figures}
1690 \newcommand*\listtablename{List of Tables}

```

`\refname`

```

\bibname 1691 <artikel>\newcommand*\refname{References}

```

```

\indexname 1692 <rapport | boek>\newcommand*\bibname{Bibliography}

```

```

1693 \newcommand*\indexname{Index}

```

`\figurename`

```

\tablename 1694 \newcommand*\figurename{Figure}

```

```

1695 \newcommand*\tablename{Table}

```

`\partname`

```

\chaptername 1696 \newcommand*\partname{Part}

```

```

\appendixname 1697 <rapport | boek>\newcommand*\chaptername{Chapter}

```

```

\abstractname 1698 \newcommand*\appendixname{Appendix}

```

```

\seename 1699 <!boek>\newcommand*\abstractname{Abstract}

```

```

\andname 1700 \newcommand*\seename{see}

```

```

1701 \newcommand*\andname{and}

```

10.2 Date

`\today` This macro uses the T_EX primitives `\month`, `\day` and `\year` to provide the date of the L^AT_EX-run.

```
1702 \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 T_EX registers that are happening later on (e.g. if some packages goes `\month=5`) are not reflected in `\today`.

```
1703 \def\today{\ifcase\month\or
1704   January\or February\or March\or April\or May\or June\or
1705   July\or August\or September\or October\or November\or December\fi
1706   \space\number\day, \number\year}
```

10.3 Two column mode

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

```
1707 \setlength\columnsep{10\p@}
```

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

```
1708 \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.

```
1709 \(!boek)\pagestyle{plain}
1710 \!boek)\pagestyle{headings}
1711 \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.

```
1712 \!artikel)
1713 \if@twoside
1714 \else
1715   \raggedbottom
1716 \fi
1717 \!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.

```
1718 \if@twocolumn
```

1719 `\twocolumn`
 1720 `\sloppy`
 1721 `\flushbottom`

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

1722 `\else`
 1723 `\onecolumn`
 1724 `\fi`

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

1725 `\frenchspacing`
 1726 `\</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.

Symbols	<code>\@dblftop</code> <u>423</u>	<code>\@idxitem</code> .. 1648, <u>1651</u>
<code>\@Roman</code> 755	<code>\@dotsep</code> <u>1392</u>	<code>\@ifundefined</code> 196
<code>\@afterheading</code>	<code>\@dottedtocline</code> 1544,	<code>\@indentset</code> <u>191</u> ,
. 823, 856, 905, 929	1551–1554,	197, 199, 687, 1678
<code>\@afterindentfalse</code> .	1557–1560, 1584	<code>\@itemdepth</code>
. 784, 881	<code>\@emptypagestyle</code> 1137, 1140, 1141
<code>\@afterindenttrue</code> 440, 443	<code>\@itemitem</code> . 1141, 1144
. 785, 1426	<code>\@endparpenalty</code> . . .	<code>\@itempenalty</code> <u>1033</u>
<code>\@allcapsfalse</code> 53 <u>1033</u> , <u>1203</u>	<code>\@latex@warning</code> .. 1627
<code>\@allcapstrue</code> 54	<code>\@endpart</code> . 845, 864, <u>865</u>	<code>\@listI</code> 102, <u>1036</u>
<code>\@author</code> 528, 559, 561,	<code>\@endtheorem</code> <u>1255</u>	<code>\@listdepth</code> . 1116, 1146
579, 631, 651, 663	<code>\@enumctr</code>	<code>\@listi</code> 102, <u>1036</u>
<code>\@beginparpenalty</code> 1111, 1114, 1115	<code>\@listii</code> <u>1049</u>
. <u>1033</u> , 1201	<code>\@enumdepth</code>	<code>\@listiii</code> <u>1049</u>
<code>\@begintheorem</code> <u>1255</u> 1107, 1110, 1111	<code>\@listiv</code> <u>1049</u>
<code>\@biblabel</code>	<code>\@eqnnum</code> <u>1317</u>	<code>\@listv</code> <u>1049</u>
. 1616, 1617, <u>1633</u>	<code>\@evenfoot</code> <u>439</u> ,	<code>\@listvi</code> <u>1049</u>
<code>\@chapapp</code> . 481, 508,	452, 454, 459, 515	<code>\@lowpenalty</code> .. <u>217</u> ,
<u>765</u> , 887, 916, 1293	<code>\@evenhead</code>	1033–1035, 1201
<code>\@chapter</code> 882, <u>883</u> <u>439</u> , 449, 460, 516	<code>\@mainaux</code> 195, 198
<code>\@cite</code> <u>1634</u>	<code>\@fnsymbol</code> 542, 602	<code>\@mainmatterfalse</code> .
<code>\@currentpagestyle</code> .	<code>\@fontswitch</code> 1383, 1384 769, 777
. 441, 443	<code>\@fpbot</code> <u>408</u>	<code>\@mainmattertrue</code> 9, 773
<code>\@date</code> 529,	<code>\@fpsep</code> <u>408</u>	<code>\@makecaption</code> <u>1356</u>
530, 567, 569,	<code>\@fptop</code> <u>408</u>	<code>\@makechapterhead</code> .
581, 633, 654, 664	<code>\@hangfrom</code> 698, 701, 731 902, 904, <u>908</u>
<code>\@dblfloat</code> . 1335, 1354	<code>\@highpenalty</code>	<code>\@makefnmark</code> . 603, <u>1687</u>
<code>\@dblfpbot</code> <u>423</u> <u>217</u> , 1478,	<code>\@makefnstext</code>
<code>\@dblfpsep</code> <u>423</u>	1498, 1506, 1517	539, 605, 608, <u>1669</u>

<code>\@makeschapterhead</code> .	<code>\@secntformat</code> 691, 723	<code>\@twosidetrue</code> 42
926, 928, <u>931</u> , 1642	<code>\@secpenalty</code>	<code>\@undefined</code> 105
<code>\@maketitle</code> . . . 615,	. . . 1477, 1497, 1526	<code>\@verbatim</code> . 1187, 1193
617, 622, 629, <u>639</u>	<code>\@sect</code> <u>675</u>	<code>\@vobeyspaces</code> 1195
<code>\@medpenalty</code> <u>217</u>	<code>\@setfontsize</code> . . 84,	<code>\@width</code> 1659, 1663
<code>\@minipagefalse</code> . . 1366	90, 96, 110, 116,	<code>\@writeindent</code> . <u>195</u> , 203
<code>\@minipagerestore</code> <u>1301</u>	122, 131, 137,	<code>\@xmakefnstext</code>
<code>\@mparswitchfalse</code> . 41	143, 151–157,	. . . 605, 1670, 1671
<code>\@mparswitchtrue</code> . . 42	160–166, 169–174	<code>\@xssect</code> 722, 736
<code>\@mpfootins</code> <u>1307</u>	<code>\@settopoint</code> . . 270,	<code>\@xverbatim</code> 1195
<code>\@nameuse</code> 441	342, 343, 348, 360	
<code>\@needwriteindenttrue</code>	<code>\@spart</code> . . . 786, 799, <u>847</u>	
. 194	<code>\@specialpagetrue</code> . 445	<code>_</code> 481,
<code>\@nobreakfalse</code>	<code>\@specialstyle</code> 446	487, 508, 1258, 1262
. 1420, 1490	<code>\@ssect</code> <u>727</u>	
<code>\@nobreaktrue</code> 1489	<code>\@startsection</code>	A
<code>\@noitemerr</code> 1626	. 945, 949, 958,	<code>\abovecaptionskip</code> .
<code>\@normalsize</code> <u>81</u>	962, 971, 975, <u>1356</u> , 1361
<code>\@oddfnfoot</code> . <u>439</u> , 450,	985, 986, 993, 994	<code>\abovedisplayshortskip</code>
454, 459, 492, 515	<code>\@starttoc</code> . . . 1409, 86,
<code>\@oddfnhead</code> <u>439</u> ,	<u>1412</u> , 1580, 1600	92, 98, 112, 118,
449, 461, 493, 517	<code>\@svsec</code> 677,	124, 133, 139, 145
<code>\@oldtocfalse</code> 11	691, 698, 702, 714	<code>\abovedisplayskip</code> 85,
<code>\@oldtoctrue</code> 51	<code>\@svsechd</code> 713, 734	91, 97, 101, 111,
<code>\@opargbegintheorem</code>	<code>\@tempa</code> 587,	117, 123, 127,
. <u>1255</u>	589, 591, 595, 839	132, 138, 144, 148
<code>\@openbib@code</code>	<code>\@tempskipa</code>	<code>abstract</code> (environ-
. . . 61, 1620, <u>1632</u>	. 693, 694, 727, 728	ment) <u>1197</u>
<code>\@openrightfalse</code> . . 48	<code>\@textsuperscript</code> .	<code>\abstractname</code>
<code>\@openrighttrue</code> . . . 47 541,	. . . 1202, 1210,
<code>\@part</code> 786, 799, <u>801</u>	604, 606, 611, 1687	1215, 1219, <u>1696</u>
<code>\@pnumwidth</code> . . . <u>1392</u> ,	<code>\@thanks</code>	<code>\addcontentsline</code> . .
1482, 1483,	. 572, 578, 624, 630 706, 715,
1485, 1510,	<code>\@thefnmark</code> . . . 541,	806, 808, 829,
1511, 1515,	604, 606, 611,	831, 888, 892, 896
1530, 1531, 1535	1670, 1685, 1687	<code>\addtocontents</code> 899, 900
<code>\@ptsizes</code> . . <u>1</u> , 38–40, 79	<code>\@title</code> 527, 550, 552,	<code>\and</code> 559,
<code>\@regtocline</code> . . <u>1422</u> ,	580, 632, 646, 658	585, 588, 637, 660
1500, 1520,	<code>\@titlecenteredfalse</code> 14	<code>\andname</code> 593, <u>1696</u>
1539, 1546,	<code>\@titlecenteredtrue</code> 52	<code>\appendix</code> <u>1283</u>
1563–1566, 1586	<code>\@titlepagefalse</code> . .	<code>\appendixname</code> 1293, <u>1696</u>
<code>\@restonecolfalse</code> 6, 46, 50	<code>\arraycolsep</code> <u>1296</u>
. . . 1270, 1402,	<code>\@titlepagetrue</code> 7, 45, 49	<code>\arrayrulewidth</code> . . <u>1298</u>
1573, 1593, 1637	<code>\@tocrmarg</code> <u>1392</u>	<code>\AtBeginDocument</code> . 1461
<code>\@restonecoltrue</code> . .	<code>\@toodeep</code> . . 1108, 1138	<code>\AtEndDocument</code> 200
. . . 1268, 1400,	<code>\@topnewpage</code> . . 902, 926	<code>\AtEndOfPackage</code> . . . 60
1571, 1591, 1639	<code>\@topnum</code> 621, 880	<code>\author</code> . . . <u>527</u> , 583, 635
<code>\@revlabeltrue</code> 540, 1673	<code>\@twocolumnfalse</code> . . 55	
<code>\@roman</code> 1097	<code>\@twocolumtrue</code> . . . 56	B
<code>\@schapter</code> . . . 882, <u>925</u>	<code>\@twosidefalse</code> 41	<code>\backmatter</code> <u>775</u>

<code>\baselineskip</code>	<u>743</u> , 804, 813, 827, 837, 884, 914	D
. 213, 214,		<code>\date</code> <u>527</u> , 584, 636
273–275, 280,	<code>\c@section</code>	<code>\dblfloatpagefraction</code> <u>379</u>
282, 946, 950,	<u>745</u> , 756, 759, 1287	<code>\dblfloatsep</code> <u>396</u>
952, 953, 959,	<code>\c@subparagraph</code> <u>745</u> , 764	<code>\dbltextfloatsep</code> .. <u>396</u>
963, 965, 966,	<code>\c@subsection</code> . <u>745</u> , 761	<code>\dbltopfraction</code> ... <u>378</u>
972, 976, 978,	<code>\c@subsubsection</code> ..	<code>\DeclareOldFontCommand</code> 1376–1382
979, 1168, 1169, <u>745</u> , 762	<code>\DeclareRobustCommand</code> . 108, 129, 151– 157, 160–166, 169–174, 1383–1385
1173, 1174,	<code>\c@table</code> <u>1337</u>	description (environ- ment) <u>1160</u>
1178, 1179,	<code>\c@tocdepth</code>	<code>\descriptionlabel</code> 1162, <u>1164</u>
1185, 1186,	.. 1423, 1476, 1496	<code>\dimen</code> 1675–1677
1256, 1260,	<code>\c@topnumber</code> <u>369</u>	<code>\displaywidowpenalty</code> <u>222</u>
1263, 1304,	<code>\c@totalnumber</code> <u>374</u>	<code>\doublerulesep</code> ... <u>1299</u>
1305, 1658, 1660	<code>\cal</code> <u>1383</u>	
<code>\baselinestretch</code> .. <u>181</u>	<code>\CaptionFonts</code> <u>1372</u>	
<code>\belowcaptionskip</code> .	<code>\CaptionLabelFont</code> .	
. <u>1356</u> , 1369 1362, 1364, <u>1370</u> , 1373	
<code>\belowdisplayshortskip</code>	<code>\CaptionTextFont</code> ..	
. 87, 1362, 1364, <u>1370</u> , 1374	
93, 99, 113, 119,	<code>\center</code> 1167	E
125, 134, 140, 146	<code>\center (environment)</code> <u>1166</u>	<code>\edef</code> ... 839, 1111, 1141
<code>\belowdisplayskip</code> .	<code>\centering</code> 834, 860, 1170	<code>\em</code> <u>1385</u>
. 101, 127, 148	<code>\changes</code> 1629	<code>\endcenter</code> 1171
<code>\bf</code> <u>1379</u>	<code>\ChapFont</code> <u>907</u> ,	<code>\endcsname</code> 683, 705, 709, 714, 718, 724, 725, 1114, 1144, 1418, 1419
<code>\bibindent</code> 62, 63, <u>1604</u>	916, 921, 938, 1013	<code>\endflushleft</code> 1176
<code>\bibname</code> 1613, 1614, <u>1691</u>	<code>\chapter</code> . <u>877</u> , 1404, 1575, 1595, 1613	<code>\endflushright</code> 1181
<code>\bottomfraction</code> ... <u>373</u>	<code>\chaptermark</code> .. 478, 504, 519, <u>737</u> , 898	<code>\endgraf</code> .. 698, 702, 731
<code>\brokenpenalty</code> <u>226</u>	<code>\chaptername</code> . 765, <u>1696</u>	<code>\endtitlepage</code> 1205
	<code>\cleardoublepage</code> ..	<code>\endverbatim</code> 1196
 768, 772, 776, 790, 878, 1266	enumerate (environ- ment) <u>1106</u>
C	<code>\clearpage</code> 776, 878, 1650	environments:
<code>\c@bottomnumber</code> <u>372</u>	<code>\clubpenalty</code> . <u>220</u> , 1624	abstract <u>1197</u>
<code>\c@chapter</code>	<code>\col@number</code> 614	center <u>1166</u>
<u>745</u> , 758, 1294,	<code>\columnsep</code> . 1646, <u>1707</u>	description ... <u>1160</u>
1314, 1325, 1344	<code>\columnseprule</code>	enumerate <u>1106</u>
<code>\c@dbltopnumber</code> <u>377</u> 1645, <u>1708</u>	figure <u>1331</u>
<code>\c@enumi</code> 1095	<code>\contentsname</code> . 1404, 1406–1408, <u>1688</u>	figure* <u>1331</u>
<code>\c@enumii</code> 1096	<code>\cs</code> 1629, 1630	flushleft <u>1172</u>
<code>\c@enumiii</code> 1097	<code>\csname</code> 683, 705, 709, 714, 718, 724, 725, 1114, 1144, 1418, 1419	flushright <u>1177</u>
<code>\c@enumiv</code>		itemize <u>1136</u>
. 1098, 1616, 1623		quotation <u>1241</u>
<code>\c@equation</code> . 1310, 1314		quote <u>1251</u>
<code>\c@figure</code> <u>1318</u>		table <u>1350</u>
<code>\c@footnote</code>		
. 542, 602, <u>1667</u>		
<code>\c@paragraph</code> .. <u>745</u> , 763		
<code>\c@part</code> <u>745</u> , 755		
<code>\c@secnumdepth</code>		
. 466, 472, 479,		
486, 498, 506,		
676, 707, 716,		

<code>\labelitemi</code>	<code>\leftmarginiv</code>	<code>\mathnormal</code>
540, 609, 1131 , 1673	.. 1020 , 1068, 1071	1384
<code>\labelitemii</code>	<code>\leftmarginv</code>	<code>\mathrm</code>
1131	.. 1020 , 1077, 1080	1376
<code>\labelitemiii</code>	<code>\leftmarginvi</code>	<code>\mathsf</code>
1131	.. 1020 , 1086, 1089	1377
<code>\labelitemiv</code>	<code>\leftskip</code>	<code>\mathtt</code>
1131	1189,	1378
<code>\labelsep</code>	1190, 1427,	<code>\maxdepth</code>
1028 ,	1434, 1513,	235
1039, 1040,	1514, 1533, 1534	<code>\mit</code>
1042, 1050,	<code>\lineskip</code>	1383
1051, 1053,	.. 178 , 556, 649, 659	N
1059, 1060,	<code>\listfigurename</code> 1575,	<code>\newblock</code> 67, 1606 , 1629
1062, 1069,	1577–1579, 1688	<code>\newif</code>
1070, 1072,	<code>\listoffigures</code> ... 1568	4, 5,
1078, 1079,	<code>\listoftables</code> 1588	8–10, 12, 13, 15, 190
1081, 1087,	<code>\listparindent</code>	<code>\newwrite</code>
1088, 1090, 64, 1220,	1418
1164, 1256,	1236, 1243–1246	<code>\nobreakspace</code> .
1258, 1260,	<code>\listtablename</code> 1595,	814,
1262, 1300, 1619	1597–1599, 1688	815, 840, 1330, 1349
<code>\labelwidth</code> ... 1028 ,	<code>\lowercase</code>	<code>\noexpand</code>
1041, 1042,	1466	839
1052, 1053,	M	<code>\normalfont</code> ...
1061, 1062,	<code>\mainmatter</code>	604,
1071, 1072,	<code>\makeatletter</code>	606, 611, 812,
1080, 1081,	<code>\makelabel</code> ...	836, 851, 862,
1089, 1090,	1119,	913, 937, 954,
1161, 1617, 1618	1121, 1122,	967, 980, 989,
<code>\LARGE</code> 150 , 548, 646, 658	1126, 1149–	997, 1135, 1165,
<code>\Large</code>	1151, 1155, 1162	1376–1382,
150 ,	<code>\MakeRobust</code> ...	1670, 1685, 1687
814, 815, 818,	105, 106	<code>\normallineskip</code> ...
819, 838, 844,	<code>\maketitle</code>	178
852, 853, 863,	<code>\MakeUppercase</code>	<code>\normalsize</code>
916, 921, 938, 1449 465, 479,	.. 81 , 184, 188,
<code>\large</code>	486, 497, 505,	967, 980, 989, 997
150 , 555,	839, 844, 1407,	<code>\num@figure</code>
565, 648, 654,	1408, 1472,	1327
659, 664, 954, 1485	1578, 1579,	<code>\num@table</code>
<code>\leftmargin</code>	1598, 1599,	1346
62,	1610, 1614, 1643	<code>\numberline</code>
1020 ,	<code>\marginparpush</code>	1440
1038, 1041, 285, 328, 336	O
1049, 1058,	<code>\marginparwidth</code> ...	<code>\oddsidemargin</code>
1068, 1077,	294	294
1086, 1161,	<code>\MarkFont</code>	<code>\onecolumn</code> 793, 1268,
1222, 1237,	.. 460, 461, 493,	1400, 1571,
1238, 1247,	516, 517, 523 , 526	1591, 1650, 1723
1252, 1618, 1619	<code>\mathbf</code>	<code>\openout</code>
<code>\leftmargini</code> ..	1379	1419
193,	<code>\mathcal</code>	<code>\ORG@hyper@linkstart</code>
1020 , 1029, 1038	1380 1464, 1466
<code>\leftmarginii</code>	<code>\mathit</code>	<code>\othermargin</code> ..
.. 1020 , 1049, 1052	1380	187 ,
<code>\leftmarginiii</code>	<code>\mathit</code>	207, 1021–1025,
.. 1020 , 1058, 1061	1380	1431, 1684, 1685
		<code>\overfullrule</code> ...
		43, 44
		P
		<code>\p@enumii</code>
		1103
		<code>\p@enumiii</code>
		1103
		<code>\p@enumiv</code> ..
		1103 , 1622

<code>\PageFont</code> . 450, 452, 460, 461, 493, 516, 517, <u>523</u> , 526		
<code>\pagenumbering</code> 770, 774, 1711		
<code>\paperheight</code> . . . 17, 20, 23, 26, 29, 32, 35, 36, 277, 353		
<code>\paperwidth</code> 18, 21, 24, 27, 30, 33, 36, 37, 249, 323, 331, 344		
<code>\ParaFont</code> 989, <u>990</u> , 1006, 1017		
<code>\paragraph</code> <u>984</u>		
<code>\paragraphmark</code> <u>737</u>		
<code>\parindent</code> . . . 192, <u>208</u> , 539, 556, 608, 811, 849, 911, 934, 993, 1221, 1301, 1432, 1482, 1510, 1530, 1644, 1672, 1684		
<code>\parsep</code> . . . 65, 1043, 1056, 1065, 1075, 1084, 1093, 1224, 1248		
<code>\parskip</code> <u>208</u> , 912, 935, 1045, 1055, 1064, 1074, 1083, 1092, 1192, 1304, 1415, 1647		
<code>\part</code> <u>779</u>		
<code>\PartFont</code> . <u>801</u> , 814, 815, 818, 819, 838, 844, 852, 853, 863, 1002, 1012		
<code>\partname</code> 814, 815, 839, <u>1696</u>		
<code>\partopsep</code> . <u>1031</u> , 1066		
<code>\postdisplaypenalty</code> <u>222</u>		
<code>\predisplaypenalty</code> . <u>222</u>		
<code>\protected</code> 1465		
<code>\protected@edef</code> . . . 691		
<code>\ps@headings</code> <u>457</u>		
<code>\ps@myheadings</code> <u>514</u>		
<code>\ps@plain</code> <u>448</u>		
	Q	
	<code>\quad</code> 467, 473, 499	
	quotation (environ- ment) <u>1241</u>	
	quote (environment) <u>1251</u>	
	R	
	<code>\raggedleft</code> 1180	
	<code>\refname</code> 1609, 1610, <u>1691</u>	
	<code>\renewenvironment</code> 1106, 1136	
	<code>\rm</code> <u>1376</u>	
	<code>\romannumeral</code> 1111, 1141	
	<code>\RunningFonts</code> <u>525</u>	
	S	
	<code>\sbox</code> 1362	
	<code>\sc</code> <u>1380</u>	
	<code>\scriptsize</code> <u>150</u>	
	<code>\scshape</code> . 982, 991, 1382	
	<code>\SectFont</code> 954, <u>955</u> , 1003, 1014, 1202, 1215, 1219	
	<code>\section</code> . <u>943</u> , 1210, 1406, 1577, 1597, 1609, 1641	
	<code>\sectionmark</code> . . 464, 485, 496, 520, <u>737</u>	
	<code>\seename</code> <u>1696</u>	
	<code>\sf</code> <u>1376</u>	
	<code>\sffamily</code> 1377	
	<code>\sl</code> <u>1380</u>	
	<code>\small</code> . . . <u>108</u> , 536, 1212	
	<code>\SParaFont</code> 997, <u>998</u> , 1007, 1018	
	<code>\SSectFont</code> 967, <u>968</u> , 1004, 1015	
	<code>\SSSectFont</code> 980, <u>981</u> , 1005, 1016	
	<code>\string</code> 196–199	
	<code>\subitem</code> <u>1651</u>	
	<code>\subparagraph</code> <u>992</u>	
	<code>\subparagraphmark</code> . <u>737</u>	
	<code>\subsection</code> <u>956</u>	
	<code>\subsectionmark</code> 470, 521, <u>737</u>	
	<code>\subsubitem</code> <u>1651</u>	
	<code>\subsubsection</code> <u>969</u>	
	<code>\subsubsectionmark</code> . <u>737</u>	
	T	
	<code>\tabbingsep</code> <u>1300</u>	
	<code>\tabcolsep</code> . . 659, <u>1297</u>	
	table (environment) <u>1350</u>	
	table* (environment) <u>1350</u>	
	<code>\tablename</code> . 1349, <u>1694</u>	
	<code>\tableofcontents</code> . <u>1397</u>	
	<code>\textasteriskcentered</code> 1133	
	<code>\textbullet</code> 1131	
	<code>\textendash</code> 540, 609, 1132, 1673	
	<code>\textfloatsep</code> <u>381</u>	
	<code>\textfraction</code> <u>375</u>	
	<code>\textheight</code> . . . <u>272</u> , 357	
	<code>\textperiodcentered</code> 1134	
	<code>\textwidth</code> <u>240</u> , 324, 332, 346	
	<code>\thanks</code> 538, 550, 576, 587, 588, 597, 627, 645	
	thebibliography (envi- ronment) . . <u>1607</u>	
	<code>\thechapter</code> . . . 481, 508, <u>755</u> , 887, 889, 916, 1294, 1314, 1325, 1344	
	<code>\theenumi</code> <u>1095</u> , 1099, 1103, 1104	
	<code>\theenumii</code> <u>1095</u> , 1100, 1104	
	<code>\theenumiii</code> <u>1095</u> , 1101, 1105	
	<code>\theenumiv</code> <u>1095</u> , 1102, 1623	
	<code>\theequation</code> <u>1310</u> , 1317	
	<code>\thefigure</code> 1320, 1324, 1330	
	<code>\tfootnote</code> . . 542, 602	
	theindex (environ- ment) <u>1635</u>	
	<code>\thepage</code> 450, 452, 460, 461, 493, 516, 517	
	<code>\theparagraph</code> <u>755</u>	
	<code>\thepart</code> . . <u>755</u> , 806, 814, 815, 829, 840	
	<code>\thesection</code> . . . 467, 487, 499, <u>755</u> , 1287	
	<code>\thesubparagraph</code> . . <u>755</u>	

