

Standard Document Classes for L^AT_EX version 2e*

Copyright (C) 1992 by Leslie Lamport
Copyright (C) 1994-97 by Frank Mittelbach Johannes Braams

1997/10/10

Contents

1	The DOCSTRIP modules	3
2	Initial Code	3
3	Declaration of Options	4
3.1	Setting Paper Sizes	4
3.2	Choosing the type size	4
3.3	Two-side or one-side printing	5
3.4	Draft option	5
3.5	Titlepage option	5
3.6	openright option	5
3.7	Twocolumn printing	6
3.8	Equation numbering on the left	6
3.9	Flush left displays	6
3.10	Open bibliography	6
4	Executing Options	7
5	Loading Packages	7
6	Document Layout	7
6.1	Fonts	7
6.2	Paragraphing	10
6.3	Page Layout	12
6.3.1	Vertical spacing	12
6.3.2	The dimension of text	13
6.3.3	Margins	14
6.3.4	Footnotes	17
6.3.5	Float placement parameters	18

*This file has version number v1.3x, last revised 1997/10/10.

6.4	Page Styles	21
6.4.1	Marking conventions	21
6.4.2	Defining the page styles	21
7	Document Markup	24
7.1	The title	24
7.2	Chapters and Sections	27
7.2.1	Building blocks	27
7.2.2	Mark commands	28
7.2.3	Define Counters	28
7.2.4	Front Matter, Main Matter, and Back Matter	29
7.2.5	Parts	30
7.2.6	Chapters	33
7.2.7	Lower level headings	35
7.3	Lists	36
7.3.1	General List Parameters	36
7.3.2	Enumerate	38
7.3.3	Itemize	39
7.3.4	Description	39
7.4	Defining new environments	39
7.4.1	Abstract	39
7.4.2	Verse	40
7.4.3	Quotation	40
7.4.4	Quote	41
7.4.5	Theorem	41
7.4.6	Titlepage	41
7.4.7	Appendix	42
7.5	Setting parameters for existing environments	43
7.5.1	Array and tabular	43
7.5.2	Tabbing	43
7.5.3	Minipage	43
7.5.4	Framed boxes	43
7.5.5	Equation and eqnarray	44
7.6	Floating objects	44
7.6.1	Figure	45
7.6.2	Table	45
7.6.3	Captions	46
7.7	Font changing	47
8	Cross Referencing	48
8.1	Table of Contents, etc.	48
8.1.1	Table of Contents	48
8.1.2	List of figures	51
8.1.3	List of tables	52
8.2	Bibliography	52
8.3	The index	54

8.4	Footnotes	54
9	Initialization	55
9.1	Words	55
9.2	Date	56
9.3	Two column mode	56
9.4	The page style	57
9.5	Single or double sided printing	57

1 The DOCSTRIP modules

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

article	produce the documentclass article
report	produce the documentclass report
size10	produce the class option for 10pt
size11	produce the class option for 11pt
size12	produce the class option for 12pt
book	produce the documentclass book
bk10	produce the book class option for 10pt
bk11	produce the book class option for 11pt
bk12	produce the book class option for 12pt
driver	produce a documentation driver file

2 Initial Code

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

<code>\@ptsize</code>	This control sequence is used to store the second digit of the pointsize we are typesetting in. So, normally, it's value is one of 0, 1 or 2. 1 <code><{*article report book}></code> 2 <code>\newcommand\@ptsize{}</code>
<code>\if@restonecol</code>	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. 3 <code>\newif\if@restonecol</code>
<code>\if@titlepage</code>	A switch to indicate if a titlepage has to be produced. For the article document class the default is not to make a separate titlepage. 4 <code>\newif\if@titlepage</code> 5 <code><{article}\@titlepagefalse</code> 6 <code><{!article}\@titlepagetrue</code>
<code>\if@openright</code>	A switch to indicate if chapters must start on a right-hand page. The default for the report class is no; for the book class it's yes. 7 <code><{!article}\newif\if@openright</code>

`\if@mainmatter` The switch `\if@mainmatter`, only available in the document class `book`, indicates whether we are processing the main material in the book.

```
8 <book>\newif\if@mainmatter \@mainmattertrue
```

3 Declaration of Options

3.1 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. In compatibility mode, these (and some of the subsequent) options are disabled, as they were not present in L^AT_EX2.09.

```
9 \if@compatibility\else
10 \DeclareOption{a4paper}
11   {\setlength\paperheight {297mm}%
12    \setlength\paperwidth  {210mm}}
13 \DeclareOption{a5paper}
14   {\setlength\paperheight {210mm}%
15    \setlength\paperwidth  {148mm}}
16 \DeclareOption{b5paper}
17   {\setlength\paperheight {250mm}%
18    \setlength\paperwidth  {176mm}}
19 \DeclareOption{letterpaper}
20   {\setlength\paperheight {11in}%
21    \setlength\paperwidth  {8.5in}}
22 \DeclareOption{legalpaper}
23   {\setlength\paperheight {14in}%
24    \setlength\paperwidth  {8.5in}}
25 \DeclareOption{executivepaper}
26   {\setlength\paperheight {10.5in}%
27    \setlength\paperwidth  {7.25in}}
```

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

```
28 \DeclareOption{landscape}
29   {\setlength@tempdima {\paperheight}%
30    \setlength\paperheight {\paperwidth}%
31    \setlength\paperwidth {\@tempdima}}
32 \fi
```

3.2 Choosing the type size

The type size options are handled by defining `\@ptsize` to contain the last digit of the size in question and branching on `\ifcase` statements. This is done for historical reasons to stay compatible with other packages that use the `\@ptsize` variable to select special actions. It makes the declarations of size options less

than 10pt difficult, although one can probably use 9 and 8 assuming that a class wont define both 8pt and 18pt options.

```
33 \ifcompatibility
34   \renewcommand\@ptsize{0}
35 \else
36 \DeclareOption{10pt}{\renewcommand\@ptsize{0}}
37 \fi
38 \DeclareOption{11pt}{\renewcommand\@ptsize{1}}
39 \DeclareOption{12pt}{\renewcommand\@ptsize{2}}
```

3.3 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.

```
40 \ifcompatibility\else
41 \DeclareOption{oneside}{\@twosidefalse \@mparswitchfalse}
42 \fi
43 \DeclareOption{twoside}{\@twosidetrue \@mparswitchtrue}
```

3.4 Draft option

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

```
44 \DeclareOption{draft}{\setlength\overfullrule{5pt}}
45 \ifcompatibility\else
46 \DeclareOption{final}{\setlength\overfullrule{0pt}}
47 \fi
```

3.5 Titlepage option

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

```
48 \DeclareOption{titlepage}{\@titlepagetrue}
49 \ifcompatibility\else
50 \DeclareOption{notitlepage}{\@titlepagefalse}
51 \fi
```

3.6 openright option

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

```
52 <!article>\ifcompatibility
53 <book>\@openrighttrue
54 <!article>\else
55 <!article>\DeclareOption{openright}{\@openrighttrue}
56 <!article>\DeclareOption{openany}{\@openrightfalse}
57 <!article>\fi
```

3.7 Twocolumn printing

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

```
58 \if@compatibility\else
59 \DeclareOption{onecolumn}{\@twocolumnfalse}
60 \fi
61 \DeclareOption{twocolumn}{\@twocolumntrue}
```

3.8 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.

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

3.9 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.

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

3.10 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`.

```
64 \DeclareOption{openbib}{%
First some hook into the bibliography environment is filled.
65 \AtEndOfPackage{%
66 \renewcommand\@openbib@code{%
67 \advance\leftmargin\bibindent
68 \itemindent -\bibindent
69 \listparindent \itemindent
70 \parsep \z@
71 }%

```

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

```
72 \renewcommand\newblock{\par}}%
73 }
```

4 Executing Options

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

```
74 <*article>
75 \ExecuteOptions{letterpaper,10pt,oneside,onecolumn,final}
76 </article>
77 <*report>
78 \ExecuteOptions{letterpaper,10pt,oneside,onecolumn,final,openany}
79 </report>
80 <*book>
81 \ExecuteOptions{letterpaper,10pt,twoside,onecolumn,final,openright}
82 </book>
```

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.

```
83 \ProcessOptions
```

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

```
84 <!book>\input{size1\@ptsize.clo}
85 <book>\input{bk1\@ptsize.clo}
86 </article | report | book>
```

5 Loading Packages

The standard class files do not load additional packages.

6 Document Layout

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

6.1 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 `\belowdisplayshortskip`.

```

87 <*10pt | 11pt | 12pt>
88 \renewcommand\normalsize{%
89 <*10pt>
90   \@setfontsize\normalsize\@xpt\@xiipt
91   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
92   \abovedisplayshortskip \z@ \@plus3\p@
93   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
94 </10pt>
95 <*11pt>
96   \@setfontsize\normalsize\@xipt{13.6}%
97   \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
98   \abovedisplayshortskip \z@ \@plus3\p@
99   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
100 </11pt>
101 <*12pt>
102   \@setfontsize\normalsize\@xiipt{14.5}%
103   \abovedisplayskip 12\p@ \@plus3\p@ \@minus7\p@
104   \abovedisplayshortskip \z@ \@plus3\p@
105   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
106 </12pt>

```

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

```

107 \belowdisplayskip \abovedisplayskip
108 \let\@listi\@listI}

```

We initially choose the `normalsize` font.

```

109 \normalsize

```

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

```

110 \newcommand\small{%
111 <*10pt>
112   \@setfontsize\small\@ixpt{11}%
113   \abovedisplayskip 8.5\p@ \@plus3\p@ \@minus4\p@
114   \abovedisplayshortskip \z@ \@plus2\p@
115   \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
116   \def\@listi{\leftmargin\leftmargini

```



```

117         \topsep 4\p@ \@plus2\p@ \@minus2\p@
118         \parsep 2\p@ \@plus\p@ \@minus\p@
119         \itemsep \parsep}%
120 </10pt>
121 <*11pt>
122   \@setfontsize\small\@xpt\@xiipt
123   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
124   \abovedisplayshortskip \z@ \@plus3\p@
125   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
126   \def\@listif\leftmargin\leftmargini
127         \topsep 6\p@ \@plus2\p@ \@minus2\p@
128         \parsep 3\p@ \@plus2\p@ \@minus\p@
129         \itemsep \parsep}%
130 </11pt>
131 <*12pt>
132   \@setfontsize\small\@xipt{13.6}%
133   \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
134   \abovedisplayshortskip \z@ \@plus3\p@
135   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
136   \def\@listif\leftmargin\leftmargini
137         \topsep 9\p@ \@plus3\p@ \@minus5\p@
138         \parsep 4.5\p@ \@plus2\p@ \@minus\p@
139         \itemsep \parsep}%
140 </12pt>
141   \belowdisplayskip \abovedisplayskip
142 }

```

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

```

143 \newcommand\footnotesize{%
144 <*10pt>
145   \@setfontsize\footnotesize\@viipt{9.5}%
146   \abovedisplayskip 6\p@ \@plus2\p@ \@minus4\p@
147   \abovedisplayshortskip \z@ \@plus\p@
148   \belowdisplayshortskip 3\p@ \@plus\p@ \@minus2\p@
149   \def\@listif\leftmargin\leftmargini
150         \topsep 3\p@ \@plus\p@ \@minus\p@
151         \parsep 2\p@ \@plus\p@ \@minus\p@
152         \itemsep \parsep}%
153 </10pt>
154 <*11pt>
155   \@setfontsize\footnotesize\@ixpt{11}%
156   \abovedisplayskip 8\p@ \@plus2\p@ \@minus4\p@
157   \abovedisplayshortskip \z@ \@plus\p@
158   \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
159   \def\@listif\leftmargin\leftmargini
160         \topsep 4\p@ \@plus2\p@ \@minus2\p@
161         \parsep 2\p@ \@plus\p@ \@minus\p@
162         \itemsep \parsep}%
163 </11pt>
164 <*12pt>

```

```

165 \setfontsize\footnotesize\@xpt\@xipt
166 \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
167 \abovedisplayshortskip \z@ \@plus3\p@
168 \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
169 \def\@listif\leftmargin\leftmargini
170 \topsep 6\p@ \@plus2\p@ \@minus2\p@
171 \parsep 3\p@ \@plus2\p@ \@minus\p@
172 \itemsep \parsep}%
173 </12pt>
174 \belowdisplayskip \abovedisplayskip
175 }
176 </10pt | 11pt | 12pt>

```

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

```

\large 177 <*10pt>
\Large 178 \newcommand\scriptsize{\@setfontsize\scriptsize\@viipt\@viipt}
\LARGE 179 \newcommand\tiny{\@setfontsize\tiny\@vpt\@vipt}
\huge 180 \newcommand\large{\@setfontsize\large\@xiipt{14}}
\Huge 181 \newcommand\Large{\@setfontsize\Large\@xivpt{18}}
182 \newcommand\LARGE{\@setfontsize\LARGE\@xviipt{22}}
183 \newcommand\huge{\@setfontsize\huge\@xxpt{25}}
184 \newcommand\Huge{\@setfontsize\Huge\@xxvpt{30}}
185 </10pt>
186 <*11pt>
187 \newcommand\scriptsize{\@setfontsize\scriptsize\@viiipt{9.5}}
188 \newcommand\tiny{\@setfontsize\tiny\@vipt\@viipt}
189 \newcommand\large{\@setfontsize\large\@xiipt{14}}
190 \newcommand\Large{\@setfontsize\Large\@xivpt{18}}
191 \newcommand\LARGE{\@setfontsize\LARGE\@xviipt{22}}
192 \newcommand\huge{\@setfontsize\huge\@xxpt{25}}
193 \newcommand\Huge{\@setfontsize\Huge\@xxvpt{30}}
194 </11pt>
195 <*12pt>
196 \newcommand\scriptsize{\@setfontsize\scriptsize\@viiipt{9.5}}
197 \newcommand\tiny{\@setfontsize\tiny\@vipt\@viipt}
198 \newcommand\large{\@setfontsize\large\@xivpt{18}}
199 \newcommand\Large{\@setfontsize\Large\@xviipt{22}}
200 \newcommand\LARGE{\@setfontsize\LARGE\@xxpt{25}}
201 \newcommand\huge{\@setfontsize\huge\@xxvpt{30}}
202 \let\Huge=\huge
203 </12pt>

```

6.2 Paragraphing

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

```

204 <*article | report | book>
205 \setlength\lineskip{1\p@}

```

```

206 \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.
207 \renewcommand\baselinestretch{}

\parskip \parskip gives extra vertical space between paragraphs and \parindent is the
\parindent width of the paragraph indentation. The value of \parindent depends on whether
we are in two column mode.
208 \setlength\parskip{0\p@ \@plus \p@}
209 </article | report | book>
210 <*10pt | 11pt | 12pt>
211 \if@twocolumn
212 \setlength\parindent{1em}
213 \else
214 <10pt> \setlength\parindent{15\p@}
215 <11pt> \setlength\parindent{17\p@}
216 <12pt> \setlength\parindent{1.5em}
217 \fi
218 </10pt | 11pt | 12pt>

\smallskipamount The values for these three parameters are set in the LATEX kernel. They should
\medskipamount perhaps vary, according to the size option specified. But as they have always
\bigskipamount had the same value regardless of the size option we do not change them to stay
compatible with both LATEX 2.09 and older releases of LATEX 2ε.
219 <*10pt | 11pt | 12pt>
220 \setlength\smallskipamount{3\p@ \@plus 1\p@ \@minus 1\p@}
221 \setlength\medskipamount{6\p@ \@plus 2\p@ \@minus 2\p@}
222 \setlength\bigskipamount{12\p@ \@plus 4\p@ \@minus 4\p@}
223 </10pt | 11pt | 12pt>

\@lowpenalty The commands \nopagebreak and \nolinebreak put in penalties to discourage
\@medpenalty these breaks at the point they are put in. They use \@lowpenalty, \@medpenalty
\@highpenalty or \@highpenalty, dependent on their argument.
224 <*article | report | book>
225 \@lowpenalty 51
226 \@medpenalty 151
227 \@highpenalty 301

\clubpenalty These penalties are use to discourage club and widow lines. Because we use their
\widowpenalty default values we only show them here, commented out.
228 % \clubpenalty 150
229 % \widowpenalty 150

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

```

```

230 % \displaywidowpenalty 50
231 % \predisplaypenalty 10000
232 % \postdisplaypenalty 0

```

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

```
233 % \interlinepenalty 0
```

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

```

234 % \brokenpenalty 100
235 </article | report | book>

```

6.3 Page Layout

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

6.3.1 Vertical spacing

`\headheight` The `\headheight` is the height of the box that will contain the running head. The `\headsep` is the distance between the bottom of the running head and the top of the text. The `\topskip` is the `\baselineskip` for the first line on a page; L^AT_EX's output routine will not work properly if it has the value 0pt, so do not do that!

```

236 <*10pt | 11pt | 12pt>
237 \setlength\headheight{12\p@}
238 (!bk)\setlength\headsep {25\p@}
239 <10pt & bk>\setlength\headsep {.25in}
240 <11pt & bk>\setlength\headsep {.275in}
241 <12pt & bk>\setlength\headsep {.275in}
242 <10pt>\setlength\topskip {10\p@}
243 <11pt>\setlength\topskip {11\p@}
244 <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`.

```

245 <!bk>\setlength\footskip{30\p@}
246 <10pt & bk>\setlength\footskip{.35in}
247 <11pt & bk>\setlength\footskip{.38in}
248 <12pt & bk>\setlength\footskip{30\p@}

```

`\maxdepth` The T_EX primitive register `\maxdepth` has a function that is similar to that of `\topskip`. The register `\@maxdepth` should always contain a copy of `\maxdepth`. This is achieved by setting it internally at `\begin{document}`. In both plain T_EX and L^AT_EX 2.09 `\maxdepth` had a fixed value of 4pt; in native L^AT_EX 2e mode we let the value depend on the typesize. We set it so that `\maxdepth + \topskip = typesize × 1.5`. As it happens, in these classes `\topskip` is equal to the typesize, therefore we set `\maxdepth` to half the value of `\topskip`.

```

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

```

6.3.2 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.

```
251 \if@compatibility
252   \if@twocolumn
253     \setlength\textwidth{410\p@}
254   \else
255     <10pt&!bk>   \setlength\textwidth{345\p@}
256     <11pt&!bk>   \setlength\textwidth{360\p@}
257     <12pt&!bk>   \setlength\textwidth{390\p@}
258     <10pt & bk>   \setlength\textwidth{4.5in}
259     <11pt & bk>   \setlength\textwidth{5in}
260     <12pt & bk>   \setlength\textwidth{5in}
261   \fi
```

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

```
262 \else
```

First, we calculate the maximum `\textwidth`, which 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 more or less suitable when Computer Modern fonts are used.

```
263 \setlength\@tempdima{\paperwidth}
264 \addtolength\@tempdima{-2in}
265 <10pt>   \setlength\@tempdimb{345\p@}
266 <11pt>   \setlength\@tempdimb{360\p@}
267 <12pt>   \setlength\@tempdimb{390\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).

```
268 \if@twocolumn
269   \ifdim\@tempdima>2\@tempdimb\relax
270     \setlength\textwidth{2\@tempdimb}
271   \else
272     \setlength\textwidth{\@tempdima}
273   \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.

```
274 \else
275   \ifdim\@tempdima>\@tempdimb\relax
276     \setlength\textwidth{\@tempdimb}
277   \else
278     \setlength\textwidth{\@tempdima}
279   \fi
```

```
280 \fi
281 \fi
```

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

```
282 \if@compatibility\else
283 \settopoint\textwidth
284 \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_EX2.09. 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.

```
285 \if@compatibility
286 <10pt&!bk> \setlength\textheight{43\baselineskip}
287 <10pt & bk> \setlength\textheight{41\baselineskip}
288 <11pt> \setlength\textheight{38\baselineskip}
289 <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.

```
290 \else
291 \setlength\@tempdima{\paperheight}
```

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

```
292 \addtolength\@tempdima{-2in}
```

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

```
293 \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.

```
294 \divide\@tempdima\baselineskip
295 \@tempcnta=\@tempdima
```

From this we can calculate the height of the text.

```
296 \setlength\textheight{\@tempcnta\baselineskip}
297 \fi
```

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

```
298 \addtolength\textheight{\topskip}
```

6.3.3 Margins

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

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

```

299 \if@twocolumn
300 \setlength\marginparsep {10\p@}
301 \else
302 <10pt&!bk> \setlength\marginparsep{11\p@}
303 <11pt&!bk> \setlength\marginparsep{10\p@}
304 <12pt&!bk> \setlength\marginparsep{10\p@}
305 <bk> \setlength\marginparsep{7\p@}
306 \fi
307 <10pt | 11pt>\setlength\marginparpush{5\p@}
308 <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 309 \if@compatibility
310 <*bk>
311 <10pt> \setlength\oddsidemargin {1.5in}
312 <11pt> \setlength\oddsidemargin {1.25in}
313 <12pt> \setlength\oddsidemargin {1.25in}
314 <10pt> \setlength\evensidemargin {1.5in}
315 <11pt> \setlength\evensidemargin {1.25in}
316 <12pt> \setlength\evensidemargin {1.25in}
317 <10pt> \setlength\marginparwidth {.75in}
318 <11pt> \setlength\marginparwidth {1in}
319 <12pt> \setlength\marginparwidth {1in}
320 </bk>
321 <!*bk>
322 \if@twoside
323 <10pt> \setlength\oddsidemargin {44\p@}
324 <11pt> \setlength\oddsidemargin {36\p@}
325 <12pt> \setlength\oddsidemargin {21\p@}
326 <10pt> \setlength\evensidemargin {82\p@}
327 <11pt> \setlength\evensidemargin {74\p@}
328 <12pt> \setlength\evensidemargin {59\p@}
329 <10pt> \setlength\marginparwidth {107\p@}
330 <11pt> \setlength\marginparwidth {100\p@}
331 <12pt> \setlength\marginparwidth {85\p@}
332 \else
333 <10pt> \setlength\oddsidemargin {63\p@}
334 <11pt> \setlength\oddsidemargin {54\p@}
335 <12pt> \setlength\oddsidemargin {39.5\p@}
336 <10pt> \setlength\evensidemargin {63\p@}
337 <11pt> \setlength\evensidemargin {54\p@}

```

Values for one-sided printing:

```

338 <12pt> \setlength\evensidemargin {39.5\p@}
339 <10pt> \setlength\marginparwidth {90\p@}
340 <11pt> \setlength\marginparwidth {83\p@}
341 <12pt> \setlength\marginparwidth {68\p@}
342 \fi
343 </!bk>

```

And values for two column mode:

```

344 \if@twocolumn
345 \setlength\oddsidemargin {30\p@}
346 \setlength\evensidemargin {30\p@}
347 \setlength\marginparwidth {48\p@}
348 \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 book) we make the inner margin smaller than the outer one.

```

349 \else
350 \if@twoside
351 \setlength\@tempdima {\paperwidth}
352 \addtolength\@tempdima {-\textwidth}
353 \setlength\oddsidemargin {.4\@tempdima}
354 \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!

```

355 \setlength\marginparwidth {.6\@tempdima}
356 \addtolength\marginparwidth {-\marginparsep}
357 \addtolength\marginparwidth {-0.4in}

```

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

```

358 \else
359 \setlength\@tempdima {\paperwidth}
360 \addtolength\@tempdima {-\textwidth}
361 \setlength\oddsidemargin {.5\@tempdima}
362 \addtolength\oddsidemargin {-1in}
363 \setlength\marginparwidth {.5\@tempdima}
364 \addtolength\marginparwidth {-\marginparsep}
365 \addtolength\marginparwidth {-0.4in}
366 \addtolength\marginparwidth {-0.4in}
367 \fi

```

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


```

368 \ifdim \marginparwidth >2in
369     \setlength\marginparwidth{2in}
370 \fi

```

Having done these calculations we make them pt values.

```

371 \@settopoint\oddsidemargin
372 \@settopoint\marginparwidth

```

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

```

373 \setlength\evensidemargin {\paperwidth}
374 \addtolength\evensidemargin{-2in}
375 \addtolength\evensidemargin{-\textwidth}
376 \addtolength\evensidemargin{-\oddsidemargin}

```

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

```

377 \@settopoint\evensidemargin
378 \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.

```

379 \if@compatibility
380 <!bk> \setlength\topmargin{27pt}
381 <10pt & bk> \setlength\topmargin{.75in}
382 <11pt & bk> \setlength\topmargin{.73in}
383 <12pt & bk> \setlength\topmargin{.73in}
384 \else
385 \setlength\topmargin{\paperheight}
386 \addtolength\topmargin{-2in}
387 \addtolength\topmargin{-\headheight}
388 \addtolength\topmargin{-\headsep}
389 \addtolength\topmargin{-\textheight}
390 \addtolength\topmargin{-\footskip} % this might be wrong!

```

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

```

391 \addtolength\topmargin{-.5\topmargin}
392 \@settopoint\topmargin
393 \fi

```

6.3.4 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.

```

394 <10pt>\setlength\footnotesep{6.65\p@}
395 <11pt>\setlength\footnotesep{7.7\p@}
396 <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.

```
397 <10pt>\setlength{\skip\footins}{9\p@ \@plus 4\p@ \@minus 2\p@}
398 <11pt>\setlength{\skip\footins}{10\p@ \@plus 4\p@ \@minus 2\p@}
399 <12pt>\setlength{\skip\footins}{10.8\p@ \@plus 4\p@ \@minus 2\p@}
400 </10pt | 11pt | 12pt>
```

6.3.5 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.

```
401 <*article | report | book>
402 \setcounter{topnumber}{2}
```

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

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

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

```
404 \setcounter{bottomnumber}{1}
```

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

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

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

```
406 \setcounter{totalnumber}{3}
```

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

```
407 \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.

```
408 \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.

```
409 \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.

```
410 \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.

```
411 \renewcommand\dblfloatpagefraction{.5}
```

```
412 </article | report | book>
```

Floats on a text page

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

`\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.

```
413 <*10pt>
```

```
414 \setlength\floatsep {12\p@ \@plus 2\p@ \@minus 2\p@}
```

```
415 \setlength\textfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
```

```
416 \setlength\intextsep {12\p@ \@plus 2\p@ \@minus 2\p@}
```

```
417 </10pt>
```

```
418 <*11pt>
```

```
419 \setlength\floatsep {12\p@ \@plus 2\p@ \@minus 2\p@}
```

```
420 \setlength\textfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
```

```
421 \setlength\intextsep {12\p@ \@plus 2\p@ \@minus 2\p@}
```

```
422 </11pt>
```

```
423 <*12pt>
```

```
424 \setlength\floatsep {12\p@ \@plus 2\p@ \@minus 4\p@}
```

```
425 \setlength\textfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
```

```
426 \setlength\intextsep {14\p@ \@plus 4\p@ \@minus 4\p@}
```

```
427 </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.

```
428 <*10pt>
```

```
429 \setlength\dblfloatsep {12\p@ \@plus 2\p@ \@minus 2\p@}
```

```
430 \setlength\dbltextfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
```

```
431 </10pt>
```

```
432 <*11pt>
```

```
433 \setlength\dblfloatsep {12\p@ \@plus 2\p@ \@minus 2\p@}
```

```
434 \setlength\dbltextfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
```

```
435 </11pt>
```

```
436 <*12pt>
```

```
437 \setlength\dblfloatsep {14\p@ \@plus 2\p@ \@minus 4\p@}
```

```
438 \setlength\dbltextfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
439 </12pt>
```

Floats on their own page or column

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

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

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

```
440 <*10pt>
441 \setlength\@fptop{0\p@ \@plus 1fil}
442 \setlength\@fpsep{8\p@ \@plus 2fil}
443 \setlength\@fpbot{0\p@ \@plus 1fil}
444 </10pt>
445 <*11pt>
446 \setlength\@fptop{0\p@ \@plus 1fil}
447 \setlength\@fpsep{8\p@ \@plus 2fil}
448 \setlength\@fpbot{0\p@ \@plus 1fil}
449 </11pt>
450 <*12pt>
451 \setlength\@fptop{0\p@ \@plus 1fil}
452 \setlength\@fpsep{10\p@ \@plus 2fil}
453 \setlength\@fpbot{0\p@ \@plus 1fil}
454 </12pt>
```

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

```
\@dblfpsep 455 <*10pt>
\@dblfpbot 456 \setlength\@dblftop{0\p@ \@plus 1fil}
457 \setlength\@dblfpsep{8\p@ \@plus 2fil}
458 \setlength\@dblfpbot{0\p@ \@plus 1fil}
459 </10pt>
460 <*11pt>
461 \setlength\@dblftop{0\p@ \@plus 1fil}
462 \setlength\@dblfpsep{8\p@ \@plus 2fil}
463 \setlength\@dblfpbot{0\p@ \@plus 1fil}
464 </11pt>
465 <*12pt>
466 \setlength\@dblftop{0\p@ \@plus 1fil}
467 \setlength\@dblfpsep{10\p@ \@plus 2fil}
468 \setlength\@dblfpbot{0\p@ \@plus 1fil}
469 </12pt>
470 <*article | report | book>
```

6.4 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
`\@evenfoot` macro to produce the contents of the heading box for odd-numbered pages. It is
`\@oddfoot` called inside an `\hbox` of width `\textwidth`.

6.4.1 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.

6.4.2 Defining the page styles

The `pagestyle`s *empty* and *plain* are defined in `latex.dtx`.

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

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

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

```
473 \let\@oddfoot\@empty\let\@evenfoot\@empty
474 \def\@evenhead{\thepage\hfil\slshape\leftmark}%
475 \def\@oddhead{\slshape\rightmark}\hfil\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.

```
476 \let\@mkboth\markboth
```

For the article document class 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.

```
477 <*article>
478 \def\sectionmark##1{%
479 \markboth {\MakeUppercase{%
480 \ifnum \c@secnumdepth >\z@
481 \thesection\quad
482 \fi
483 ##1}}{}}%
484 \def\subsectionmark##1{%
485 \markright {%
486 \ifnum \c@secnumdepth >\@ne
487 \thesubsection\quad
488 \fi
489 ##1}}
490 </article>
```

In the report and book 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.

```
491 <*report | book>
492 \def\chaptermark##1{%
493 \markboth {\MakeUppercase{%
494 \ifnum \c@secnumdepth >\m@ne
495 <book> \if@mainmatter
496 \@chapapp\ \thechapter. \ %
497 <book> \fi
498 \fi
499 ##1}}{}}%
500 \def\sectionmark##1{%
501 \markright {\MakeUppercase{%
502 \ifnum \c@secnumdepth >\z@
503 \thesection. \ %
504 \fi
505 ##1}}}}
506 </report | book>
```

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...`

```

507 \else
508   \def\ps@headings{%
509     \let\@oddfoot\@empty
510     \def\@oddhead{\slshape\rightmark}\hfil\thepage}%
511     \let\@mkboth\markboth

```

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

```

512 < *article >
513   \def\sectionmark##1{%
514     \markright {\MakeUppercase{%
515       \ifnum \c@secnumdepth >\m@ne
516         \thesection\quad
517         \fi
518       ##1}}}}
519 < /article >
520 < *report | book >
521   \def\chaptermark##1{%
522     \markright {\MakeUppercase{%
523       \ifnum \c@secnumdepth >\m@ne
524 < book >         \if@mainmatter
525                 \@chapapp\ thechapter. \ %
526 < book >         \fi
527                 \fi
528                 ##1}}}}
529 < /report | book >
530 \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.

```

531 \def\ps@myheadings{%
532   \let\@oddfoot\@empty\let\@evenfoot\@empty
533   \def\@evenhead{\thepage\hfil\slshape\leftmark}%
534   \def\@oddhead{\slshape\rightmark}\hfil\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).

```

535   \let\@mkboth\@gobbletwo
536 < !article >   \let\chaptermark\@gobble
537   \let\sectionmark\@gobble
538 < article >   \let\subsectionmark\@gobble
539   }

```

7 Document Markup

7.1 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.

```
540 % \newcommand*\title}[1]{\gdef\@title{#1}}
541 % \newcommand*\author}[1]{\gdef\@author{#1}}
542 % \newcommand*\date}[1]{\gdef\@date{#1}}
```

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

```
543 % \date{\today}
```

`\maketitle` The definition of `\maketitle` depends on whether a separate title page is made.
This is the default for the report and book document classes, but for the article
class it is optional.

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; these changes affect all footnotes.

```
544 \if@titlepage
545 \newcommand\maketitle{\begin{titlepage}%
546 \let\footnotesize\small
547 \let\footnoterule\relax
548 \let \footnote \thanks
```

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 by the `titlepage`
environment to keep the behaviour of L^AT_EX 2.09 style files.)

```
549 \null\vfil
550 \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.

```
551 \begin{center}%
552   {\LARGE \@title \par}%
553   \vskip 3em%
554   {\large
555     \lineskip .75em%
556     \begin{tabular}[t]{c}%
557       \@author
558     \end{tabular}\par}%
559   \vskip 1.5em%
560   {\large \@date \par}%           % Set date in \large size.
561 \end{center}\par
```


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

```
562 \@thanks
563 \vfil\null
564 \end{titlepage}%
```

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

```
565 \setcounter{footnote}{0}%
566 \global\let\thanks\relax
567 \global\let\maketitle\relax
568 \global\let\@thanks\@empty
569 \global\let\@author\@empty
570 \global\let\@date\@empty
571 \global\let\@title\@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.

```
572 \global\let\title\relax
573 \global\let\author\relax
574 \global\let\date\relax
575 \global\let\and\relax
576 }
```

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).

```
577 \else
578 \newcommand\maketitle{\par
579 \begingroup
580 \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
581 \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
582 \long\def\@makefntext##1{\parindent 1em\noindent
583 \hb@xt@1.8em{\%
584 \hss\@textsuperscript{\normalfont\@thefnmark}}##1}%
```

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`.

```
585 \iftwocolumn
586 \ifnum \col@number=\@ne
587 \@maketitle
588 \else
589 \twocolumn[\@maketitle]%
590 \fi
591 \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.

```
592     \newpage
593     \global\@topnum\z@    % Prevents figures from going at top of page.
594     \@maketitle
595     \fi
```

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

```
596     \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.

```
597 \endgroup
598 \setcounter{footnote}{0}%
599 \global\let\thanks\relax
600 \global\let\maketitle\relax
601 \global\let\@maketitle\relax
602 \global\let\@thanks\@empty
603 \global\let\@author\@empty
604 \global\let\@date\@empty
605 \global\let\@title\@empty
606 \global\let\title\relax
607 \global\let\author\relax
608 \global\let\date\relax
609 \global\let\and\relax
610 }
```

`\@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.

```
611 \def\@maketitle{%
612   \newpage
613   \null
614   \vskip 2em%
615   \begin{center}%
616     \let \footnote \thanks
617     {\LARGE \@title \par}%
618     \vskip 1.5em%
619     {\large
620      \lineskip .5em%
621      \begin{tabular}[t]{c}%
622        \@author
623      \end{tabular}\par}%
624     \vskip 1em%
625     {\large \@date}%
626   \end{center}%
627   \par
628   \vskip 1.5em}
```

7.2 Chapters and Sections

7.2.1 Building blocks

The definitions in this part of the class file make use of two internal macros, `\@startsection` and `\secdef`. 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>** <= 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.

***** 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 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*{...}

```

7.2.2 Mark commands

<code>\chaptermark</code>	Default initializations of <code>\...mark</code> commands. These commands are used in the
<code>\sectionmark</code>	definition of the page styles (see section 6.4.2) Most of them are already defined
<code>\subsectionmark</code>	by <code>latex.dtx</code> , so they are only shown here.
<code>\subsubsectionmark</code>	630 <code>\newcommand*\chaptermark[1]{}</code>
<code>\paragraphmark</code>	631 <code>\newcommand*\sectionmark[1]{}</code>
<code>\subparagraphmark</code>	632 <code>\newcommand*\subsectionmark[1]{}</code>
	633 <code>\newcommand*\subsubsectionmark[1]{}</code>
	634 <code>\newcommand*\paragraphmark[1]{}</code>
	635 <code>\newcommand*\subparagraphmark[1]{}</code>

7.2.3 Define Counters

<code>\c@secnumdepth</code>	The value of the counter <code>secnumdepth</code> gives the depth of the highest-level sectioning command that is to produce section numbers.
	636 <code>\setcounter{secnumdepth}{3}</code>
	637 <code>\setcounter{secnumdepth}{2}</code>
<code>\c@part</code>	These counters are used for the section numbers. The macro
<code>\c@chapter</code>	<code>\newcounter{<newctr>}[<oldctr>]</code>
<code>\c@section</code>	defines <code><newctr></code> to be a counter, which is reset to zero when counter <code><oldctr></code> is
<code>\c@subsection</code>	stepped. Counter <code><oldctr></code> must already be defined.
<code>\c@subsubsection</code>	638 <code>\newcounter {part}</code>
<code>\c@paragraph</code>	639 <code>\newcounter {section}</code>
<code>\c@subparagraph</code>	640 <code>\newcounter {chapter}</code>
	641 <code>\newcounter {section}[chapter]</code>
	642 <code>\newcounter {section}[chapter]</code>
	643 <code>\newcounter {subsection}[section]</code>
	644 <code>\newcounter {subsubsection}[subsection]</code>
	645 <code>\newcounter {paragraph}[subsubsection]</code>
	646 <code>\newcounter {paragraph}[subsubsection]</code>
	647 <code>\newcounter {subparagraph}[paragraph]</code>
<code>\thepart</code>	For any counter <code>CTR</code> , <code>\theCTR</code> is a macro that defines the printed version of
<code>\thechapter</code>	counter <code>CTR</code> . It is defined in terms of the following macros:
<code>\thesection</code>	<code>\arabic{COUNTER}</code> prints the value of <code>COUNTER</code> as an arabic numeral.
<code>\thesubsection</code>	<code>\roman{COUNTER}</code> prints the value of <code>COUNTER</code> as a lowercase roman num-
<code>\thesubsubsection</code>	beral.
<code>\theparagraph</code>	
<code>\thesubparagraph</code>	

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

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

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

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

```
648 \renewcommand \thepart {\@Roman\c@part}
649 \langle article \rangle \renewcommand \thesection {\@arabic\c@section}
650 \langle *report | book \rangle
651 \renewcommand \thechapter {\@arabic\c@chapter}
652 \renewcommand \thesection {\thechapter.\@arabic\c@section}
653 \langle /report | book \rangle
654 \renewcommand \thesubsection {\thesection.\@arabic\c@subsection}
655 \renewcommand \thesubsubsection {\thesubsection.\@arabic\c@subsubsection}
656 \renewcommand \theparagraph {\thesubsubsection.\@arabic\c@paragraph}
657 \renewcommand \thesubparagraph {\theparagraph.\@arabic\c@subparagraph}
```

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

```
658 \langle report | book \rangle \newcommand \@chapapp {\chaptername}
```

7.2.4 Front Matter, Main Matter, and Back Matter

A book contains these three (logical) sections. The switch `\@mainmatter` 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.

```
659 \langle *book \rangle
660 \newcommand \frontmatter {%
661   \if@openright
662     \cleardoublepage
663   \else
664     \clearpage
665   \fi
666   \@mainmatterfalse
667   \pagenumbering {roman}}
```

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

```
668 \newcommand \mainmatter {%
669   \if@openright
670     \cleardoublepage
671   \else
672     \clearpage
```

```

673 \fi
674 \@mainmattertrue
675 \pagenumbering{arabic}}

```

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

```

676 \newcommand\backmatter{%
677 \if@openright
678 \cleardoublepage
679 \else
680 \clearpage
681 \fi
682 \@mainmatterfalse}
683 </book>

```

7.2.5 Parts

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

In the article class the definition of `\part` is rather simple; we start a new paragraph, add a little white space, suppress the indentation of the first paragraph and make use of `\secdef`.

```

684 <*article>
685 \newcommand\part{\par
686 \addvspace{4ex}%
687 \@afterindentfalse
688 \secdef\@part\@spart}
689 </article>

```

For the report and book classes we things a bit different.

We start a new (righthand) page and use the *plain* pagestyle.

```

690 <*report | book>
691 \newcommand\part{%
692 \if@openright
693 \cleardoublepage
694 \else
695 \clearpage
696 \fi
697 \thispagestyle{plain}%

```

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.

```

698 \if@twocolumn
699 \onecolumn
700 \@tempswatrue
701 \else
702 \@tempswafalse
703 \fi

```

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

```

704 \null\vfil

```

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

```
705 \secdef\@part\@spart}
706 </report | book>
```

`\@part` This macro does the actual formatting of the title of the part. Again the macro is differently defined for the article document class than for the document classes report and book.

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

```
707 <*article>
708 \def\@part[#1]#2{%
709     \ifnum \c@secnumdepth >\m@ne
710         \refstepcounter{part}%
711         \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
712     \else
713         \addcontentsline{toc}{part}{#1}%
714     \fi
```

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

```
715     {\parindent \z@ \raggedright
716     \interlinepenalty \@M
717     \normalfont
```

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

```
718     \ifnum \c@secnumdepth >\m@ne
719         \Large\bfseries \partname~\thepart
720         \par\nobreak
721     \fi
722     \huge \bfseries #2%
```

Now we empty the mark registers, leave some white space and let `\@afterheading` take care of suppressing the indentation.

```
723     \markboth{}{\par}%
724     \nobreak
725     \vskip 3ex
726     \@afterheading}
727 </article>
```

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

```
728 <*report | book>
729 \def\@part[#1]#2{%
730     \ifnum \c@secnumdepth >-2\relax
731         \refstepcounter{part}%
732         \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
733     \else
734         \addcontentsline{toc}{part}{#1}%
735     \fi
```

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

```
736 \markboth{}{}%
737 {\centering
738 \interlinepenalty \@M
739 \normalfont
```

When this is a numbered part we have to print the number.

```
740 \ifnum \c@secnumdepth >-2\relax
741 \huge\bfseries \partname~\thepart
742 \par
```

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

```
743 \vskip 20\p@
744 \fi
745 \Huge \bfseries #2\par}%
746 \@endpart}
747 </report | book>
```

`\@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 article document class and in the report and book document classes are similar as they were for `\@part`.

```
748 <*article>
749 \def\@spart#1{%
750 \parindent \z@ \raggedright
751 \interlinepenalty \@M
752 \normalfont
753 \huge \bfseries #1\par}%
754 \nobreak
755 \vskip 3ex
756 \@afterheading}
757 </article>
758 <*report | book>
759 \def\@spart#1{%
760 {\centering
761 \interlinepenalty \@M
762 \normalfont
763 \Huge \bfseries #1\par}%
764 \@endpart}
765 </report | book>
```

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

```
766 <*report | book>
767 \def\@endpart{\vfil\newpage
```


Then, when we are in twosided mode, we produce a completely blank page.

```

768         \if@twoside
769         \null
770         \thispagestyle{empty}%
771         \newpage
772         \fi

```

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

```

773         \if@tempwa
774         \twocolumn
775         \fi}
776 \</report | book>

```

7.2.6 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*.

```

777 <*report | book>
778 \newcommand\chapter{\if@openright\cleardoublepage\else\clearpage\fi
779         \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.

```

780         \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.

```

781         \@afterindentfalse
782         \secdef\@chapter\@schapter}

```

`\@chapter` This macro is called when we have a numbered chapter. When `secnumdepth` is larger than `-1` and, in the book 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.

```

783 \def\@chapter[#1]#2{\ifnum \c@secnumdepth >\m@ne
784 (book)         \if@mainmatter
785                 \refstepcounter{chapter}%
786                 \typeout{\@chapapp\space\thechapter.}%
787                 \addcontentsline{toc}{chapter}%
788                 {\protect\numberline{\thechapter}#1}%
789 <*book>
790                 \else
791                 \addcontentsline{toc}{chapter}{#1}%
792                 \fi
793 </book>
794                 \else
795                 \addcontentsline{toc}{chapter}{#1}%
796                 \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.

```
797         \chaptermark{#1}%
798         \addtocontents{lof}{\protect\addvspace{10\p@}}%
799         \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.

```
800         \iftwocolumn
801             \topnewpage[\@makechapterhead{#2}]%
802         \else
803             \@makechapterhead{#2}%
804             \@afterheading
805         \fi}
```

`\@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.

```
806 \def\@makechapterhead#1{%
807   \vspace*{50\p@}%
808   {\parindent \z@ \raggedright \normalfont
```

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

```
809     \ifnum \c@secnumdepth >\m@ne
810 (book)     \ifmainmatter
811             \huge\bfseries \@chapapp\space \thechapter
812             \par\nobreak
813             \vskip 20\p@
814 (book)     \fi
815     \fi}
```

Now we set the title in a large bold font. We prevent a pagebreak from occurring in the middle of or after the title. Finally we leave some whitespace before the text begins.

```
816     \interlinepenalty\@M
817     \Huge \bfseries #1\par\nobreak
818     \vskip 40\p@
819   }}
```

`\@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.

```
820 \def\@schapter#1{\iftwocolumn
821     \topnewpage[\@makeschapterhead{#1}]%
```

```

822             \else
823             \@makeschapterhead{#1}%
824             \@afterheading
825             \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.

```

826 \def\@makeschapterhead#1{%
827   \vspace*{50\p@}%
828   {\parindent \z@ \raggedright
829     \normalfont
830     \interlinepenalty\@M
831     \Huge \bfseries #1\par\nobreak
832     \vskip 40\p@
833   }}
834 \</report | book>

```

7.2.7 Lower level headings

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

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

```

835 \newcommand\section{\@startsection {section}{1}{\z@}%
836                               {-3.5ex \@plus -1ex \@minus -.2ex}%
837                               {2.3ex \@plus .2ex}%
838                               {\normalfont\Large\bfseries}}

```

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

```

839 \newcommand\subsection{\@startsection{subsection}{2}{\z@}%
840                               {-3.25ex\@plus -1ex \@minus -.2ex}%
841                               {1.5ex \@plus .2ex}%
842                               {\normalfont\large\bfseries}}

```

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

```

843 \newcommand\subsubsection{\@startsection{subsubsection}{3}{\z@}%
844                               {-3.25ex\@plus -1ex \@minus -.2ex}%
845                               {1.5ex \@plus .2ex}%
846                               {\normalfont\normalsize\bfseries}}

```

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

```

847 \newcommand\paragraph{\@startsection{paragraph}{4}{\z@}%
848                               {3.25ex \@plus 1ex \@minus .2ex}%
849                               {-1em}%
850                               {\normalfont\normalsize\bfseries}}

```

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

```
851 \newcommand\subparagraph{\@startsection{subparagraph}{5}{\parindent}%
852                                     {3.25ex \@plus1ex \@minus .2ex}%
853                                     {-1em}%
854                                     {\normalfont\normalsize\bfseries}}
```

7.3 Lists

7.3.1 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', 'i', ... , 'vi'. (I.e., `\@listiii` is called for a third-level list.) By convention, `\@listK` should set `\leftmargin` to `\leftmarginK`.

```
\leftmargin When we are in two column mode some of the margins are set somewhat smaller.
\leftmargini 855 \if@twocolumn
\leftmarginii 856 \setlength\leftmargini {2em}
\leftmarginiii 857 \else
\leftmarginiv 858 \setlength\leftmargini {2.5em}
\leftmarginv 859 \fi
```

`\leftmarginvi` Until the whole of the parameter setting in these files is rationalised, we need to set the value of `\leftmargin` at this outer level.

```
860 \leftmargin \leftmargini
```

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

```
861 \setlength\leftmarginii {2.2em}
862 \setlength\leftmarginiii {1.87em}
863 \setlength\leftmarginiv {1.7em}
864 \if@twocolumn
865 \setlength\leftmarginv {.5em}
866 \setlength\leftmarginvi {.5em}
867 \else
868 \setlength\leftmarginv {1em}
869 \setlength\leftmarginvi {1em}
870 \fi
```

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

```
871 \setlength \labelsep {.5em}
872 \setlength \labelwidth{\leftmargini}
873 \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`.

```

874 </article | report | book>
875 <10pt>\setlength\partopsep{2\p@ \@plus 1\p@ \@minus 1\p@}
876 <11pt>\setlength\partopsep{3\p@ \@plus 1\p@ \@minus 1\p@}
877 <12pt>\setlength\partopsep{3\p@ \@plus 2\p@ \@minus 2\p@}

```

`\@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.

```

878 <*article | report | book>
879 \@beginparpenalty -\@lowpenalty
880 \@endparpenalty -\@lowpenalty
881 \@itempenalty -\@lowpenalty
882 </article | report | book>

```

`\@listi` `\@listi` defines the values of `\leftmargin`, `\parsep`, `\topsep`, `\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.

```

883 <*10pt | 11pt | 12pt>
884 \def\@listi{\leftmargin\leftmarginI
885 <*10pt>
886         \parsep 4\p@ \@plus2\p@ \@minus\p@
887         \topsep 8\p@ \@plus2\p@ \@minus4\p@
888         \itemsep4\p@ \@plus2\p@ \@minus\p@}
889 </10pt>
890 <*11pt>
891         \parsep 4.5\p@ \@plus2\p@ \@minus\p@
892         \topsep 9\p@ \@plus3\p@ \@minus5\p@
893         \itemsep4.5\p@ \@plus2\p@ \@minus\p@}
894 </11pt>
895 <*12pt>
896         \parsep 5\p@ \@plus2.5\p@ \@minus\p@
897         \topsep 10\p@ \@plus4\p@ \@minus6\p@
898         \itemsep5\p@ \@plus2.5\p@ \@minus\p@}
899 </12pt>
900 \let\@listI\@listi

```

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

```

901 \@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 902 \def\@listii {\leftmargin\leftmarginii

```

```

903         \labelwidth\leftmarginii
904         \advance\labelwidth-\labelsep
905 < *10pt >
906         \topsep    4\p@ \@plus2\p@ \@minus\p@
907         \parsep    2\p@ \@plus\p@  \@minus\p@
908 < /10pt >
909 < *11pt >
910         \topsep    4.5\p@ \@plus2\p@ \@minus\p@
911         \parsep    2\p@  \@plus\p@  \@minus\p@
912 < /11pt >
913 < *12pt >
914         \topsep    5\p@  \@plus2.5\p@ \@minus\p@
915         \parsep    2.5\p@ \@plus\p@  \@minus\p@
916 < /12pt >
917         \itemsep    \parsep}
918 \def\@listiii{\leftmargin\leftmarginiii
919         \labelwidth\leftmarginiii
920         \advance\labelwidth-\labelsep
921 < 10pt >         \topsep    2\p@ \@plus\p@\@minus\p@
922 < 11pt >         \topsep    2\p@ \@plus\p@\@minus\p@
923 < 12pt >         \topsep    2.5\p@\@plus\p@\@minus\p@
924         \parsep    \z@
925         \partopsep \p@ \@plus\z@ \@minus\p@
926         \itemsep    \topsep}
927 \def\@listiv {\leftmargin\leftmarginiv
928         \labelwidth\leftmarginiv
929         \advance\labelwidth-\labelsep}
930 \def\@listv  {\leftmargin\leftmarginv
931         \labelwidth\leftmarginv
932         \advance\labelwidth-\labelsep}
933 \def\@listvi {\leftmargin\leftmarginvi
934         \labelwidth\leftmarginvi
935         \advance\labelwidth-\labelsep}
936 < /10pt | 11pt | 12pt >

```

7.3.2 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 937 < *article | report | book >
\theenumiv 938 \renewcommand\theenumi{\@arabic\c@enumi}
939 \renewcommand\theenumii{\@alph\c@enumii}
940 \renewcommand\theenumiii{\@roman\c@enumiii}
941 \renewcommand\theenumiv{\@Alph\c@enumiv}

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

```

```

942 \newcommand\labelenumi{\theenumi.}
943 \newcommand\labelenumii{(\theenumii)}
944 \newcommand\labelenumiii{\theenumiii.}
945 \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 946 \renewcommand\p@enumii{\theenumi}
          947 \renewcommand\p@enumiii{\theenumi(\theenumii)}
          948 \renewcommand\p@enumiv{\p@enumiii\theenumiii}

```

7.3.3 Itemize

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

```

949 \newcommand\labelitemi{\textbullet}
950 \newcommand\labelitemii{\normalfont\bfseries \textendash}
951 \newcommand\labelitemiii{\textasteriskcentered}
952 \newcommand\labelitemiv{\textperiodcentered}

```

7.3.4 Description

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

```

953 \newenvironment{description}
954         {\list{}{\labelwidth\z@ \itemindent-\leftmargin
955                 \let\makelabel\descriptionlabel}}
956         {\endlist}

```

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

```

957 \newcommand*\descriptionlabel[1]{\hspace\labelsep
958                               \normalfont\bfseries #1}

```

7.4 Defining new environments

7.4.1 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.

```

959 % \changes{v1.3m}{1995/10/23}{Added setting of \cs{beginparpenalty} to
960 %   discourage page break before abstract heading.}
961 <*article | report>
962 \if@titlepage
963   \newenvironment{abstract}{%
964     \titlepage

```

```

965     \null\vfil
966     \@beginparpenalty\@lowpenalty
967     \begin{center}%
968         \bfseries \abstractname
969     \@endparpenalty\@M
970     \end{center}}%
971     {\par\vfil\null\endtitlepage}

```

When we are not making a separate titlepage—the default for the article document class—we have to check if we are in twocolumn mode. In that case the abstract is as a `\section*`, otherwise the quotation environment is used to typeset the abstract.

```

972 \else
973   \newenvironment{abstract}{%
974     \if@twocolumn
975       \section*{\abstractname}%
976     \else
977       \small
978       \begin{center}%
979         {\bfseries \abstractname\vspace{-.5em}\vspace{\z@}}%
980       \end{center}%
981       \quotation
982     \fi}
983   {\if@twocolumn\else\endquotation\fi}
984 \fi
985 </article | report>

```

7.4.2 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`.

```

986 \newenvironment{verse}
987     {\let\\ \@centercr
988     \list{}{\itemsep      \z@
989             \itemindent   -1.5em%
990             \listparindent\itemindent
991             \rightmargin  \leftmargin
992             \advance\leftmargin 1.5em}%
993     \item\relax}
994     {\endlist}

```

7.4.3 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.

```

995 \newenvironment{quotation}
996     {\list{}{\listparindent 1.5em%

```



```

997             \itemindent   \listparindent
998             \rightmargin  \leftmargin
999             \parsep       \z@ \@plus\p@}%
1000         \item\relax}
1001     {\endlist}

```

7.4.4 Quote

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

```

1002 \newenvironment{quote}
1003     {\list{}{\rightmargin\leftmargin}%
1004     \item\relax}
1005     {\endlist}

```

7.4.5 Theorem

This document class does not define it's own theorem environments, the defaults, supplied by `latex.dtx` are available.

7.4.6 Titlepage

`titlepage` In the normal environments, the titlepage environment does nothing but start and end a page, and inhibit page numbers. In the report style, it also resets the page number to one, and then sets it back to one at the end. In compatibility mode, it sets 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.

First we do give the definition for compatibility mode.

```

1006 \if@compatibility
1007 \newenvironment{titlepage}
1008     {%
1009 <book>     \cleardoublepage
1010     \if@twocolumn
1011         \@restonecoltrue\onecolumn
1012     \else
1013         \@restonecolfalse\newpage
1014     \fi
1015     \thispagestyle{empty}%
1016     \setcounter{page}\z@
1017     }%
1018     {\if@restonecol\twocolumn \else \newpage \fi
1019     }

```

And here is the one for native L^AT_EX 2_ε.

```

1020 \else
1021 \newenvironment{titlepage}
1022     {%
1023 <book>     \cleardoublepage

```

```

1024     \if@twocolumn
1025         \@restonecoltrue\onecolumn
1026     \else
1027         \@restonecolfalse\newpage
1028     \fi
1029     \thispagestyle{empty}%
1030     \setcounter{page}\@ne
1031 }%
1032 {\if@restonecol\twocolumn \else \newpage \fi

```

If we are not in two-side mode the first page after the title page should also get page number 1.

```

1033     \if@twoside\else
1034         \setcounter{page}\@ne
1035     \fi
1036 }
1037 \fi

```

7.4.7 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 article document class the `\appendix` command must do the following:

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

```

1038 <*article>
1039 \newcommand\appendix{\par
1040   \setcounter{section}{0}%
1041   \setcounter{subsection}{0}%
1042   \renewcommand\thesection{\@Alph\c@section}}
1043 </article>

```

In the report and book 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.

```

1044 <*report | book>
1045 \newcommand\appendix{\par
1046   \setcounter{chapter}{0}%
1047   \setcounter{section}{0}%
1048   \renewcommand\@chapapp{\appendixname}%
1049   \renewcommand\thechapter{\@Alph\c@chapter}}
1050 </report | book>

```

7.5 Setting parameters for existing environments

7.5.1 Array and tabular

`\arraycolsep` The columns in an array environment are separated by `2\arraycolsep`.
1051 `\setlength\arraycolsep{5\p@}`

`\tabcolsep` The columns in a tabular environment are separated by `2\tabcolsep`.
1052 `\setlength\tabcolsep{6\p@}`

`\arrayrulewidth` The width of rules in the array and tabular environments is given by `\arrayrulewidth`.
1053 `\setlength\arrayrulewidth{.4\p@}`

`\doublerulesep` The space between adjacent rules in the array and tabular environments is given by `\doublerulesep`.
1054 `\setlength\doublerulesep{2\p@}`

7.5.2 Tabbing

`\tabbingsep` This controls the space that the `\'` command puts in. (See L^AT_EX manual for an explanation.)
1055 `\setlength\tabbingsep{\labelsep}`

7.5.3 Minipage

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

`\@mpfootins` Minipages have their own footnotes; `\skip\@mpfootins` plays same rôle for footnotes in a minipage as `\skip\footins` does for ordinary footnotes.
1056 `\skip\@mpfootins = \skip\footins`

7.5.4 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`.
1057 `\setlength\fboxsep{3\p@}`
1058 `\setlength\fboxrule{.4\p@}`

7.5.5 Equation and eqnarray

`\theequation` When within chapters, the equation counter will be reset at the 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.

```
1059 <article>\renewcommand \theequation {\@arabic\c@equation}
1060 <*report | book>
1061 \@addtoreset {equation}{chapter}
1062 \renewcommand\theequation
1063   {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@equation}
1064 (/report | book)
```

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

```
1065 % \setlength\jot{3pt}
```

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

```
1066 % \def\@eqnnum{(\theequation)}
```

7.6 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 `\endfloat`, which are defined in `latex.dtx`.

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

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

7.6.1 Figure

Here is the implementation of the figure environment.

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

```
1067 <*article>
1068 \newcounter{figure}
1069 \renewcommand \thefigure {\@arabic\c@figure}
1070 </article>
1071 <*report | book>
1072 \newcounter{figure}[chapter]
1073 \renewcommand \thefigure
1074     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@figure}
1075 </report | book>
```

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

```
\ftype@figure 1076 \def\fps@figure{tbp}
\ext@figure 1077 \def\ftype@figure{1}
\num@figure 1078 \def\ext@figure{lof}
1079 \def\fnum@figure{\figurename~\thefigure}
```

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

```
1080 \newenvironment{figure}
1081     {\@float{figure}}
1082     {\end@float}
1083 \newenvironment{figure*}
1084     {\@dblfloat{figure}}
1085     {\end@dblfloat}
```

7.6.2 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 report and book document classes tables within chapters are numbered per chapter.

```
1086 <*article>
1087 \newcounter{table}
1088 \renewcommand \thetable {\@arabic\c@table}
1089 </article>
1090 <*report | book>
```

```

1091 \newcounter{table}[chapter]
1092 \renewcommand \thetable
1093     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@table}
1094 (/report | book)

```

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

```

\ftype@table 1095 \def\fps@table{tbp}
\ext@table 1096 \def\ftype@table{2}
\num@table 1097 \def\ext@table{lot}
1098 \def\fnm@table{\tablename~\thetable}

```

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

```

1099 \newenvironment{table}
1100     {\@float{table}}
1101     {\end@float}
1102 \newenvironment{table*}
1103     {\@dblfloat{table}}
1104     {\end@dblfloat}

```

7.6.3 Captions

`\@makecaption` The `\caption` command calls `\@makecaption` to format the caption of floating objects. It gets two arguments, $\langle number \rangle$, the number of the floating object and $\langle text \rangle$, the text of the caption. Usually $\langle number \rangle$ 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.

```

1105 \newlength\abovecaptionskip
1106 \newlength\belowcaptionskip
1107 \setlength\abovecaptionskip{10\p@}
1108 \setlength\belowcaptionskip{0\p@}

```

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

```

1109 \long\def\@makecaption#1#2{%
1110     \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.

```

1111     \sbox\@tempboxa{#1: #2}%

```

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

```

1112     \ifdim \wd\@tempboxa >\hsize
1113         #1: #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.

```

1114 \else
1115   \global \@minipagefalse
1116   \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
1117 \fi
1118 \vskip\belowcaptionskip}

```

7.7 Font changing

Here we supply the declarative font changing commands that were common in \LaTeX version 2.09 and earlier. These commands work in text mode *and* in math mode. They are provided for compatibility, but one should start using the `\text...` and `\math...` commands instead. These commands are defined using `\DeclareTextFontCommand`, 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.

```

\sf 1119 \DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
      1120 \DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
      1121 \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.

```

1122 \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 `\sc` in math mode. However, we do warn the user that the selection will not have any effect. One should use `\upshape` to explicitly change back to the upright shape.

```

1123 \DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
1124 \DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
1125 \DeclareOldFontCommand{\sc}{\normalfont\scshape}{\@nomath\sc}

```

`\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’.

```

1126 \DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}
1127 \DeclareRobustCommand*\mit{\@fontswitch\relax\mathnormal}

```

8 Cross Referencing

8.1 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 the `\@dottedtocline` command, which works as follows.

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

```
1128 \newcommand\@pnumwidth{1.55em}
```

```
1129 \newcommand\@tocrmarg{2.55em}
```

```
1130 \newcommand\@dotsep{4.5}
```

```
1131 \article\setcounter{tocdepth}{3}
```

```
1132 \!article\setcounter{tocdepth}{2}
```

8.1.1 Table of Contents

`\tableofcontents` This macro is used to request that \LaTeX produces a table of contents. In the report and book document classes the tables of contents, figures etc. are always set in single-column style.


```

1133 \newcommand\tableofcontents{%
1134 <*report | book>
1135     \if@twocolumn
1136         \@restonecoltrue\onecolumn
1137     \else
1138         \@restonecolfalse
1139     \fi

```

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

```

1140     \chapter*{\contentsname
1141 </report | book>
1142 <article>     \section*{\contentsname
1143             \@mkboth{%
1144                 \MakeUppercase\contentsname}\MakeUppercase\contentsname}}%

```

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

```

1145     \@starttoc{toc}%
1146 <!article>     \if@restonecol\twocolumn\fi
1147     }

```

`\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.

```

1148 \newcommand*\l@part[2]{%
1149     \ifnum \c@tocdepth >-2\relax
1150 <article>     \addpenalty\@secpenalty
1151 <!article>     \addpenalty{-\@highpenalty}%
1152     \addvspace{2.25em \@plus\p@}%
1153     \begingroup

```

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

```

1154         \parindent \z@ \rightskip \@pnumwidth
1155         \parfillskip -\@pnumwidth

```

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

```

1156         {\leavevmode
1157         \large \bfseries #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par

```

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

```

1158         \nobreak
1159 <article>     \if@compatibility
1160         \global\@nobreaktrue

```

```

1161     \everypar{\global\@nobeakfalse\everypar{}}%
1162 <article>     \fi
1163     \endgroup
1164     \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.

```

1165 <*report | book>
1166 \newcommand*\l@chapter[2]{%
1167   \ifnum \c@tocdepth >\m@ne
1168     \addpenalty{-\@highpenalty}%
1169     \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.

```

1170   \setlength\@tempdima{1.5em}%
1171   \begingroup
1172     \parindent \z@ \rightskip \@pnumwidth
1173     \parfillskip -\@pnumwidth

```

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

```

1174     \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.

```

1175     \advance\leftskip\@tempdima
1176     \hskip -\leftskip
1177     #1\nobreak\hfil \nobreak\hb@xt@\@pnumwidth{\hss #2}\par
1178     \penalty\@highpenalty
1179   \endgroup
1180   \fi}
1181 </report | book>

```

`\l@section` In the article document class the entry in the table of contents for sections looks much like the chapter entries for the report and book 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.

```

1182 <*article>
1183 \newcommand*\l@section[2]{%
1184   \ifnum \c@tocdepth >\z@
1185     \addpenalty\@secpenalty
1186     \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.

```

1187 \setlength\@tempdima{1.5em}%
1188 \begingroup
1189 \parindent \z@ \rightskip \@pnumwidth
1190 \parfillskip -\@pnumwidth

```

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

```

1191 \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.

```

1192 \advance\leftskip\@tempdima
1193 \hskip -\leftskip
1194 #1\nobreak\hfil \nobreak\hb@xt@\@pnumwidthf\hss #2\par
1195 \endgroup
1196 \fi}
1197 </article>

```

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

```

1198 <*report | book>
1199 \newcommand*\l@section{\@dottedtocline{1}{1.5em}{2.3em}}
1200 </report | book>

```

```

\l@subsection All lower level entries are defined using the macro \@dottedtocline (see above).
\l@subsubsection 1201 <*article>
\l@paragraph 1202 \newcommand*\l@subsection{\@dottedtocline{2}{1.5em}{2.3em}}
\l@subparagraph 1203 \newcommand*\l@subsubsection{\@dottedtocline{3}{3.8em}{3.2em}}
1204 \newcommand*\l@paragraph{\@dottedtocline{4}{7.0em}{4.1em}}
1205 \newcommand*\l@subparagraph{\@dottedtocline{5}{10em}{5em}}
1206 </article>
1207 <*report | book>
1208 \newcommand*\l@subsection{\@dottedtocline{2}{3.8em}{3.2em}}
1209 \newcommand*\l@subsubsection{\@dottedtocline{3}{7.0em}{4.1em}}
1210 \newcommand*\l@paragraph{\@dottedtocline{4}{10em}{5em}}
1211 \newcommand*\l@subparagraph{\@dottedtocline{5}{12em}{6em}}
1212 </report | book>

```

8.1.2 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`.

```

1213 \newcommand\listoffigures{%
1214 <*report | book>
1215 \if@twocolumn
1216 \@restonecoltrue\onecolumn
1217 \else
1218 \@restonecolfalse
1219 \fi
1220 \chapter*{\listfigurename

```

```

1221 </report | book>
1222 <article>    \section*{\listfigurename
1223             \mkboth{\MakeUppercase\listfigurename}%
1224                 {\MakeUppercase\listfigurename}}%
1225             \@starttoc{lof}%
1226 <report | book>    \if@restonecol\twocolumn\fi
1227     }

```

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

```
1228 \newcommand*\l@figure{\@dottedtocline{1}{1.5em}{2.3em}}
```

8.1.3 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`.

```

1229 \newcommand\listoftables{%
1230 <*report | book>
1231     \if@twocolumn
1232         \@restonecoltrue\onecolumn
1233     \else
1234         \@restonecolfalse
1235     \fi
1236     \chapter*{\listtablename}
1237 </report | book>
1238 <article>    \section*{\listtablename
1239             \mkboth{%
1240                 \MakeUppercase\listtablename}\MakeUppercase\listtablename}}%
1241             \@starttoc{lot}%
1242 <report | book>    \if@restonecol\twocolumn\fi
1243     }

```

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

```
1244 \let\l@table\l@figure
```

8.2 Bibliography

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

```

1245 \newdimen\bibindent
1246 \setlength\bibindent{1.5em}

```

`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.

```

1247 \newenvironment{thebibliography}[1]
1248 (*article)
1249     {\section*{\refname
1250         \@mkboth{\MakeUppercase\refname}{\MakeUppercase\refname}}}%
1251 (/article)
1252 (*!article)
1253     {\chapter*{\bibname
1254         \@mkboth{\MakeUppercase\bibname}{\MakeUppercase\bibname}}}%
1255 (/!article)
1256     \list{\@biblabel{\@arabic\c@enumiv}}%
1257         {\settowidth\labelwidth{\@biblabel{#1}}%
1258         \leftmargin\labelwidth
1259         \advance\leftmargin\labelsep
1260         \@openbib@code
1261         \usecounter{enumiv}}%
1262         \let\p@enumiv\@empty
1263         \renewcommand\theenumiv{\@arabic\c@enumiv}}%
1264     \sloppy

This is setting the normal (non-infinite) value of \clubpenalty for the whole of
this environment, so we must reset its stored value also. (Why is there a % after
the second 4000 below?)

1265     \clubpenalty4000
1266     \@clubpenalty \clubpenalty
1267     \widowpenalty4000%
1268     \sfcode'\.\@m}
1269 {\def\@noitemerr
1270     {\@latex@warning{Empty ‘thebibliography’ environment}}}%
1271     \endlist}

```

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

```
1272 \newcommand\newblock{\hspace .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.

```
1273 \let\@openbib@code\@empty
```

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

```
1274 % \renewcommand*{\@biblabel}[1]{[#1]\hfill}
```

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

```
1275 % \renewcommand*{\@cite}[1]{[#1]}
```

8.3 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.

```
1276 \newenvironment{theindex}
1277     {\if@twocolumn
1278      \@restonecolfalse
1279      \else
1280      \@restonecoltrue
1281      \fi
1282      \columnseprule \z@
1283      \columnsep 35\p@
1284 <article>      \twocolumn[\section*{\indexname}]%
1285 <!article>     \twocolumn[\@makeschapterhead{\indexname}]%
1286              \@mkboth{\MakeUppercase\indexname}%
1287                  {\MakeUppercase\indexname}%
1288              \thispagestyle{plain}\parindent\z@
1289              \parskip\z@ \@plus .3\p@\relax
1290              \let\item\@idxitem}
```

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.

```
1291     {\if@restonecol\onecolumn\else\clearpage\fi}
```

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

```
\subitem 1292 \newcommand\@idxitem{\par\hangindent 40\p@}
\subsubitem 1293 \newcommand\subitem{\@idxitem \hspace*{20\p@}}
1294 \newcommand\subsubitem{\@idxitem \hspace*{30\p@}}
```

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

```
1295 \newcommand\indexspace{\par \vskip 10\p@ \@plus5\p@ \@minus3\p@\relax}
```

8.4 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`. We have to make sure that the rule takes no vertical space (see `plain.tex`) so we compensate for the natural height of the rule of 0.4pt by adding the right amount of vertical skip.

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

```
1296 \renewcommand\footnoterule{%
1297   \kern-3\p@
1298   \hrule\@width.4\columnwidth
1299   \kern2.6\p@}
```

`\c@footnote` Footnotes are numbered within chapters in the report and book document styles.
1300 `\(!article)\@addtoreset{footnote}{chapter}`

`\@makefnmark` The footnote mechanism of L^AT_EX calls the macro `\@makefnmark` 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 `\@makefnmark` is called when effectively inside a `\parbox` of width `\columnwidth` (i.e., with `\hsize = \columnwidth`).

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

```
\newcommand\@makefnmark[1]{%
  \setpar{\@par
    \tempdima = \hsize
    \advance\tempdima-10pt
    \parshape \@ne 10pt \tempdima}%
  \par
  \parindent 1em\noindent
  \hbox to \z@\hss\@makefnmark#1}
```

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

In these document classes we use a simpler macro, in which the footnote text is set like an ordinary text paragraph, with no indentation except on the first line of a paragraph, and the first line of the footnote. Thus, all the macro must do is set `\parindent` to the appropriate value for succeeding paragraphs and put the proper indentation before the mark.

```
1301 \newcommand\@makefnmark[1]{%
1302   \parindent 1em%
1303   \noindent
1304   \hb@xt@1.8em{\hss\@makefnmark#1}
```

`\@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.

```
1305 %\renewcommand\@makefnmark{\hbox{\@textsuperscript
1306 %                               {\normalfont\@thefnmark}}}
```

9 Initialization

9.1 Words

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

These commands may be redefined in any class or package that is customising L^AT_EX for use with non-English languages.

```

\contentsname
\listfigurename 1307 \newcommand\contentsname{Contents}
\listtablename 1308 \newcommand\listfigurename{List of Figures}
                  1309 \newcommand\listtablename{List of Tables}

\refname
\bibname 1310 <article>\newcommand\refname{References}
\indexname 1311 <report | book>\newcommand\bibname{Bibliography}
            1312 \newcommand\indexname{Index}

\figurename
\tablename 1313 \newcommand\figurename{Figure}
            1314 \newcommand\tablename{Table}

\partname
\chaptername 1315 \newcommand\partname{Part}
\appendixname 1316 <report | book>\newcommand\chaptername{Chapter}
\abstractname 1317 \newcommand\appendixname{Appendix}
            1318 <!book>\newcommand\abstractname{Abstract}

```

9.2 Date

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

At `\begin{document}` this definition will be optimised so that the names of all the ‘wrong’ months are not stored. This optimisation is not done here as that would ‘freeze’ `\today` in any special purpose format made by loading the class file into the format file.

```

1319 \def\today{\ifcase\month\or
1320   January\or February\or March\or April\or May\or June\or
1321   July\or August\or September\or October\or November\or December\fi
1322   \space\number\day, \number\year}

```

9.3 Two column mode

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

```
1323 \setlength\columnsep{10\p@}
```

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

```
1324 \setlength\columnseprule{0\p@}
```


9.4 The page style

We have *plain* pages in the document classes `article` and `report` unless the user specified otherwise. In the ‘book’ document class we use the page style *headings* by default. We use arabic pagenumbers.

```
1325 <!book> \pagestyle{plain}
1326 <book> \pagestyle{headings}
1327 \pagenumbering{arabic}
```

9.5 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.

```
1328 \if@twoside
1329 \else
1330   \raggedbottom
1331 \fi
```

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.

```
1332 \if@twocolumn
1333   \twocolumn
1334   \sloppy
1335   \flushbottom
```

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

```
1336 \else
1337   \onecolumn
1338 \fi
1339 </article | report | book>
```

Index

Numbers written in *italic* refer to the page where the corresponding entry is described, the ones underlined to the code line of the definition, the rest to the code lines where the entry is used.

Symbols	<code>\@biblabel</code>	<code>\@dblfpbot</code> <u>455</u>
<code>\@Roman</code> 648	. . 1256, 1257, <u>1274</u>	<code>\@dblfpsep</code> <u>455</u>
<code>\@afterheading</code>	<code>\@chapapp</code> . 496, 525,	<code>\@dblftop</code> <u>455</u>
. 726, 756, 804, 824	<u>658</u> , 786, 811, 1048	<code>\@dotsep</code> <u>1128</u>
<code>\@afterindentfalse</code>	<code>\@chapter</code> 782, <u>783</u>	<code>\@dottedtocline</code> 1199,
. 687, 781	<code>\@cite</code> <u>1275</u>	1202–1205,
<code>\@author</code> 541,	<code>\@clubpenalty</code> 1266	1208–1211, 1228
557, 569, 603, 622	<code>\@date</code> 542,	<code>\@endparpenalty</code> <u>878</u> , 969
<code>\@beginparpenalty</code>	560, 570, 604, 625	<code>\@endpart</code> . 746, 764, <u>766</u>
. <u>878</u> , 966	<code>\@dblfloat</code> . 1084, 1103	<code>\@eqnnum</code> <u>1066</u>

<code>\@evenfoot</code>	471, 473, 532	<code>\@openbib@code</code>	<code>\@twosidefalse</code> 41
<code>\@evenhead</code>	471, 474, 533	...	66, 1260, <u>1273</u>	<code>\@twosidettrue</code> 43
<code>\@fnsymbol</code> 580	<code>\@openrightfalse</code>	.. 56	<code>\@width</code> 1298
<code>\@fontswitch</code>	1126, 1127	<code>\@openrighttrue</code>	. 53, 55	A	
<code>\@fpbot</code> <u>440</u>	<code>\@part</code> 688, 705, <u>707</u>	<code>\abovecaptionskip</code>	..
<code>\@fpsep</code> <u>440</u>	<code>\@pnumwidth</code>	... <u>1128</u> ,	<u>1105</u> , 1110
<code>\@fptop</code> <u>440</u>	1154,	1155,	<code>\abovedisplayskip</code> 92, 98,
<code>\@highpenalty</code>	. <u>224</u> ,	1157,	1172,	104, 114, 124,	
1151, 1168, 1178		1173,	1177,	134, 147, 157, 167	
<code>\@idxitem</code>	.. 1290, <u>1292</u>	1189, 1190, 1194		<code>\abovedisplayskip</code>	..
<code>\@itempenalty</code> <u>878</u>	<code>\@ptsize</code> <u>1</u> , 34,	91, 97,
<code>\@latex@warning</code>	.. 1270	36, 38, 39, 84, 85		<code>\abstract</code>	(environ-
<code>\@listI</code> 108, <u>883</u>	<code>\@restonecolfalse</code>	..	ment)	<u>959</u>
<code>\@listi</code> 108,	1013,	<code>\abstractname</code>
116, 126, 136,		1027, 1138,		968, 975, 979, <u>1315</u>	
149, 159, 169, <u>883</u>		1218, 1234, 1278		<code>\addcontentsline</code>	..
<code>\@listii</code> <u>902</u>	<code>\@restonecoltrue</code> 711, 713, 732,	
<code>\@listiii</code> <u>902</u>	1011,	734, 787, 791, 795	
<code>\@listiv</code> <u>902</u>	1025, 1136,		<code>\addcontents</code>	798, 799
<code>\@listv</code> <u>902</u>	1216, 1232, 1280		<code>\and</code> 575, 609
<code>\@listvi</code> <u>902</u>	<code>\@roman</code> 940	<code>\appendix</code> <u>1038</u>
<code>\@lowpenalty</code>	<code>\@schapter</code>	... 782, <u>820</u>	<code>\appendixname</code>	1048, <u>1315</u>
. <u>224</u> , 879–881, 966		<code>\@secpenalty</code>	1150, 1185	<code>\arraycolsep</code> <u>1051</u>
<code>\@mainmatterfalse</code>	..	<code>\@setfontsize</code>	90, 96,	<code>\arrayrulewidth</code>	.. <u>1053</u>
..... 666, 682		102, 112, 122,		<code>\AtEndOfPackage</code>	... 65
<code>\@mainmattertrue</code>	8, 674	132, 145, 155,		<code>\author</code>	... <u>540</u> , 573, 607
<code>\@makecaption</code> <u>1105</u>	165, 178–184,		B	
<code>\@makechapterhead</code>	..	187–193, 196–201		<code>\backmatter</code> <u>676</u>
.... 801, 803, <u>806</u>		<code>\@settopoint</code>	.. 283,	<code>\baselineskip</code>
<code>\@makefnmark</code>	371, 372, 377, 392		.. 286–289, 294, 296	
.. 581, 1304, <u>1305</u>		<code>\@spart</code>	... 688, 705, <u>748</u>	<code>\baselinestretch</code>	.. <u>207</u>
<code>\@makefntext</code>	. 582, <u>1301</u>	<code>\@startsection</code>	835,	<code>\belowcaptionskip</code>	..
<code>\@makeschapterhead</code>	..	839, 843, 847, 851		<u>1105</u> , 1118
821, 823, <u>826</u> , 1285		<code>\@starttoc</code>	<code>\belowdisplayskip</code>
<code>\@maketitle</code>	... 587,	.. 1145, 1225, 1241		93, 99,
589, 594, 601, <u>611</u>		<code>\@textsuperscript</code>	..	105, 115, 125,	
<code>\@medpenalty</code> <u>224</u>	... 581, 584, 1305		135, 148, 158, 168	
<code>\@minipagefalse</code>	.. 1115	<code>\@thanks</code>	<code>\belowdisplayskip</code>	..
<code>\@minipagerestore</code>	<u>1056</u>	.. 562, 568, 596, 602		107, 141, 174
<code>\@mparswitchfalse</code>	. 41	<code>\@thefnmark</code>	<code>\bf</code> <u>1122</u>
<code>\@mparswitchtrue</code>	.. 43	... 581, 584, 1306		<code>\bibindent</code>	67, 68, <u>1245</u>
<code>\@mpfootins</code> <u>1056</u>	<code>\@title</code> 540,	<code>\bibname</code>	1253, 1254, <u>1310</u>
<code>\@nobreakfalse</code>	... 1161	552, 571, 605, 617		<code>\bigskipamount</code> <u>1219</u>
<code>\@nobreaktrue</code> 1160	<code>\@titlepagefalse</code>	. 5, 50	<code>\bottomfraction</code>	... <u>405</u>
<code>\@noitemerr</code> 1269	<code>\@titlepagetrue</code>	.. 6, 48		
<code>\@normalsize</code> <u>87</u>	<code>\@tocrmarg</code> <u>1128</u>		
<code>\@oddfnhead</code>	<code>\@topnewpage</code>	.. 801, 821		
. 471, 473, 509, 532		<code>\@topnum</code> 593, 780		
<code>\@odddhead</code>	<code>\@twocolumnfalse</code>	.. 59		
. <u>471</u> , 475, 510, 534		<code>\@twocolumntrue</code>	... 61		

<code>\brokenpenalty</code>	<u>234</u>	<code>\columnseprule</code>	figure* (environment)	<u>1080</u>
C		<code>\columnwidth</code>	<code>\figurename</code>	1079, <u>1313</u>
<code>\c@bottomnumber</code>	<u>404</u>	<code>\contentsname</code>	<code>\floatpagefraction</code>	<u>408</u>
<code>\c@chapter</code>	<u>638</u> , 651, 1049, 1063, 1074, 1093	1142, 1144, <u>1307</u>	<code>\floatsep</code>	<u>413</u>
<code>\c@dbltopnumber</code>	<u>409</u>	<code>\cs</code>	<code>\flushbottom</code>	1335
<code>\c@enumi</code>	938	D		<code>\fnum@figure</code>
<code>\c@enumii</code>	939	<code>\date</code>	<code>\fnum@table</code>	1079
<code>\c@enumiii</code>	940	<code>\dblfloatpagefraction</code>	<code>\footins</code>	<u>397</u> , 1056
<code>\c@enumiv</code> 941, 1256, 1263		<code>\dblfloatset</code>	<code>\footnote</code>	548, 616
<code>\cequation</code>	1059, 1063	<code>\dblfloatsep</code>	<code>\footnoterule</code>	547, <u>1296</u>
<code>\c@figure</code>	<u>1067</u>	<code>\dbltextfloatsep</code>	<code>\footnotesep</code>	<u>394</u>
<code>\c@footnote</code>	580, <u>1300</u>	<code>\dbltopfraction</code>	<code>\footnotesize</code>	<u>143</u> , 546
<code>\c@paragraph</code>	<u>638</u> , 656	<code>\DeclareOldFontCommand</code>	<code>\footskip</code>	<u>245</u> , 390
<code>\c@part</code>	<u>638</u> , 648	1119–1125	<code>\fps@figure</code>	<u>1076</u>
<code>\c@secnumdepth</code>	480, 486, 494, 502, 515, 523, <u>636</u> , 709, 718, 730, 740, 783, 809	<code>\DeclareRobustCommand</code>	<code>\fps@table</code>	<u>1095</u>
<code>\c@section</code>	<u>638</u> , 649, 652, 1042	1126, 1127	<code>\frontmatter</code>	<u>659</u>
<code>\c@subparagraph</code> <u>638</u> , 657		description (environment)	<code>\ftype@figure</code>	<u>1076</u>
<code>\c@subsection</code>	<u>638</u> , 654	<code>\descriptionlabel</code>	<code>\ftype@table</code>	<u>1095</u>
<code>\c@subsubsection</code>	<u>638</u> , 655	955, <u>957</u>	H	
<code>\c@table</code>	<u>1086</u>	<code>\displaywidowpenalty</code>	<code>\hb@xt@</code>	583, 1116, 1157, 1177, 1194, 1304
<code>\c@tocdepth</code>	1149, 1167, 1184	230	<code>\headheight</code>	<u>236</u> , 387
<code>\c@topnumber</code>	<u>401</u>	<code>\doublerulesep</code>	<code>\headsep</code>	<u>236</u> , 388
<code>\c@totalnumber</code>	<u>406</u>	E		<code>\Huge</code>
<code>\cal</code>	<u>1126</u>	<code>\endquotation</code>	745, 763, 817, 831	<code>\huge</code>
<code>\centering</code>	737, 760	<code>\endttitlepage</code>	722, 741, 753, 811	
<code>\changes</code>	959	environments:	I	
<code>\chapter</code>	<u>777</u> , 1140, 1220, 1236, 1253	abstract	<code>\if@compatibility</code>	9, 33, 40, 45, 49, 52, 58, 249, 251, 282, 285, 309, 379, 1006, 1159
<code>\chaptermark</code>	492, 521, 536, <u>630</u> , 797	description	<code>\if@mainmatter</code>	8, 495, 524, 784, 810
<code>\chaptername</code>	658, <u>1315</u>	figure*	<code>\if@openright</code> <u>7</u> , 661, 669, 677, 692, 778	
<code>\cleardoublepage</code>	662, 670, 678, 693, 778, 1009, 1023	figure	<code>\if@restonecol</code>	3, 1018, 1032, 1146, 1226, 1242, 1291
<code>\clearpage</code> 664, 672, 680, 695, 778, 1291		quotation	<code>\if@titlepage</code> <u>4</u> , 544, 962	
<code>\clubpenalty</code>	<u>228</u> , 1265, 1266	quote	<code>\if@twocolumn</code>	211, 252, 268, 299, 344, 585, 698, 800, 820, 855,
<code>\col@number</code>	586	table*		
<code>\columnsep</code>	1283, <u>1323</u>	table		
		thebibliography <u>52</u>		
		theindex		
		titlepage		
		verse		
		<code>\evensidemargin</code>		
		309		
		<code>\ext@figure</code>		
		<u>1076</u>		
		<code>\ext@table</code>		
		<u>1095</u>		
		F		
		<code>\fboxrule</code>		
		<u>1057</u>		
		<code>\fboxsep</code>		
		<u>1057</u>		
		figure (environment) <u>1080</u>		

864, 974, 983,	931, 932, 934,	<code>\marginparsep</code>
1010, 1024,	935, 954, 1257, 1258 299 , 356, 364
1135, 1215,	<code>\LARGE</code> 177 , 552, 617	<code>\marginparwidth</code> 309
1231, 1277, 1332	<code>\Large</code> 177 , 719, 838	<code>\mathbf</code> 1122
<code>\if@twoside</code> 322, 350,	<code>\large</code> 177 , 554, 560,	<code>\mathcal</code> 1126
471, 768, 1033, 1328	619, 625, 842, 1157	<code>\mathit</code> 1123
<code>\indexname</code>	<code>\leftmargin</code>	<code>\mathnormal</code> 1127
. 1284–1287, 1310 67, 116, 126,	<code>\mathrm</code> 1119
<code>\indexspace</code> 1295	136, 149, 159,	<code>\mathsf</code> 1120
<code>\interlinepenalty</code> .	169, 855 , 884,	<code>\mathtt</code> 1121
. 233 , 716, 738,	902, 918, 927,	<code>\maxdepth</code> 249
751, 761, 816, 830	930, 933, 954,	<code>\medskipamount</code> 219
<code>\intextsep</code> 413	991, 992, 998,	<code>\mit</code> 1126
<code>\it</code> 1123	1003, 1258, 1259	
<code>\item</code> 993, 1000, 1004, 1290	<code>\leftmargini</code> 116, 126,	N
<code>\itemindent</code> 68, 69,	136, 149, 159,	<code>\newblock</code> 72, 1272
954, 989, 990, 997	169, 855 , 872, 884	<code>\newif</code> 3, 4, 7, 8
<code>\itemsep</code> 119, 129,	<code>\leftmarginii</code>	<code>\normalfont</code>
139, 152, 162, 855 , 902, 903 581, 584, 717,
172, 888, 893,	<code>\leftmarginiii</code>	739, 752, 762,
898, 917, 926, 988 855 , 918, 919	808, 829, 838,
<code>\itshape</code> 1123	<code>\leftmarginiv</code>	842, 846, 850,
 855 , 927, 928	854, 950, 958,
J	<code>\leftmarginv</code>	1119–1125, 1306
<code>\jot</code> 1065 855 , 930, 931	<code>\normallineskip</code> 204
	<code>\leftmarginvi</code>	<code>\normalsize</code>
L 855 , 933, 934 87 , 846, 850, 854
<code>\l@chapter</code> 1165	<code>\leftskip</code> 1175,	<code>\num@figure</code> 1076
<code>\l@figure</code> 1228 , 1244	1176, 1192, 1193	<code>\num@table</code> 1095
<code>\l@paragraph</code> 1201	<code>\lineskip</code> 204 , 555, 620	O
<code>\l@part</code> 1148	<code>\listfigurename</code> 1220,	<code>\oddsidemargin</code> 309
<code>\l@section</code> 1182	1222–1224, 1307	<code>\onecolumn</code> 699,
<code>\l@subparagraph</code> 1201	<code>\listoffigures</code> 1213	1011, 1025,
<code>\l@subsection</code> 1201	<code>\listoftables</code> 1229	1136, 1216,
<code>\l@subsubsection</code> 1201	<code>\listparindent</code>	1232, 1291, 1337
<code>\l@table</code> 1244 69, 990, 996, 997	<code>\overfullrule</code> 44, 46
<code>\labelenumi</code> 942	<code>\listtablename</code> 1236,	
<code>\labelenumii</code> 942	1238, 1240, 1307	P
<code>\labelenumiii</code> 942		<code>\p@enumii</code> 946
<code>\labelenumiv</code> 942	M	<code>\p@enumiii</code> 946
<code>\labelitemi</code> 949	<code>\mainmatter</code> 668	<code>\p@enumiv</code> 946 , 1262
<code>\labelitemii</code> 949	<code>\makelabel</code> 955	<code>\pagenumbering</code>
<code>\labelitemiii</code> 949	<code>\maketitle</code> 544 667, 675, 1327
<code>\labelitemiv</code> 949	<code>\MakeUppercase</code>	<code>\paperheight</code> 11,
<code>\labelsep</code> 871 , 904, 479, 493, 501,	14, 17, 20, 23,
920, 929, 932,	514, 522, 1144,	26, 29, 30, 291, 385
935, 957, 1055, 1259	1223, 1224,	<code>\paperwidth</code>
<code>\labelwidth</code> 871 ,	1240, 1250, 12, 15, 18, 21,
903, 904, 919,	1254, 1286, 1287	24, 27, 30, 31,
920, 928, 929,	<code>\marginparpush</code> 299	263, 351, 359, 373

Change History

v1.0d	General: Added use of <code>\NeedsTeXFormat</code> 1	<code>\listoftables</code> : Moved the setting of <code>\@restonecolfalse</code> 52
	Made equation and eqnarray environments in the fleqn option up to date with latex.dtx 1	<code>\maketitle</code> : Removed the setting of the page number, when not in compatibility mode 24
	remove <code>\@in</code> , made option <code>makeindex</code> a synonym for option <code>makeidx</code> 1	<code>\mit</code> : Macro added 47
	removed <code>\@minus</code> , <code>\@plus</code> , <code>\@settopoint</code> , <code>\@setfontsize</code> ; they are now in the kernel 1	<code>\tableofcontents</code> : Moved the setting of <code>\@restonecolfalse</code> 48
	Replaced <code>\bf</code> with <code>\bfseries</code> ; <code>\rm</code> with <code>\rmfamily</code> 1	v1.0h
v1.0e	<code>\@normalsize</code> : <code>\normalsize</code> doesn't exist, so use <code>\newcommand</code> 8	<code>\@normalsize</code> : <code>\normalsize</code> is now defined in the kernel, so use <code>\renewcommand</code> . ASAJ. 8
v1.0f	General: Made all lines shorter than 72 characters 1	General: Corrected some typos. ASAJ. 6
	<code>\bf</code> : Macro added 47	Made the definitions of the font- and size-changing commands use <code>\renew</code> rather than <code>\new</code> . Defined the float parameters with <code>\renewcommand</code> rather than <code>\newcommand</code> . Corrected some typos in the fleqn option. Replaced two occurrences of <code>-\@secpenalty</code> by <code>\@secpenalty</code> . ASAJ. 1
	<code>\it</code> : Macro added 47	These are now defined in the kernel, so use <code>\@renewfontswitch</code> . Compatibility mode defines <code>\@renewfontswitch</code> to do nothing, so we don't need to check for compatibility mode any more. 47
	<code>\rm</code> : Macro added 47	<code>\bottomfraction</code> : Replaced <code>\newcommand</code> with <code>\renewcommand</code> . ASAJ. 18
	<code>\sc</code> : Macro added 47	<code>\dblfloatpagefraction</code> : Replaced <code>\newcommand</code> with <code>\renewcommand</code> . ASAJ. 19
	<code>\sf</code> : Macro added 47	<code>\dbltopfraction</code> : Replaced <code>\newcommand</code> with <code>\renewcommand</code> . ASAJ. 18
	<code>\sl</code> : Macro added 47	<code>\floatpagefraction</code> : Replaced <code>\newcommand</code> with <code>\renewcommand</code> . ASAJ. 18
	<code>\tt</code> : Macro added 47	<code>\footnotesize</code> : <code>\footnotesize</code> is now defined in the kernel, so use <code>\renewcommand</code> . ASAJ. 9
v1.0g	General: Distinguished between compatibility and 'normal' mode for the font changing commands. 47	
	Made change in eqnarray for the fleqn option, as suggested by Rainer. 1	
	Removed typo, A4 is not 279 mm high 4	
	<code>\cal</code> : Macro added 47	
	<code>theindex</code> : Moved the setting of <code>\@restonecoltrue</code> 54	
	<code>titlepage</code> : Moved the setting of <code>\@restonecolfalse</code> 41	
	<code>\listoffigures</code> : Moved the setting of <code>\@restonecolfalse</code> 51	

<code>\Huge</code> : These are now defined in the kernel, so use <code>\renewcommand</code> . ASAJ.	10	<code>\partopsep</code> : <code>\partopsep</code> should be different, depending on the pointsize	37
<code>\l@part</code> : Replaced <code>-\@secpenalty</code> by <code>\@secpenalty</code> . ASAJ.	49	v1.0n General: Removed code for <code>makeidx</code> option and made it a separate package; removed use of <code>\setlength</code> from list parameters.	1
<code>\l@section</code> : Replaced <code>-\@secpenalty</code> by <code>\@secpenalty</code> . ASAJ.	50	v1.0o <code>\normalsize</code> : <code>\normalsize</code> now defined in the kernel	8
<code>\small</code> : <code>\small</code> is now defined in the kernel, so use <code>\renewcommand</code> . ASAJ.	8	General: Small documentation changes	1
<code>\textfraction</code> : Replaced <code>\newcommand</code> with <code>\renewcommand</code> . ASAJ.	18	v1.0q General: Small documentation changes	1
<code>\topfraction</code> : Replaced <code>\newcommand</code> with <code>\renewcommand</code> . ASAJ.	18	v1.1a <code>\@makefnmark</code> : Use <code>\@makefnmark</code> to generate footnote marker	55
v1.0i <code>thebibliography</code> : Corrected definition of <code>thebibliography</code> for article	53	General: Removed <code>\typeout</code> messages	1
v1.0j General: Added <code>\normalfont</code> back in the definitions of <code>\rm</code> etc. as this should be the default behaviour	47	<code>\brokenpenalty</code> : Show correct default which is 100	12
Added <code>\ProvidesFile</code> to size files	1	<code>\marginparwidth</code> : New algorithm for <code>\marginparwidth</code>	16
Use <code>\cmd</code> in change entries	1	New algorithm for <code>\oddsidemargin</code>	16
v1.0k General: Removed some typos/bugs	1	<code>\textwidth</code> : Have old values for width in native mode	13
v1.0l General: add the extension to the names of the files	1	v1.1f General: Inserted forgotten line break	1
Changed version numbering; moved <code>leqno</code> and <code>fleqn</code> options to an external file.	1	v1.2a General: Added <code>openright</code> option. (LL)	1
<code>\marginparwidth</code> : Computing of <code>\evensidemargin</code> should only occur in compatibility mode	17	v1.2b General: Added the <code>...matter</code> commands. (LL)	1
v1.0m <code>\labelwidth</code> : Use em instead of pt to remain compatible with old styles	36	v1.2c General: Fixed page numbering in <code>titlepage</code> env. (LL)	1
<code>\leftmarginvi</code> : Use em instead of pt to remain compatible with old styles	36	<code>titlepage</code> : <code>page := 0</code> only in compatibility mode (LL)	41
<code>\parindent</code> : <code>\parindent</code> should be different, depending on the pointsize	11	<code>\maketitle</code> : Removed setting of page number, now done in <code>titlepage</code> environment	24
		v1.2d General: Checked the file for long lines and wrapped them when	

necessary; made a slight implementation modification to the openright and openany options.	1	v1.2o	General: Changed file information	1
<code>\titlepage</code> : Moved <code>\cleardoublepage</code> inside definition of <code>\titlepage</code> environment	41	v1.2p	General: Moved identification and driver to the front of the file	1
v1.2e		v1.2q	<code>\@makecaption</code> : Use <code>\sbox\@tempboxa</code> instead of <code>\setbox\@tempboxa\hbox</code> to make this colour safe	46
General: <code>\@renewfontswitch</code> has become <code>\DeclareOldFontCommand</code>	47	v1.2s	<code>\maketitle</code> : Reset <code>\@makefntext</code>	25
<code>\footnotesize</code> : use <code>\newcommand</code> again	9	v1.2t	General: Refrased a few sentences to prevent overfull hboxes	1
<code>\Huge</code> : use <code>\newcommand</code> again	10	v1.2u	<code>\topmargin</code> : Moved rounding of <code>\topmargin</code> to native mode	17
<code>\small</code> : <code>\small</code> is no longer defined in the kernel; use <code>\newcommand</code>	8	v1.2v	<code>\@dotsep</code> : Changed documentation from <code>></code> or <code>=</code> to <code>≥</code>	48
v1.2g		<code>\@makechapterhead</code> :	Added <code>\interlinepenalty\@M</code> to prevent a pagebreak in the middle of a title	34
<code>\sc</code> : Added warning if used in math mode	47		Added a <code>\nobreak</code> to prevent a pagebreak between the chapter-number and the chaptertitle	34
<code>\sl</code> : Added warning if used in math mode	47	<code>\@makeschapterhead</code> :	Added <code>\interlinepenalty\@M</code> to prevent a pagebreak in the middle of a title	35
v1.2i		General:	Made the onside option work for the book class	1
General: Use LaTeX instead of LaTeX2e in messages	1		removed typo	57
<code>\l@chapter</code> : Don't print a toc line when the tocdepth counter is less than 0	50	<code>\if@mainmatter</code> :	Moved the allocation of <code>\if@mainmatter</code> here	4
<code>\l@part</code> : Don't print a toc line when the tocdepth counter is less than -1	49	v1.2w	General: Use <code>\newcommand*</code> for commands with arguments	1
<code>\l@section</code> : Don't print a toc line when the tocdepth counter is less than 1	50	<code>\mit</code> :	Now define <code>\cal</code> and <code>\mit</code> using <code>\DeclareRobustCommand*</code>	47
v1.2j		v1.2x	<code>\@makecaption</code> : Due to a change in the way floats are handled we need to set the <code>\if@minipage</code> switch to false	47
General: Removed the use of <code>\fileversion</code> c.s.	1	v1.2y	<code>\descriptionlabel</code> : made command short	39
v1.2k				
<code>\descriptionlabel</code> : Inserted <code>\normalfont</code>	39			
<code>\labelitemii</code> : Inserted <code>\normalfont</code>	39			
<code>\maketitle</code> : Added check on number of columns in use locally	25			
<code>\maxdepth</code> : Added setting of <code>\maxdepth</code> and <code>\@maxdepth</code>	12			
v1.2l				
General: changed some <code>\changes</code> entries	1			
v1.2m				
General: Forgot a few entries	1			

v1.2z	General: Always use <code>\cs</code> in <code>\changes</code> entries	1	v1.3f	<code>\subsubitem</code> : Corrected error in definition of <code>\@idxitem</code>	54
	<code>thebibliography</code> : added a missing percent character	53	v1.3g	<code>\maketitle</code> : Fix definition of <code>\@makefnmark</code> and <code>\@makefnmark</code> and <code>\@makefnmark</code> to a) work and b) without using math	25
	<code>\marginparwidth</code> : Also take <code>\marginparsep</code> into account here	16	v1.3h	General: Split up to save save stack /1742	56
	<code>\ps@headings</code> : Removed extra dot after <code>\thesection</code> (PR 1519)	22	v1.3i	General: Moved code for generic class options <code>leqno</code> and <code>fleqn</code> to kernel file	57
v1.3a	General: Replaced all <code>\hbox</code> to by <code>\hb@xt@</code>	1	<code>titlepage</code> : New implementation with support for <code>twoside</code> and <code>openright</code> option	41	
	<code>\footnoterule</code> : use <code>\@width</code>	54	v1.3j	<code>\@endpart</code> : move <code>docstrip</code> guard to avoid defining <code>\@endpart</code> in article	32
	<code>\maketitle</code> : Use <code>\@makefnmark</code> in definition of <code>\@makefnmark</code>	25	<code>\@makechapterhead</code> : replace braces by <code>\space</code>	34	
v1.3b	<code>thebibliography</code> : Added missing braces in definition of <code>thebibliography</code> environment.	53	<code>\@part</code> : add missing percent	32	
	<code>\l@chapter</code> : Added missing braces around argument to <code>\addpenalty</code>	50	<code>verse</code> : stop <code>\item</code> scanning for [with <code>\relax</code>	40	
	<code>\l@part</code> : Added missing braces around argument to <code>\addpenalty</code>	49	<code>thebibliography</code> : remove surplus spaces	53	
v1.3c	<code>\@makechapterhead</code> : replace <code>\reset@font</code> with <code>\normalfont</code>	34	<code>\l@part</code> : Add missing percent	49	
	<code>\@makeschapterhead</code> : replace <code>\reset@font</code> with <code>\normalfont</code>	35	<code>\maketitle</code> : use <code>\let</code> to save space	25, 26	
	<code>\@part</code> : replace <code>\reset@font</code> with <code>\normalfont</code>	31, 32	<code>\maxdepth</code> : Take setting of <code>\@maxdepth</code> out again	12	
	<code>\@spart</code> : replace <code>\reset@font</code> with <code>\normalfont</code>	32	<code>\mit</code> : Remove surplus braces	47	
	<code>\ps@headings</code> : Replace <code>\hskip 1em\relax</code> with <code>\quad</code>	22	<code>\part</code> : Replace <code>\hbox</code> by <code>\null</code>	30	
	<code>\section</code> : replace <code>\reset@font</code> with <code>\normalfont</code>	35	<code>\subsubitem</code> : use <code>\@idxitem</code> to save space	54	
v1.3d	General: Replaced all <code>\uppercase</code> by <code>\MakeUppercase</code>	1	<code>\today</code> : use <code>\edef</code> to save a lot of space	56	
v1.3e	<code>abstract</code> : Added setting of <code>\@endparpenalty</code> to avoid page break after abstract heading.	39	<code>quotation</code> : stop <code>\item</code> scanning for [with <code>\relax</code>	40	
			<code>quote</code> : stop <code>\item</code> scanning for [with <code>\relax</code>	41	
			v1.3k	<code>\@openbib@code</code> : Macro added	53
			General: Macro <code>\if@openbib</code> removed	4	
			openbib option reimplemented	6	

<code>\thebibliography</code> : Code for open-bib changed	53	<code>\mainmatter</code> : Make this command react to the option <code>openany</code>	29
<code>\maketitle</code> : Disable <code>\title</code> and similar decls	25, 26	<code>\part</code> : Make this command react to the option <code>openany</code>	30
<code>\newblock</code> : Default changed.	53	v1.3s	
v1.3l		<code>\labelitemii</code> : Replaced -- with <code>\textendash</code>	39
General: Disabled in compatibility mode all options that are new in L ^A T _E X 2 _ε .	1	v1.3t	
v1.3n		<code>\thebibliography</code> : Added setting value of <code>\clubpenalty</code>	53
<code>\bigskipamount</code> : Added setting the values of <code>\...skipamount</code>	11	v1.3u	
<code>\maketitle</code> : Empty <code>\@date</code> as well	25, 26	<code>\c@figure</code> : Added test for non-zero chapter number	45
v1.3o		<code>\c@table</code> : Added test for non-zero chapter number	45
<code>\@maketitle</code> : (CAR) Make <code>\footnote</code> always work in title, etc	26	<code>\labelitemii</code> : Changed to <code>\textbullet</code> , <code>\textasteriskcentered</code> and <code>\textperiodcentered</code>	39
<code>\maketitle</code> : (CAR) Make <code>\footnote</code> always work in title, etc	24	<code>\theequation</code> : Added test for non-zero chapter number	44
v1.3q		v1.3v	
<code>\leftmarginvi</code> : Temporary(?) fix: revert to setting <code>\leftmargin</code> at outer level	36	General: Documentation fixes.	1
v1.3r		v1.3w	
<code>\frontmatter</code> : Make this command react to the option <code>openany</code>	29	<code>\today</code> : use <code>\def</code> again, latex/2620	56
		v1.3x	
		<code>\l@part</code> : Removed setting of <code>\@tempdima</code> as this macro does not use <code>\numberline</code> to set the toc line.	49