

Document Class **refman** for L^AT_EX version 2e*

Copyright (C) 1988 by Hubert Partl
Copyright (C) 1994-99 by Axel Kielhorn

1999/03/20

Contents

1	The DOCSTRIP modules	1
2	Initial Code	1
3	Declaration of Options	2
3.1	Setting Paper Sizes	2
3.2	Choosing the type size	2
3.3	Two-side or one-side printing	3
3.4	Draft option	3
3.5	Titlepage option	3
3.6	openright option	3
3.7	Twocolumn printing	3
3.8	Equation numbering on the left	3
3.9	Flush left displays	4
3.10	Open bibliography	4
3.11	User flags	4
4	Executing Options	4
5	Loading Packages	5
6	Document Layout	5
6.1	Fonts	5
6.2	Paragraphing	6
6.3	Page Layout	6
6.3.1	Vertical spacing	6
6.3.2	The dimension of text	7
6.3.3	Horizontal margins	7
6.3.4	Vertical margins	8
6.3.5	Float placement parameters	9
6.4	Page Styles	10
6.4.1	Marking conventions	10
6.4.2	Defining the page styles	11

*This file has version number v2.0c, last revised 1999/03/20.

7	Document Markup	15
7.1	The title	15
7.2	Chapters and Sections	18
7.2.1	Building blocks	18
7.2.2	Mark commands	19
7.2.3	Define Counters	19
7.2.4	Parts	20
7.2.5	Chapters	22
7.2.6	Lower level headings	23
7.3	Lists	24
7.3.1	General List Parameters	24
7.3.2	Enumerate	26
7.3.3	Itemize	26
7.3.4	Description	27
7.4	Defining new environments	27
7.4.1	Abstract	27
7.4.2	Verse	28
7.4.3	Quotation	28
7.4.4	Quote	28
7.4.5	Example	28
7.4.6	Theorem	29
7.4.7	Titlepage	29
7.4.8	Appendix	29
7.5	Setting parameters for existing environments	30
7.5.1	Array and tabular	30
7.5.2	Tabbing	30
7.5.3	Minipage	30
7.5.4	Framed boxes	30
7.5.5	Equation and eqnarray	31
7.6	Floating objects	31
7.6.1	Figure	32
7.6.2	Table	32
7.6.3	Captions	33
7.7	Font changing	33
8	Cross Referencing	34
8.1	Table of Contents, etc.	34
8.1.1	Table of Contents	35
8.1.2	List of figures	37
8.1.3	List of tables	38
8.2	Bibliography	38
8.3	The index	39
8.4	Footnotes	40
9	New commands	42
9.1	Margin commands	42
9.2	Rules	43
9.3	Pages	43
9.4	Miscellaneous	44
9.5	Obsolete commands	44

9.6	Future commands	44
10	Initialization	45
10.1	Words	45
10.2	Date	45
10.3	Two column mode	46
10.4	The page style	46
10.5	Single or double sided printing	46

1 The DOCSTRIP modules

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

refart	produce the documentclass refart
refrep	produce the documentclass refrep
driver	produce a documentation driver file

2 Initial Code

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

```

\@ptsize This control sequence is used to store the second digit of the pointsize we are
typesetting in. So, normally, it's value is one of 0, 1 or 2.
1 (*refart | refrep)
2 \newcommand\@ptsize{}

\if@restonecol Only the index is printed in twocolumn-layout.
3 \newif\if@restonecol

\if@titlepage A switch to indicate if a titlepage has to be produced. For the refart document
class the default is not to make a separate titlepage.
4 \newif\if@titlepage
5 (+refart)\@titlepagefalse
6 (+refrep)\@titlepagetrue

\if@openright A switch to indicate if chapters must start on a right-hand page. The default for
the refrep class is no. There are no chapters in the refart class
7 (+refrep)\newif\if@openright

```

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. Since `repbook` does not exist you may change the papersizes in your document if needed.

```

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

```

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

```

26 \DeclareOption{landscape}
27   {\setlength@tempdima  {\paperheight}%
28    \setlength\paperheight {\paperwidth}%
29    \setlength\paperwidth  {\@tempdima}}

```

The option `square` assigns the values of `\paperwidth` to `\paperheight`, which will result in a square layout. If you use `landscape` first you will get a square layout which uses the height of your original paper.

```

30 \DeclareOption{square}
31   {\setlength\paperheight {\paperwidth}}

```

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 8 assuming that a class won't define both 8pt and 18pt options.

```

32 \DeclareOption{10pt}{\renewcommand\@ptsize{0}}
33 \DeclareOption{11pt}{\renewcommand\@ptsize{1}}
34 \DeclareOption{12pt}{\renewcommand\@ptsize{2}}

```

3.3 Two-side or one-side printing

For two-sided printing we use the switch `\if@twoside`. We set `\if@mparswitch` which does nothing now but is kept for compatibility reasons.

```

35 \DeclareOption{oneside}{\@twosidefalse \@mparswitchfalse}
36 \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.

```
37 \DeclareOption{draft}{\setlength\overfullrule{5pt}}
38 \DeclareOption{final}{\setlength\overfullrule{0pt}}
```

3.5 Titlepage option

The `refart` usually has no separate titlepage, but the user can request one.

```
39 \DeclareOption{titlepage} {\@titlepagetrue}
40 \DeclareOption{notitlepage}{\@titlepagefalse}
```

3.6 openright option

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

```
41 (+refrep)\DeclareOption{openright}{\@openrighttrue}
42 (+refrep)\DeclareOption{openany} {\@openrightfalse}
```

3.7 Twocolumn printing

Two-column is used in the index. There is no user command or option to request twocolumn printing. Therefore `twocolumn` will lead to an error message.

```
43 \DeclareOption{onecolumn}{\@twocolumnfalse}
44 \DeclareOption{twocolumn}{%
45 (+refart) \ClassError{Refart}
46 (+refrep) \ClassError{Refrep}
47 {There is no twocolumn layout in this class!}
48 {Can you imagine how twocolumn layout will look\MessageBreak
49 in this class? That's why!}
50 \@twocolumnfalse}
```

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.

```
51 \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.

```
52 \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`.

```
53 \DeclareOption{openbib}{%
```

First some hook into the bibliography environment is filled.

```
54 \AtEndOfClass{%
55 \renewcommand\@openbib@code{%
56 \advance\leftmargin\bibindent
57 \itemindent -\bibindent
58 \listparindent \itemindent
59 \parsep \z@
60 }%
```

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

```
61 \renewcommand\newblock{\par}}%
62 }
```

3.11 User flags

There are some flags the user may change to control the behaviour of some commands:

`\ifdescriptioncolon` This switch controls whether there is a colon in the description item or not. The default is to include a colon.

```
63 \newif\ifdescriptioncolon \descriptioncolontrue
```

`\ifdescriptionleft` This switch controls whether the description items are set left bound or right bound. The default is right bound.

```
64 \newif\ifdescriptionleft \descriptionleftfalse
```

`\ifmaxipagerule` This switch controls whether there is a rule at the beginning and end of a maxipage. This flag may later be used to select rules at other places (like part or chapter) as well.

```
65 \newif\ifmaxipagerule \maxipageruletrue
```

4 Executing Options

Here we execute the default options to initialize certain variables.

```
66 (*refart)
67 \ExecuteOptions{letterpaper,10pt,oneside,onecolumn,final}
68 (/refart)
69 (*refrep)
70 \ExecuteOptions{letterpaper,10pt,oneside,onecolumn,final,openany}
71 (/refrep)
```

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.

```
72 \ProcessOptions
```

Now that all the options have been executed we can load the chosen class option file that contains all size dependent code. We are using the `sizexx.clo` Files from `classes.dtx` now and do the page layout calculation inside the class-file.

```
73 (*refart | refrep)
```

```
74 \input{size1\@ptsize.clo}
75 </refart | refrep>
```

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:

```
\@vpt      5      \@vipt    6      \@viipt   7
\@viiipt   8      \@ixpt    9      \@xpt     10
\@xipt     10.95  \@xiipt  12     \@xivpt   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.

See `classes.dtx` for documentaion on `sizexx.clo`

6.2 Paragraphing

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

```
76 (*refart | refrep)
77 \setlength\lineskip{1\p@}
78 \setlength\normallineskip{1\p@}
```

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

```
79 \renewcommand\baselinestretch{}
```

`\parskip` `\parindent` gives extra vertical space between paragraphs and `\parindent` is the width of the paragraph indentation. The value of `\parindent` is set to 0.

```
80 \setlength\parskip {0.5\baselineskip \@plus 2\p@}
81 \setlength\parindent {\z@}
```

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

```
82 \@lowpenalty 51
83 \@medpenalty 151
84 \@highpenalty 301
```

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

```
85 % \clubpenalty 150
86 % \widowpenalty 150
```

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

```
87 % \displaywidowpenalty 50
88 % \predisplaypenalty 10000
89 % \postdisplaypenalty 0
```

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

```
90 % \interlinepenalty 0
```

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

```
91 % \brokenpenalty 100
```

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` `\headsep` `\topskip` The `\headheight` is the height of the box that will contain the running head. The `\headsep` is the distance between the bottom of the running head and the top of the text. `\topskip` is the `\baselineskip` for the first line on a page. Only the definition of `\headsep` differs from `sizexx` and has to be changed.

```
92 \setlength\headsep {\baselineskip}
```

6.3.2 The dimension of text

`\fullwidth` `\textwidth` `\leftmarginwidth` There is no need to supply a compatibility mode since the independend `refman.sty` was never released to the public.

We will set the dimensions differently, taking into account the paper size for instance.

First, we calculate the maximum `textwidth`, which will fit on the selected paper and store it in `\@tempdima`.


```
93 \newdimen\leftmarginwidth
94 \newdimen\fullwidth
```

`\emptyfoottopmargin` `\emptyheadtopmargin` Either your document uses *footings* or *headings*. Depending on this, the whole page is shifted up or down by one line.

```
95 \newdimen\emptyfoottopmargin
96 \newdimen\emptyheadtopmargin
```

`\setttextfraction` You can specify how much of the `\fullwidth` will be used for the text by using the `\setttextfraction` command. The argument should be between 0 and 1. The remaining width is used for the left margin.

```
97 \newcommand\setttextfraction[1]%
98 {
99   \setlength\@tempdima{\paperwidth}
100  \addtolength\@tempdima{-2in}
```

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

```
101 \if@twocolumn
102   \setlength\textwidth{\@tempdima}
103 \else
104   \setlength\fullwidth{\@tempdima}
105   \settopoint\fullwidth
106   \setlength\textwidth{#1\fullwidth} % This may change
107 \fi
```

Here we modify the width of the text a little to be a whole number of points and calculate the remaining margin.

```
108 \settopoint\textwidth
109 \setlength\leftmarginwidth{\fullwidth}
110 \addtolength\leftmarginwidth{-\textwidth}
```

6.3.3 Horizontal margins

`\oddsidemargin` `\evensidemargin` `\marginparwidth` The values for `\oddsidemargin` and `\marginparwidth` will be set independently on the status of the `\if@twoside`. (We have the same layout on odd and even pages.)

For one-sided printing we center the text on the page, by calculating the difference between `\textwidth` and `\paperwidth-2in`. Half of that difference is then used for the margin. The amount of space that can be used for marginal notes is `\leftmarginwidth - marginparsep` to which we add any ‘leftover’ space.

```
111 \setlength\@tempdima      {\paperwidth}
112 \addtolength\@tempdima   {-2in}
113 \addtolength\@tempdima   {-\fullwidth}
114 \setlength\oddsidemargin {0.5\@tempdima}
115 \addtolength\oddsidemargin {\leftmarginwidth}
116 \settopoint\oddsidemargin
```

Then `\evensidemargin` and `\marginparwidth` are set to `\oddsidemargin`. `\marginparwidth` will be modified later.

```
117 \setlength\evensidemargin {\oddsidemargin}
118 \setlength\marginparwidth {\oddsidemargin}
```

`\marginparsep` The horizontal space between the main text and marginal notes is determined by `\marginparsep` (defined in `sizexx`), the minimum vertical separation between two marginal notes is controlled by `\marginparpush` which is set to 0 because we will have lots of margin notes. The width of the `marginpar` is reduced by `\marginparsep` to produce flushleft pages.

```
119 \addtolength\marginparwidth {-\marginparsep}
120 \setlength\marginparpush {0\p@}
121 }
```

Now we call `\settextrfraction` with the default value of 0.7

```
122 \@onlypreamble\settextrfraction
123 \settextrfraction {0.7}
```

`\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).

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.

```
124 \setlength\@tempdima {\paperheight}
```

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

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

The running headers and footers extend partly into the top and bottom margins.

```
126 \addtolength\@tempdima{-.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.

```
127 \divide\@tempdima\baselineskip
128 \@tempcnta=\@tempdima
```

From this we can calculate the height of the text.

```
129 \setlength\textheight{\@tempcnta\baselineskip}
```

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

```
130 \advance\textheight by \topskip
```

6.3.4 Vertical margins

`\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 and rounded to full points.

```
131 \setlength\topmargin{\paperheight}
132 \addtolength\topmargin{-2in}
133 \addtolength\topmargin{-\headheight}
134 \addtolength\topmargin{-\headsep}
135 \addtolength\topmargin{-\textheight}
136 \addtolength\topmargin{-\footskip} % this might be wrong!
137 \addtolength\topmargin{-.5\topmargin}
138 \@settopoint\topmargin
```

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

The contents of the page is shifted up or down by one `\baselineskip` depending on the pagestyle. Do not combine headings and footings in one document!

```
139 \setlength\emptyfoottopmargin {\topmargin}
140 \addtolength\emptyfoottopmargin{\baselineskip}
141 \setlength\emptyheadtopmargin {\topmargin}
142 \addtolength\emptyheadtopmargin{-\baselineskip}
```

6.3.5 Float placement parameters

All float parameters are given default values in the $\text{\LaTeX} 2_{\epsilon}$ kernel. For this reason counters only need to be set with `\setcounter` and other parameters are set using `\renewcommand`.

Limits for the placement of floating objects

<code>\c@topnumber</code>	The <i>topnumber</i> counter holds the maximum number of floats that can appear on the top of a text page.
143	<code>\setcounter{topnumber} {2}</code>
<code>\topfraction</code>	This indicates the maximum part of a text page that can be occupied by floats at the top.
144	<code>\renewcommand\topfraction {.7}</code>
<code>\c@bottomnumber</code>	The <i>bottomnumber</i> counter holds the maximum number of floats that can appear on the bottom of a text page.
145	<code>\setcounter{bottomnumber} {1}</code>
<code>\bottomfraction</code>	This indicates the maximum part of a text page that can be occupied by floats at the bottom.
146	<code>\renewcommand\bottomfraction {.3}</code>
<code>\c@totalnumber</code>	This indicates the maximum number of floats that can appear on any text page.
147	<code>\setcounter{totalnumber} {3}</code>
<code>\textfraction</code>	This indicates the minimum part of a text page that has to be occupied by text.
148	<code>\renewcommand\textfraction {.2}</code>
<code>\floatpagefraction</code>	This indicates the minimum part of a page that has to be occupied by floating objects before a ‘float page’ is produced.
149	<code>\renewcommand\floatpagefraction {.5}</code>
<code>\c@dbltopnumber</code>	The <i>dbltopnumber</i> counter holds the maximum number of two column floats that can appear on the top of a two column text page.
150	<code>\setcounter{dbltopnumber} {2}</code>
<code>\dbltopfraction</code>	This indicates the maximum part of a two column text page that can be occupied by two column floats at the top.
151	<code>\renewcommand\dbltopfraction {.7}</code>
<code>\dblfloatpagefraction</code>	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.
152	<code>\renewcommand\dblfloatpagefraction {.5}</code>
153	<code>\refart refrep)</code>

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.

```
154 % %%\mark{ } % Initializes TeX's marks <--- can vanish
```

6.4.2 Defining the page styles

The pagestyles *empty* is defined in `latex.dtx`.

`\ps@plain` We have to redefine *plain* to support twoside layout.

```
155 (*refart | refrep)
156 \if@twoside
157   \def\ps@plain{%
158     \let\mkboth\gobbletwo
159     \let\oddhead\empty
160     \let\evenhead\empty
```

```

161     \def\@oddfoot{\normalfont\hfil\thepage}
162     \def\@evenfoot{\normalfont\thepage\hfil}}
163 \else
164     \def\ps@plain{%
165         \let\@mkboth\@gobbletwo
166         \let\@oddhead\@empty
167         \let\@evenhead\@empty
168         \def\@oddfoot{\normalfont\hfil\thepage}
169         \let\@evenfoot\@oddfoot}
170 \fi
171 % \end{macrocode}
172 % \end{macro}
173 %
174 % \begin{macro}{\ps@headings}
175 %     The definition of the page style \pstyle{headings} has to be
176 %     different for two sided printing than it is for one sided
177 %     printing.
178 %
179 %     \begin{macrocode}
180 \if@twoside
181     \def\ps@headings{%

```

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

```

182         \let\@oddfoot\@empty\let\@evenfoot\@empty
183         \def\@evenhead{\hss\vbox to \z@\vss\hsize=\fullwidth
184         \hb@xt@\fullwidth{\thepage\hfil\slshape\leftmark}
185         \vskip 3\p@ \hrule}}%
186         \def\@oddhead{\hss\vbox to \z@\vss\hsize=\fullwidth
187         \hb@xt@\fullwidth{\slshape\rightmark}\hfil\thepage}
188         \vskip 3\p@ \hrule}}%

```

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.

```

189     \let\@mkboth\markboth

```

We shift the page one \baselineskip to the bottom to compensate for the headings.

```

190     \topmargin\emptyfoottopmargin
191 \refart | refrep)

```

For the refart 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.

```

192 (*refart)
193     \def\sectionmark##1{%
194         \markboth {\ifnum \c@secnumdepth >\z@
195         \thesection\quad\fi
196         ##1}{}}%
197     \def\subsectionmark##1{%
198         \markright {\ifnum \c@secnumdepth >\@ne
199         \thesubsection\quad\fi

```

```

200         ##1}}
201 </refart>
    In the refrep document class 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.
202 (*refrep)
203     \def\chaptermark##1{%
204         \markboth {\ifnum \c@secnumdepth >\m@ne
205             \@chapapp\ thechapter \ fi
206             ##1}{}}%
207     \def\sectionmark##1{%
208         \markright {\ifnum \c@secnumdepth >\z@
209             \thesection \ fi
210             ##1}}
211 </refrep)

```

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....

```

212 (*refart | refrep)
213 \else
214     \def\ps@headings{%
215         \let\@oddfoot\@empty
216         \def\@oddhead{\hss\vbox to \z@{\vss\hsize=\fullwidth
217             \hbxt@\fullwidth{\slshape\rightmark}\hfil\thepage}
218             \vskip 3\p@ \hrule}}%
219         \let\@mkboth\markboth

```

We shift the page one \baselineskip to the bottom to compensate for the headings.

```

220         \topmargin\emptyfoottopmargin
221 </refart | refrep)

```

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

```

222 (*refart)
223     \def\sectionmark##1{%
224         \markright {\ifnum \c@secnumdepth >\m@ne
225             \thesection\quad\fi
226             ##1}}
227 </refart)
228 (*refrep)
229     \def\chaptermark##1{%
230         \markright {\ifnum \c@secnumdepth >\m@ne
231             \@chapapp\ thechapter \ fi
232             ##1}}
233 </refrep)
234 (*refart | refrep)
235 \fi

```

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

```

236 \if@twoside
237     \def\ps@footings{%

```

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

```

238 \let\@oddhead\@empty\let\@evenhead\@empty
239 \def\@evenfoot{\hss\vbox to \z@\vss\hsize=\fullwidth
240 \hrule \vskip 3\p@
241 \hb@xt@\fullwidth{\thepage\hfil\slshape\leftmark}}}%
242 \def\@oddfoot{\hss\vbox to \z@\vss\hsize=\fullwidth
243 \hrule \vskip 3\p@
244 \hb@xt@\fullwidth{\slshape\rightmark}\hfil\thepage}}}%

```

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

```
245 \let\@mkboth\markboth
```

We shift the page one `\baselineskip` to the top to compensate for the footings.

```

246 \topmargin\emptyheadtopmargin
247 \refart | refrep)

```

For the `refart` 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.

```

248 (*refart)
249 \def\sectionmark##1{%
250 \markboth {\ifnum \c@secnumdepth >\z@
251 \thesection\quad\fi
252 ##1}\fi}%
253 \def\subsectionmark##1{%
254 \markright {\ifnum \c@secnumdepth >\@ne
255 \thesubsection\quad\fi
256 ##1}\fi}}
257 \refart)

```

In the `refrep` document class 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@footings` is executed.

```

258 (*refrep)
259 \def\chaptermark##1{%
260 \markboth {\ifnum \c@secnumdepth >\m@ne
261 \@chapapp\ thechapter \ \fi
262 ##1}\fi}%
263 \def\sectionmark##1{%
264 \markright {\ifnum \c@secnumdepth >\z@
265 \thesection \ \fi
266 ##1}\fi}}
267 \refrep)

```

The definition of `\ps@footings` 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...`

```

268 (*refart | refrep)
269 \else

```

```

270 \def\ps@footings{%
271   \let\@oddhead\@empty
272   \def\@oddfoot{\hss\vbox to \z@\vss\hsize=\fullwidth
273     \hrule \vskip 3\p@
274     \hb@xt@\fullwidth{\slshape\rightmark}\hfil\thepage}}%
275   \let\@mkboth\markboth

```

We shift the page one \baselineskip to the top to compensate for the footings.

```

276   \topmargin\emptyheadtopmargin
277 \refart | refrep)

```

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

```

278 (*refart)
279   \def\sectionmark##1{%
280     \markright {\ifnum \c@secnumdepth >\m@ne
281       \thesection\quad\fi
282       ##1}}
283 \refart)
284 (*refrep)
285   \def\chaptermark##1{%
286     \markright {\ifnum \c@secnumdepth >\m@ne
287       \@chapapp\ thechapter \ \fi
288       ##1}}
289 \refrep)
290 (*refart | refrep)
291 \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.

```

292 \def\ps@myheadings{%
293   \let\@oddfoot\@empty\let\@evenfoot\@empty
294   \def\@evenhead{\hss\vbox to \z@\vss\hsize=\fullwidth
295     \hb@xt@\fullwidth{\thepage\hfil\slshape\leftmark}
296     \vskip 3\p@ \hrule}}%
297   \def\@oddhead{\hss\vbox to \z@\vss\hsize=\fullwidth
298     \hb@xt@\fullwidth{\slshape\rightmark}\hfil\thepage}
299     \vskip 3\p@ \hrule}}%

```

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).

```

300   \let\@mkboth\@gobbletwo
301 (+refrep)   \let\chaptermark\@gobble
302   \let\sectionmark\@gobble
303 (+refart)   \let\subsectionmark\@gobble

```

We shift the page one \baselineskip to the bottom to compensate for the headings.

```

304   \topmargin\emptyfoottopmargin
305   }

```


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

```

306 \def\ps@myfootings{%
307     \let\@oddhead\@empty\let\@evenhead\@empty
308     \def\@evenfoot{\hss\vbox to \z@\vss\hsize=\fullwidth
309         \hrule \vskip 3\p@
310         \hb@xt@\fullwidth{\thepage\hfil\slshape\leftmark}}}%
311     \def\@oddfont{\hss\vbox to \z@\vss\hsize=\fullwidth
312         \hrule \vskip 3\p@
313         \hb@xt@\fullwidth{\slshape\rightmark}\hfil\thepage}}}%

```

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

```

314     \let\@mkboth\@gobbletwo
315 (+refrep) \let\chaptermark\@gobble
316     \let\sectionmark\@gobble
317 (+refart) \let\subsectionmark\@gobble

```

We shift the page one `\baselineskip` to the top to compensate for the footings.

```

318     \topmargin\emptyheadtopmargin
319 }

```

7 Document Markup

7.1 The title

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

```

320 % \newcommand*\title}[1]{\gdef\@title{#1}}
321 % \newcommand*\author}[1]{\gdef\@author{#1}}
322 % \newcommand*\date}[1]{\gdef\@date{#1}}

```

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

```

323 % \gdef\@date{\today}

```

`\maketitle` The definition of `\maketitle` depends on whether a separate title page is made. This is the default for `refrep`. If you want a titlepage with `refart` you can enable it using the `titlepage` option.

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.

```

324 \if@titlepage
325     \newcommand\maketitle{\begin{titlepage}}%
326     \let\footnotesize\small
327     \let\footnoterule\relax
328     \let\footnote\thanks
329     \renewcommand\thefootnote{\@fnsymbol\c@footnote}%

```

```

330 \def\@makefnmark%
331   {\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
332 \long\def\@makefntext##1{%
333   \@setpar{\@par
334     \@tempdima = \hsize
335     \advance\@tempdima -1em
336     \parshape \@ne 1em \@tempdima}%
337   \par\parindent 1em \noindent
338   \hb@xt@\z@{\hss\@textsuperscript{\normalfont\@thefnmark}\,}##1}

```

We center the entire title vertically; the centering is set off a little by adding a `\vskip`. In compatibility mode the pagenumber is set to 0 to keep the behaviour of L^AT_EX 2.09 style files

```

339 \null\vfil
340 \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.

```

341 \begin{center}%
342   {\LARGE \@title \par}%
343   \vskip 3em%
344   {\large
345     \lineskip .75em%
346     \begin{tabular}[t]{c}%
347       \@author
348     \end{tabular}\par}%
349   \vskip 1.5em%
350   {\large \@date \par}%           % Set date in \large size.
351 \end{center}\par

```

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

```

352 \@thanks
353 \vfil\null
354 \end{titlepage}%

```

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

```

355 \setcounter{footnote}{0}%
356 \global\let\thanks\relax
357 \global\let\maketitle\relax
358 \global\let\@thanks\@empty
359 \global\let\@author\@empty
360 \global\let\@date\@empty
361 \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.

```

362 \global\let\title\relax
363 \global\let\author\relax
364 \global\let\date\relax
365 \global\let\and\relax
366 }

```

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).

```

367 \else
368   \newcommand\maketitle{\par
369     \begingroup
370     \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
371     \def\@makefnmark%
372       {\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
373     \long\def\@makefntext##1{%
374       \@setpar{\@@par
375         \@tempdima = \hspace
376         \advance\@tempdima -1em
377         \parshape \@ne 1em \@tempdima}%
378       \par\parindent 1em \noindent
379       \hb@xt@\z@{\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`.

```

380   \iftwocolumn
381     \ifnum \col@number=\@ne
382       \@maketitle
383     \else
384       \twocolumn[\@maketitle]%
385     \fi
386   \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.

```

387     \newpage
388     \global\@topnum\z@ % Prevents figures from going at top of page.
389     \@maketitle
390     \fi

```

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

```

391   \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.

```

392   \endgroup
393   \setcounter{footnote}{0}%
394   \global\let\thanks\relax
395   \global\let\maketitle\relax
396   \global\let\@maketitle\relax
397   \global\let\@thanks\@empty
398   \global\let\@author\@empty
399   \global\let\@date\@empty
400   \global\let\@title\@empty
401   \global\let\title\relax
402   \global\let\author\relax

```

```

403 \global\let\date\relax
404 \global\let\and\relax
405 }

```

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

We always start a new page and put the title flushleft using a `\Large` bold font with thick rules above and below. Then we put the author information flushright in slanted type. This title will always show the date unless it is set to nothing, using the `\date{}` command.

```

406 \def\@maketitle{%
407   \newpage
408   \null
409   \longthickrule\vskip1.5em%
410   \let \footnote \thanks
411   {\secshape \parskip\z@ \parindent\z@
412   \Large\bfseries \@title \par}%
413   \vskip1.5em\longthickrule\vskip1.5em%
414   {\normalsize
415     \lineskip .5em%
416     \begin{flushright}%
417       {\slshape\@author\par}
418       \vskip 1em%
419       {\@date}%
420     \end{flushright}\par}%
421   \vskip 1.5em}
422 \fi

```

7.2 Chapters and Sections

7.2.1 Building blocks

The definitions in this part of the class file make use of two macros, `\@startsection` and `\secdef`, which are defined by `latex.dtx`. They are not described here, see the `classes.dtx` for more information.

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>	423 <code>\newcommand*\chaptermark[1]{}</code>
<code>\paragraphmark</code>	424 <code>\newcommand*\sectionmark[1]{}</code>
<code>\subparagraphmark</code>	425 <code>\newcommand*\subsectionmark[1]{}</code>
	426 <code>\newcommand*\subsubsectionmark[1]{}</code>
	427 <code>\newcommand*\paragraphmark[1]{}</code>
	428 <code>\newcommand*\subparagraphmark[1]{}</code>

7.2.3 Define Counters

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

```

429 (+refart)\setcounter{secnumdepth}{3}
430 (+refrep)\setcounter{secnumdepth}{2}

```

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

```

\c@subsection 431 \newcounter {part}
\c@subsubsection 432 (+refart)\newcounter {section}
\c@paragraph 433 (*refrep)
\c@subparagraph 434 \newcounter {chapter}
435 \newcounter {section}[chapter]
436 (/refrep)
437 \newcounter {subsection}[section]
438 \newcounter {subsubsection}[subsection]
439 \newcounter {paragraph}[subsubsection]
440 \newcounter {subparagraph}[paragraph]

```

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

`\thesection` `\arabic{COUNTER}` prints the value of `COUNTER` as an arabic numeral.
`\thesubsection` `\roman{COUNTER}` prints the value of `COUNTER` as a lowercase roman num-
`\thesubsubsection` beral.
`\theparagraph` `\Roman{COUNTER}` prints the value of `COUNTER` as an uppercase roman
`\thesubparagraph` numeral.
`\alph{COUNTER}` prints the value of `COUNTER` as a lowercase letter: 1 = a,
2 = b, etc.
`\Alph{COUNTER}` prints the value of `COUNTER` as an uppercase letter:
1 = A, 2 = B, etc.

```

441 \renewcommand\thepart          {\@Roman\c@part}
442 (+refart)\renewcommand\thesection  {\@arabic\c@section}
443 (*refrep)
444 \renewcommand\thechapter       {\@arabic\c@chapter}
445 \renewcommand\thesection       {\thechapter.\@arabic\c@section}
446 (/refrep)
447 \renewcommand\thesubsection    {\thesection.\@arabic\c@subsection}
448 \renewcommand\thesubsubsection {\thesubsection .\@arabic\c@subsubsection}
449 \renewcommand\theparagraph     {\thesubsubsection.\@arabic\c@paragraph}
450 \renewcommand\thesubparagraph  {\theparagraph.\@arabic\c@subparagraph}

```

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

```

451 (+refrep)\newcommand\@chapapp{}

```

7.2.4 Parts

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

In the refart 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`. As in other sectioning commands (cf. `\@startsection` in the L^AT_EX 2_ε kernel), we need to check the `@noskipsec` switch and force horizontal mode if it is set.

```

452 (*refart)
453 \newcommand\part{%
454   \ifnoskipsec \leavevmode \fi
455   \par
456   \addvspace{4ex}%
457   \@afterindentfalse
458   \secdef\@part\@spart}
459 (/refart)

```

For the refrep class things are a bit different.
 We start a new (righthand) page and use the *plain*.

```

460 (*refrep)
461 \newcommand\part{%
462   \if@openright
463     \cleardoublepage
464   \else
465     \clearpage
466   \fi
467   \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.

```

468     \if@twocolumn
469       \onecolumn
470       \@tempswatrue
471     \else
472       \@tempswafalse
473     \fi

```

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

```

474     \null\vfil

```

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

```

475     \secdef\@part\@spart}
476 (/refrep)

```

`\@part` This macro does the actual formatting of the title of the part. Again the macro is differently defined for the refart document class than for the document class refrep.

When `secnumdepth` is larger than -1 for the document class refart or -2 for the document class refrep, we have a numbered part, otherwise it is unnumbered.

```

477 (*refart | refrep)
478 \def\@part[#1]#2{%
479 (+refart)   \ifnum \c@secnumdepth >\m@ne
480 (+refrep)   \ifnum \c@secnumdepth >-2\relax
481     \refstepcounter{part}%
482     \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
483   \else
484     \addcontentsline{toc}{part}{#1}%
485   \fi

```

We print the title flush left, we also prevent breaking between lines and reset the font.

```

486     \longrule\medskip

```

```

487   {\parindent \z@ \raggedright
488     \interlinepenalty \@M
489     \normalfont

```

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

```

490     \Large
491 (+refart)     \ifnum \c@secnumdepth >\m@ne
492 (+refrep)     \ifnum \c@secnumdepth >-2 \relax
493     \thepart.\quad
494     \fi
495     #2\par \medskip
496     \longrule\bigskip%

```

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

```

497     \markboth{}{\par}%
498     \nobreak
499     \vskip 3ex
500     \@afterheading}
501 (/refart | refrep)

```

`\@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.

```

502 (*refart | refrep)
503 \def\@spart#1{%
504     \longrule\medskip
505     {\parindent \z@ \raggedright
506       \interlinepenalty \@M
507       \normalfont
508       \Large #1\par}%
509     \medskip\longrule
510     \nobreak
511     \vskip 3ex
512     \@afterheading}
513 (/refart | refrep)

```

7.2.5 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*.

```

514 (*refrep)
515 \newcommand\chapter{\if@openright\cleardoublepage\else\clearpage\fi
516                   \if@pageperchapter\setcounter{page}{1}\fi
517                   \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.

```

518         \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.

```

519             \@afterindentfalse
520             \secdef\@chapter\@schapter}

```

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

```

521 \def\@chapter[#1]#2{\ifnum \c@secnumdepth >\m@ne
522             \refstepcounter{chapter}%
523             \typeout{\@chapapp\space\thechapter.}%
524             \addcontentsline{toc}{chapter}%
525                 {\protect\numberline{\thechapter}#1}%
526             \else
527             \addcontentsline{toc}{chapter}{#1}
528             \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.

```

529             \chaptermark{#1}%
530             \addtocontents{lof}{\protect\addvspace{10\p@}}%
531             \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.

```

532             \if@twocolumn
533             \@topnewpage[\@makechapterhead{#2}]%
534             \else
535             \@makechapterhead{#2}%
536             \@afterheading
537             \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.

```

538 \def\@makechapterhead#1{%
539     \longthickrule\bigskip%
540     {\parindent \z@ \secshape \normalfont \Large\bfseries

```

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

```

541     \@hangfrom{\ifnum \c@secnumdepth >\m@ne
542         \@chapapp\space \thechapter\quad
543     \fi}%

```

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

```

544     #1\par}
545     \bigskip\longthickrule\bigskip
546 }

```


`\@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.

```
547 \def\@schapter#1{\if@twocolumn
548     \@topnewpage[\@makeschapterhead{#1}]%
549     \else
550     \@makeschapterhead{#1}%
551     \@afterheading
552     \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.

```
553 \def\@makeschapterhead#1{%
554     \longthickrule\bigskip%
555     {\parindent \z@ \secshape \normalfont
556     \Large \bfseries #1\par}
557     \bigskip\longthickrule\bigskip
558     }
559 \refrep)
```

7.2.6 Lower level headings

`\secshape`

```
560 \newcommand\secshape{\leftskip=-\leftmarginwidth%
561     \rightskip=\@flushglue%
562     \hyphenpenalty=2000}
```

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.

```
563 \newcommand\section{\@startsection {section}{1}{\z@}%
564     {-2ex \@plus -1ex \@minus -.2ex}%
565     {0.5ex \@plus .2ex}%
566     {\secshape\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.

```
567 \newcommand\subsection{\@startsection{subsection}{2}{\z@}%
568     {-1.5ex\@plus -.5ex \@minus -.2ex}%
569     {0.5ex \@plus .2ex}%
570     {\secshape\normalfont\normalsize\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.

```
571 \newcommand\subsubsection{\@startsection{subsubsection}{3}{\z@}%
572     {-1.5ex\@plus -.5ex \@minus -.2ex}%
573     {0.5ex \@plus .2ex}%
574     {\secshape\normalfont\normalsize\mdseries}}
```

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

```

575 \newcommand\paragraph{\@startsection{paragraph}{4}{\z0}%
576                               {2ex\@plus 1ex \@minus .2ex}%
577                               {-1em}%
578                               {\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`.

```

579 \newcommand\subparagraph{\@startsection{subparagraph}{5}{\parindent}%
580                               {2ex \@plus 1ex \@minus .2ex}%
581                               {-1em}%
582                               {\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', 'ii', ... , 'vi'. (I.e., `\@listiii` is called for a third-level list.) By convention, `\@listK` should set `\leftmarginK` to `\leftmarginK`.

```

\leftmargin For efficiency, level-one list's values are defined at top level, and \@listi is defined
\leftmargini to set only \leftmargin.
\leftmarginii When we are in two column mode some of the margins are set somewhat
\leftmarginiii smaller.
\leftmarginiv 583 \if@twocolumn
\leftmarginv 584 \setlength\leftmargini {2em}
\leftmarginvi 585 \else
586 \setlength\leftmargini {2.5em}
587 \fi

```

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.').

```

588 \setlength\leftmarginii {2.2em}
589 \setlength\leftmarginiii {1.87em}
590 \setlength\leftmarginiv {1.7em}
591 \if@twocolumn
592 \setlength\leftmarginv {.5em}
593 \setlength\leftmarginvi {.5em}
594 \else
595 \setlength\leftmarginv {1em}
596 \setlength\leftmarginvi {1em}
597 \fi

```

Here we set the top level `\leftmargin`.

```

598 \setlength\leftmargin {\leftmargini}

```

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

```

599 \setlength \labelsep {.5em}
600 \setlength \labelwidth{\leftmargini}

```

```

601 \addtolength\labelwidth{-\labelsep}

\@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.
602 (*refart | refrep)
603 \@beginparpenalty -\@lowpenalty
604 \@endparpenalty   -\@lowpenalty
605 \@itempenalty     -\@lowpenalty
606 (/refart | refrep)

\@listI \@listI defines top level and \@listi values of \leftmargin, \parsep, \topsep,
\@listi and \itemsep
607 (*refart | refrep)
608 \def\@listI{\leftmargin\leftmarginI
609             \parsep \parskip
610             \topsep \z@
611             \itemsep\z@}
612 \let\@listi\@listI

We have to initialise these parameters.
613 \@listi

\@listii Here are the same macros for the higher level lists.
\@listiii 614 \def\@listii {\leftmargin\leftmarginii
\@listiv 615             \labelwidth\leftmarginii
\@listv 616             \advance\labelwidth-\labelsep
\@listvi 617             \topsep \z@
618             \parsep \parskip
619             \itemsep \z@}
620 \def\@listiii{\leftmargin\leftmarginiii
621             \labelwidth\leftmarginiii
622             \advance\labelwidth-\labelsep
623             \topsep \z@
624             \parsep \parskip
625             \partopsep \z@
626             \itemsep \topsep}
627 \def\@listiv {\leftmargin\leftmarginiv
628             \labelwidth\leftmarginiv
629             \advance\labelwidth-\labelsep}
630 \def\@listv {\leftmargin\leftmarginv
631             \labelwidth\leftmarginv
632             \advance\labelwidth-\labelsep}
633 \def\@listvi {\leftmargin\leftmarginvi
634             \labelwidth\leftmarginvi
635             \advance\labelwidth-\labelsep}
636 (*refart | refrep)

```

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 here.

```

\theenumiii 637 (*refart | refrep)
\theenumiv 638 \renewcommand\theenumi {\@arabic\c@enumi}
639 \renewcommand\theenumii {\@alph\c@enumii}
640 \renewcommand\theenumiii{\@roman\c@enumiii}
641 \renewcommand\theenumiv {\@Alph\c@enumiv}

```

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

```

\labelenumii 642 \newcommand\labelenumi {\theenumi.}
\labelenumiii 643 \newcommand\labelenumii {(\theenumii)}
\labelenumiv 644 \newcommand\labelenumiii{\theenumiii.}
645 \newcommand\labelenumiv {\theenumiv.}

```

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

```

\p@enumiv 646 \renewcommand\p@enumii {\theenumi}
647 \renewcommand\p@enumiii {\theenumi(\theenumii)}
648 \renewcommand\p@enumiv {\p@enumiii\theenumiii}

```

7.3.3 Itemize

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

```

649 \newcommand\labelitemi {\textbullet}
650 \newcommand\labelitemii {\normalfont\bfseries \textendash}
651 \newcommand\labelitemiii{\textasteriskcentered}
652 \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`.

```

653 \newenvironment{description}
654     {\list{}{}}
655     {\labelsep\marginparsep
656     \labelwidth\leftmarginwidth
657     \advance\labelwidth by \leftmargin
658     \advance\labelwidth by -\labelsep
659     \let\makelabel\descriptionlabel}}
660     {\endlist}

```

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

```

661 \newcommand*\descriptionlabel[1]{%
662     \ifdescriptionleft\else \hfil\fi
663     \normalfont #1 \ifdescriptioncolon : \fi
664     \ifdescriptionleft \hfil \fi}

```

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.

```
665 \if@titlepage
666   \newenvironment{abstract}{%
667     \titlepage
668     \null\vfil
669     \@beginparpenalty\@lowpenalty
670     \begin{center}
671       \bfseries \abstractname
672       \@endparpenalty\@M
673     \end{center}}
674   {\par\vfil\null\endtitlepage}
```

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

```
675 \else
676   \newenvironment{abstract}{%
677     \if@twocolumn
678       \section*{\abstractname}%
679     \else
680       \small
681       \begin{center}%
682         {\bfseries \abstractname\vspace{-.5em}\vspace{\z@}}%
683       \end{center}%
684       \quote
685     \fi}
686   {\if@twocolumn\else\endquote\fi}
687 \fi
```

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`.

```
688 \newenvironment{verse}
689   {\let\\=\@centercr
690    \list{}{\itemsep \z@
691             \itemindent -1.5em%
692             \listparindent\itemindent
693             \rightmargin \leftmargin
694             \advance\leftmargin 1.5em}%
695    \item\relax}
696   {\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.

```
697 \newenvironment{quotation}
698     {\list{}{\listparindent 1.5em%
699         \itemindent \listparindent
700         \rightmargin \leftmargin
701         \parsep \z@ \@plus\p@}%
702     \item\relax}
703 {\endlist}
```

7.4.4 Quote

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

```
704 \newenvironment{quote}
705     {\list{}{\rightmargin\leftmargin}%
706     \item\relax}
707 {\endlist}
```

7.4.5 Example

\example The example environment is a verse environment with tt font which tries to avoid pagebrakes at the `\begin{example}`.

```
708 \newenvironment{example}
709     {\@beginparpenalty=\highpenalty
710     \let\=\@centercr
711     \list{}{\itemsep \z@
712         \itemindent -1.5em%
713         \listparindent\itemindent
714         \rightmargin \leftmargin
715         \advance\leftmargin 1.5em}%
716     \ttfamily
717     \item\relax}
718 {\endlist}
```

7.4.6 Theorem

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

7.4.7 Titlepage

titlepage In the normal environments, the titlepage environment does nothing but start and end a page, and inhibit page numbers. It also resets the page number to zero. In two-column style, it still makes a one-column page.

```
719 \newenvironment{titlepage}
720     {\if@twocolumn
721         \@restonecoltrue\onecolumn
722         \else
723         \@restonecolfalse\newpage
724         \fi
725     \thispagestyle{empty}%
726     \setcounter{page}\@ne
```

```

727     }
728     {\if@restonecol\twocolumn \else \newpage \fi
729     \if@twoside\else
730     \setcounter{page}\@ne
731     \fi
732     }

```

7.4.8 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.

```

733 (*refart)
734 \newcommand\appendix{\par
735 \setcounter{section}{0}%
736 \setcounter{subsection}{0}%
737 \gdef\thesection{\@Alph\c@section}}
738 (/refart)

```

In the report and book document classes the `\appendix` command must do the following:

- issue a `\newpage` if `pageperchapter` is defined, otherwise the `pagenumber` would come out wrong.
- 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.

```

739 (*refrep)
740 \newcommand\appendix{\par
741 \if@pageperchapter\newpage\fi
742 \setcounter{chapter}{0}%
743 \setcounter{section}{0}%
744 \gdef\@chapapp{\appendixname}%
745 \gdef\thechapter{\@Alph\c@chapter}}
746 (/refrep)

```

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`.

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

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

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

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

`\doublerulesep` The space between adjacent rules in the array and tabular environments is given by `\doublerulesep`.
750 `\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.)
751 `\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.
752 `\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`.
753 `\setlength\fboxsep {3\p@}`
754 `\setlength\fboxrule{.4\p@}`

7.5.5 Equation and eqnarray

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

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

```
755 (+refart)\renewcommand\theequation{\@arabic\c@equation}
756 (*refrep)
757 \@addtoreset{equation}{chapter}
758 \renewcommand\theequation
759 {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@equation}
760 (/refrep)
```

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

```
761 % \setlength\jot{3pt}
```

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

```
762 % \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 `\end@float`, which are defined in `latex.dtx`.

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

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

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 the figures are numbered per chapter.

```
763 <refart>
764 \newcounter{figure}
765 \renewcommand \thefigure {\@arabic\c@figure}
766 </refart>
767 <refrep>
768 \newcounter{figure}[chapter]
769 \renewcommand\thefigure
770     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@figure}
771 </refrep>
```

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

```
\ftype@figure 772 \def\fps@figure{tbp}
\ext@figure    773 \def\ftype@figure{1}
\num@figure   774 \def\ext@figure{lof}
               775 \def\fnum@figure{\figurename~\thefigure}
```

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

```
776 \newenvironment{figure}
777     {\@float{figure}}
778     {\end@float}
779 \newenvironment{figure*}
780     {\@dblfloat{figure}}
781     {\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 the tables are numbered per chapter.

```
782 \refart
783 \newcounter{table}
784 \renewcommand\thetable{\@arabic\c@table}
785 \refrep
786 \refrep
787 \newcounter{table}[chapter]
788 \renewcommand\thetable%
789     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@table}
790 \refrep
```

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

```
\ftype@table 791 \def\fps@table{tbp}
\ext@table    792 \def\ftype@table{2}
\num@table   793 \def\ext@table{lot}
              794 \def\fnm@table{\tablename~\thetable}
```

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

```
795 \newenvironment{table}
796     {\@float{table}}
797     {\end@float}
798 \newenvironment{table*}
799     {\@dblfloat{table}}
800     {\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.

```
801 \newlength\abovecaptionskip
```

```

802 \newlength\belowcaptionskip
803 \setlength\abovecaptionskip{10\p@}
804 \setlength\belowcaptionskip{0\p@}

```

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

```

805 \long\def\@makecaption#1#2{%
806   \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.

```

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

```

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

```

808   \ifdim \wd\@tempboxa >\hsize
809     #1: #2\par

```

If the caption fits, we center it.

```

810   \else
811     \global \@minipagefalse
812     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
813     \fi
814     \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 `\@newfontswitch`, 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 815 \DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
816 \DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
817 \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.

```

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

```

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

```

819 \DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
820 \DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
821 \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’.

```
822 \DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}
823 \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 μ units. Should be defined as a number like 2 or 1.7

```
824 \newcommand\@pnumwidth{1.55em}
825 \newcommand\@tocrmarg {2.55em}
826 \newcommand\@dotsep   {4.5}
827 (+refart)\setcounter{tocdepth}{3}
828 (+refrep)\setcounter{tocdepth}{2}
```

8.1.1 Table of Contents

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

```
829 \newcommand\tableofcontents{%
830 (*refrep)
831   \if@twocolumn
832     \@restonecoltrue\onecolumn
833   \else
834     \@restonecolfalse
835   \fi
```

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

```
836   \chapter*{\contentsname
837 //refrep)
838 (+refart)   \section*{\contentsname
839             \@mkboth{\contentsname}{\contentsname}}%
```

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

```
840   \@starttoc{toc}%
841   \if@restonecol\twocolumn\fi
842 }
```

`\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.

```
843 \newcommand*\l@part [2]{%
844   \ifnum \c@tocdepth >-2\relax
845 (+refart)   \addpenalty\secpenalty
846 (+refrep)   \addpenalty{-\@highpenalty}%
847   \addvspace{2.25em \@plus\p@}%
848   \begingroup
```

The 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.

```
849     \parindent \z@ \rightskip \@pnumwidth
850     \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.

```
851     {\leavevmode
852     \large \bfseries #1\hfil \hbox to\@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.

```
853     \nobreak
854     \global\@nobreaktrue
855     \everypar{\global\@nobreakfalse\everypar{}}%
856   \endgroup
```

857 `\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.

```
858 (*refrep)
859 \newcommand*\l@chapter[2]{%
860   \ifnum \c@tocdepth >\m@ne
861     \addpenalty{-\@highpenalty}%
862     \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.

```
863   \setlength\@tempdima{1.5em}%
864   \begingroup
865   \parindent \z@ \rightskip \@pnumwidth
866   \parfillskip -\@pnumwidth
```

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

```
867   \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.

```
868   \advance\leftskip\@tempdima
869   \hskip -\leftskip
870   #1\nobreak\hfil \nobreak\hbxt@\@pnumwidth{\hss #2}\par
871   \penalty\@highpenalty
872   \endgroup
873   \fi}
874 (/refrep)
```

`\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.

```
875 (*refart)
876 \newcommand*\l@section[2]{%
877   \ifnum \c@tocdepth >\z@
878     \addpenalty\@secpenalty
879     \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.

```
880   \setlength\@tempdima{1.5em}%
881   \begingroup
882   \parindent \z@ \rightskip \@pnumwidth
883   \parfillskip -\@pnumwidth
```

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

```
884   \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.

```
885 \advance\leftskip\@tempdima
886 \hskip -\leftskip
887 #1\nobreak\hfil \nobreak\hb@xt@\@pnumwidth{\hss #2}\par
888 \endgroup
889 \fi}
890 \refart}
```

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

```
891 (*refrep)
892 \newcommand*\l@section {\@dottedtocline{1}{1.5em}{2.3em}}
893 \refrep}
```

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

```
\l@subsection 894 (*refart)
\l@subsection 895 \newcommand*\l@subsection {\@dottedtocline{2}{1.5em}{2.3em}}
\l@paragraph 896 \newcommand*\l@subsection{\@dottedtocline{3}{3.8em}{3.2em}}
\l@subparagraph 897 \newcommand*\l@paragraph {\@dottedtocline{4}{7.0em}{4.1em}}
898 \newcommand*\l@subparagraph {\@dottedtocline{5}{10em}{5em}}
899 \refart}
900 (*refrep)
901 \newcommand*\l@subsection {\@dottedtocline{2}{3.8em}{3.2em}}
902 \newcommand*\l@subsubsection{\@dottedtocline{3}{7.0em}{4.1em}}
903 \newcommand*\l@paragraph {\@dottedtocline{4}{10em}{5em}}
904 \newcommand*\l@subparagraph {\@dottedtocline{5}{12em}{6em}}
905 \refrep}
```

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`.

```
906 \newcommand\listoffigures{%
907 (*refrep)
908 \if@twocolumn
909 \@restonecoltrue\onecolumn
910 \else
911 \@restonecolfalse
912 \fi
913 \chapter*\listfigurename
914 \refart}
915 (+refart) \section*\listfigurename
916 \mkboth{\listfigurename}%
917 {\listfigurename}%
918 \starttoc{lof}%
919 (+refrep) \if@restonecol\twocolumn\fi
920 }
```

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

```
921 \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`.

```
922 \newcommand\listoftables{%
923   \*refrep
924   \if@twocolumn
925     \@restonecoltrue\onecolumn
926   \else
927     \@restonecolfalse
928   \fi
929   \chapter*{\listtablename}
930 \*refrep
931 \+refart \section*{\listtablename}
932 \mkboth{\listtablename}{\listtablename}}%
933 \starttoc{lot}%
934 \+refrep \if@restonecol\twocolumn\fi
935 }
```

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

```
936 \let\l@table\l@figure
```

8.2 Bibliography

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

```
937 \newdimen\bibindent
938 \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.

```
939 \newenvironment{thebibliography}[1]
940 \+refart \section*{\refname}
941 \+refart \mkboth{\refname}{\refname}}%
942 \+refrep \chapter*{\bibname}
943 \+refrep \mkboth{\bibname}{\bibname}}%
944 \list{\@biblabel{\@arabic\c@enumiv}}%
945 \settowidth\labelwidth{\@biblabel{#1}}%
946 \leftmargin\labelwidth
947 \advance\leftmargin\labelsep
948 \openbib@code
949 \usecounter{enumiv}%
950 \let\p@enumiv\empty
951 \renewcommand\theenumiv{\@arabic\c@enumiv}}%
952 \sloppy
```



```

953         \clubpenalty4000
954         \@clubpenalty \clubpenalty
955         \widowpenalty4000%
956     \sfcode'\.=\@m}
957     {\def\@noitemerr
958      {\@latex@warning{Empty 'thebibliography' environment}}}%
959     \endlist}

```

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

```

960 \newcommand\newblock{\hskip .11em\@plus.33em\@minus.07em}

```

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

```

961 \let\@openbib@code\@empty

```

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

```

962 % \renewcommand*\@biblabel}[1]{[#1]\hfill}
963 %( \end{macrocode}
964 % \end{macro}
965 %
966 % \begin{macro}{\@cite}
967 %   The output of the |\cite| command is produced by this macro. The
968 %   default from \file{latex.dtx} is used.
969 %   \begin{macrocode}
970 % \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.

```

971 \newenvironment{theindex}
972     {\if@twocolumn
973      \@restonecolfalse
974      \else
975      \@restonecoltrue
976      \fi
977      \begin{fullpage}
978      \let\twocolumn\REF@twocolumn
979      \columnseprule \z@
980      \columnsep 35\p@
981 (+refart)      \twocolumn[\section*{\indexname}]%
982 (*refrep)
983      \if@pageperchapter
984      \setcounter{page}{1}
985      \ifnum \c@secnumdepth >\m@ne
986      \refstepcounter{chapter}%
987      \typeout{\@chapapp\space\thechapter.}%
988      \addcontentsline{toc}{chapter}
989      {\protect\numberline{\thechapter}\indexname}%

```

```

990             \else
991                 \addcontentsline{toc}{chapter}{\indexname}%
992             \fi
993             \addtocontents{lof}{\protect\addvspace{10\p@}}%
994             \addtocontents{lot}{\protect\addvspace{10\p@}}%
995             \twocolumn[\@makechapterhead{\indexname}]%
996         \else
997             \twocolumn[\@makeschapterhead{\indexname}]%
998         \fi
999 \end{refrep}
1000         \mkboth{\indexname}%
1001             {\indexname}%
1002         \thispagestyle{plain}\parindent\z@
1003         \parskip\z@ \@plus .3\p@\relax
1004         \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.

```

1005         {\end{fullpage}}\if@restonecol\onecolumn\else\clearpage\fi

```

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

```

\subitem 1006 \newcommand\@idxitem {\par\hangindent 40\p@}
\subsubitem 1007 \newcommand\subitem {\@idxitem \hspace*{20\p@}}
1008 \newcommand\subsubitem{\@idxitem \hspace*{30\p@}}

```

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

```

1009 \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.

```

1010 \renewcommand\footnoterule{%
1011     \kern-3\p@
1012     \hrule\@width.4\columnwidth
1013     \kern 2.6\p@}

```

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

```

1014 % \newcounter{footnote}
1015 \refrep\@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 \TeX code.

```

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

```

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

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

```

1016 \long\def\@makefnmark#1{%
1017   \setpar{\@par
1018     \@tempdima = \hsize
1019     \advance\@tempdima -1em
1020     \parshape \@ne 1em \@tempdima}%
1021   \par\parindent 1em\noindent
1022   \hb@xt@ \z@{\hss\@textsuperscript{\normalfont\@thefnmark}\,}#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.

```

1023 %\def\@makefnmark{\hbox{\@textsuperscript{\normalfont\@thefnmark}}}

```

9 New commands

`\@addmarginpar` Redefine the `\@addmarginpar` command to only use the left margin.

```

1024 \def\@addmarginpar{\@next\@marbox\@currlistf\@cons\@freelist\@marbox
1025   \@cons\@freelist\@currbox}\@latexbug\@tempcnta\@ne
1026   \if@twocolumn
1027     \if@firstcolumn \@tempcnta\m@ne \fi
1028   \else
1029     \@tempcnta\m@ne
1030   \fi
1031   \ifnum\@tempcnta <\z@ \global\setbox\@marbox\box\@currbox \fi
1032   \@tempdima\@mparbottom
1033   \advance\@tempdima -\@pageht
1034   \advance\@tempdima\ht\@marbox
1035   \ifdim\@tempdima >\z@
1036     \@@warning{Marginpar on page \thepage\space moved}%
1037   \else
1038     \@tempdima\z@

```

```

1039 \fi
1040 \global\@mparbottom\@pageht
1041 \global\advance\@mparbottom\@tempdima
1042 \global\advance\@mparbottom\dp\@marbox
1043 \global\advance\@mparbottom\marginparpush
1044 \advance\@tempdima -\ht\@marbox
1045 \global\setbox \@marbox
1046 \vbox {\vskip \@tempdima \box \@marbox}%
1047 \global \ht\@marbox \z@
1048 \global \dp\@marbox \z@
1049 \kern -\@pagedp
1050 \nointerlineskip
1051 \hb@xt@\columnwidth
1052   {\ifnum \@tempcnta >\z@
1053     \hskip\columnwidth \hskip\marginparsep
1054   \else
1055     \hskip -\marginparsep \hskip -\marginparwidth
1056   \fi
1057   \box\@marbox \hss}%
1058 \nointerlineskip
1059 \hbox{\vrule \@height\z@ \@width\z@ \@depth\@pagedp}}

```

9.1 Margin commands

`\marginlable` This defines three commands to put information in the margin: `\marginlabel` puts the argument into a flushright marginpar, `\attention` puts `\attentionsymbol` to the left of the text to mark an important piece of text and `\seealso` puts a `→` to the left of the margin to mark a reference within the text. `\attentionsymbol` is defined as `! →` but can be changed with a `\renewcommand{\attentionsymbol}{:-)}` command.

```

1060 \newcommand*{\marginlabel}[1]
1061 {\mbox{} \marginpar{\raggedleft #1}\ignorespaces}
1062 \newcommand*{\seealso}[1]
1063   {\mbox{} \marginpar{\small $\rightarrow$ #1}\ignorespaces}
1064 \newcommand*{\attention}[1][\attentionsymbol]
1065   {\mbox{} \marginpar{\raggedleft #1}}
1066 \newcommand*{\attentionsymbol}{\large\bfseries ! $\rightarrow$}

```

9.2 Rules

`\longrule` These rules are used in several places, like the title, new parts and chapters and
`\longthickrule` for maxi and fullpages.

```

1067 \def\longrule{\par\hb@xt@\linewidth{\hss
1068   \vrule width \fullwidth height 0.4\p@ depth \z@}\par}
1069 \def\longthickrule{\par\hb@xt@\linewidth{\hss
1070   \vrule width \fullwidth height 1.0\p@ depth \z@}\par}

```

9.3 Pages

`maxipage` The `\maxipage` is a minipage which uses the full width of the page with optional
`fullpage` rules on the top and bottom. A maxipage can not split over pages. You can use

it for wide tables, long math equations and the like. It can be used in floats.

The `\fullpage` changes the page layout such that normal text and all environments use the full width of the page. Inside the `\fullpage`-environment, the `\leftmarginwidth` is reset to 0, thus it is possible to start a new chapter inside a `\fullpage`. This will be used in the index.

```

1071 \newenvironment{maxipage}{\par
1072     \mbox{} \kern-\leftmarginwidth %\kern-\@totalleftmargin
1073     \begin{minipage}{\fullwidth}
1074     \medskip \ifmaxipagerule \hrule\medskip \fi
1075     \parskip = 0.5\baselineskip
1076     \def\marginpar{%
1077 (+refart)         \ClassError{Refart}
1078 (+refrep)         \ClassError{Refrep}
1079     {Marginpar not allowed within Maxipage.}
1080     {Where should I put them?\MessageBreak
1081     I'm using the full pagewidth.}}
1082     {\par \vskip\parskip
1083     \medskip \ifmaxipagerule \hrule\medskip \fi
1084     \end{minipage}}\par}
1085 \newenvironment{fullpage}{%
1086     \clearpage
1087     \textwidth=\fullwidth
1088     \addtolength\oddsidemargin {-\leftmarginwidth}
1089     \setlength\evensidemargin{\oddsidemargin}
1090     \leftmarginwidth=\z@
1091     \hsize=\fullwidth
1092     \linewidth=\fullwidth
1093     \columnwidth=\fullwidth
1094     \def\marginpar{%
1095 (+refart)         \ClassError{Refart}
1096 (+refrep)         \ClassError{Refrep}
1097     {Marginpar not allowed within Fullpage.}
1098     {Where should I put them? I'm already\MessageBreak
1099     using the whole page for text.}}
1100     {\clearpage}

```

9.4 Miscellaneous

`\condbreak` The `\condbreak{length}` controls pagebreaks: If less than length is left on this page it will be moved to the next page. Thus it will remain together, either on this page or on the next.

`\noparskip` removes the vertical `\parskip` like `\noindent` removes the `\parindent`.

```

1101 \def\condbreak#1{\vskip \z@ plus #1\pagebreak[3]\vskip \z@ plus -#1\relax}
1102 \def\noparskip{\vskip-\parskip}

```

`\REF@twocolumn` Since this layout does not support `\twocolumn` the command is disabled but saved in `\REF@twocolumn`. The saved version will be used in the index. This is still experimental! Don't rely on it in future releases.

```

1103 \let\REF@twocolumn\twocolumn
1104 \def\twocolumn{%
1105 (+refart)\ClassError{Refart}
1106 (+refrep)\ClassError{Refrep}

```

```

1107 {Sorry, there is no twocolumn layout in this class}
1108 {Can you imagine how twocolumn layout will look?\MessageBreak
1109 That's why!}}

```

9.5 Obsolete commands

Well, these comands are not really obsolete, but they are not implemented in this version and will not be implemented later unless there is popular demand.

`\makeauthor`: The author is printed when `\maketitel` is executed thus there is no need for this command.

`\setleftmarginwidth` has been used in version 1.1 to change the horizontal layout. I would prefer to set the `leftmarginfraction` instead but I'm still open to suggestions from users.

9.6 Future commands

The following commands are not yet implemented but sound like a good idea.

`\ppc` This gives you a pagecount per chapter like 1-1, 1-2, 2-1. Since this is often requested and would be usefull in a reference manual style. `\pageperchapter` is only supported in `refrep.cls`.

This version redefines the L^AT_EX `\@wrindex` command which writes the indexentry. This hack is needed to keep MakeIndex happy when processing the index-file. The `\ppc` command is responsible to extract the chapternumber from the index-entry and reformat it. The chapternumber can be a Roman or Alpha number but the page has to be arabic.

```

1110 (+refrep)\newif\if@pageperchapter \@pageperchapterfalse
1111 (+refrep)\newcommand{\pageperchapter}
1112 (+refrep)  {\@pageperchaptertrue
1113 (+refrep)  \let\ppthepage=\thepage
1114 (+refrep)  \renewcommand\@pnumwidth{2.55em}
1115 (+refrep)  \@openrighttrue
1116 (+refrep)  \renewcommand\thepage{%
1117 (+refrep)  \ifnum \c@chapter = \z@
1118 (+refrep)  \ppthepage
1119 (+refrep)  \else
1120 (+refrep)  \thechapter\ -- \arabic{page}
1121 (+refrep)  \fi
1122 (+refrep)  }
1123 (+refrep)  \def\@wrindex##1{%
1124 (+refrep)  \ifnum \c@chapter = \z@
1125 (+refrep)  \protected@write\@indexfile{%
1126 (+refrep)  {\string\indexentry{##1}{\arabic{page}}}%
1127 (+refrep)  \else
1128 (+refrep)  \protected@write\@indexfile{%
1129 (+refrep)  {\string\indexentry{##1/ppc{\thechapter}}%
1130 (+refrep)  {\arabic{page}}}%
1131 (+refrep)  \fi
1132 (+refrep)  \endgroup
1133 (+refrep)  \@esphack
1134 (+refrep)  }
1135 (+refrep)  \def\ppc##1##2{##1 -- ##2}

```

1136 (+refrep) }

`\leftmarginfraction` This provides an interface to change the horizontal layout. In this version the margin is set to 0.3 fullwidth, this may change in future versions.

10 Initialization

10.1 Words

`\contentsname` 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
`\listfigurename`
`\listtablename` English words that require replacement are defined below in command names.

```
\refname 1137 \newcommand\contentsname{Contents}
\bibname  1138 \newcommand\listfigurename{List of Figures}
\indexname 1139 \newcommand\listtablename{List of Tables}
\figurename 1140 (+refart)\newcommand\refname{References}
\tablename 1141 (+refrep)\newcommand\bibname{Bibliography}
\partname  1142 \newcommand\indexname{Index}
\chaptername 1143 \newcommand\figurename{Figure}
\appendixname 1144 \newcommand\tablename{Table}
\abstractname 1145 \newcommand\partname{Part}
1146 (+refrep)\newcommand\chaptername{Chapter}
1147 \newcommand\appendixname{Appendix}
1148 \newcommand\abstractname{Abstract}
```

10.2 Date

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

```
1149 \def\today{\ifcase\month\or
1150   January\or February\or March\or April\or May\or June\or
1151   July\or August\or September\or October\or November\or December\fi
1152   \space\number\day, \number\year}
```

10.3 Two column mode

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

```
1153 \setlength\columnsep{10\p@}
```

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

```
1154 \setlength\columnseprule{0\p@}
```

10.4 The page style

We have *plain* pages in the document classes *refart* and *refrep* unless the user specified otherwise. We use arabic pagenumbers.

```
1155 \pagestyle{plain}
1156 \pagenumbering{arabic} % Arabic page numbers
```

10.5 Single or double sided printing

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

```
1157 \if@twoside
1158 \else
1159   \raggedbottom
1160 \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.

```
1161 \if@twocolumn
1162   \twocolumn
1163   \sloppy
1164   \flushbottom
```

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

```
1165 \else
1166   \onecolumn
1167 \fi
1168 </refart | refrep>
```