

# The Octavo Package

Stefan A. Revets

Version v1.1, 2003/09/06

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background . . . . .	1
1.2	Rationale . . . . .	2
1.3	Some notes on making books with octavo . . . . .	3
1.4	Acknowledgements . . . . .	4
<b>2</b>	<b>Preliminaries</b>	<b>4</b>
<b>3</b>	<b>Class options</b>	<b>5</b>
3.1	Declaring options . . . . .	5
3.2	Executing options . . . . .	7
<b>4</b>	<b>Document layout</b>	<b>7</b>
4.1	Fount sizes . . . . .	7
4.1.1	Fount changes . . . . .	10
4.2	Paragraphs . . . . .	10
4.3	Page layout . . . . .	11
4.3.1	Vertical spacing . . . . .	11
4.3.2	Text dimensions . . . . .	12
4.3.3	Margin dimensions . . . . .	12
4.3.4	Footnotes . . . . .	13
4.3.5	Float placement . . . . .	13
4.4	Page styles . . . . .	16
4.4.1	Marking conventions . . . . .	16
4.4.2	The headings and myheadings styles . . . . .	16
<b>5</b>	<b>Document markup</b>	<b>17</b>
5.1	The title . . . . .	17
5.2	Chapters and section . . . . .	18
5.2.1	Definition of counters . . . . .	19
5.2.2	Front, main and back matter . . . . .	20
5.2.3	Parts . . . . .	20
5.2.4	Chapters . . . . .	22
5.2.5	Lower-level headings . . . . .	24
5.3	Lists . . . . .	25
5.3.1	List parameters . . . . .	25
5.3.2	Enumerate . . . . .	27

5.3.3	Itemise . . . . .	27
5.3.4	Description . . . . .	27
5.4	Defining new environments . . . . .	28
5.4.1	Verse . . . . .	28
5.4.2	Quotation . . . . .	28
5.4.3	Quote . . . . .	28
5.4.4	Titlepage . . . . .	28
5.4.5	Appendix . . . . .	29
5.5	Setting environment parameters . . . . .	29
5.6	Floating objects . . . . .	30
<b>6</b>	<b>Cross referencing</b>	<b>31</b>
6.1	Table of contents, etc. . . . .	32
6.2	Bibliography . . . . .	34
6.3	The index . . . . .	35
6.4	Footnotes . . . . .	36
<b>7</b>	<b>Initialisation</b>	<b>36</b>
7.1	Heading names . . . . .	36
7.2	Date . . . . .	37
7.3	Double columns . . . . .	37
7.4	Page style . . . . .	37

# 1 Introduction

Donald Knuth developed T<sub>E</sub>X because of the dissatisfaction he felt with the commercial typesetting of his books ‘The Art of Computer programming’ [1]. The program grew into a typesetting system, with a great deal of attention and ingenuity devoted to the calculations of the placement of letters, words and paragraphs on a page. More people began to use T<sub>E</sub>X, and, with the advent of L<sup>A</sup>T<sub>E</sub>X, ready-made class files became part of the system. These style files tend to show off the capabilities of the system, rather than conform to good design, or, as Philip Taylor [3, 4] put it ‘...all of which shriek “T<sub>E</sub>X” from every page’.

Class files have been proposed and are available from CTAN which give the more fastidious user the means with which to moderate at least some of the excesses of the standard L<sup>A</sup>T<sub>E</sub>X styles. The KOMA package is a good example, and so is the excellent Memoir class of Peter Wilson [8]. The manual of this package is a worthwhile read, especially since works on design and style in book making are not all that easy to obtain.

## 1.1 Background

To *make* a book is an interesting and somewhat involved process [5]. The text is set in type and printed on paper, the pages are gathered and folded into signatures, and these are then bound and covered.

The printing is done on large sheets of paper, which may have to be cut before they are folded into signatures. These are passed on to the bookbinder who sews the signatures into the book proper. As is so often the case with venerable crafts, a florid terminology developed which facilitated contacts between the guilds

of printers and binders, while at the same time inspiring some sort of grudging respect in the uninitiated, keeping them at bay.

The sheets of paper, the broadsides, came in a limited number of more or less standard sizes. These sheets are known under the names Foolscap, Crown, Post, Demy, Medium, Royal, Super Royal and Imperial. Folding the sheets adds another set of terms: a single fold yields a Folio, a double fold a Quarto, a triple fold an Octavo and so on. Hence, the combination of a sheetname and the number of folds denotes a particular and precise size of the book [5, 7], for example, a crown octavo measures 191x127 mm.

Jan Tschichold devoted a lifetime to the design and production of books. Fortunately, he wrote many of his views down. Later in life, he assembled his findings in an overview [6], discussing in a more coherent, concentrated fashion the questions which face anyone wishing to make a book. Of particular interest is his analysis of size and proportions of page, text and margins. Through his own historical investigations, he made, and continues to make, a very strong and convincing case against the more ‘modern’ tendency of making squarish books. His elegant, eloquent and well illustrated discourse guides us away from the ugliness of the convenient but rather square A format (proportion 1:1.414) back to the more rectangular formats with proportions tending towards the Golden Ratio (from 1:1.5 to 1:1.618). It is certainly no accident that the classical sheets of paper traditionally used in the printing of books show proportions which fall very nicely within this range.

Tschichold analysed the size and disposition of the text block relative to the page as well. By 1953, he published his discovery of the late gothic workshop secrets of determining the margins on a page. Rosarivo [2] confirmed this determination through his own, independent discovery of a different method which happily yielded the same results. Rosarivo’s method is very easy to use and implement automatically: divide the height and width of the page into an equal number, assign one part to make up inner and upper margin, and two parts to make up outer and lower margin, with the remainder defining the area available for printing the body text.

The amount of leading between lines of print depends on the nature of the fount, its size, and the length of the line. Also, it should remain fixed for the entire text, yielding a constant, regular grid of lines. Tschichold discussed the leading of lines of text in some detail, and similar views can be found in Williamson’s book [7].

It should be clear by now that the standard book class included in  $\text{\LaTeX}$  requires a fair amount of adjustment to reflect more closely the design principles just highlighted. The Octavo class is an attempt to provide such a class file.

## 1.2 Rationale

The Octavo class is in essence a modification of the standard  $\text{\LaTeX}$  book class, with a similar, limited amount of options and choices. These limitations are there because a number of inbuilt choices have been made on the basis of design principles and guidelines garnered from a variety of authors and books. The class is christened Octavo because its express aim is to implement many of these classical design elements.

First of all, the option of page sizes are the classical octavo sizes. The four smallest ones, i.e., foolscap, crown, post and largepost can be made to fit two

by two on A4 paper, allowing one to produce books in these formats on printers routinely available.

The size of the margins follows the analysis by Tschichold, but instead of adopting the 6:9 ratio of division of the page (which yields only a 44% covering of a page by text), a 7:10 ratio is here implemented (yielding a 49% cover of a page by text).

Including a title page is very necessary when making a book, but the way in which to do this is not as straightforward a matter as that may seem at first. Title pages (plural) are much better when they are treated, and made, as an integral part of the endpapers. When binding (good) books, endpapers fulfill a functional and aesthetical role, and a separate folio bearing the titling, adds to the strength, and beauty, of the endpapers [5].

Ideally, a separate folio should be made, bearing on the first recto page the bastard title, on the verso possibly a frontispiece, or failing something suitable, left blank. The next recto bears title, author, publisher and the like, and the final verso provides space for the biblio or imprint. From a design perspective, it is not really feasible, nor desirable, to lay down hard and fast rules for the organisation or the placement of the various elements needed on a titlepage. In practice, it is much easier to do this as a separate document, especially when the text is to be organised into signatures.

An attempt is made here to try and maintain a uniform printing grid across the text. Amongst the  $\text{\TeX}$  algorithms, Knuth developed a set with the aim of keeping the overall text block on each page the same size. If necessary, this means allowing the change of spacing between paragraphs, around floating objects, or indeed the leading within a paragraph. To counteract this behaviour completely would require a rewriting of part of the  $\text{\TeX}$  code, which really is too daunting a task. However, it seems possible, through a judicious setting of parameters, lengths and sizes, almost to achieve the same result. In principle, the `octavo` class will maintain the regularity of the grid within the text body, i.e., the leading should remain constant, and successive paragraphs should not become separated by extra space. Adjustments, if and when they become necessary, will be made to the spacing around headings, lists and floats. Nevertheless, there will be times when the actual text, the parameters defined by the `octavo` class, and the  $\text{\TeX}$  kernel will conspire to produce truly horrible results. Such misfortunes call for direct intervention, or, the user willing, suggestions of improving the definitions of the parameters proposed here for the `octavo` class.

### 1.3 Some notes on making books with octavo

Thanks to excellent works on bookbinding such as the book by Lawrence Town [5], the physical making of books lies well within reach of anyone caring to make the effort. Thanks to a number of additional programs, also available from CTAN, it is a straightforward matter to organise and print out a text in signatures. Here are some of the means which have proved to be both useful and effective.

As already mentioned, the formats foolscap, crown, post and largepost can be made to sit on an A5 page. This means that four pages can be put on a single A4 sheet, a set of which can be made to form a signature. To do so requires a few careful but simple adjustments.

A problem comes up with the imposition of the textpages on an A4 sheet. The four formats mentioned are all narrower than an A5 sheet, and hence ex-

tra space is left protruding at the outer margins. This is not a problem with the odd-numbered pages, as it is their inner margin which is made to coincide with the inner margin of the A5 sheet. However, the even-numbered pages have their outer margin made to coincide with the outer margin of the A5 sheet, the latter of course being too wide. What should happen is to have the inner margin of the even-numbered sheet coincide with the inner margins of the A5 sheet. The simplest solution is to add a single line to the preamble of the document: `\addtolength{\evensidemargin}{148.5mm-\textwidth}`, rather than try and reposition only the even-numbered pages with `dvidvi`, `dvips`, and the like.

Organising the pages into signatures can be done quite easily. With `dvips`:

1. `dvips -ta5 <file>`
2. `psbook -s16 <file>.ps <file>.tmp`
3. `psnup -pa4 -Pa5 -l -2 -s1.0 <file>.tmp <file>.ps`

or with `dvihtmlj` or similar, using the switches:

```
prthpljh <file> +columns:2 +rows:1 +section-size:4
+full-last-section +page-width:148.5mm +page-height:210mm
+landscape-mode +double-sided:odd[even]
```

These sets of commands and switches yield signatures made up of 16 pages on 4 sheets. Such an arrangement is very effective and handy when binding the book. It may be necessary to instruct the printer to shift each page a little on the sheet to ensure perfect register, and a little experimenting may be called for.

## 1.4 Acknowledgements

Octavo is a modification of `classes.dtx` written by Leslie Lamport (1992), Frank Mittelbach (1994-97) and Johannes Braams (1994-97). As can be seen from the code, my own input is restricted to a tweaking of some parameters and true credit is due to Lamport, Mittelbach and Braams for their monumental efforts.

## References

- [1] Knuth, D. 1998. *Digital Typography*. CSLI Publications, Stanford.
- [2] Rosarivo, R. 1961. *Divina proportio typographica*. Scherpe, Krefeld.
- [3] Taylor, P. 1998. *Book design for T<sub>E</sub>X users, Part 1: Theory*. TUGBoat, 19:65–74.
- [4] Taylor, P. 1999. *Book design for T<sub>E</sub>X users, Part 2: Practice*. TUGBoat, 20:378–389.
- [5] Town, L. *Bookbinding by hand*. Faber & Faber, London.
- [6] Tschichold, J. 1987. *Ausgewählte Aufsätze über Fragen der Gestalt des Buches und der Typographie*. Birkhäuser Verlag, Basel.
- [7] Williamson, H. 1966. *Methods of book design*. Oxford University Press, Oxford.
- [8] Wilson, P. 2001. *The Memoir class for configurable typesetting*. CTAN. `macros\latex\contrib\memoir`

## 2 Preliminaries

We begin by declaring a number of commands, parameters and control sequences which will be used later on.

<code>\@ptsize</code>	The controlsequence stores the second digit of the pointsize in which the body of the text will be typeset. <pre>1 \*octavo 2 \newcommand\@ptsize{}</pre>
<code>\if@restonecol</code>	In texts printed in two columns, this switch is used to remember to switch back to two column if there has been a temporary change to single column mode. <pre>3 \newif\if@restonecol</pre>
<code>\if@titlepage</code>	Usually, a titlepage is produced with a book. However, if a set of endpapers is made, titlepages can be made separately from the body of the book, and it is useful to omit the production of a titlepage. <pre>4 \newif\if@titlepage 5 \@titlepagetrue</pre>
<code>\if@openright</code> <code>\if@openbib</code>	Another set of switches, to indicate if new chapters must start on a right-hand page. Normally they do, but novels and the like may well benefit from having new chapters start on either side. <pre>6 \newif\if@openright 7 \newif\if@openbib 8 \@openbibfalse</pre>
<code>\if@mainmatter</code>	A switch to indicate when we are processing the main material of the book. <pre>9 \newif\if@mainmatter 10 \@mainmattertrue</pre>

## 3 Class options

### 3.1 Declaring options

The options available to the octavo class are essentially the same as those defined for the standard book class. What is different and new is a set of papersize options.

<code>\paperheight</code> <code>\paperwidth</code>	We begin with the papersizes. These are largely the traditional octavo sizes. The option <code>largepost</code> is slightly different: it is an example of Tschichold's pentagon-based proportion (1:1.538) applied to a page height of an a5 page. <pre>11 \DeclareOption{foolscap} 12   {\setlength\paperheight{171mm}% 13    \setlength\paperwidth{108mm}} 14 \DeclareOption{crown} 15   {\setlength\paperheight{191mm}% 16    \setlength\paperwidth{127mm}} 17 \DeclareOption{post} 18   {\setlength\paperheight{194mm}% 19    \setlength\paperwidth{122mm}} 20 \DeclareOption{largepost} 21   {\setlength\paperheight{210mm}%</pre>
---	---

```

22 \setlength\paperwidth{137mm}
23 \DeclareOption{demy}
24   {\setlength\paperheight{222mm}%
25    \setlength\paperwidth{143mm}}
26 \DeclareOption{medium}
27   {\setlength\paperheight{229mm}%
28    \setlength\paperwidth{146mm}}
29 \DeclareOption{royal}
30   {\setlength\paperheight{254mm}%
31    \setlength\paperwidth{159mm}}
32 \DeclareOption{superroyal}
33   {\setlength\paperheight{267mm}%
34    \setlength\paperwidth{171mm}}
35 \DeclareOption{imperial}
36   {\setlength\paperheight{279mm}%
37    \setlength\paperwidth{191mm}}

```

**\@ptsize** The choice of type size is kept through the last digit of the size in **\@ptsize**. There are apparently good, or at least, historical reasons for doing so. Since this may well ensure compatibility with other packages, the method is maintained here.

```

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

```

**twoside** When printing books, two-sided printing is the norm. This means that we have to set the **if@mparswitch** to get any margin paragraphs into the outside margin.

```

41 \DeclareOption{twoside}{\@twosidettrue \@mparswitchtrue}

```

**draft** When preparing a text for final tweaking, the **draft** mode may come in handy, as it will show on the page where overfull boxes appear.

```

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

```

**titlepage** Having argued that it is much better to deal with the titlepage separately from the book itself, we do leave the choice of a titlepage as integral part of the book as an option. However, the default will be to have no titlepage.

```

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

```

**openright** New chapters normally start on a right-hand page in books, but novels, or poetry books may well be better off by allowing chapters to begin on either side.

```

46 \DeclareOption{openright}{\@openrighttrue}
47 \DeclareOption{openany}{\@openrightfalse}

```

**twocolumn** Single or double column arrangement of the body text is chosen with this option. The choice does require some care, as the width of a line of text that can be read comfortably depends on the total number of ens, and this is a function of the fount, the fount size and the text width. The larger formats almost always benefit from being typeset in two columns.

```

48 \DeclareOption{onecolumn}{\@twocolumnfalse}
49 \DeclareOption{twocolumn}{\@twocolumntrue}

```

- leqno** When including mathematical formulæ, equation numbers are usually placed to the right, but the **leqno** option places them on the left of the equation. The **fleqn** option redefines the math display environment so that equations are typeset flush with the left margin and indented by `\mathindent` from the prevailing left margin.
- ```
50 \DeclareOption{leqno}{\input{leqno.clo}}
51 \DeclareOption{fleqn}{\input{fleqn.clo}}
```
- openbib** The format of the bibliography is set to an ‘open’ one, that is, each block starts on a new line, with successive lines in a block indented by the amount defined by `\bibindent`.
- ```
52 \DeclareOption{openbib}{\@openbibtrue}
```

## 3.2 Executing options

Here, default options are executed in order to initialise a number of variables. The `\ProcessOptions` command then ensures the execution of the code for every option declared and for which the user included an option in the `\documentclass` command.

```
53 \ExecuteOptions{crown,10pt,twoside,final,notitlepage,openright,onecolumn}
54 \ProcessOptions
```

After the execution of all the options, we can now load the class option file which contains all the size dependent code.

```
55 \input{oct1\@ptsize.clo}
56 \</octavo>
```

## 4 Document layout

### 4.1 Fount sizes

The user has access to a number of commands which change the size of the fount, relative to the ‘main’ size used for the bulk of the text. These `\size` commands issue a `\setfontsize\size<font-size>\baselineskip` where:

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

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

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

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

- \normalsize** The user command to obtain the ‘main’ size is `\normalsize`. L<sup>A</sup>T<sub>E</sub>X uses `\@normalsize` when referring to the main size and maintains this value even if `\normalsize` is redefined. The `\normalsize` macro also sets values for `\abovedisplayskip`, `\abovedisplayshortskip` and `\belowdisplayshortskip`.



```

57 <*10pt | 11pt | 12pt>
58 \renewcommand\normalsize{%
59 <*10pt>
60   \@setfontsize\normalsize\@xpt\@xipt
61   \abovedisplayskip 12\p@ \@plus 2\p@ \@minus 2\p@
62   \abovedisplayshortskip \z@ \@plus 3\p@
63   \belowdisplayshortskip 6\p@ \@plus 3\p@ \@minus 3\p@
64 </10pt>
65 <*11pt>
66   \@setfontsize\normalsize\@xipt{13.2}%
67   \abovedisplayskip 13.2\p@ \@plus 3\p@ \@minus 3\p@
68   \abovedisplayshortskip \z@ \@plus 3\p@
69   \belowdisplayshortskip 6.6\p@ \@plus 3\p@ \@minus 3\p@
70 </11pt>
71 <*12pt>
72   \@setfontsize\normalsize\@xipt\@xivpt
73   \abovedisplayskip 14.4\p@ \@plus 3\p@ \@minus 3\p@
74   \abovedisplayshortskip \z@ \@plus 3\p@
75   \belowdisplayshortskip 7.2\p@ \@plus 3\p@ \@minus 3\p@
76 </12pt>

```

The `\belowdisplayskip` equals `\abovedisplayskip`. The parameters of the first-level list are always given by `\@listI`.

```

77   \belowdisplayskip \abovedisplayskip
78   \let\@listi\@listI}

```

Initially, the `normalsize` font is put into use for the document

```

79 \normalsize

```

`\small` Similar to `\normalsize`

```

80 \newcommand\small{%
81 <*10pt>
82   \@setfontsize\small\@ixpt\@xipt%
83   \abovedisplayskip 11\p@ \@plus 2\p@ \@minus 2\p@
84   \abovedisplayshortskip \z@ \@plus 3\p@
85   \belowdisplayshortskip 5.5\p@ \@plus 2\p@ \@minus 2\p@
86   \def\@listi{\leftmargin\leftmarginI
87     \topsep 5.5\p@ \@plus 2\p@ \@minus 2\p@
88     \parsep \z@
89     \itemsep \z@}%
90 </10pt>
91 <*11pt>
92   \@setfontsize\small\@xpt\@xipt
93   \abovedisplayskip 12\p@ \@plus 2\p@ \@minus 2\p@
94   \abovedisplayshortskip \z@ \@plus 3\p@
95   \belowdisplayshortskip 6\p@ \@plus 3\p@ \@minus 3\p@
96   \def\@listi{\leftmargin\leftmarginI
97     \topsep 6\p@ \@plus 2\p@ \@minus 2\p@
98     \parsep \z@
99     \itemsep \z@}%
100 </11pt>
101 <*12pt>
102   \@setfontsize\small\@xipt{13.2}%
103   \abovedisplayskip 13.2\p@ \@plus 3\p@ \@minus 3\p@
104   \abovedisplayshortskip \z@ \@plus 3\p@

```

```

105     \belowdisplayshortskip 6.6\p@ \@plus 3\p@ \@minus 3\p@
106     \def\@listi{\leftmargin\leftmargini
107         \topsep 6.6\p@ \@plus 3\p@ \@minus 3\p@
108         \parsep \z@
109         \itemsep \z@}%
110 \
```

`\footnotesize` Again, very similar to `\normalsize`

```

112 \newcommand\footnotesize{%
113 (*10pt)
114     \@setfontsize\footnotesize\@viipt{9.5}%
115     \abovedisplayskip 9.5\p@ \@plus 2\p@ \@minus 2\p@
116     \abovedisplayshortskip \z@ \@plus 2\p@
117     \belowdisplayshortskip 5\p@ \@plus 2\p@ \@minus 2\p@
118     \def\@listi{\leftmargin\leftmargini
119         \topsep 5\p@ \@plus 2\p@ \@minus 2\p@
120         \parsep \z@
121         \itemsep \z@}%
122 \
```

`\scriptsize` These macros select new font sizes, but leave display and lists alone.

```

123 (*11pt)
124     \@setfontsize\footnotesize\@ixpt\@xipt%
125     \abovedisplayskip 11\p@ \@plus 2\p@ \@minus 2\p@
126     \abovedisplayshortskip \z@ \@plus 2\p@
127     \belowdisplayshortskip 5.5\p@ \@plus 2\p@ \@minus 2\p@
128     \def\@listi{\leftmargin\leftmargini
129         \topsep 5.5\p@ \@plus 2\p@ \@minus 2\p@
130         \parsep \z@
131         \itemsep \z@}%
132 \
```

```

133 (*12pt)
134     \@setfontsize\footnotesize\@xpt\@xipt
135     \abovedisplayskip 12\p@ \@plus 2\p@ \@minus 2\p@
136     \abovedisplayshortskip \z@ \@plus 3\p@
137     \belowdisplayshortskip 6\p@ \@plus 3\p@ \@minus 3\p@
138     \def\@listi{\leftmargin\leftmargini
139         \topsep 6\p@ \@plus 2\p@ \@minus 2\p@
140         \parsep \z@
141         \itemsep \z@}%
142 \
```

```

143     \belowdisplayskip \abovedisplayskip}

\scriptsize \tiny 144 (*10pt)
\large 145 \newcommand\scriptsize{\@setfontsize\scriptsize\@viipt\@viipt}
\Large 146 \newcommand\tiny{\@setfontsize\tiny\@vpt\@vpt}
\LARGE 147 \newcommand\large{\@setfontsize\large\@xiipt{14}}
\huge 148 \newcommand\Large{\@setfontsize\Large\@xivpt{18}}
\Huge 149 \newcommand\LARGE{\@setfontsize\LARGE\@xxvpt{22}}
150 \newcommand\huge{\@setfontsize\huge\@xxvpt{25}}
151 \newcommand\Huge{\@setfontsize\Huge\@xxvpt{30}}
152 \
```

```

153 (*11pt)
154 \newcommand\scriptsize{\@setfontsize\scriptsize\@viipt{9.5}}
```

```

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

```

#### 4.1.1 Fount changes

Fount changes should really be done with the  $\text{\LaTeX} 2_{\epsilon}$  commands, but quite a few packages rely on the older versions of these commands. Therefore, for the sake of compatibility, the old commands are retained.

```

\rm These are commands to change the fount family.
\tt 172 <octavo>
\sf 173 \DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
    174 \DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
    175 \DeclareOldFontCommand{\tt}{\normalfont\ttfamily}{\mathtt}

\bf The command to change to the bold series
    176 \DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}

\sl And finally, the commands to change the shape of the founts.
\it 177 \DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
\sc 178 \DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
    179 \DeclareOldFontCommand{\sc}{\normalfont\scshape}{\@nomath\sc}

```

## 4.2 Paragraphs

The parameters which control  $\text{\TeX}$ 's behaviour when typesetting paragraphs receive a bit of a tweak here. Contrary to the usual behaviour of modifying the grid with glue when difficulties are encountered with vertical space, here we shall try to counteract these tendencies and enforce as much as possible uniformity of the grid of lines.

```

\lineskip These two parameters influence  $\text{\TeX}$  when two lines come too close.
\normallineskip 180 \setlength\lineskip{1\p@}
    181 \setlength\normallineskip{1\p@}

\baselinestretch Used by  $\text{\TeX}$  as a multiplier for \baselineskip. Default behaviour is not to
stretch the baselines. If this command does not resolve to 'empty', any plus or
minus part in the specification of \baselineskip is ignored.

```

```

182 \renewcommand\baselinestretch{}
183 </octavo>

\parskip Extra vertical space can be put between successive paragraphs with \parskip as
it is usually defined as a zero rubber length. Here, it is set to zero, without rubber
extention, hopefully helping to maintain the regular grid.
184 < *10pt | 11pt | 12pt>
185 \setlength\parskip{0\p@}

\parindent The value of \parindent was set to a variety of sizes depending on the fount size
and whether single- or double-column is in use. Using the em measure relates the
indent to the fount in use, and there is little merit in having a marginally wider
indent when typesetting in a single column.
186 \setlength\parindent{1em}
187 </10pt | 11pt | 12pt>

\@lowpenalty Command such as \nopagebreak or \nolinebreak put in penalties to discourage
\@medpenalty the placement of these breaks. Depending on the argument with which they are
\@highpenalty put, a low, medium or high penalty is chosen.
188 < *octavo>
189 \@lowpenalty 51
190 \@medpenalty 151
191 \@highpenalty 301
192 </octavo>

```

## 4.3 Page layout

Here the class file deviates more noticeably from the standard book class. To begin with, all dimensions are measured from the true upper left corner of the page. Dimensions and placement of the text area is directly related to and determined as proportions of the dimensions of the page.

```

\hoffset First of all, the origin of the measurements is placed at the upper left corner of
\voffset the page, against the TEX standard of having it 1 inch away.
193 < *10pt | 11pt | 12pt>
194 \setlength\hoffset{-1in}
195 \setlength\voffset{-1in}

```

### 4.3.1 Vertical spacing

```

\headheight The \headheight is the height of the box containing the running head. The
\headsep \headsep is the distance between the bottom of the box with the running head
\topskip and the top of the text. The \topskip is the \baselineskip for the first line on
the page.
196 \setlength\headheight{12\p@}
197 <10pt>\setlength\headsep{6\p@}
198 <11pt>\setlength\headsep{7\p@}
199 <12pt>\setlength\headsep{8\p@}
200 <10pt>\setlength\topskip{10\p@}
201 <11pt>\setlength\topskip{11\p@}
202 <12pt>\setlength\topskip{12\p@}

```

`\footskip` This is the distance from the baseline of the box containing the running foot to the bottom of the text.

```
203 <10pt>\setlength\footskip{25\p@}
204 <11pt>\setlength\footskip{27.5\p@}
205 <12pt>\setlength\footskip{30\p@}
```

`\maxdepth` The `\maxdepth` register has a function similar to that of `\topskip`. The register `\@maxdepth` should always contain a copy of `\maxdepth`, and this is ensured by setting it when issuing the command `\begin{document}`. In  $\text{\TeX}$  and  $\text{\LaTeX}2.09$  it was fixed at 4pt. Following  $\text{\LaTeX}2_{\epsilon}$ , it now depends on the typesize, and calculation simplifies it to half the `\topskip`.

```
206 \setlength\maxdepth{0.5\topskip}
```

### 4.3.2 Text dimensions

Here we implement the proposals and calculations by Tschichold. In particular, the proportions of text to page are set so that 49% of the page area is covered by the text block.

`\topmargin` The `\topmargin` is the distance between the top of the text block and the top of the box containing the running head. The running head is not part of the text block, and it goes into the top margin. As `\topmargin` is measured from the top of the box, we have to correct for header height and header separation.

```
207 \setlength{\topmargin}{0.1\paperheight}
208     \addtolength{\topmargin}{-\headheight}
209     \addtolength{\topmargin}{-\headsep}
```

`\textwidth` Define the width of the text block to 0.7 of the page width, and make calculations a little easier by adjusting the calculated width to a whole number of points.

```
210 \setlength{\textwidth}{0.7\paperwidth}
211     \@settopoint\textwidth
```

`\textheight` The height of the text block itself is set to 0.7 times the page height. This amount is then adjusted to ensure that a whole number of lines makes up the text block, and does so exactly.

```
212 \setlength\@tempdima{0.7\paperheight}
```

take away the first line, which is a bit shorter than the `\baselineskip`,

```
213     \addtolength\@tempdima{-\topskip}
```

this length may be very close, but just a little too small to accommodate one more line, so we add a small amount,

```
214     \addtolength\@tempdima{5\p@}
```

and calculate the number of lines in this length,

```
215     \divide\@tempdima\baselineskip
```

```
216     \@tempcnta=\@tempdima
```

The correct `\textheight` comes to the number of lines just calculated, multiplied by the height of text lines, `\baselineskip`, and with the addition of the `\topskip` we took away initially.

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

```
218     \addtolength\textheight{\topskip}
```

### 4.3.3 Margin dimensions

Now that we have set the size of the text block, the amount of space available for margins is set as well. The remaining white space is divided in a 1:2 ratio, hence the proportions between margins and text become 1:7:2.

`\evensidemargin` Since we are typesetting books, both even and odd side margins have to be set.

`\oddsidemargin` 219 `\setlength{\evensidemargin}{0.2\paperwidth}`  
 220 `\setlength{\oddsidemargin}{0.1\paperwidth}`

`\marginparsep` The horizontal space between the text block and marginal notes is set by

`\marginparpush` `\marginparsep`, while the minimal vertical separation between successive marginal notes is controlled by `\marginparpush`.

221 `\if@twocolumn`  
 222 `\setlength\marginparsep{10\p@}`  
 223 `\else`  
 224 `\langle 10pt \rangle \setlength\marginparsep{10\p@}`  
 225 `\langle 11pt \rangle \setlength\marginparsep{11\p@}`  
 226 `\langle 12pt \rangle \setlength\marginparsep{12\p@}`  
 227 `\fi`  
 228 `\langle 10pt | 11pt \rangle \setlength\marginparpush{5\p@}`  
 229 `\langle 12pt \rangle \setlength\marginparpush{7\p@}`

`\marginparwidth` The amount of space available for marginal notes is the outer margin. From this we have to subtract the separation between notes and the text body, as well as another, marginal outer margin. In principle, marginal text could run up all the way to the outer margin, but this is both ugly and impractical for the book binder. Since the margins have to be cut, it is very wise to leave sufficient space: 30pt seems a good minimum.

230 `\setlength\@tempdima{\paperwidth}`  
 231 `\addtolength\@tempdima{-\textwidth}`  
 232 `\addtolength\@tempdima{-\oddsidemargin}`  
 233 `\addtolength\@tempdima{-\marginparsep}`  
 234 `\addtolength\@tempdima{-30\p@}`  
 235 `\setlength\marginparwidth{\@tempdima}`  
 236 `\@settopoint\marginparwidth`

### 4.3.4 Footnotes

`\footnotesep` The height of the strut placed at the beginning of every footnote. For each size, `\footnotesep` is set to the height of the corresponding `\footnotesize` strut, so there is no extra space between successive footnotes.

237 `\langle 10pt \rangle \setlength\footnotesep{6.65\p@}`  
 238 `\langle 11pt \rangle \setlength\footnotesep{7.7\p@}`  
 239 `\langle 12pt \rangle \setlength\footnotesep{8.4\p@}`

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

240 `\langle 10pt \rangle \setlength{\skip\footins}{9\p@ \@plus 4\p@ \@minus 2\p@}`  
 241 `\langle 11pt \rangle \setlength{\skip\footins}{10\p@ \@plus 4\p@ \@minus 2\p@}`  
 242 `\langle 12pt \rangle \setlength{\skip\footins}{11\p@ \@plus 4\p@ \@minus 2\p@}`  
 243 `\langle /10pt | 11pt | 12pt \rangle`

### 4.3.5 Float placement

Float placement parameters are given values by L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>, therefore any parameters which are not counters have to be (re)set with `\renewcommand`.

<code>topnumber</code>	The maximum number of floats allowed at the top of a page. 244 <code>\*octavo</code> 245 <code>\setcounter{topnumber}{2}</code>
<code>\topfraction</code>	The maximum part of a text page that may be occupied by floats at the top of a page. 246 <code>\renewcommand\topfraction{.7}</code>
<code>bottomnumber</code>	The equivalent of <code>topnumber</code> , but for the maximum number of floats allowed at the bottom of a page. 247 <code>\setcounter{bottomnumber}{1}</code>
<code>\bottomfraction</code>	The maximum part of a text page that may be occupied by floats at the bottom of a page. 248 <code>\renewcommand\bottomfraction{.5}</code>
<code>totalnumber</code>	The maximum number of floats in total which may appear on a page. 249 <code>\setcounter{totalnumber}{3}</code>
<code>\textfraction</code>	The minimum part of a text page which has to be occupied by text. 250 <code>\renewcommand\textfraction{.2}</code>
<code>\floatpagefraction</code>	The minimum part of a page which has to be occupied by floating object before a ‘float page’ is produced. 251 <code>\renewcommand\floatpagefraction{.5}</code>
<code>dbltopnumber</code>	The maximum number of two-column floats which may appear at the top of a two-column text page. 252 <code>\setcounter{dbltopnumber}{2}</code>
<code>\dbltopfraction</code>	The maximum part of a two-column text page which may be occupied at the top by two-column floats. 253 <code>\renewcommand\dbltopfraction{.7}</code>
<code>\dblfloatpagefraction</code>	The minimum part of a page which has to be occupied by two-column floating objects before a ‘float page’ is produced. 254 <code>\renewcommand\dblfloatpagefraction{.5}</code> 255 <code>\*octavo</code>
<code>\floatsep</code> <code>\textfloatsep</code> <code>\intextsep</code>	These parameters control the separation between a float object and the other objects on the same page. They are used in both single-column and double-column mode.  <code>\floatsep</code> is the amount of space between adjacent floats moved to the top or to the bottom of the text page.  <code>\textfloatsep</code> is the amount of space between the main text and the floats at the top or at the bottom of the text page.

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

```

256 (*10pt)
257 \setlength\floatsep{12\p@ \@plus 2\p@ \@minus 2\p@}
258 \setlength\textfloatsep{24\p@ \@plus 2\p@ \@minus 4\p@}
259 \setlength\intextsep{12\p@ \@plus 2\p@ \@minus 2\p@}
260 (/10pt)
261 (*11pt)
262 \setlength\floatsep{13.2\p@ \@plus 2\p@ \@minus 2\p@}
263 \setlength\textfloatsep{26\p@ \@plus 2\p@ \@minus 4\p@}
264 \setlength\intextsep{13.2\p@ \@plus 2\p@ \@minus 2\p@}
265 (/11pt)
266 (*12pt)
267 \setlength\floatsep{14.4\p@ \@plus 2\p@ \@minus 2\p@}
268 \setlength\textfloatsep{28\p@ \@plus 2\p@ \@minus 4\p@}
269 \setlength\intextsep{14.4\p@ \@plus 4\p@ \@minus 2\p@}
270 (/12pt)

```

`\dblfloatsep` In two-column mode, floats spanning the entire `\textwidth` are placed with the  
`\dbltextfloatsep` parameters `\dblfloatsep` and `\dbltextfloatsep` in control.

`\dblfloatsep` is the amount of space between adjacent floats at the top or at the bottom of the text page

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

```

271 (*10pt)
272 \setlength\dblfloatsep{12\p@ \@plus 2\p@ \@minus 2\p@}
273 \setlength\dbltextfloatsep{24\p@ \@plus 2\p@ \@minus 4\p@}
274 (/10pt)
275 (*11pt)
276 \setlength\dblfloatsep{12\p@ \@plus 2\p@ \@minus 2\p@}
277 \setlength\dbltextfloatsep{26\p@ \@plus 2\p@ \@minus 4\p@}
278 (/11pt)
279 (*12pt)
280 \setlength\dblfloatsep{14\p@ \@plus 2\p@ \@minus 2\p@}
281 \setlength\dbltextfloatsep{28\p@ \@plus 2\p@ \@minus 4\p@}
282 (/12pt)

```

`\@fptop` The layout of floating objects on separate, ‘float’ pages is controlled by the pa-  
`\@fpsep` rameters `\@fptop`, `\@fpsep` and `\@fpbot`. These parameters insert stretchable  
`\@fpbot` amounts of white space between the top and the bottom of the page and the floats, while `\@fpsep` inserts white space between adjacent floats. In order to ensure that any remaining space is filled, at least one of `\@fptop` or `\@fpbot` has to contain stretchable space.

```

283 (*10pt)
284 \setlength\@fptop{0\p@ \@plus 1fil}
285 \setlength\@fpsep{10\p@ \@plus 2fil}
286 \setlength\@fpbot{0\p@ \@plus 1fil}
287 (/10pt)
288 (*11pt)
289 \setlength\@fptop{0\p@ \@plus 1fil}
290 \setlength\@fpsep{11\p@ \@plus 2fil}
291 \setlength\@fpbot{0\p@ \@plus 1fil}

```



```

292 </11pt>
293 <*12pt>
294 \setlength\@fptop{0\p@ \@plus 1fil}
295 \setlength\@fpsep{12\p@ \@plus 2fil}
296 \setlength\@fpbot{0\p@ \@plus 1fil}
297 </12pt>

\@dblftop The layout of floating objects in double-column mode on separate ‘float’ pages is
\@dblfpsep handled in a similar way to that used for single-column mode pages just described.
\@dblfpbot 298 <*10pt>
299 \setlength\@dblftop{0\p@ \@plus 1fil}
300 \setlength\@dblfpsep{10\p@ \@plus 2fil}
301 \setlength\@dblfpbot{0\p@ \@plus 1fil}
302 </10pt>
303 <*11pt>
304 \setlength\@dblftop{0\p@ \@plus 1fil}
305 \setlength\@dblfpsep{11\p@ \@plus 2fil}
306 \setlength\@dblfpbot{0\p@ \@plus 1fil}
307 </11pt>
308 <*12pt>
309 \setlength\@dblftop{0\p@ \@plus 1fil}
310 \setlength\@dblfpsep{12\p@ \@plus 2fil}
311 \setlength\@dblfpbot{0\p@ \@plus 1fil}
312 </12pt>

```

## 4.4 Page styles

A number of page styles are defined by the class file. Of course, more elaborate styles can be designed using the `fancyheadings` package, but once again, here we opt for the more restrained and discrete means of guiding the reader. The page styles included in the standard  $\text{\LaTeX}$  book class are either `headings`, `plain`, `empty` or `myheadings`. The standard `headings` is here modified a little, so that the folios now appear centered at the foot of the page.

The page style *style* is defined by defining the command `\ps@style`. Definitions should be local only, and any stray spaces in definitions had better be avoided as they may turn up unexpectedly in the output.

```

\@evenhead The \ps@style commands define the macros \@evenhead, \@evenfoot and their
\@oddhead odd-page equivalents to define the running head and foot. The contents are pro-
\@evenfoot duced inside a \hbox of width \textwidth.
\@oddfoot

```

### 4.4.1 Marking conventions

The making of headings reflecting the sectioning commands, a page style has to define commands such as `\chaptermark`, `\sectionmark` and so on so that e.g., `\chapter` calls `\chaptermark{TEXT}`. These commands are all defined with the help of the following macros, which have extended in  $\text{\LaTeX}$  a bit as compared to the original  $\text{\TeX}$  `\mark` facility. So we now have at our disposal:

`\markboth{Left}{Right}` Adds both marks

`\markright{Right}` Adds a righthand-side mark

`\leftmark` used in the `\SIDEhead` and `\SIDEfoot` macros, getting the current lefthand-side mark

`\rightmark` also used in the `\SIDEhead` and `\SIDEfoot` macros, but getting the current righthand-side mark.

#### 4.4.2 The headings and myheadings styles

The two other styles, `plain` and `empty` are defined by L<sup>A</sup>T<sub>E</sub>X.

`\ps@headings` To begin with, the folios are centered at the foot of the pages, while the running heads contain one of the marks. The contents of the running head is determined both by the chapter and the section titles, hence we let `\@mkboth` be `\markboth`.

```
313 \*octavo
314 \def\ps@headings{%
315     \def\@oddfoot{\hfil\thepage\hfil}%
316     \def\@evenfoot{\hfil\thepage\hfil}%
317     \def\@evenhead{\hfil{\scshape\leftmark}\hfil}%
318     \def\@oddhead{\hfil{\scshape\rightmark}\hfil}%
319     \let\@mkboth\markboth
```

`\chaptermark` As we use `\chaptermark` and `\sectionmark` to fill the running heads we define these here. The use of `##1` as parameter of the commands, and these will be defined when `\ps@headings` is executed.

`\sectionmark`

```
320 \def\chaptermark##1{%
321     \markboth {%
322         \ifnum \c@secnumdepth >\m@ne
323             \if@mainmatter
324                 \@chapapp\ thechapter. \ %
325             \fi
326         \fi
327     }##1}{}}%
328 \def\sectionmark##1{%
329     \markright {%
330         \ifnum \c@secnumdepth >\z@
331             \thesection. \ %
332         \fi
333     }##1}}}
```

`\ps@myheadings` The definition of this page style is straightforward, as it is up to the user to determine what goes into the running heads by setting the `\markboth` and `\markright` commands in the text. We do, however, maintain the presence of the centered folios at the foot of the pages. To ensure that the marking commands used by chapter and section commands are not used, we set them to a macro gobbling up the arguments.

```
334 \def\ps@myheadings{%
335     \def\@oddfoot{\hfil\thepage\hfil}%
336     \def\@evenfoot{\hfil\thepage\hfil}%
337     \def\@evenhead{\hfil{\scshape\leftmark}\hfil}%
338     \def\@oddhead{\hfil{\scshape\rightmark}\hfil}%
339     \let\@mkboth\@gobbletwo
340     \let\chaptermark\@gobble
341     \let\sectionmark\@gobble}
```

## 5 Document markup

### 5.1 The title

`\maketitle` The three macros `\title`, `\author` and `\date` are part of L<sup>A</sup>T<sub>E</sub>X, and that info is therefore stored away in internal control sequences. The `\date` macro gets today's date by default, so it may have to be set explicitly to a particular date by the user.

If there is a request for a title, we will generate a simple, separate titlepage. First, a local redefinition of `\footnotesize` and `footnoterule`, so that we get a different appearance of the `\thanks` command. These changes affect all footnotes, but since there really is no need for true footnotes on a title page, this is hardly a problem.

```
342 \if@titlepage
343   \newcommand\maketitle{\begin{titlepage}%
344     \let\footnotesize\small
345     \let\footnoterule\relax
346     \let\footnote\thanks
```

The title itself is centered vertically, with a little offset brought by a `\vskip`.

```
347   \null\vfil
348   \vskip 60\p@
```

Now we can set the title, using the `\LARGE` fount, followed by a little bit of vertical space, and then the authors, set in `\large` fount. To ensure everything lines up nicely in a single column, we resort to the tabular environment. A little vertical space is then followed by the date.

```
349   \begin{center}%
350     {\LARGE \@title \par}%
351     \vskip 3em%
352     {\large
353       \lineskip .75em%
354       \begin{tabular}[t]{c}%
355         \@author
356       \end{tabular}\par}%
357     \vskip 1.5em%
358     {\large \@date \par}%
359   \end{center}\par
```

Calling `\@thanks` prints the information to go into the footnote, and completes the printing on the page.

```
360   \@thanks
361   \vfil\null
362   \end{titlepage}%
```

Having finished the printing of the title page, we now clean up.

```
363   \setcounter{footnote}{0}%
364   \let\thanks\relax
365   \let\maketitle\relax
366   \gdef\@thanks{}
367   \gdef\@author{}
368   \gdef\@title{}
369 \fi
```

## 5.2 Chapters and section

The coming definitions rely on two internal macros, `\@startsection` and `\secdef`. Their syntax is simple but a bit involved. The `\@startsection` macro has 7 required arguments and 2 optional ones. The full set looks like this:

```
\@startsection<name><level><indent><beforeskip> <afterskip><style>
[*] [(<altheading>)] <heading>
```

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

*<level>* A number denoting the depth of the section, chapter=1, section=2, etc. A section number will be printed only if *<level>* is equal or smaller than the value of *secnumdepth*.

*<indent>* The indentation of the heading from the left margin

*<beforeskip>* The absolute value of this argument is the skip to leave above the heading. If it is negative, then the paragraph indent of the text following the heading is suppressed.

*<afterskip>* If positive, it is the skip to leave below the heading, else it is the skip to the right of a run-in heading.

*<style>* Sets the style of the heading.

\* When this is missing the heading is numbered and the corresponding counter is incremented.

*<altheading>* Gives an alternative heading to use in the table of contents and in the running heads. This should be present when the \* form is used.

*<heading>* The heading of the new section.

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

```
\secdef<unstarcmds><starcmds>
```

*<unstarcmds>* Used for the normal form of the sectioning command.

*<starcmds>* Used for the \*-form of the sectioning command.

### 5.2.1 Definition of counters

`\c@secnumdepth` The *secnumdepth* counter indicates to what level sectioning commands have to make and include section numbers. More often than not, such numbers are of little real value to the reader of a book: therefore, it is herein set to a value that eliminates such number. The user can always bring them back in by resetting *secnumdepth* in the preamble of the document.

```
370 \setcounter{secnumdepth}{-2}
```

<code>\c@part</code>	Nevertheless, all counters will be defined in case they are wanted. The particular construction <code>\newcounter{&lt;meta&gt;{&lt;newcount&gt;}}[&lt;meta&gt;{&lt;oldcount&gt;}]</code> defined
<code>\c@chapter</code>	<i>&lt;newcount&gt;</i> as a counter which is reset every time <i>&lt;oldcount&gt;</i> is stepped. Of course,
<code>\c@section</code>	the counter <i>&lt;oldcount&gt;</i> must have been defined.
<code>\c@subsection</code>	
<code>\c@subsubsection</code>	
<code>\c@paragraph</code>	
<code>\c@subparagraph</code>	

```

371 \newcounter{part}
372 \newcounter{chapter}
373 \newcounter{section}[chapter]
374 \newcounter{subsection}[section]
375 \newcounter{subsubsection}[subsection]
376 \newcounter{paragraph}[subsubsection]
377 \newcounter{subparagraph}[paragraph]

\thepart For each counter, the \theCTR macro defines how the counter CTR is printed.
\thechapter 378 \renewcommand\thepart{\@Roman\c@part}
\thesection 379 \renewcommand\thechapter{\@arabic\c@chapter}
\thesubsection 380 \renewcommand\thesection{\thechapter.\@arabic\c@section}
\thesubsubsection 381 \renewcommand\thesubsection{\thesection.\@arabic\c@subsection}
\theparagraph 382 \renewcommand\thesubsubsection{\thesubsection.\@arabic\c@subsubsection}
\thesubparagraph 383 \renewcommand\theparagraph{\thesubsubsection.\@arabic\c@paragraph}
384 \renewcommand\thesubparagraph{\theparagraph.\@arabic\c@subparagraph}

\@chapapp The macro \@chapapp is defined to be \chaptername. But when the \appendix
command is issued, it is changed to \appendixname
385 \newcommand\@chapapp{\chaptername}

```

### 5.2.2 Front, main and back matter

Books are traditionally made up from three logical sections, known as front matter, main matter and back matter. The actual *text* of a book is the main matter. The separation into different ‘matters’ allows slightly different ways of typesetting these sections.

```

\frontmatter Front matter is typeset with roman page numbering and without numbers given
to any of its chapters.
386 \newcommand\frontmatter{\cleardoublepage
387     \@mainmatterfalse\pagenumbering{roman}}

\mainmatter In main matter, we first clear the page, start page numbering with arabic numerals
and switch on chapter numbering (which will be visible only if secnumdepth has
been set to a sufficiently high value, of course.
388 \newcommand\mainmatter{\cleardoublepage
389     \@mainmattertrue\pagenumbering{arabic}}

\backmatter Back matter clears the page, turns off any chapter numbering, but leaves the page
numbering unchanged.
390 \newcommand\backmatter{\cleardoublepage
391     \@mainmatterfalse}

```

### 5.2.3 Parts

The text of books may well be divided into a number of parts by the writer(s). Here are the macros to deal with these major divisions of a text, should they be required.

```

\part A part is shown in the book by beginning on a new right-hand page and we use a
plain pagestyle.
392 \newcommand\part{\cleardoublepage
393     \thispagestyle{plain}}%

```

In case we are dealing with a double-column text, the page will be in a single column but we have to remember we switch back to double-column mode.

```

394 \if@twocolumn
395     \onecolumn
396     \@tempwattrue
397 \else
398     \@tempwafalse
399 \fi

```

To ensure that the fill glue does not disappear, we need an empty box.

```
400 \hbox{}\vfil
```

To make the actual heading, we issue a `\secdef` here.

```
401 \secdef\@part\@spart}
```

`\@part` This is the macro which carries out the actual formatting of the title of the part. If *secnumdepth* equals -2 or larger, a number is given to the part.

```

402 \def\@part[#1]#2{%
403     \ifnum \c@secnumdepth >-2\relax
404         \refstepcounter{part}%
405         \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
406     \else
407         \addcontentsline{toc}{part}{#1}%
408     \fi

```

Empty the mark registers, center the title on the page, and prevent any breaking between lines, and finally reset the fount.

```

409     \markboth{}{}%
410     {\centering
411     \interlinepenalty \@M
412     \normalfont

```

If we are dealing with a numbered part, print the number.

```

413     \ifnum \c@secnumdepth >-2\relax
414         \huge \partname~\thepart
415         \par

```

Leave some space before printing the title, and clear up with `\@endpart`.

```

416         \vskip 20\p@
417     \fi
418     \Huge #2\par}%
419     \@endpart}

```

`\@spart` A macro to deal with a part\*. If this sectioning command was used, we do not print a number, but otherwise the format is the same as for the unstarred part.

```

420 \def\@spart#1{%
421     {\centering
422     \interlinepenalty \@M
423     \normalfont
424     \Huge #1\par}%
425     \@endpart}

```

`\@endpart` A macro to finishes the part page. First, fill the current page,

```

426 \def\@endpart{%
427     \vfil\newpage

```

and, as we are dealing with a two-sided document, insert a blank page.

```
428 \hbox{}%
429 \thispagestyle{empty}%
430 \newpage
```

If we are dealing with a two-column document, switch back to double column mode.

```
431 \if@tempswa
432 \twocolumn
433 \fi}
```

### 5.2.4 Chapters

`\chapter` Chapters start on a new page, and the page style is set to *plain*

```
434 \newcommand\chapter{%
435 \if@openright\cleardoublepage
436 \else\clearpage
437 \fi
438 \thispagestyle{plain}%
```

Ensure that no float appears on the top of this page: it really would not do to have something like a float appearing above a chapter title.

```
439 \global\@topnum\z@
```

Also, suppress any indentation of the first paragraph and issue the `\secdef` which will take care of making the chapter title.

```
440 \@afterindentfalse
441 \secdef\@chapter\@schapter}
```

`\@chapter` The macro for numbered chapters. If *secnumdepth* is larger than -1, and `\@mainmatter` is true, display the chapter number. Write a message to the terminal to let the user know that a new chapter is about to be typeset.

```
442 \def\@chapter[#1]#2{%
443 \ifnum \c@secnumdepth >\m@ne
444 \if@mainmatter
445 \refstepcounter{chapter}%
446 \typeout{\@chapapp\space\thechapter.}%
447 \addcontentsline{toc}{chapter}%
448 {\protect\numberline{\thechapter}#1}%
449 \else
450 \addcontentsline{toc}{chapter}{#1}%
451 \fi
452 \else
453 \addcontentsline{toc}{chapter}{#1}%
454 \fi
```

Having written the entry to the table of contents, store the (alternative) title of the chapter with `\chaptermark`, add some white space to the lists of figures and tables

```
455 \chaptermark{#1}%
456 \addtocontents{lof}{\protect\addvspace{10\p@}}%
457 \addtocontents{lot}{\protect\addvspace{10\p@}}%
```

Now call `\@makechapterhead` to typeset the actual chapter title. When dealing with text in double columns, this requires a bit of attention as we want the title to use the whole of the width of the text. The `\@afterheading` macro suppresses the indentation.

```

458   \if@twocolumn
459     \topnewpage[\@makechapterhead{#2}]%
460   \else
461     \@makechapterhead{#2}%
462     \@afterheading
463   \fi}

```

**\@makechapterhead** The macro to typeset the actual chapter heading. Before we begin, bear in mind that we have to retain the grid, so the total amount of space occupied by the heading should be a whole number of lines. So, first some white space. Then open a group with no indenting, with ragged right text, and reset the font.

```

464 \def\@makechapterhead#1{%
465   \vspace*{4\baselineskip}%
466   {\parindent \z@ \raggedright \normalfont

```

Find out if the chapter number has to be printed, and, if that is the case, leave some white space between number and title.

```

467   \ifnum \c@secnumdepth >\m@ne
468     \if@mainmatter
469       \Large \@chapapp{} \thechapter
470       \par\nobreak
471       \ifnum \@ptsize < 1
472         \vskip 26\p@
473       \else
474         \vskip 22\p@
475       \fi
476     \fi
477   \fi

```

Set the title in Large font, prevent a pagebreak from happening in the middle or after the title, and leave a bit of white space before the text.

```

478   \interlinepenalty\@M
479   \Large #1\par\nobreak
480   \vskip 34\p@ \@minus 4\p@
481   }}

```

**\@schapter** The macro which takes care of unnumbered chapters, which is much more straightforward, as it only has to typeset the chapter title.

```

482 \def\@schapter#1{%
483   \if@twocolumn
484     \topnewpage[\@makeschapterhead{#1}]%
485   \else
486     \@makeschapterhead{#1}%
487     \@afterheading
488   \fi}

```

**\@makeschapterhead** The equivalent macro of `\@makechapterhead` but for typesetting the heading of an unnumbered chapter.

```

489 \def\@makeschapterhead#1{%

```



```

490 \vspace*{4\baselineskip}%
491 {\parindent \z@ \raggedright
492 \normalfont
493 \interlinepenalty\@M
494 \Large #1\par\nobreak
495 \ifnum \@ptsize < 1
496 \vskip 30\p@
497 \else
498 \vskip 34.8\p@
499 \fi
500 }}

```

### 5.2.5 Lower-level headings

`\section` All the lower-level headings are generated by using the `\@startsection`. Because  
`\subsection` these various headings can occur anywhere on a page, some care has gone into  
`\subsubsection` defining the vertical spacing surrounding the headings. The reason for the dif-  
`\paragraph` ferent approach, compared to the standard book class, is that we are trying our  
`\subparagraph` utmost to maintain the regularity of the printing grid of text lines. Through the  
use of `\baselineskip` multiples and fractions, the space given to headings and  
surrounding white space will always be a multiple of print lines. We take the  
opportunity here of toning down the ‘loudness’ of some of the headings compared  
to the standard book class.

```

501 \newcommand\section{%
502   \@startsection{section}%
503   {1}%
504   {0em}%
505   {-1.166\baselineskip \@minus -2\p@}%
506   {0.835\baselineskip \@minus 2\p@}%
507   {\centering\large\normalfont\scshape}}
508 \newcommand\subsection{%
509   \@startsection{subsection}%
510   {2}%
511   {1em}%
512   {-0.666\baselineskip \@minus -2\p@}%
513   {0.333\baselineskip \@minus 2\p@}%
514   {\normalfont\normalsize\scshape}}
515 \newcommand\subsubsection{%
516   \@startsection{subsubsection}%
517   {3}%
518   {1em}%
519   {-0.666\baselineskip \@minus -2\p@}%
520   {0.333\baselineskip \@minus 2\p@}%
521   {\normalfont\normalsize\itshape}}
522 \newcommand\paragraph{%
523   \@startsection{paragraph}%
524   {4}%
525   {\z@}%
526   {0\p@}%
527   {-1em}%
528   {\normalfont\normalsize}}
529 \newcommand\subparagraph{%
530   \@startsection{subparagraph}%

```

```

531    {5}%
532    {\parindent}%
533    {0\p@}%
534    {-1em}%
535    {\normalfont\normalsize}}

```

## 5.3 Lists

### 5.3.1 List parameters

The type setting of the list environment is governed by a considerable number of parameters, including `\leftmargin`, `\rightmargin`, `\listparindent`, `\itemindent`, `\labelwidth`, `\labelsep` and `\partopsep`. By default, `\rightmargin`, `\listparindent` and `itemindent` are set to 0pt. For a list at level *L*, the command `\@listL` is called, with *L* denoting ‘i’, ‘ii’, ... ‘vi’. By convention, `@listL` should set `\leftmargin` to `\leftmarginL`.

```

\listmargin In double-column mode, margins are set somewhat smaller.
\listmargini 536 \if@twocolumn
\listmarginii 537 \setlength\leftmargini{2em}
\listmarginiii 538 \else
\listmarginiv 539 \setlength\leftmargini{2.5em}
\listmarginv 540 \fi
\listmarginvi 541 \setlength\leftmargin{\leftmargini}

```

The next three levels are set so that they are larger than the sum of `\labelsep` and the width of the default labels, being ‘(m)’, ‘vii.’ and ‘M.’

```

542 \setlength\leftmarginii{2.2em}
543 \setlength\leftmarginiii{1.87em}
544 \setlength\leftmarginiv{1.7em}
545 \if@twocolumn
546 \setlength\leftmarginv{.5em}
547 \setlength\leftmarginvi{.5em}
548 \else
549 \setlength\leftmarginv{1em}
550 \setlength\leftmarginvi{1em}
551 \fi

```

```

\labelsep The distance between label and text of an item is set by \labelsep, while the
\labelwidth width of the label is set by \labelwidth

```

```

552 \setlength\labelsep{.5em}
553 \setlength\labelwidth{\leftmargini}
554 \addtolength\labelwidth{-\labelsep}

```

```

\partopsep When a blank line is left before the list environment, we add an extra vertical
space of \partopsep, in addition to \parskip and \topsep. Bearing in mind our
attempts to maintain the regularity of the line grid, the exact value of this space
has to be set with some care.

```

```

555 </octavo>
556 <*10pt | 11pt | 12pt>
557 \setlength\partopsep{0.5\baselineskip \@plus 2\p@ \@minus 2\p@}
558 </10pt | 11pt | 12pt>

```

```

\@beginparpenalty These are the penalties incurred before and after a list or paragraph environment,
\@endparpenalty   and set to a bonus value to encourage a page break at these points.
\@itempenalty     559 \*octavo>
                  560 \@beginparpenalty -\@lowpenalty
                  561 \@endparpenalty -\@lowpenalty
                  562 \@itempenalty -\@lowpenalty
                  563 \</octavo>

\@listi The macro \@listi sets the values of \leftmargin, \parsep, \topsep, \itemsep,
\@listI and so forth, for the lists that appear at top level. These definitions are usually
modified in function of the font-size in use. As we wish to maintain the printing
grid, these adjustments are not used here. Just to be on the safe side though, we
define listI to hold a copy of listi so that \normalsize can switch all parameters
back.

564 \*10pt|11pt|12pt>
565 \def\@listi{%
566     \leftmargin\leftmarginI

567 \*10pt>
568     \parsep \z@
569     \topsep 6\p@ \@plus 2\p@ \@minus 2\p@
570     \itemsep \z@}
571 \</10pt>

572 \*11pt>
573     \parsep \z@
574     \topsep 6.6\p@ \@plus 2\p@ \@minus 2\p@
575     \itemsep \z@}
576 \</11pt>

577 \*12pt>
578     \parsep \z@
579     \topsep 7.2\p@ \@plus 3\p@ \@minus 3\p@
580     \itemsep \z@}
581 \</12pt>

582 \let\@listI\@listi
583 \@listi

\@listii Lists at higher levels need their own macros, and here they are. These do not have
\@listiii a saved version, and are not modified by font size commands.

\@listiv 584 \def\@listii{%
\@listv 585     \leftmargin\leftmarginii
\@listvi 586     \labelwidth\leftmarginii
587     \advance\labelwidth-\labelsep
588     \itemsep \parsep}
589 \def\@listiii{%
590     \leftmargin\leftmarginiii
591     \labelwidth\leftmarginiii
592     \advance\labelwidth-\labelsep
593     \itemsep \parsep}
594 \def\@listiv{%
595     \leftmargin\leftmarginiv
596     \labelwidth\leftmarginiv
597     \advance\labelwidth-\labelsep

```

```

598 \itemsep \parsep}
599 \def\@listv{%
600 \leftmargin\leftmarginv
601 \labelwidth\leftmarginv
602 \advance\labelwidth-\labelsep
603 \itemsep \parsep}
604 \def\@listvi{%
605 \leftmargin\leftmarginvi
606 \labelwidth\leftmarginvi
607 \advance\labelwidth-\labelsep
608 \itemsep \parsep}
609 </10pt|11pt|12pt>

```

### 5.3.2 Enumerate

`\theenumi` The *enumerate* environment uses four counters, where *enumN* controls the numbering at level N of the enumeration. These counters have been defined by L<sup>A</sup>T<sub>E</sub>X, but we use a different representation here.

`\theenumii`

`\theenumiii`

`\theenumiv` 610 `{*octavo}`

611 `\renewcommand\theenumi{\@arabic\c@enumi}`

612 `\renewcommand\theenumii{\@alph\c@enumii}`

613 `\renewcommand\theenumiii{\@roman\c@enumiii}`

614 `\renewcommand\theenumiv{\@Alph\c@enumiv}`

`\labelenumi` The label for each item in generated by these commands

`\labelenumii` 615 `\newcommand\labelenumi{\theenumi.}`

`\labelenumiii` 616 `\newcommand\labelenumii{(\theenumii)}`

`\labelenumiv` 617 `\newcommand\labelenumiii{\theenumiii.}`

618 `\newcommand\labelenumiv{\theenumiv.}`

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

`\p@enumiii`

`\p@enumiv` 619 `\renewcommand\p@enumii{\theenumi}`

620 `\renewcommand\p@enumiii{\theenumi(\theenumii)}`

621 `\renewcommand\p@enumiv{\p@enumiii\theenumiii}`

### 5.3.3 Itemise

`\labelitemi` Itemising is governed by four commands, `\labelitemN`, defining the labels at the various levels.

`\labelitemii`

`\labelitemiii` 622 `\newcommand\labelitemi{\normalfont\bfseries --}`

`\labelitemiv` 623 `\newcommand\labelitemii{\m@th\cdot}`

624 `\newcommand\labelitemiii{\m@th\bullet}`

625 `\newcommand\labelitemiv{\m@th\ast}`

### 5.3.4 Description

`description` The *itemize* and *enumerate* environments are defined by L<sup>A</sup>T<sub>E</sub>X, but we have to define the description environment here.

626 `\newenvironment{description}`

627 `{\list{}}%`

628 `{\labelwidth\z@%`

629 `\itemindent-\leftmargin%`

```

630 \let\makelabel\descriptionlabel}}%
631 {\endlist}

```

`\descriptionlabel` To change the format of the label, redefine `\descriptionlabel`

```

632 \newcommand*\descriptionlabel[1]{%
633 \hspace\labelsep%
634 \normalfont\bfseries #1}

```

## 5.4 Defining new environments

A number of separate environments for specific bodies of text can be specified at class level. In the context of writing and making books, here are a number of such environments.

### 5.4.1 Verse

**verse** A simple way of implementing the requirements of typesetting verse. The environment is defined by tweaking some parameters of the list environment. Each line of verse ends with a `\\`, and the macro takes this to be `\@centercr`.

```

635 \newenvironment{verse}
636 {\let\\=\@centercr
637 \list{}\itemsep \z@
638 \itemindent -1.5em%
639 \listparindent\itemindent
640 \rightmargin \leftmargin
641 \advance\leftmargin 1.5em}%
642 \item[]}
643 {\endlist}

```

### 5.4.2 Quotation

**quotation** Also an implementation based on the list environment. Lines of the quotation are set more narrowly.

```

644 \newenvironment{quotation}
645 {\list{}\listparindent 1.5em%
646 \itemindent \listparindent
647 \rightmargin \leftmargin
648 \parsep \z@}%
649 \item[]}
650 {\endlist}

```

### 5.4.3 Quote

**quote** For short quotations, the text is set as a quotation but without the narrowing of the line length.

```

651 \newenvironment{quote}
652 {\list{}%
653 {\rightmargin\leftmargin}%
654 \item[]}
655 {\endlist}

```

#### 5.4.4 Titlepage

**titlepage** A titlepage starts and ends a page, and omits the folio. If the text is typeset in double columns, this page alone reverts to single column mode.

```
656 \newenvironment{titlepage}{%
657     \cleardoublepage
658     \if@twocolumn
659         \@restonecoltrue\onecolumn
660     \else
661         \@restonecolfalse\newpage
662     \fi
663     \thispagestyle{empty}}%
664     {\if@restonecol
665         \twocolumn
666     \else
667         \newpage
668     \fi}
```

#### 5.4.5 Appendix

**\appendix** Typesetting appendices is somewhat different from the other chapters. We have to reset section and subsection counters and redefine the macro **\thesection** so that it produces an alphabetic enumeration of the appendices. We also set **\@chapapp** to **\appendixname**.

```
669 \newcommand\appendix{\par
670     \setcounter{chapter}{0}%
671     \setcounter{section}{0}%
672     \renewcommand\@chapapp{\appendixname}%
673     \renewcommand\thechapter{\@Alph{c}chapter}}
```

### 5.5 Setting environment parameters

**\arraycolsep** Columns in an **array** environment are separated by 2 **\arraycolsep**

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

**\tabcolsep** Columns in a **tabular** environment are separated by 2 **\tabcolsep**

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

**\arrayrulewidth** The width of the lines in **array** and **tabular** environments

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

**\doublerulesep** When rules are adjacent in **array** and **tabular** environments, this is how far apart they are drawn

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

**\tabbingsep** Defines the space requested by the **\'** command in the text

```
678 \setlength\tabbingsep{\labelsep}
```

**\@mpfootins** The equivalent of **\skip\footins** for footnotes appearing in minipages.

```
679 \skip\@mpfootins = \skip\footins
```

`\fboxsep` When using `\fbox` or `\framebox`, these macros define the amount of space left between box and text, and the width of the rules in the box.

```

680 \setlength\fboxsep{3\p@}
681 \setlength\fboxrule{.4\p@}

```

`\theequation` Contrary to what is often seen in mathematical texts, here we number equations continuously throughout the text. Cross-referencing is just so easy in TeX that there is not really any need for a different numbering system.

```

682 \renewcommand\theequation{\@arabic\c@equation}

```

## 5.6 Floating objects

Floats cover a variety of object types. Here, we deal with two different types: figure, and table. The typesetting of floats is of course governed by a number of specific macros.

`\fps@TYPE` The default placement of TYPE floats

`\ftype@TYPE` The type number of floats or type TYPE. These are unique, powers of 2. For example, the type number of figures might be 1, tables 2, programs 4.

`\ext@TYPE` File extension of the file in which the contents list of the TYPE floats is stored (example: `\ext@figure=‘lof’`)

`fnum@TYPE` The macro which generates the number for the caption of the TYPE float.

`\@makecaption<num><text>` The macro to make a caption with `<num>` the value produced by `\fnum@TYPE` and `<text>` the text of the caption (in a `\parbox` of appropriate width). It is used for all floating objects.

`\c@figure` The format of the counter of figures. Often, the chapter number is part of the number, as this supposedly makes it easier to find the figure again, but that seems rather a poor justification, especially in the light of the excellent capabilities of TeX of making cross references.

```

683 \newcounter{figure}
684 \renewcommand\thefigure{\@arabic\c@figure}

```

`\fps@figure` These are the parameters dealing with the ‘figure’ floats.

```

\ftype@figure 685 \def\fps@figure{tbp}
\ext@figure    686 \def\ftype@figure{1}
\num@figure    687 \def\ext@figure{lof}
               688 \def\fnum@figure{\figurename~\thefigure}

```

`figure` Here we define the actual floating figure environments. The starred version is used for double-column figures.

```

689 \newenvironment{figure}
690   {\@float{figure}}
691   {\end@float}
692 \newenvironment{figure*}
693   {\@dblfloat{figure}}
694   {\end@dblfloat}

```

`\c@table` And now for tables as floats. The numbering of tables follows the same ideas as that of figures.

```

695 \newcounter{table}
696 \renewcommand\thetable{\@arabic\c@table}

```

`\fps@table` Define the parameters for table floats.

```

\ftype@table 697 \def\fps@table{tbp}
\ext@table 698 \def\ftype@table{2}
\num@table 699 \def\ext@table{lot}
700 \def\fnm@table{\tablename~\thetable}

```

`table` Here we define the actual floating table environments. The starred version is used for double-column tables.

```

701 \newenvironment{table}
702   {\@float{table}}
703   {\end@float}
704 \newenvironment{table*}
705   {\@dblfloat{table}}
706   {\end@dblfloat}

```

`\makecaption` The `\caption` macro calls `\@makecaption` to typeset the caption of the floating object. It uses the arguments *number* and *text*, so that number and captiontext can be typeset. The macro assumes, rightly, that it is called inside a `\parbox` of appropriate width.

`\abovecaptionskip`

`\belowcaptionskip`

```

707 \newlength\abovecaptionskip
708 \newlength\belowcaptionskip
709 \setlength\abovecaptionskip{10\p@}
710 \setlength\belowcaptionskip{0\p@}

```

The definition of the macro is `\long`, allowing more than a single paragraph to be put into the caption.

```

711 \long\def\@makecaption#1#2{%
712   \vskip\abovecaptionskip

```

In order to see if the caption text fits on a single line, we typeset it temporarily into a box.

```

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

```

Now measure the width of the box. If it exceeds the current `\hsize` then the caption text has to be set as an ordinary paragraph.

```

714   \ifdim \wd\@tempboxa >\hsize
715     #1: #2\par

```

If the text fits, center it on the line. This means using a `\hbox` in vertical mode, hence the `\everypar` tokens are not executed. What could be needed, if at all, is a resetting of the ‘minipage’ flag, so we do that here explicitly.

```

716   \else
717     \global \@minipagefalse
718     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
719   \fi
720   \vskip\belowcaptionskip}

```



## 6 Cross referencing

Commands like `\section` or `\caption` write a contentsline to an appropriate file. The line `\contentsline{section}{<title>}{<page>}` is written to the `.toc` file by the `\section` command. If sections are numbered, the `<title>` will show up as `\numberline{<num> }{<heading>}`, the `<num>` is the number generated by `\thesection`.

Similarly, the `\caption` command in a figure environment writes to a `.lof` file the line `\contentsline{figure}{\numberline{<num>}{<caption>}}{<page>}`.

The `\contentsline{<name>}` command expands to `\l@<name>`. Hence, we have to define `\l@chapter`, `\l@section` and so on, and for figures the similar `\l@figure`. These can be defined conveniently with the command `\dottedtocline{<level>}{<indent>}{<numwidth> }{<title>}{<page>}`.

`<level>` An entry is only produced if `<level>` is smaller or equal to the `tocdepth` counter, with chapters being level 0, sections level 1 and so on.

`<indent>` Defines the indentation from the outer left margin of the contentsline.

`<numwidth>` Defines the width of the box containing the section number, if the `<title>` includes a `\numberline`.

### 6.1 Table of contents, etc.

`\@pnumwidth` The three parameters control the appearance of the contents. Their definition through the `\newcommand` allows the use of the ‘em’ measure, meaning that the sizes defined can depend on the font used. The role of `\@pnumwidth` is to define the width of the box containing the page number. `\@tocrmargin` defines the right margin for multi-line entries, obviously it is a good idea to have `\@tocrmarg ≤ \@pnumwidth`. Finally, the separation between the dots on contentslines in mu units is given as a real number by `\@dotsep`. From a design perspective, these dots are very unattractive, and an easy way of avoiding them appearing is to set the separation to a very high number.

```
721 \newcommand\@pnumwidth{1.55em}
722 \newcommand\@tocrmarg{2.55em}
723 \newcommand\@dotsep{4500}
```

`\tableofcontents` This is the request for the production of a table of contents. Such a table is always typeset as a single column, so it might be necessary to switch temporarily to single-column mode.

```
724 \setcounter{tocdepth}{2}
725 \newcommand\tableofcontents{%
726     \if@twocolumn
727         \@restonecoltrue\onecolumn
728     \else
729         \@restonecolfalse
730     \fi
```

The title is set with `\chapter*`, to ensure that a possible running head contains the correct information

```
731     \chapter*{\contentsname
732         \mkboth%
```

```

733      \MakeUppercase\contentsname}
734      {\MakeUppercase\contentsname}}%

```

The actual table of contents itself is typeset by calling `\@starttoc{toc}`. If necessary, we also switch back to double-column mode.

```

735      \@starttoc{toc}%
736      \if@restonecol
737          \twocolumn
738      \fi}

```

**\l@part** The sectioning commands require additional macros to format the entries in the table of contents. Of these various entries, the entry for parts is somewhat more involved. First of all, ensure that if a pagebreak may be imminent, it should occur before the entry of a part. The addition of a little white space and the beginning of a group to keep changes local occupies us here as well.

```

739 \newcommand*\l@part[2]{%
740     \ifnum \c@tocdepth >-2\relax
741     \addpenalty{-\@highpenalty}%
742     \addvspace{2.25em \@plus\p@}%
743     \begingroup

```

Set `\parindent` to 0pt, and use `\rightskip` to reserve enough room for the page numbers. The `\parfillskip` is made negative so that no overfull box messages are issued.

```

744         \setlength\@tempdima{3em}%
745         \parindent \z@ \rightskip \@pnumwidth
746         \parfillskip -\@pnumwidth

```

Set the entry in large font, and do a bit of housekeeping before finishing the macro.

```

747         {\leavevmode
748         \large #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
749         \nobreak
750         \global\@nobreaktrue
751         \everypar{\global\@nobreakfalse\everypar{}}
752     \endgroup
753     \fi}

```

**\l@chapter** The entries of chapters is very similar to that of parts, and the definition of the macro follows the same lines.

```

754 \newcommand*\l@chapter[2]{%
755     \ifnum \c@tocdepth >\m@ne
756     \addpenalty{-\@highpenalty}%
757     \vskip 1.0em \@plus\p@

```

The `\numberline` macro needs the width of the box holding the number of the chapter to be in the scratch `\@tempdima`. Also, begin a new group and change some of the paragraph parameters.

```

758     \setlength\@tempdima{1.5em}%
759     \begingroup
760     \parindent \z@ \rightskip \@pnumwidth
761     \parfillskip -\@pnumwidth
762     \leavevmode

```

As we do not use `\numberline`, some manual adjustments are needed before typesetting the entry. A pagebreak following a chapter entry should be discouraged but not forbidden.

```

763      \advance\leftskip\@tempdima
764      \hskip -\leftskip
765      #1\nobreak\hfil \nobreak\hb@xt@\@pnumwidth{\hss #2}\par
766      \penalty\@highpenalty
767      \endgroup
768      \fi}

```

`\l@section` The lower level entries are all governed by the `\@dottedtocline` macro.

```

\l@section 769 \newcommand*\l@section{\@dottedtocline{1}{1.5em}{2.3em}}
\l@subsection 770 \newcommand*\l@subsection{\@dottedtocline{2}{3.8em}{3.2em}}
\l@subsubsection 771 \newcommand*\l@subsubsection{\@dottedtocline{3}{7.0em}{4.1em}}
\l@paragraph 772 \newcommand*\l@paragraph{\@dottedtocline{4}{10em}{5em}}
\l@subparagraph 773 \newcommand*\l@subparagraph{\@dottedtocline{5}{12em}{6em}}

```

`\listoffigures` This is the request to produce a list of figures, and very similar in construction to `\tableofcontents`.

```

774 \newcommand\listoffigures{%
775     \if@twocolumn
776         \@restonecoltrue\onecolumn
777     \else
778         \@restonecolfalse
779     \fi
780     \chapter*{\listfigurename
781         \@mkboth{\MakeUppercase\listfigurename}%
782         {\MakeUppercase\listfigurename}}%
783     \@starttoc{lof}%
784     \if@restonecol\twocolumn
785     \fi}

```

`\l@figure` The macro which produces the actual entry in the list of figures.

```

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

```

`\listoftables` The request to make a list of tables is given through this macro. Once again, it is very similar to the `\tableofcontents` macro.

```

787 \newcommand\listoftables{%
788     \if@twocolumn
789         \@restonecoltrue\onecolumn
790     \else
791         \@restonecolfalse
792     \fi
793     \chapter*{\listtablename
794         \@mkboth{%
795             \MakeUppercase\listtablename}{\MakeUppercase\listtablename}}%
796     \@starttoc{lot}%
797     \if@restonecol\twocolumn
798     \fi}

```

`\l@table` The macro which produces the actual entry in the list of tables.

```

799 \let\l@table\l@figure

```

## 6.2 Bibliography

`\bibindent` The open bibliography format uses the indentation set by `\bibindent`.

```
800 \newdimen\bibindent
801 \bibindent=1.5em
```

`thebibliography` The environment is based on a generic list environment, using the *enumiv* as internal counter to generate the labels of the list. If an empty *thebibliography* environment is found, issue a warning. The environment executes a number of commands: `\sloppy` is used because it is hard to do line breaks in bibliographies, `\sfcode'\.=1000\relax` is issued so that a period does not produce an end-of-sequence space.

```
802 \newenvironment{thebibliography}[1]
803   {\chapter*{\bibname
804     \mkboth{\MakeUppercase\bibname}
805     {\MakeUppercase\bibname}}%
806   \list{\@biblabel{\@arabic\c@enumiv}}%
807     {\settowidth\labelwidth{\@biblabel{#1}}%
808     \leftmargin\labelwidth
809     \advance\leftmargin\labelsep
810     \if@openbib
811       \advance\leftmargin\bibindent
812       \itemindent -\bibindent
813       \listparindent \itemindent
814       \parsep \z@
815     \fi
816     \usecounter{enumiv}%
817     \let\p@enumiv\empty
818     \renewcommand\theenumiv{\@arabic\c@enumiv}}%
819     \if@openbib
820       \renewcommand\newblock{\par}%
821     \else
822       \renewcommand\newblock{\hskip .11em \@plus.33em \@minus.07em}%
823     \fi
824     \sloppy

Reset parameters to their previous values before leaving the environment
825     \clubpenalty4000
826     \@clubpenalty \clubpenalty
827     \widowpenalty4000
828     \sfcode'\.=\@m}
829     {\def\@noitemerr
830     {\@latex@warning{Empty 'thebibliography' environment}}}%
831   \endlist}
832 \newcommand\newblock{}
```

## 6.3 The index

`theindex` The index environment produces a two-column index, with each new entry having its own paragraph. In the text, the commands `\item`, `\subitem` and `\subsubitem` are available to produce index entries at three levels. Each time a new letter of the alphabet is encountered, an `\indexspace` amount of white space can be inserted.

```
833 \newenvironment{theindex}
```

```

834     {\if@twocolumn
835       \@restonecolfalse
836     \else
837       \@restonecoltrue
838     \fi
839     \columnseprule \z@
840     \columnsep 35\p@
841     \twocolumn[\@makeschapterhead{\indexname}]%
842     \@mkboth{\MakeUppercase\indexname}%
843     {\MakeUppercase\indexname}%
844     \thispagestyle{plain}\parindent\z@
845     \parskip\z@ \@plus .3\p@\relax
846     \let\item\@idxitem}

```

In case more text follows the index, and the document was typeset in single-column mode, switch back to the one column mode before finishing the index.

```

847     {\if@restonecol
848       \onecolumn
849     \else
850       \clearpage
851     \fi}

```

```

\@idxitem The macros which typeset the actual index entries
\subitem 852 \newcommand\@idxitem{\par\hangindent 40\p@}
\subsubitem 853 \newcommand\subitem{\par\hangindent 40\p@ \hspace*{20\p@}}
854 \newcommand\subsubitem{\par\hangindent 40\p@ \hspace*{30\p@}}

```

\indexspace Define the amount of white space to insert between letter blocks in the index.

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

## 6.4 Footnotes

\footnoterule Footnotes are often separated from the body of the text by a small rule. To ensure that such a rule does not take up any vertical space, we have to compensate for the height of the rule (of 0.4pt) by adding some vertical skip. But we also have to avoid the rule from colliding with the footnote text, so we add first a bit of negative skip, and make sure that we end up eventually at the point where we began.

```

856 \renewcommand\footnoterule{%
857   \kern-3\p@
858   \hrule\@width.4\columnwidth
859   \kern2.6\p@}

```

\c@footnote Footnotes are numbered within chapters, so we have to restart the numbering when beginning a new chapter.

```
860 \@addtoreset{footnote}{chapter}
```

\@makefnmark The macro which produces the footnote. We use a simple macro, in which the footnote text is typeset like a normal paragraph, without indentation except for its first line. The footnote markers printed in the text, which refer to the actual footnotes are produced by the macro \@makefnmark. Here we take over the default definition without change.

```

861 \newcommand\@makefnmark[1]{%
862     \parindent 1em%
863     \noindent
864     \hb@xt@1.8em{\hss\@makefnmark}#1}

```

## 7 Initialisation

### 7.1 Heading names

The names of most of the headings can be (re)defined, and here we set these names to their English equivalent. They can be renamed in the text file through the `\renewcommand`.

```

865 \newcommand\contentsname{Contents}
866 \newcommand\listfigurename{List of Figures}
867 \newcommand\listtablename{List of Tables}
868 \newcommand\bibname{Bibliography}
869 \newcommand\indexname{Index}
870 \newcommand\figurename{Figure}
871 \newcommand\tablename{Table}
872 \newcommand\partname{Part}
873 \newcommand\chaptername{Chapter}
874 \newcommand\appendixname{Appendix}

```

### 7.2 Date

`\today` The macro `\today` relies on the T<sub>E</sub>X macros `\month`, `\day` and `\year` generate the date of the L<sup>A</sup>T<sub>E</sub>X run. The macro has been re-arranged from the standard L<sup>A</sup>T<sub>E</sub>X format to conform to European usage, that is, day month year sequence.

```

875 \newcommand\today{\number\day \ \ifcase\month\or
876   January\or February\or March\or April\or May\or June\or
877   July\or August\or September\or October\or November\or December\fi
878   \ \number\year}

```

### 7.3 Double columns

`\columnsep` The macros define the distance between the two columns, and the width of rule  
`\columnseprule` separating the columns. Such a rule is usually unattractive and serves no real purpose for the reader: it is therefore set to 0pt.

```

879 \setlength\columnsep{10\p@}
880 \setlength\columnseprule{0\p@}

```

### 7.4 Page style

Set the default page style to *heading* and set the folio style to *arabic*

```

881 \pagestyle{headings}
882 \pagenumbering{arabic}

```

If `twocolumn` was chosen as an option for typesetting the text, activate the double column mode with `\twocolumn`. Try to keep the columns balanced but do not make things too hard, hence issue a `\sloppy` command as well. Normally, call the `\onecolumn` macro to typeset in a single column.

```
883 \if@twocolumn
884   \twocolumn
885   \sloppy
886   \flushbottom
887 \else
888   \onecolumn
889 \fi
890 \</octavo>
      End of file 'octavo.cls'.
```