

L^AT_EX for ISO Standards: Source code*

Peter Wilson
Catholic University of America[†]
Now at peter.r.wilson@boeing.com

2002/07/22

Contents

1	Introduction	3
2	A driver for this document	4
3	Identification	6
4	Initial Code	6
5	Declaration of Options	7
5.1	Setting Paper Sizes	7
5.2	Choosing the type size	8
5.3	Two-side or one-side printing	8
5.4	Two column printing	8
5.5	The copyright option	8
5.6	Document kind options is, dis, cd, wd, techrep, otherdoc etc.	9
5.7	The draft option	11
6	Executing Options	11
7	Loading Packages	12
8	Document Layout	12
8.1	Fonts	12
8.2	Paragraphing	17
8.3	Page Layout	18

*This file has version number v2.4, last revised 2002/07/22.

[†]This work was originally performed as a Guest Researcher at the National Institute of Standards and Technology.

8.3.1	Vertical spacing	18
8.3.2	The dimension of text	19
8.3.3	Margins	19
8.3.4	Footnotes	20
8.3.5	Float placement parameters	20
8.4	Page Styles	22
8.4.1	Marking conventions	22
8.4.2	Defining the page styles	24
9	Document Markup	25
9.1	The title	25
9.2	The cover	26
9.3	Clauses	26
9.3.1	Building blocks	26
9.3.2	Overview	27
9.3.3	Hyperref ToC levels	28
9.3.4	Define Counters	28
9.3.5	Clauses	29
9.3.6	Lower level headings	30
9.3.7	Annexes	31
9.4	Lists	33
9.4.1	General List Parameters	33
9.4.2	Enumerate	35
9.4.3	Itemize	36
9.4.4	Description	37
9.5	Defining new environments	37
9.5.1	Quotation	37
9.5.2	Quote	37
9.5.3	Theorem	37
9.5.4	Notes	38
9.5.5	Examples	40
9.5.6	Listing of references	41
9.5.7	Listing of definitions	43
9.5.8	Listing of symbols and abbreviations	45
9.5.9	Listing of scope items	45
9.6	Setting parameters for existing environments	46
9.6.1	Array and tabular	46
9.6.2	Tabbing	46
9.6.3	Minipage	46
9.6.4	Framed boxes	46
9.6.5	Equation and eqnarray	47
9.7	Floating objects	47
9.7.1	Figure	48
9.7.2	Table	49
9.7.3	A bottom float	49
9.7.4	Captions	50

9.8	Font changing	51
9.9	Urls, etc	52
10	Cross Referencing	52
10.1	Label referencing	52
10.2	Table of Contents, etc.	52
10.2.1	Table of Contents	53
10.2.2	List of figures	55
10.2.3	List of tables	55
10.2.4	ToC and clause numbering	56
10.3	Bibliography	59
10.4	The index	59
10.5	Footnotes	60
11	Version control tools	62
11.1	Print control	62
11.2	Change marking	62
12	Structure and boilerplate	63
12.1	Structural elements	63
12.2	Boilerplate	65
13	Initialization	69
13.1	Words and phrases	69
13.2	Date	71
13.3	Two column mode	71
13.4	The page style	71
13.5	Single or double sided printing	72
14	The askinc package	72

1 Introduction

This document provides the commented source for L^AT_EX class and package files designed for the typesetting of documents according to the rules for ISO international standards. A separate document provides the user manual [Wil96]. This manual is typeset according to the conventions of the L^AT_EX DOCSTRIP utility which enables the automatic extraction of the L^AT_EX macro source files [GMS94].

The original version of this class was used for the production of camera ready copy for the ISO 10303 standard *Product data representation and exchange*. The initial release of ISO 10303:1994 consisted of twelve parts and over 2400 pages. The editorial board of the ISO Central Secretariat in Geneva accepted the typographic conventions embodied in those macros.

ISO (the International Organization for Standardisation) specify their document layout requirements in ISO Directives [ISO97]. Unfortunately these Directives do not completely define the document layout, leaving several aspects open

to interpretation by the document editor and re-interpretation by the ISO editorial board. At the request of the editors of ISO 10303, and no doubt others as well, ISO has clarified the intent of their Directives [ISO01]. Also, since they were published ISO has been considering how best to accept and use electronic manuscripts instead of camera ready paper copy. At the time of writing (July 2001) they will accept documents in PDF format. This has also led to some changes in requirements.

The following specifications are a re-implementation of the class macros published in July 2000.

This manual is provided as a service for future developers and maintainers of the class and packages for ISO standards. It is assumed that any such person is L^AT_EX literate and accustomed to supporting complex class and package files [GMS94].

Sections 2 through 4 describe some administrative elements and code for general use later in the specification. The macros forming the class file are defined in sections 5 through 13. These are principally revisions of the report class to meet ISO typographic requirements and many new macros to support specific structural elements of an ISO standard to provide logical markup capabilities. Section 14 describes the macros for the `askinc` package for interactive file inclusion.

2 A driver for this document

The next series of code contains the documentation driver file for L^AT_EX, i.e., the file that will produce the documentation you are currently reading. This will be extracted from this file by the DOCSTRIP program.

```

1 {*driver}
2 \documentclass{ltxdoc}

   We do not want the following basic elements to appear in the index.
3 \DoNotIndex{\', \., \@M, \@input, \@addtoreset, \@arabic, \@badmath}
4 \DoNotIndex{\@centercr, \@cite}
5 \DoNotIndex{\@dotsep, \@empty, \@float, \@gobble, \@gobbletwo, \@ignoretrue}
6 \DoNotIndex{\@input, \@ixpt, \@m}
7 \DoNotIndex{\@minus, \@mkboth, \@ne, \@nil, \@nomath, \@plus, \@set@topoint}
8 \DoNotIndex{\@tempboxa, \@tempcnta, \@tempdima, \@tempdimb}
9 \DoNotIndex{\@tempswafalse, \@tempswatruer, \@viipt, \@viipt, \@vipt}
10 \DoNotIndex{\@vpt, \@warning, \@xipt, \@xipt, \@xivpt, \@xpt, \@xvipt}
11 \DoNotIndex{\@xxpt, \@xxvpt, \, \ , \addpenalty, \addtolength, \addvspace}
12 \DoNotIndex{\advance, \Alph, \alph}
13 \DoNotIndex{\arabic, \ast, \begin, \beginngroup, \bfseries, \bgroup, \box}
14 \DoNotIndex{\bullet}
15 \DoNotIndex{\cdot, \cite, \CodelineIndex, \cr, \day, \DeclareOption}
16 \DoNotIndex{\def, \DisableCrossrefs, \divide, \DocInput, \documentclass}
17 \DoNotIndex{\DoNotIndex, \egroup, \ifdim, \else, \fi, \em, \endtrivlist}
18 \DoNotIndex{\EnableCrossrefs, \end, \end@dblfloat, \end@float, \endgroup}
19 \DoNotIndex{\endlist, \everycr, \everypar, \ExecuteOptions, \expandafter}
20 \DoNotIndex{\fbox}

```

```

21 \DoNotIndex{\filedate, \filename, \fileversion, \fontsize, \framebox, \gdef}
22 \DoNotIndex{\global, \halign, \hangindent, \hbox, \hfil, \hfill, \hrule}
23 \DoNotIndex{\hspace, \hskip, \hspace, \hss, \if@tempswa, \ifcase, \or, \fi, \fi}
24 \DoNotIndex{\ifhmode, \ifvmode, \ifnum, \iftrue, \ifx, \fi, \fi, \fi, \fi}
25 \DoNotIndex{\input}
26 \DoNotIndex{\jobname, \kern, \leavevmode, \let, \leftmark}
27 \DoNotIndex{\list, \llap, \long, \m@ne, \m@th, \mark, \markboth, \markright}
28 \DoNotIndex{\month, \newcommand, \newcounter, \newenvironment}
29 \DoNotIndex{\NeedsTeXFormat, \newdimen}
30 \DoNotIndex{\newlength, \newpage, \nobreak, \noindent, \null, \number}
31 \DoNotIndex{\numberline, \OldMakeindex, \OnlyDescription, \p@}
32 \DoNotIndex{\pagestyle, \par, \paragraph, \paragraphmark, \parfillskip}
33 \DoNotIndex{\penalty, \PrintChanges, \PrintIndex, \ProcessOptions}
34 \DoNotIndex{\protect, \ProvidesClass, \raggedbottom, \raggedright}
35 \DoNotIndex{\refstepcounter, \relax, \renewcommand, \reset@font}
36 \DoNotIndex{\rightmargin, \rightmark, \rightskip, \rlap, \rmfamily, \roman}
37 \DoNotIndex{\roman, \secdef, \selectfont, \setbox, \setcounter, \setlength}
38 \DoNotIndex{\settowidth, \sfcode, \skip, \sloppy, \slshape, \space}
39 \DoNotIndex{\symbol, \the, \trivlist, \typeout, \tw@, \undefined, \uppercase}
40 \DoNotIndex{\usecounter, \usefont, \usepackage, \vfil, \vfill, \viipt}
41 \DoNotIndex{\vipt, \vipt, \vskip, \vspace}
42 \DoNotIndex{\wd, \xiipt, \year, \z@}

```

We do want an index, using linenumbers, but not update information.

```

43 \EnableCrossrefs
44 \CodelineIndex
45 %% \RecordChanges

```

We use so many docstrip modules that we set the StandardModuleDepth counter to 1.

```

46 \setcounter{StandardModuleDepth}{1}

```

Some commonly used abbreviations

```

47 \newcommand*{\Lopt}[1]{\textsf {#1}}           % typeset an option
48 \newcommand*{\file}[1]{\texttt {#1}}          % typeset a file
49 \newcommand*{\Lcount}[1]{\textsl {#1}}        % typeset a counter
50 \newcommand*{\pstyle}[1]{\textsl {#1}}        % typeset a pagestyle
51 \newcommand*{\Lenv}[1]{\texttt {#1}}          % typeset an environment
52 \newcommand*{\Lpack}[1]{\textsf {#1}}         % typeset a package

```

We want the full details printed.

```

53 \begin{document}
54 \DocInput{isoe.dtx}
55 \PrintIndex
56 %% \PrintChanges
57 \end{document}
58 </driver>

```

3 Identification

The `iso` document class can only be used with L^AT_EX2e, so we make sure that an appropriate message is displayed when another T_EX format is used.

```
59 <iso>\NeedsTeXFormat{LaTeX2e}
```

Announce the name, option files and version for L^AT_EX2e files:

```
60 <iso>\ProvidesClass{isov2}[2002/07/22 v2.4 LaTeX ISO document class]
61 <9pt>\ProvidesFile{iso9.clo}[1997/11/30 v1.1 ISO class size option]
62 <10pt>\ProvidesFile{iso10.clo}[1997/11/30 v1.1 ISO class size option]
63 <11pt>\ProvidesFile{iso11.clo}[1997/11/30 v1.1 ISO class size option]
64 <inc>\ProvidesPackage{askincv1}[1995/05/31 Interactive include package]
65 <fwd1>\ProvidesFile{isofwdbp.tex}[2002/01/10 ISO Foreword boilerplate]
66 <trfwd1>\ProvidesFile{trfwd1.tex}[2002/01/10 PAS/TS Foreword boilerplate]
```

4 Initial Code

```
67 <*iso>
```

The class requires the `url` package, so make sure that it is loaded.

```
68 \RequirePackage{url}
```

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.

```
69 \newcommand{\@ptsize}{}
```

`\if@restonecol` When the document has to be printed in two columns, we sometimes have to temporarily switch to one column. This switch is used to remember to switch back.

```
70 \newif\if@restonecol
```

`\isostringsequal` The command `\isostringsequal` is based on code in Stephan von Bechtolsheim *T_EX in Practice*, vol III page 334. It enables the definition of specific commands for testing whether two strings are equal.

`\isoemptystring`

```
71 \def\isostringsequal #1#2{%
72   TT\fi
73   \edef\@is@str@ngsequali{#1}%
74   \edef\@is@str@ngsequalii{#2}%
75   \ifx\@is@str@ngsequali\@is@str@ngsequalii}
```

Now we define the `\isoemptystring` command for use in testing for an empty parameter.

```
76 \def\isoemptystring #1{%
77   TT\fi
78   \if\isostringsequal{#1}{}}
```

`\fillline` This command draws a horizontal line across the page.

```
79 \newcommand{\fillline}{\mbox{}}\hrulefill\mbox{}}
```

`\makecommand` The `\makecommand` macro is like the `\newcommand` macro except that it always (re)defines a command. It is equivalent to the pair of commands:
`\providecommand{\com}...\renewcommand{\com}...`
The code for `\make@command` is a simplified version of the code for `\renew@command` in file `ltdefns.dtx`.

```
80 \newcommand{\makecommand}{\@star@or@long\make@command}
81 \newcommand{\make@command}[1]{%
82   \let\@ifdefinable\@rc@ifdefinable
83   \new@command#1}
```

`\ifpdf` This can be used to check whether or not a document is being processed by LaTeX or pdfLaTeX.

```
84 \newif\ifpdf
85 \ifx\pdfoutput\undefined
86   \pdffalse
87 \else
88   \pdftrue
89 \fi
```

`\ifisohyper` This can be used to check, after `\begin{document}` to check if the `hyperref` package has been used.

```
90 \newif\ifisohyper
91 \isohyperfalse
92 \AtBeginDocument{%
93   \@ifpackageloaded{hyperref}%
94     {\isohypertrue}%
95     {\newcommand{\hyperpage}[1]{#1}}%
96 }
97
```

5 Declaration of Options

5.1 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. We assume that the document will only be printed on either ISO standard A4 paper (option `a4paper`) or on the most common of the US paper sizes (option `letterpaper`).

Option `a4paper` will be the default.

`\if@us` A flag for the paper size option.

```
98 \newif\if@us\@usfalse
```

Declare the paper size options.

```
99 \DeclareOption{a4paper}
100   {\setlength\paperheight {297mm}%   %% 11.69in
```

```

101 \setlength\paperwidth {210mm}} %% 8.27in
102 \DeclareOption{letterpaper}
103 {\setlength\paperheight {11in}} %% 279mm
104 \setlength\paperwidth {8.5in}} %% 216mm
105 \@ustrue}

```

5.2 Choosing the type size

The type size options are handled by defining `\@ptsize` to contain the last digit of the size in question and branching on `\ifcase` statements. This is done for historical reasons to stay compatible with other packages that use the `\@ptsize` variable to select special actions. It makes the declarations of size options less than 10pt difficult, although one can probably use 9 assuming that a class will not define both 9pt and 19pt options.

Option 11pt will be the default.

```

106 \renewcommand{\@ptsize}{1}
107 \DeclareOption{9pt}{\renewcommand{\@ptsize}{9}}
108 \DeclareOption{10pt}{\renewcommand{\@ptsize}{0}}
109 \DeclareOption{11pt}{\renewcommand{\@ptsize}{1}}

```

5.3 Two-side or one-side printing

For two-sided printing we use the switch `\if@twoside`. In addition we have to set the `\if@mparswitch` to get any margin paragraphs into the outside margin. In this class we always use two-sided printing with marginal notes on the outside.

```

\if@twoside
\if@mparswitch 110 \@twosidetrue \@mparswitchtrue

```

5.4 Two column printing

Two-column and one-column printing is again realized via a switch which is defined in the kernel. The default is single column printing.

```

\if@twocolumn
111 \DeclareOption{onecolumn}{\@twocolumnfalse}
112 \DeclareOption{twocolumn}{\@twocolumntrue}

```

5.5 The copyright option

The default is not to print ISO copyright notices. This option enables copyright notice printing. As usual, we employ a flag.

```

\ifc@pyrightopt c@pyrightopt stores the user's option, while c@pyright will be used to control
printing of copyright notices and symbols in the body of the document.
113 \newif\ifc@pyright\c@pyrightfalse
114 \newif\ifc@pyrightopt\c@pyrightoptfalse

```



```

115 \DeclareOption{copyright}{\c@pyrightopttrue}
116 \DeclareOption{notcopyright}{\c@pyrightoptfalse}

```

5.6 Document kind options `is`, `dis`, `cd`, `wd`, `techrep`, `otherdoc` etc.

The default is to assume that an ISO standard in preparation is to be printed (effectively this is the `otherdoc` option). The `is` option declares that an International Standard (IS) is to be printed. The `fdis` option declares that a Final Draft International Standard (FDIS) is to be printed, and similarly the `dis` option declares that a Draft International Standard (DIS) is to be printed. The `cd` option is for Committee Draft (CD) documents and the option `wd` is for Working Drafts.

The `techrep` option declares that a Technical Report (probably type 1 or 2) is to be printed.

The `otherdoc` option indicates that the document is not intended to become an ISO standard (e.g., is an ISO internal report).

```

\ifisstandard We use flags for remembering which option is in effect.
\iffdisstandard 117 \newif\ifisstandard\isstandardfalse
\ifdisstandard 118 \newif\iffdisstandard\fdisstandardfalse
\ifcdstandard 119 \newif\ifdisstandard\disstandardfalse
\ifwdstandard 120 \newif\ifcdstandard\cdstandardfalse
\iftechrep 121 \newif\ifwdstandard\wdstandardfalse
\ifotherdoc 122 \newif\iftechrep\techrepfalse
123 \newif\ifotherdoc\otherdocfalse

\iftechspec Flags for the techspec Technical Specification and pas Publicly Available Specification options.
\ifpaspec
124 \newif\iftechspec\techspecfalse
125 \newif\ifpaspec\paspecfalse

```

Now declare the options (including an `is` option just for completeness). We need to ensure (later) that, whatever copyright option has been used, copyright notices are not printed for certain kinds of documents.

```

126 \DeclareOption{is}{\isstandardtrue
127 \fdisstandardfalse
128 \disstandardfalse
129 \cdstandardfalse
130 \wdstandardfalse
131 \techrepfalse
132 \techspecfalse
133 \paspecfalse
134 \otherdocfalse}
135 \DeclareOption{fdis}{\isstandardfalse
136 \fdisstandardtrue
137 \disstandardfalse
138 \cdstandardfalse
139 \wdstandardfalse

```

```

140             \techrepfalse
141             \techspecfalse
142             \paspecfalse
143             \otherdocfalse}
144 \DeclareOption{dis}{\isstandardfalse
145             \fdisstandardfalse
146             \disstandardtrue
147             \cdstandardfalse
148             \wdstandardfalse
149             \techrepfalse
150             \techspecfalse
151             \paspecfalse
152             \otherdocfalse}
153 \DeclareOption{cd}{\isstandardfalse
154             \fdisstandardfalse
155             \disstandardfalse
156             \cdstandardtrue
157             \wdstandardfalse
158             \techrepfalse
159             \techspecfalse
160             \paspecfalse
161             \otherdocfalse
162             \copyrightfalse}
163 \DeclareOption{wd}{\isstandardfalse
164             \fdisstandardfalse
165             \disstandardfalse
166             \cdstandardfalse
167             \wdstandardtrue
168             \techrepfalse
169             \techspecfalse
170             \paspecfalse
171             \otherdocfalse
172             \copyrightfalse}
173 \DeclareOption{techrep}{\isstandardfalse
174             \fdisstandardfalse
175             \disstandardfalse
176             \cdstandardfalse
177             \wdstandardfalse
178             \techreptrue
179             \techspecfalse
180             \paspecfalse
181             \otherdocfalse}
182 \DeclareOption{techspec}{\isstandardfalse
183             \fdisstandardfalse
184             \disstandardfalse
185             \cdstandardfalse
186             \wdstandardfalse
187             \techrepfalse
188             \techspectrue
189             \paspecfalse

```

```

190             \otherdocfalse}
191 \DeclareOption{pas}{\isstandardfalse
192             \fdisstandardfalse
193             \disstandardfalse
194             \cdstandardfalse
195             \wdstandardfalse
196             \techreptrue
197             \techspecfalse
198             \paspectrue
199             \otherdocfalse}
200 \DeclareOption{otherdoc}{\isstandardfalse
201             \fdisstandardfalse
202             \disstandardfalse
203             \cdstandardfalse
204             \wdstandardfalse
205             \techrepfalse
206             \techspecfalse
207             \paspecfalse
208             \otherdoctrue
209             \copyrightfalse}

```

5.7 The draft option

If the user requests `draft` we show any overfull boxes, marginal notes are allowed, and any copyright notices are not printed. For symmetry, we also define a `final` option which is the default.

```

\ifdr@ftd@c
210 \newif\ifdr@ftd@c\dr@ftd@cfalse
211 \setlength{\overfullrule}{\z@}
212 \DeclareOption{final}{\setlength{\overfullrule}{\z@}
213             \dr@ftd@cfalse}
214 \DeclareOption{draft}{\setlength{\overfullrule}{5pt}%
215             \dr@ftd@ctrue}

```

6 Executing Options

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

```

216 \ExecuteOptions{notcopyright,otherdoc,final,a4paper,11pt,onecolumn}

```

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.

```

217 \ProcessOptions

```

`\ifc@pyright` Ensure that we have the correct value of `\ifc@pyright` no matter the ordering in which the options are processed.

```
218 \c@pyrightfalse
219 \ifc@pyrightopt
220 \c@pyrighttrue
221 \fi
```

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

```
222 \ifnum\@ptsize < \tw@
223   \input{iso1\@ptsize.clo}
224 \else
225   \input{iso\@ptsize.clo}
226 \fi
```

7 Loading Packages

This class file does not load additional package files.

8 Document Layout

In this section we deal with the more difficult typographical details.

8.1 Fonts

L^AT_EX offers the user commands to change the size of the font, relative to the ‘main’ size. Each relative size changing command `\size` executes the command `\setfontsize\size⟨font-size⟩⟨baselineskip⟩` where:

⟨font-size⟩ The absolute size of the font to use from now on.

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

A number of commands, defined in the L^AT_EX kernel, shorten the following definitions and are used throughout. They are:

<code>\@vpt</code>	5	<code>\@vipt</code>	6	<code>\@viipt</code>	7
<code>\@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
<code>\@xvipt</code>	17.28	<code>\@xxpt</code>	20.74	<code>\@xxvpt</code>	24.88

`\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 9pt/11pt size.

The `\normalsize` macro also sets new values for `\abovedisplayskip`, `\abovedisplayshortskip` and `\belowdisplayshortskip`.

```

227 </iso>
228 <*9pt | 10pt | 11pt>
229 \renewcommand{\normalsize}{%
230 <*9pt>
231   \setfontsize\normalsize\@ixpt\@xpt
232   \abovedisplayskip 9\p@ \@plus 2\p@ \@minus 4.5\p@
233   \abovedisplayshortskip \z@ \@plus 3\p@
234   \belowdisplayshortskip 5.5\p@ \@plus 2.5\p@ \@minus 3\p@
235 </9pt>
236 <*10pt>
237   \setfontsize\normalsize\@xpt\@xipt
238   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
239   \abovedisplayshortskip \z@ \@plus3\p@
240   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
241 </10pt>
242 <*11pt>
243   \setfontsize\normalsize\@xipt{13.6}%
244   \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
245   \abovedisplayshortskip \z@ \@plus3\p@
246   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
247 </11pt>

```

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

```

248 \belowdisplayskip \abovedisplayskip
249 \let\@listi\@listI}

```

We initially choose the `normalsize` font.

```

250 \normalsize

```

`\smidgeon` ISO typesetting is grid based, which is not something that \LaTeX is good at. We use some ‘fixed’ skips for before and after headings, plus a flexible `smidgeon`.
`\parskip` For the grid, we want a fixed size `\parskip`, dependant only on the normal font, of one blank line (i.e., the `\baselineskip`).
`\onelineskip` Just in case the value of `\parskip` gets changed, also keep a similar value in `\onelineskip`.

```

251 \newlength{\@smidgeon}
252 \setlength{\@smidgeon}{0.5\p@ \@plus 1\p@ \@minus 1\p@}
253 \newlength{\@onelineskip}
254 <9pt>\parskip \@xpt\p@
255 <9pt>\setlength{\@onelineskip}{\@xpt\p@}
256 <10pt>\parskip \@xipt\p@
257 <10pt>\setlength{\@onelineskip}{\@xipt\p@}
258 <11pt>\parskip 13.6\p@
259 <11pt>\setlength{\@onelineskip}{13.6\p@}

```

`\small` This code is similar to that for `\normalsize`.

```

260 \newcommand{\small}{%
261  <*9pt>
262    \@setfontsize\small\@viiipt{9}
263    \abovedisplayskip 6\p@ \@plus 2\p@ \@minus 4\p@
264    \abovedisplayshortskip \z@ \@plus 2\p@
265    \belowdisplayshortskip 4\p@ \@plus 2\p@ \@minus 2\p@
266    \def\@listif{\leftmargin\leftmargini
267      \topsep 2\p@ \@plus 2\p@ \@minus 2\p@
268      \parsep 1\p@ \@plus\p@ \@minus\p@
269      \itemsep \parsep
270      \itemindent\z@
271      }%
272 </9pt>
273 <*10pt>
274   \@setfontsize\small\@ixpt{11}%
275   \abovedisplayskip 8.5\p@ \@plus3\p@ \@minus4\p@
276   \abovedisplayshortskip \z@ \@plus2\p@
277   \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
278   \def\@listif{\leftmargin\leftmargini
279     \topsep 4\p@ \@plus2\p@ \@minus2\p@
280     \parsep 2\p@ \@plus\p@ \@minus\p@
281     \itemsep \parsep
282     \itemindent\z@
283     }%
284 </10pt>
285 <*11pt>
286   \@setfontsize\small\@xpt\@xiipt
287   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
288   \abovedisplayshortskip \z@ \@plus3\p@
289   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
290   \def\@listif{\leftmargin\leftmargini
291     \topsep 6\p@ \@plus2\p@ \@minus2\p@
292     \parsep 3\p@ \@plus2\p@ \@minus\p@
293     \itemsep \parsep
294     \itemindent\z@
295     }%
296 </11pt>
297   \belowdisplayskip \abovedisplayskip
298 }

```

`\footnotesize` This code is similar to that for `\normalsize`.

```

299 \newcommand{\footnotesize}{%
300  <*9pt>
301    \@setfontsize\footnotesize\@viiipt{9}
302    \abovedisplayskip 6\p@ \@plus 2\p@ \@minus 4\p@
303    \abovedisplayshortskip \z@ \@plus 2\p@
304    \belowdisplayshortskip 4\p@ \@plus 2\p@ \@minus 2\p@
305    \def\@listif{\leftmargin\leftmargini
306      \topsep 2\p@ \@plus 2\p@ \@minus 2\p@
307      \parsep 1\p@ \@plus\p@ \@minus\p@

```

```

308             \itemsep \parsep
309             \itemindent\z@
310         }%
311 </9pt>
312 <*10pt>
313     \@setfontsize\footnotesize\@viipt{9.5}%
314     \abovedisplayskip 6\p@ \@plus2\p@ \@minus4\p@
315     \abovedisplayshortskip \z@ \@plus\p@
316     \belowdisplayshortskip 3\p@ \@plus\p@ \@minus2\p@
317     \def\@listif\leftmargin\leftmargini
318         \topsep 3\p@ \@plus\p@ \@minus\p@
319         \parsep 2\p@ \@plus\p@ \@minus\p@
320         \itemsep \parsep
321         \itemindent\z@
322     }%
323 </10pt>
324 <*11pt>
325     \@setfontsize\footnotesize\@ixpt{11}%
326     \abovedisplayskip 8\p@ \@plus2\p@ \@minus4\p@
327     \abovedisplayshortskip \z@ \@plus\p@
328     \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
329     \def\@listif\leftmargin\leftmargini
330         \topsep 4\p@ \@plus2\p@ \@minus2\p@
331         \parsep 2\p@ \@plus\p@ \@minus\p@
332         \itemsep \parsep
333         \itemindent\z@
334     }%
335 </11pt>
336     \belowdisplayskip \abovedisplayskip
337 }

```

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

```

\large 338 <*9pt>
\Large 339 \newcommand{\tiny}{\@setfontsize\tiny\@vpt{6}}
\LARGE 340 \newcommand{\scriptsize}{\@setfontsize\scriptsize\@viipt{8}}
\huge 341 \newcommand{\large}{\@setfontsize\large\@xpt{11}}
\Huge 342 \newcommand{\Large}{\@setfontsize\Large\@xipt{12}}
343 \newcommand{\LARGE}{\@setfontsize\LARGE\@xivpt{18}}
344 \newcommand{\huge}{\@setfontsize\huge\@xviipt{22}}
345 \newcommand{\Huge}{\@setfontsize\Huge\@xxpt{25}}
346 </9pt>
347 <*10pt>
348 \newcommand{\tiny}{\@setfontsize\tiny\@vipt{7}}
349 \newcommand{\scriptsize}{\@setfontsize\scriptsize\@viipt{9}}
350 \newcommand{\large}{\@setfontsize\large\@xipt{12}}
351 \newcommand{\Large}{\@setfontsize\Large\@xipt{14}}
352 \newcommand{\LARGE}{\@setfontsize\LARGE\@xivpt{18}}
353 \newcommand{\huge}{\@setfontsize\huge\@xviipt{22}}
354 \newcommand{\Huge}{\@setfontsize\Huge\@xxvpt{30}}

```

```

355 </10pt>
356 <*11pt>
357 \newcommand{\tiny}{\@setfontsize\tiny\@vipt{7}}
358 \newcommand{\scriptsize}{\@setfontsize\scriptsize\@viiipt{9}}
359 \newcommand{\large}{\@setfontsize\large\@xiipt{14}}
360 \newcommand{\Large}{\@setfontsize\Large\@xivpt{18}}
361 \newcommand{\LARGE}{\@setfontsize\LARGE\@xviipt{22}}
362 \newcommand{\huge}{\@setfontsize\huge\@xxpt{25}}
363 \newcommand{\Huge}{\@setfontsize\Huge\@xxvpt{30}}
364 </11pt>

```

\Gfont Define the font sizes for headings, captions, etc. **\Gfont** is the normal size font
\Tfont for general text. **\Tfont** is for the title of the standard. **\Cfont** is for clause
\Cfont headings. Similarly **\SCfont** and **\SSCfont** are for subheadings. **\Nfont** is for
\SCfont notes, examples, footers, footnotes, copyright. **\Efont** is for code in typewriter
\SSCfont font.

```

\Nfont 365 \newcommand{\Gfont}{\normalsize}
\Efont 366 \newcommand{\Nfont}{\small}
367 \newcommand{\Efont}{\small}
368 <*9pt>
369 %%\newcommand{\Tfont}{\huge}
370 \newcommand{\Tfont}{\@setfontsize\Tfont\@xviipt{22}\bfseries}
371 \newcommand{\Cfont}{\Large\bfseries}
372 \newcommand{\SCfont}{\large\bfseries}
373 \newcommand{\SSCfont}{\normalsize\bfseries}
374
375 </9pt>
376 <*10pt>
377 %%\newcommand{\Tfont}{\huge}
378 \newcommand{\Tfont}{\@setfontsize\Tfont\@xviipt{22}\bfseries}
379 \newcommand{\Cfont}{\Large\bfseries}
380 \newcommand{\SCfont}{\large\bfseries}
381 \newcommand{\SSCfont}{\normalsize\bfseries}
382
383 </10pt>
384 <*11pt>
385 %%\newcommand{\Tfont}{\LARGE}
386 \newcommand{\Tfont}{\LARGE\bfseries}
387 \newcommand{\Cfont}{\Large\bfseries}
388 \newcommand{\SCfont}{\large\bfseries}
389 \newcommand{\SSCfont}{\normalsize\bfseries}
390
391 </11pt>
392

```

\beforecskip We define skips for before and after headings. ISO wants two blank lines before a
\aftercskip clause and one afterwards. For lower level sectioning the spacing is one blank line
\beforesscskip before and one after.
\afterscskip Remember that \LaTeX automatically adds **\parskip** before and after headings.
\beforesscskip
\aftersscskip


```

393 \newlength{\beforecskip}
394 \setlength{\beforecskip}{\@smidgeon}
395 \addtolength{\beforecskip}{2\@onelineskip}
396 \addtolength{\beforecskip}{-\parskip}
397 \newlength{\aftercskip}
398 \setlength{\aftercskip}{\@smidgeon}
399 \addtolength{\aftercskip}{\@onelineskip}
400 \addtolength{\aftercskip}{-\parskip}
401 \newlength{\beforecscskip}
402 \setlength{\beforecscskip}{\@smidgeon}
403 \addtolength{\beforecscskip}{\@onelineskip}
404 \addtolength{\beforecscskip}{-\parskip}
405 \newlength{\aftercscskip}
406 \setlength{\aftercscskip}{\@smidgeon}
407 \addtolength{\aftercscskip}{\@onelineskip}
408 \addtolength{\aftercscskip}{-\parskip}
409 \newlength{\beforesscskip}
410 \setlength{\beforesscskip}{\@smidgeon}
411 \addtolength{\beforesscskip}{\@onelineskip}
412 \addtolength{\beforesscskip}{-\parskip}
413 \newlength{\aftersscskip}
414 \setlength{\aftersscskip}{\@smidgeon}
415 \addtolength{\aftersscskip}{\@onelineskip}
416 \addtolength{\aftersscskip}{-\parskip}
417

418 </9pt | 10pt | 11pt>
419 <*iso>

```

`\captionsize` This internal command holds the font size for captions. Its value depends on the `uglycaption` option.

```
420 \newcommand{\captionsize}{\normalsize}
```

8.2 Paragraphing

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

`\normallineskip`

```
421 \setlength\lineskip{1\p@}
422 \setlength\normallineskip{1\p@}
```

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

```
423 \renewcommand{\baselinestretch}{}
```

`\parindent` `\parskip` gives extra vertical space between paragraphs and `\parindent` is the width of the paragraph indentation. (`\parskip` is defined in the `.clo` file.)

```
424 \setlength\parindent{\z@}
```

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

```
425 \@lowpenalty 51
426 \@medpenalty 151
427 \@highpenalty 301
```

`\clubpenalty` These penalties are used to discourage club and widow lines. The default values
`\widowpenalty` are 150 each, but we want stronger discouragement.

```
428 \clubpenalty 1000
429 \widowpenalty 1000
```

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

```
430 % \displaywidowpenalty 50
431 % \predisplaypenalty 10000
432 % \postdisplaypenalty 0
```

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

```
433 % \interlinepenalty 0
```

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

```
434 % \brokenpenalty 100
```

8.3 Page Layout

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

The ISO layout on A4 paper (297 by 210 mm) is 25mm sidemargins (make that 25.4mm for simplicity) 12mm above and below the header and footer, at least one blank line between the typeblock and headers/footers. This leads to `\dotsidemargin = 0`, and `\textwidth = 159.2mm = 160mm` for convenience, and `\topmargin = -13.5mm`.

Make `\headheight`, `\headskip` and `footheight` each be 12pt, then `\footskip = 24pt`. The total height of the typeblock is then 256mm; subtracting the `\topskip` (say 12pt = 4mm) gives `\textheight = 252mm`.

8.3.1 Vertical spacing

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

```
435 \setlength\headheight{12\p@}
436 \setlength\headsep{12\p@}
437 </iso>
```

```

438 <9pt>\setlength\topskip{12\p@}
439 <10pt>\setlength\topskip{12\p@}
440 <11pt>\setlength\topskip{12\p@}
441 <*iso>

```

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

```
442 \setlength\footskip{24\p@}
```

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

```
443 \setlength\maxdepth{.5\topskip}
444 \setlength\@maxdepth\maxdepth
```

8.3.2 The dimension of text

`\textwidth` The width and height of the text which are fixed in this class. Also, the gutter width when in two column mode.

```

\columnsep 445 \setlength\textwidth{160mm}
446 %%\setlength\textheight{221.5mm}
447 \setlength\textheight{252mm}
448 \setlength\columnsep{10mm}

```

8.3.3 Margins

`\topmargin` The margins are fixed in this class.

```

\oddsidemargin 449 %%\setlength\topmargin{0mm}
\evensidemargin 450 \setlength\topmargin{-13.5mm}
\marginparwidth 451 \setlength\oddsidemargin{0mm}
\marginparsep 452 \setlength\evensidemargin{0mm}
\marginparpush 453 \setlength\marginparwidth{0pt}
454 \setlength\marginparsep{0pt}
455 \setlength\marginparpush{3mm}

```

However, some of the options can change these values. The `draft` option allows marginal notes.

```

456 \ifdr@ftd@c
457 \setlength\marginparwidth{20mm}
458 \setlength\marginparsep{0.5mm}
459 \fi

```

The `letterpaper` (279 by 216 mm) option rearranges the text block on the page, trying to center it horizontally.

```

460 \if@us
461 %% \setlength\topmargin{-9.4mm}

```

```

462 %%% \setlength\oddsidemargin{1.55mm}
463 %%% \setlength\evensidemargin{1.55mm}
464 \addtolength{\topmargin}{-9mm}
465 \setlength\oddsidemargin{2mm}
466 \setlength\evensidemargin{2mm}
467 \typeout{ }
468 \typeout{*****}
469 \typeout{* Warning: You have used the letterpage option. *****}
470 \typeout{* This will not be acceptable as ISO camera ready copy. *}
471 \typeout{*****}
472 \typeout{ }
473 \fi

```

8.3.4 Footnotes

`\footnotesep` `\footnotesep` is the height of the strut placed at the beginning of every footnote.

```
474 \setlength\footnotesep{12\p@}
```

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

```
475 \setlength{\skip\footins}{6\p@ \@plus 2\p@ \@minus 2\p@}
```

8.3.5 Float placement parameters

All float parameters are given default values in the L^AT_EX₂e 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

`\c@topnumber` The *topnumber* counter holds the maximum number of floats that can appear on the top of a text page (classically 2)

```
476 \setcounter{topnumber}{2}
```

`\topfraction` This indicates the maximum part of a text page that can be occupied by floats at the top (classically 0.7).

```
477 \renewcommand{\topfraction}{.8}
```

`\c@bottomnumber` The *bottomnumber* counter holds the maximum number of floats that can appear on the bottom of a text page (classically 1).

```
478 \setcounter{bottomnumber}{2}
```

`\bottomfraction` This indicates the maximum part of a text page that can be occupied by floats at the bottom (classically 0.3).

```
479 \renewcommand{\bottomfraction}{.5}
```

`\c@totalnumber` This indicates the maximum number of floats that can appear on any text page (classically 3).

```
480 \setcounter{totalnumber}{4}
```

`\textfraction` This indicates the minimum part of a text page that has to be occupied by text (classically 0.2).
481 `\renewcommand{\textfraction}{.1}`

`\floatpagefraction` This indicates the minimum part of a page that has to be occupied by floating objects before a ‘float page’ is produced (classically 0.5).
482 `\renewcommand{\floatpagefraction}{.7}`

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

`\dbltopfraction` This indicates the maximum part of a two column text page that can be occupied by two column floats at the top (classically 0.7).
484 `\renewcommand{\dbltopfraction}{.8}`

`\dblfloatpagefraction` This indicates the minimum part of a page that has to be occupied by two column wide floating objects before a ‘float page’ is produced (classically 0.5).
485 `\renewcommand{\dblfloatpagefraction}{.7}`

Floats on a text page

`\floatsep` When a floating object is placed on a page with text, these parameters control the separation between the float and the other objects on the page. These parameters
`\textfloatsep` are used for both one-column mode and single-column floats in two-column mode.
`\intextsep` `\floatsep` is the space between adjacent floats that are moved to the top or bottom of the text page.
`\textfloatsep` is the space between the main text and floats at the top or bottom of the page.
`\intextsep` is the space between in-text floats and the text.

486 `\setlength\floatsep {12\p@ \@plus 2\p@ \@minus 2\p@}`
487 `\setlength\textfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}`
488 `\setlength\intextsep {12\p@ \@plus 2\p@ \@minus 2\p@}`

`\dblfloatsep` When floating objects that span the whole `\textwidth` are placed on a text page
`\dbltextfloatsep` and L^AT_EX is in twocolumn mode the separation between the float and the text is controlled by `\dblfloatsep` and `\dbltextfloatsep`.
`\dblfloatsep` is the space between adjacent floats that are moved to the top or bottom of the text page.
`\dbltextfloatsep` is the space between the main text and floats at the top or bottom of the page.

489 `\setlength\dblfloatsep {12\p@ \@plus 2\p@ \@minus 2\p@}`
490 `\setlength\dbltextfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}`

Floats on their own page or column

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

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

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

```
491 \setlength\@fptop{0\p@ \@plus 1fil}
492 \setlength\@fpsep{8\p@ \@plus 2fil}
493 \setlength\@fpbot{0\p@ \@plus 1fil}
```

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

```
\@dblfpsep 494 \setlength\@dblftop{0\p@ \@plus 1fil}
\@dblfpbot 495 \setlength\@dblfpsep{8\p@ \@plus 2fil}
496 \setlength\@dblfpbot{0\p@ \@plus 1fil}
```

8.4 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.

`\@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`.

8.4.1 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.

```
497 %%\mark{%-}% % Initializes TeX's marks <--- can vanish
```

`\standard` `\yearofedition` `\languageofedition` These commands are to be used in the document preamble. They are used to specify the identification of the standard, the year of the standard and the language of the standard. For example, for a DIS printed in 1995 in English:

```
\standard{ISO/DIS 10303-321}
\yearofedition{1995}
\languageofedition{(E)}
```

`\thestandard` `\thesyear` `\theslanguage` `\@runninghead` `\thestandard` and `\thesyear` hold the number and year of the standard being documented. `\theslanguage` holds the ISO identification of the publication language; note that this must include parentheses around the code letter.

```
498 \gdef\thestandard{}
499 \gdef\thesyear{}
500 \gdef\theslanguage{}
501 \def\standard#1{\gdef\thestandard{#1}}
502 \def\yearofedition#1{\gdef\thesyear{#1}}
503 \def\languageofedition#1{\gdef\theslanguage{#1}}
```

`\@runninghead` contains the document identification text for the running head. Its value depends on the `otherdoc` option.

```
504 \ifotherdoc
```

This is not intended to be a standard, so just use `\thestandard` text.

```
505 \newcommand{\@runninghead}{\thestandard}
506 \else
```

It either is, or is intended to become, a standard, ‘so the year and language are required as well; note the colon.

```
507 \newcommand{\@runninghead}{\thestandard:\thesyear\theslanguage}
508 \fi
509
```

`\copyrighthead` `\copyrighthead` contains the text for a copyright mark in a heading. However, it should be blank if the document is not copyrighted.

```
510 \newcommand{\copyrighthead}{\ifc@pyright
511 {\mbox{\copyright \textsc{\copyrightname} \thesyear} --- All rights reserved}}
```

```

512 \else
513   \mbox{}
514 \fi}
515

```

`\extrahead` `\extrahead` puts its contents into the page header (e.g., a document number). Use it in the preamble as `\renewcommand{\extrahead}{N5496}`.

```

516 \newcommand{\extrahead}{\mbox{}}
517

```

8.4.2 Defining the page styles

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

`\ps@headings` *headings* is the typical pagestyle throughout the document. The header contains the identification of the standard. The footer has the page number at the outer edge and a copyright notice at the inner.

```

518 \newcommand{\ps@headings}{%
519   \def\@oddhead{\bfseries\extrahead\hfil\@runninghead}%
520   \def\@evenhead{\bfseries\@runninghead\hfil\extrahead}%
521   \def\@oddfoot{\copyrighthead\hfil\thepage}%
522   \def\@evenfoot{\thepage\hfil\copyrighthead}}

```

`\ps@startpage` The *startpage* page style is similar to *headings* but without a copyright notice.

```

523 \newcommand{\ps@startpage}{%
524   \def\@oddhead{\bfseries\extrahead\hfil\@runninghead}%
525   \def\@evenhead{\bfseries\@runninghead\hfil\extrahead}%
526   \def\@oddfoot{\hfil\thepage}%
527   \def\@evenfoot{\thepage\hfil}}

```

`\ps@nohead` Pagestyle *nohead* has no headers or footers.

```

528 \newcommand{\ps@nohead}{%
529   \def\@oddhead{}%
530   \def\@evenhead{}%
531   \def\@oddfoot{}%
532   \def\@evenfoot{}}

```

`\rectoisotitlehead` *isotitlehead* is a special pagestyle for the title page of a standard. `\rectoisotitlehead` and `\versoisotitlehead` contain the relevant texts.

```

\ps@isotitlehead 533 \newcommand{\rectoisotitlehead}{%
534   \fillline\vspace{0.1\baselineskip}\linebreak%
535   {\bfseries \uppercase{ISname}}
536 %%   \mbox{\ifc@pyright\space\copyright {\scshape \copyrightname}\else
537 %%     \space{\scshape (\copyrightname)}\fi}
538   \hfil {\bfseries \@runninghead}%
539   \vspace{-0.5\baselineskip}\linebreak\fillline}

```



```

540 \newcommand{\versoistitlehead}{%
541   \fillline\vspace{0.1\baselineskip}\linebreak%
542   {\bfseries \@runninghead} \hfil
543   {\bfseries \uppercase{\ISname}}
544 %%   \mbox{\ifc@pyright\space\copyright {\scshape \copyrightname}\else
545 %%     \space{\scshape (\copyrightname)}\fi}
546   \vspace{-0.5\baselineskip}\linebreak\fillline}

547 \def\ps@isotitlehead{%
548   \def\@oddhead{\parbox{\textwidth}{\protect\rectoisotitlehead}}%
549   \def\@evenhead{\parbox{\textwidth}{\protect\versoistitlehead}}%
550 %%   \def\@oddfoot{\hfil\thepage}%
551 %%   \def\@evenfoot{\thepage\hfil}}
552 \def\@oddfoot{\copyrighthead\hfil\thepage}%
553 \def\@evenfoot{\thepage\hfil\copyrighthead}

```

9 Document Markup

9.1 The title

In this class the `\title` command is somewhat different to that in the standard classes.

```

\title The command \title{\intro}{\main}{\comp} produces a macro \thetitle
\thetitle which is used when generating the first normative clause.
\introelement First define a default \thetitle.
\mainelement 554 \gdef\thetitle{}
\compelement Define the elements to be used in the title.

555 \newcommand{\introelement}[1]{\if\isoemptystring{#1}\else {#1 ---\newline}\fi}
556 \newcommand{\mainelement}[1]{#1}
557 \newcommand{\compelement}[1]{\if\isoemptystring{#1}\else { --- \newline #1}\fi}

The \title command starts a new recto page with arabic numbering and ini-
tialises the counters. It also uses the isotitlehead.

558 \renewcommand{\title}[3]{%
559   \cleardoublepage\pagenumbering{arabic}%
560   \setcounter{clause}{0}%
561   \ifotherdoc \else %
562     \protect\thispagestyle{isotitlehead}
563   \fi
564   \gdef\thetitle{{\Tfont \introelement{#1} %
565     \mainelement{#2} %
566     \compelement{#3}\par}}
567   \if@twocolumn
568     \twocolumn[\vspace*{2\baselineskip}\vbox to 35mm{\thetitle}]
569   \else
570     \vspace*{2\baselineskip}\vbox to 35mm{\thetitle}
571   \fi}

```

9.2 The cover

ISO will produce the cover (pages 1 and 2) for any documents they publish. It can be useful for editors to be able to provide their own, informal, cover sheet.

cover The `cover` environment is for typesetting an informal cover sheet. there is no restriction on what can go into it, except that if used it must be the first element in the document and the contents must not exceed a single page.

```
572 \newenvironment{cover}{%
573   \if@twocolumn
574     \@restonecoltrue\onecolumn
575   \else
576     \@restonecolfalse
577   \fi
578   \setcounter{page}{1} \pagenumbering{roman}
579   \thispagestyle{empty}}{%
```

A copyright notice has to go at the foot of the second page.

```
580 %% \clearpage
581 \thispagestyle{startpage}
582 \mbox{}
583 \if@pyright\@copyrighttext\fi
584 \newpage
585 \if@restonecol\twocolumn\fi}
586
```

9.3 Clauses

9.3.1 Building blocks

The definitions in this part of a class file usually make use of two internal macros, `\@startsection` and `\secdef`. To understand what is going on here, we describe their syntax.

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

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

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

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

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

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

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

⟨*afterskip*⟩ If positive, this gives the skip to leave below the heading, else it gives the skip to leave to the right of a run-in heading.

⟨*style*⟩ Commands to set the style of the heading.

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

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

⟨*heading*⟩ The heading of the new section.

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

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

```
\secdef⟨unstarcmds⟩⟨starcmds⟩
```

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

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

You can use `\secdef` as follows:

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

9.3.2 Overview

ISO terminology uses ‘clause’ instead of the typical terms for subdivisions in a document, although they do use the term ‘section’. Accordingly, we have defined new terms for the document sectioning commands. We also use the shorthand ‘ss’ for ‘subsub’, and so on.

<u>L^AT_EX</u>	<u>ISO</u>	<u>level</u>
chapter	clause, annex	1
section	sclause	2
subsection	ssclause	3
subsubsection	sssclause	4
paragraph	ssssclause	5
subparagraph	sssssclause	6

We also provide ‘annex’ commands, which are equivalent to a ‘clause’ command.

9.3.3 Hyperref ToC levels

`\toclevel@clause` The hyperref package needs to know ToC entry levels.
`\toclevel@sclause` 587 `\newcommand*{\toclevel@clause}{1}`
`\toclevel@ssclause` 588 `\newcommand*{\toclevel@sclause}{2}`
`\toclevel@sssclause` 589 `\newcommand*{\toclevel@ssclause}{3}`
`\toclevel@ssssclause` 590 `\newcommand*{\toclevel@sssclause}{4}`
`\toclevel@sssssclause` 591 `\newcommand*{\toclevel@ssssclause}{5}`
`\toclevel@annex` 592 `\newcommand*{\toclevel@sssssclause}{6}`
`\toclevel@index` 593 `\newcommand*{\toclevel@annex}{1}`
594 `\newcommand*{\toclevel@index}{1}`
595

9.3.4 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.
596 `\setcounter{secnumdepth}{6}`

The macro
`\newcounter{<newctr>}[<oldctr>]`
defines `<newctr>` to be a counter, which is reset to zero when counter `<oldctr>` is stepped. Counter `<oldctr>` must already be defined.

`\c@annex` These counters are used for the sectioning numbers. Clause and annex are the top
`\c@clause` level document divisions.
`\c@fibicl@use` 597 `\newcounter{annex}`
598 `\newcounter{clause}`
599 `\newcounter{fibicl@use}`

`\c@sclause` The lower level divisions get reset by higher level divisions.
`\c@ssclause` 600 `\newcounter{sclause}[clause]`
`\c@sssclause` 601 `\newcounter{ssclause}[sclause]`
`\c@sssssclause` 602 `\newcounter{sssclause}[ssclause]`
`\c@sssssclause` 603 `\newcounter{sssssclause}[sssssclause]`
604 `\newcounter{sssssclause}[sssssclause]`

`\c@yextra` We need an extra counter for the hyperref package.
605 `\newcounter{yextra}`
606

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

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

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

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

`\theannex` The top level division numbers.

```
\theclause 607 \renewcommand{\theannex}{\Alph{annex}}
\thefibicl@use 608 \renewcommand{\theclause}{\arabic{clause}}
609 \renewcommand{\thefibicl@use}{\arabic{fibicl@use}}
```

`\thesclause` The lower level division number representations.

```
\thessclause 610 \renewcommand{\thesclause}{\theclause.\arabic{sclause}}
\thesssclause 611 \renewcommand{\thessclause}{\thesclause.\arabic{ssclause}}
\thessssclause 612 \renewcommand{\thessssclause}{\thessclause.\arabic{ssssclause}}
\thessssssclause 613 \renewcommand{\thessssssclause}{\thessssclause.\arabic{sssssclause}}
614 \renewcommand{\thessssssclause}{\thessssssclause.\arabic{ssssssclause}}
615
```

`\theHannex` For hyperref we have to specify a similar set of number representations.

```
\theHclause 616 \newcommand{\theHannex}{\Alph{annex}}
\theHsclause 617 \newcommand{\theHclause}{\arabic{clause}}
\theHsssclause 618 \newcommand{\theHsclause}{\theHclause.\arabic{sclause}}
\theHssssclause 619 \newcommand{\theHsssclause}{\theHsclause.\arabic{ssclause}}
\theHsssssclause 620 \newcommand{\theHssssclause}{\theHsssclause.\arabic{ssssclause}}
\theHssssssclause 621 \newcommand{\theHssssssclause}{\theHssssclause.\arabic{sssssclause}}
622 \newcommand{\theHssssssclause}{\theHssssssclause.\arabic{ssssssclause}}
623
```

9.3.5 Clauses

`\zerocounters` At the start of each document division counters like for notes and examples are zeroed.

```
624 \newcommand{\zerocounters}{%
625 \setcounter{note}{0}\setcounter{example}{0}}
```

`\@hangfrom` Multiline clause headings are flushleft (block paragraph style).

```
626 \renewcommand{\@hangfrom}[1]{#1}
627
```

`\clause` The command to start a new clause.

```
628 \newcommand{\clause}{\zerocounters
629 \addtocounter{clause}{1}
630 \typeout{Clause: \theclause}
631 \addtocounter{clause}{-1}
632 \tocskip{\tocentryskip}
633 \@startsection{clause}{1}%
634 {\z@}%
635 {\beforecskip}%
```

```

636     {\aftercskip}%
637 %%     {\raggedright\Cfont\bfseries}}
638     {\raggedright\Cfont}}

```

`\fibicl@use` Document divisions like the Foreword and the Bibliography are effectively unnumbered clauses, but which appear in the ToC. In order to ease support for the `tex4ht` package, the `\fibicl@use` command is defined, but should only be used in its starred form.

```

639 \newcommand{\fibicl@use}{%
640   \@startsection{fibicl@use}{1}%
641   {\z@}%
642   {\beforecskip}%
643   {\aftercskip}%
644 %%     {\raggedright\Cfont\bfseries}}
645   {\raggedright\Cfont}}

```

9.3.6 Lower level headings

These commands all make use of `\@startsection`. They also reinitialize the note and example counters.

```

\sclause
\ssclause 646 \newcommand{\sclause}{\zerocounters
\sssclause 647   \@startsection{sclause}{2}%
\sssssclause 648     {\z@}%
\sssssclause 649     {\beforescskip}%
650     {\afterscskip}%
651     {\raggedright\SCfont}}

652 \newcommand{\ssclause}{\zerocounters
653   \@startsection{ssclause}{3}%
654   {\z@}%
655   {\beforesscscskip}%
656   {\afterssscscskip}%
657   {\raggedright\SSCfont}}

658 \newcommand{\sssclause}{\zerocounters
659   \@startsection{sssclause}{4}%
660   {\z@}%
661   {\beforesscscskip}%
662   {\afterssscscskip}%
663   {\raggedright\SSCfont}}

664 \newcommand{\sssssclause}{\zerocounters
665   \@startsection{sssssclause}{5}%
666   {\z@}%
667   {\beforesscscskip}%
668   {\afterssscscskip}%
669   {\raggedright\SSCfont}}

670 \newcommand{\ssssscclause}{\zerocounters
671   \@startsection{ssssscclause}{6}%

```

```

672      {\z@}%
673      {\beforesscskip}%
674      {\aftersscskip}%
675      {\raggedright\SSCfont}}

```

Preloaded definitions.

```

676 \def\clausemark#1{}
677 \def\sclausemark#1{}
678 \def\ssclausemark#1{}
679 \def\ssscclausemark#1{}
680 \def\sssscclausemark#1{}
681 \def\ssssclausemark#1{}

```

9.3.7 Annexes

`\init@nnex` As an annex command has to do quite a lot, we define the internal `\init@nnex` command as a worker. It has to:

- clear the page;
- reset the table and figure counters to zero;
- redefine the `\thefigure` and the `\thetable` to precede them with the annex letter;
- reset the `sclause` counter to zero;
- test for annexes I and O since these are not allowed by ISO.

Use as: `\@annex{<title>}{<typeset body kind>}{<typeset toc kind>}`

```

682 \newcommand{\init@nnex}{%
683   \clearpage

```

Reset the counters and test for illegal annex numbering

```

684   \setcounter{table}{0}
685   \setcounter{figure}{0}
686   \setcounter{sclause}{0}
687   \zerocounters
688   \refstepcounter{annex}
689   \ifnum 9=\value{annex} \refstepcounter{annex}\fi
690   \ifnum 15=\value{annex} \refstepcounter{annex}\fi

```

Reset the numbering scheme, but not just when first called.

```

691 %%%   \ifnum 1=\value{annex}
692       \renewcommand{\clause}{%
693         \ClassWarning{iso}{%
694           \protect\clause\space commands are not allowed after starting Annexes}{%
695             Type \space <return> to proceed and I'll ignore your \protect\clause.}}
696       \renewcommand{\thesclause}{\theannex.\arabic{sclause}}
697       \renewcommand{\thetable}{\theannex.\arabic{table}}
698       \renewcommand{\thefigure}{\theannex.\arabic{figure}}

```

```

699     \renewcommand{\theHsclause}{\theHannex.\arabic{sclause}}
700     \ifisohyper
701         \renewcommand{\theHtable}{\theHannex.\arabic{table}}
702         \renewcommand{\theHfigure}{\theHannex.\arabic{figure}}
703     \fi
704 %%%     \fi

```

Prevent floats appearing before the title.

```

705     \global\@topnum\z@
706     \@afterindentfalse
707 }

```

`\makepreannexhead` Command to typeset the first part of an annex title. Use as `\makepreannexhead{<type>}`.

```

708 \newcommand{\makepreannexhead}[1]{%
709   \begin{center}
710     {\Cfont \annexname~\theannex}\Large #1}
711   \end{center}
712 }

```

`\makeannexhead` Typeset the title name of an annex. Use as `\makeannexhead{<title>}`.

```

713 \newcommand{\makeannexhead}[1]{%
714   \centerline{\Cfont #1}
715   \vskip 0.5\baselineskip
716 }

```

`\addannextotoc` Add an annex title to the ToC. Use as `\addannextotoc{<type>}{<title>}`.

```

717 \newcommand{\addannextotoc}[2]{%
718   \tocskip{\tocentryskip}
719   \addcontentsline{toc}{annex}{\ifnum2>\c@secnumdepth \else
720     \protect\numberline{\annexname~\theannex\space #1}\fi #2}%
721 }

```

`\@infannex` Three kinds of annexes are provided. `\infannex` is an informative annex and `\normannex` is a normative annex. Just to round things out, `\repannex` is neither of these.

`\normannex` All the titles are centered, together with the kind of annex.

`\@repannex` Here are the informative annex commands.

```

\repannex 722 \newcommand{\@infannex}[1]{%
723   \makepreannexhead{(\informativename)}
724   \makeannexhead{#1}
725   \addannextotoc{(\informativename)}{#1}
726 }
727 \newcommand{\infannex}[1]{%
728   \init@nnex
729   \@infannex{#1}
730   \typeout{Informative annex: #1}
731 }

```

Here are the normative annex commands.

```

732 \newcommand{\@normannex}[1]{%

```



```

733 \makepreannexhead{(\normativename)}
734 \makeannexhead{#1}
735 \addannextotoc{(\normativename)}{#1}
736 }
737 \newcommand{\normannex}[1]{%
738 \init@nnex
739 \@normannex{#1}
740 \typeout{Normative annex: #1}
741 }

```

Here are the other annex commands.

```

742 \newcommand{\@repannex}[1]{%
743 \makepreannexhead{}
744 \makeannexhead{#1}
745 \addannextotoc{}{#1}
746 }
747 \newcommand{\repannex}[1]{%
748 \init@nnex
749 \@repannex{#1}
750 \typeout{Annex: #1}
751 }

```

9.4 Lists

9.4.1 General List Parameters

The following commands are used to set the default values for the list environment's parameters. See the L^AT_EX manual for an explanation of the meanings of the parameters. Defaults for the list environment are set as follows. First, `\rightmargin`, `\listparindent` and `\itemindent` are set to 0pt. Then, for a Kth level list, the command `\@listK` is called, where 'K' denotes 'i', 'ii', ... , 'vi'. (I.e., `\@listiii` is called for a third-level list.) By convention, `\@listK` should set `\leftmargin` to `\leftmarginK`.

```

\leftmargin For efficiency, level-one list's values are defined at top level, and \@listi is defined
\leftmargin to set only \leftmargin.
\leftmarginii 752 \setlength{\leftmarginii}{2em}
\leftmarginiii The value of \leftmargin has to be set at this outer level.
\leftmarginiv 753 \leftmargin \leftmargini
\leftmarginvi For ISO, all lists are indented the same amount.
754 \setlength{\leftmarginii}{\leftmargini}
755 \setlength{\leftmarginiii}{\leftmargini}
756 \setlength{\leftmarginiv}{\leftmargini}
757 \setlength{\leftmarginv}{\leftmargini}
758 \setlength{\leftmarginvi}{\leftmargini}

\itemindent Here we set the \itemindent which is the extra indentation before a label.
759 \setlength{\itemindent}{\z@}

```

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

```

760 \setlength{\labelsep}{0.5em}
761 \setlength{\labelwidth}{\leftmargini}
762 \addtolength{\labelwidth}{-\labelsep}

```

`\partopsep` When the user leaves a blank line before the environment an extra vertical space
of `\partopsep` is inserted, in addition to `\parskip` and `\topsep`.

```

763 </iso>
764 <*9pt | 10pt | 11pt>
765 \setlength\partopsep{2\p@ \@plus 1\p@ \@minus 1\p@}
766 </9pt | 10pt | 11pt>
767 <*iso>

```

`\@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.

```

768 \@beginparpenalty -\@lowpenalty
769 \@endparpenalty -\@lowpenalty
770 \@itempenalty -\@lowpenalty

```

`\@setitemparams` Lists may be called within other list environments with differing layouts. We use
a routine to set the layout for `itemize` and `enumerate` lists.

```

771 </iso>
772 <*9pt | 10pt | 11pt>
773 \newcommand{\@setitemparams}{%
774 \setlength{\labelsep}{0.5em}
775 \setlength{\labelwidth}{\leftmargini}
776 \addtolength{\labelwidth}{-\labelsep}
777 \setlength{\itemindent}{\z@}
778 \setlength{\parsep}{\baselineskip}
779 \topsep \z@ \@plus1\p@ \@minus1\p@
780 \itemsep \z@ \@plus1\p@ \@minus1\p@}

```

`\@listI` `\@listI` defines top level and `\@listi` values of `\leftmargin`, `\parsep`, `\topsep`,
`\@listi` and `\itemsep`

```

781 \def\@listi{\leftmargin\leftmargini
782 %%% \itemindent\labelsep
783 %% \itemindent\z@
784 %% \parsep\baselineskip
785 %% \topsep 0\p@ \@plus1\p@ \@minus1\p@
786 %% \itemsep0\p@ \@plus1\p@ \@minus1\p@}
787 \@setitemparams}
788 \let\@listI\@listi

```

We have to initialise these parameters.

```

789 \@listi

```

```

\@listii Here are the same macros for the lower level lists.
\@listiii 790 \def\@listii {\leftmargin\leftmarginii
\@listiv 791 %%% \itemindent\labelsep}
\@listv 792 %% \itemindent\z@
\@listvi 793 \@setitemparams
794 }
795 \def\@listiii{\leftmargin\leftmarginiii
796 %%% \itemindent\labelsep}
797 %% \itemindent\z@
798 \@setitemparams
799 }
800 \def\@listiv {\leftmargin\leftmarginiv
801 %%% \itemindent\labelsep}
802 %% \itemindent\z@
803 \@setitemparams
804 }
805 \def\@listv {\leftmargin\leftmarginv
806 %%% \itemindent\labelsep}
807 %% \itemindent\z@
808 \@setitemparams
809 }
810 \def\@listvi {\leftmargin\leftmarginvi
811 %%% \itemindent\labelsep}
812 %% \itemindent\z@
813 \@setitemparams
814 }
815 </9pt | 10pt | 11pt>
816 <*iso>

```

9.4.2 Enumerate

ISO only requires two levels of enumeration labelled ‘a’) and ‘1)’. We include a third level and fourth labelled ‘i’) and ‘A)’, just in case. ISO has printed ISO 10303:1994 which includes all three levels defined here. The enumerate environment uses four counters: *enumi*, *enumii*, *enumiii* and *enumiv*, where *enumN* controls the numbering of the Nth level enumeration.

```

\theenumi The counters are already defined in latex.dtx, but their representation is changed
\theenumii here.
\theenumiii 817 \renewcommand{\theenumi}{\alph{enumi}}
\theenumiv 818 \renewcommand{\theenumii}{\arabic{enumii}}
819 \renewcommand{\theenumiii}{\roman{enumiii}}
820 \renewcommand{\theenumiv}{\Roman{enumiv}}

\labelenumi The label for each item is generated by the commands
\labelenumii \labelenumi ... \labelenumiv.
\labelenumiii 821 \newcommand{\labelenumi}{\theenumi}
\labelenumiv 822 \newcommand{\labelenumii}{\theenumii}
823 \newcommand{\labelenumiii}{\theenumiii}

```

```

824 \newcommand{\labelenumiv}{\theenumiv}}

\p@enumii The expansion of \p@enumN\theenumN defines the output of a \ref command
\p@enumiii when referencing an item of the Nth level of an enumerated list.
\p@enumiv 825 \renewcommand{\p@enumii}{\theenumi}
826 \renewcommand{\p@enumiii}{\p@enumii\theenumii}
827 \renewcommand{\p@enumiv}{\p@enumiii\theenumiii}

```

enumerate We modify the default `enumerate` environment to make labels flush left in the label box.

```

828 \def\enumerate{%
829   \ifnum \@enumdepth >\thr@@\toodeep\else
830     \advance\@enumdepth\@ne
831     \edef\@enumctr{enum\romannumeral\the\@enumdepth}%
832
833     \expandafter
834     \list
835     \csname label\@enumctr\endcsname
836     {\usecounter\@enumctr\def\makelabel##1{##1\hfill}}%
837   \fi}
838 \let\endenumerate =\endlist

```

9.4.3 Itemize

ISO only requires one level labelled with either a long dash or a bullet. We provide four levels, three of which have been used in ISO 10303:1994.

```

\labelitemi Itemization is controlled by the commands: \labelitemi, \labelitemii, etc.,
\labelitemii which define the labels of the various itemization levels: the symbols used are
\labelitemiii bold em-dash, bullet, asterisk, and centered period.
\labelitemiiii
839 \newcommand{\labelitemi}{\normalfont\bfseries \textemdash\hfill}
840 %\newcommand{\labelitemii}{\textbullet\hfill}
841 %\newcommand{\labelitemiii}{\textasteriskcentered}
842 %\newcommand{\labelitemiv}{\textperiodcentered}
843 \newcommand{\labelitemii}{\labelitemi}
844 \newcommand{\labelitemiii}{\labelitemi}
845 \newcommand{\labelitemiv}{\labelitemi}

```

itemize We modify the default `itemize` environment to make the labels flush left in the label box.

```

846 \def\itemize{%
847   \ifnum \@itemdepth >\thr@@\toodeep\else
848     \advance\@itemdepth\@ne
849     \edef\@itemitem{labelitem\romannumeral\the\@itemdepth}%
850
851     \expandafter
852     \list
853     \csname\@itemitem\endcsname

```

```

854      {\def\makelabel##1{##1\hfill}}%
855  \fi}
856 \let\enditemize =\endlist

```

9.4.4 Description

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

```

857 \newenvironment{description}%
858      {\list{}{\labelwidth\z@ \itemindent 0.5em \labelsep 0.5em
859              \let\makelabel\descriptionlabel}}%
860      {\endlist}

```

`\descriptionlabel` To change the formatting of the label, you must redefine `\descriptionlabel`. Note that the label includes a colon.

```

861 \newcommand*{\descriptionlabel}[1]{\normalfont\bfseries #1:\hfill}

```

9.5 Defining new environments

9.5.1 Quotation

This is not required by ISO, but we leave it in anyway.

quotation The quotation environment is 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.

```

862 \newenvironment{quotation}%
863      {\list{}{\listparindent 1.5em%
864              \itemindent \listparindent
865              \rightmargin \leftmargin
866              \parsep \z@ \@plus\p@}%
867      \item[]}%
868      {\endlist}

```

9.5.2 Quote

This is also not an ISO requirement, but leave it in anyway.

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

```

869 \newenvironment{quote}%
870      {\list{}{\rightmargin\leftmargin}%
871      \item[]}%
872      {\endlist}

```

9.5.3 Theorem

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

9.5.4 Notes

ISO requires that information which is essential to the understanding of a standard but which is not a requirement is to be given in the form of a note. In the Directives edition 2, there were three styles of note:

1. isolated notes which are marked NOTE - 1, NOTE - 2, etc.
2. a local grouping of notes marked
NOTES
1 - ...
2 - ...
3. an isolated note that is not numbered because it is the only one in that (sub-) clause of the document.

The 3rd edition removed the local grouping.

```
\ifinfloat Special consideration has to be given when notes appear within a float.
873 \newif\ifinfloat\infloatfalse

\c@note Define note counters, where the counter note for body notes gets reset within each
\c@floatnote new clause and notes within floats have their own numbering scheme via floatnote.
\thenote 874 \newcounter{note}[clause]
\thefloatnote 875 \renewcommand{\thenote}{\arabic{note}}
876 \newcounter{floatnote}
877 \renewcommand{\thefloatnote}{\arabic{floatnote}}

\theHnote We also need hyperref representations.
\theHfloatnote 878 \newcommand{\theHnote}{\thenote.\arabic{yextra}}
879 \newcommand{\theHfloatnote}{\thefloatnote.\arabic{yextra}}
880

\notetlabel Labeling of notes (and examples).
881 \newcommand{\notetlabel}[1]{\#1\hfill}

notes This environment produced a fixed heading followed by a numbered list. The
environment is defined in terms of a general list.
Use as:

\begin{notes}
\begin{note}Text of first note ... \end{note}
\begin{note}Text of second note ... \end{note}
\end{notes}

With the 3rd edition of the ISO Directives, this has been made a no-op and is
only retained for compatability. The original code was:

\newif\ifinnotes\innotesfalse
\newenvironment{notes}{\list{}}%
```

```

{\ifinfloat \leftmargin 0em \else \leftmargin 2em\fi
\itemindent 0.5em \labelwidth 0em
\labelsep 0.5em \listparindent 0em
\let\makelabel\notelabel}
\innotesttrue
\Nfont\item[\notesname]\mbox{}\nopagebreak[2]}%
{\innotesfalse\endlist}

```

`\@setnoteparams` Because notes, and examples, have the same basic layout we use a routine to set the various parameters.

```

882 \newcommand{\@setnoteparams}{%
883 \setlength{\partopsep}{\z@}
884 \setlength{\topsep}{\z@}
885 \setlength{\labelsep}{1em}
886 \setlength{\itemindent}{\labelsep}
887 \setlength{\labelwidth}{\z@}
888 \setlength{\listparindent}{\z@}
889 \setlength{\leftmargin}{\z@} % added in v2.3
890 }

```

`anote` An isolated un-numbered note.

```

891 \newenvironment{anote}{\list{}{%
892 %% \ifinfloat \setlength{\leftmargin}{\z@} \else
893 %% \setlength{\leftmargin}{2em} \fi
894 \@setnoteparams}
895 \Nfont\item[\notename]}%
896 {\endlist}

```

`note` A numbered note.

```

897 \newenvironment{note}{\list{}{%

```

Use the appropriate counter: normally *note* but *floatnote* when in a floating environment.

```

898 \stepcounter{yextra}
899 \ifinfloat
900 \refstepcounter{floatnote}
901 \let\thenote\thefloatnote
902 \else
903 \refstepcounter{note}
904 \fi

```

Originally we adjusted the margins according to whether we were in a notes environment or not.

```

905 %% \ifinfloat \setlength{\leftmargin}{\z@} \else
906 %% \setlength{\leftmargin}{2em} \fi
907 \@setnoteparams}
908 \Nfont\item[\notename~\thenote]}%
909 {\endlist}

```

9.5.5 Examples

ISO Directives part 3 (2nd edition) had no rules on how to display an example, but it did use examples itself; these examples were displayed in a format similar to notes.

We provided two styles of example:

1. isolated examples which are marked EXAMPLE - 1, EXAMPLE - 2, etc.
2. a local grouping of examples marked
EXAMPLES
1 - ...
2 - ...

The 3rd edition of the Directives does specify some options for typesetting examples. A single example in a (sub) clause is preceded by the word 'EXAMPLE'. If there are several examples, then each is numbered (e.g., 'EXAMPLE 3'). It also states that all lines of an example shall be inset from the margin or set in a smaller font, so that its extent can be determined.

For now, we choose both options.

Implementation is very similar to that for notes.

```
\c@example Define example counter. Example numbering is only continuous within a (sub)
\theexample clause (we used to have it continuous throughout the document).
\theHexample 910 \newcounter{example}[clause]
911 \renewcommand{\theexample}{\arabic{example}}
912 \newcommand{\theHexample}{theexample.\arabic{yextra}}
```

examples Originally, this environment produces a fixed heading followed by a numbered list. The environment is defined in terms of a general list.

Use as:

```
\begin{examples}
\begin{example}Text of first ... \end{example}
\begin{example}Text of second ... \end{example}
\end{examples}
```

With the 3rd edition of the ISO Directives the environment has been made a no-op, but is retained for compatibility. The code used to be:

```
\newif\ifinexamples\inexamplesfalse
\newenvironment{examples}{\list{}}%
{\leftmargin 2em
\itemindent 0.5em \labelwidth 0em
\labelsep 0.5em \listparindent 0em
\let\makelabel\notlabel}
\inexamplestrue
\Nfont\item[\examplesname]\mbox{}\nopagebreak[2]}%
{\inexamplesfalse\endlist}
```


anexample An isolated un-numbered example.

```

913 \newenvironment{anexample}{\list{}}{%
914 %% \ifinfloat \setlength{\leftmargin}{\z@} \else
915 %% \setlength{\leftmargin}{2em} \fi
916 \setnoteparams}
917 \Nfont\item[\examplename]}\endlist}

```

example Like the note environment.

```

918 \newenvironment{example}{\list{}}{%
919 \stepcounter{yextra}
920 \refstepcounter{example}
921 %% \ifinfloat \setlength{\leftmargin}{\z@} \else
922 %% \setlength{\leftmargin}{2em} \fi
923 \setnoteparams}
924 \Nfont\item[\examplename~\theexample]}%
925 {\endlist}

```

9.5.6 Listing of references

ISO has three kinds of literature references, broken into two categories. The categories are normative and informative references. Within the normative category, references are to either published or ‘unpublished’ standards (IS or DIS in ISO terminology).

nreferences The **nreferences** environment is for listing normative references. It is implemented as a list.

\nreferencelabel Labelling of normative references.

```

926 \newcommand{\nreferencelabel}[1]{#1, \hfill}

```

Define the environment. It is used as:

```

\begin{nreferences}
\isref{id}{published standard title}
\disref{id}{unpublished standard title}
...
\end{nreferences}

```

```

927 \newenvironment{nreferences}{\list{}}%
928   {\leftmargin 0pt \itemindent 0.5em
929   \labelwidth\z@ \labelsep 0.5em
930   \let\makelabel\nreferencelabel}}%
931   {\endlist}

```

\isref This is a two parameter command for printing a normative reference to a published standard.

```

932 \newcommand{\isref}[2]{\item[#1]{\itshape #2}}

```

`\disref` This is a two parameter command for printing a normative reference to an unpublished standard. ISO requires that each unpublished standard should be footnoted as ‘unpublished’. Awkwardly, only one footnote is permitted. This means we have to fiddle with the footnote counter.

`\ifdis` A flag to denote if there have been any previous disrefs.

```
933 \newif\ifdis\d@isfalse
```

Now define the `\disref` command.

```
934 \newcommand{\disref}[2]{\begingroup
935   \ifdis
```

This is not the first call to `\disref`, so just footnote the entry

```
936     {\item[#1\protect\@footnotemark]{\itshape #2}}
937   \else
```

This is the first call, so we have to make the footnote

```
938     \addtocounter{footnote}{1}
939     \xdef\@thefnmark{\thefootnote}
940     \item[#1\protect\@footnotemark]{\itshape #2}%
941     \footnotetext[\value{footnote}]{\tbpname}
942     \d@istrue
943   \fi
944   \endgroup\d@istrue}
```

`references` The `references` environment is for listing informative references. It is implemented as a list.

`\c@infrefctr` Informative references are labelled with a number enclosed in square brackets.

`\p@infrefctr` In the body of the text, a reference to an informatively listed document `n` has to be printed as `[n]`. Use the standard L^AT_EX `\label` command and the `\bref` command for this.

`\labelinfref`

```
945 \newcounter{infrefctr}
946 \renewcommand{\p@infrefctr}{}
947 \renewcommand{\theinfrefctr}{\arabic{infrefctr}}
948 \newcommand{\labelinfref}{[\arabic{infrefctr}]}
```

Define the environment. It is used as:

```
\begin{references}
\reference{authors}{title}{publisher and date}
....
\end{references}
```

```
949 \newenvironment{references}{\list{\labelinfref}{\usecounter{infrefctr}}
950   \leftmargin Opt \itemindent 0.5em
951   \labelwidth\z@ \labelsep 0.5em}%
952   {\endlist}
```

`\reference` This is a three parameter command for printing an informatively listed reference document.

```
\reference{<authors>}{<title>}{<publisher and date>}
```

```
953 \newcommand{\reference}[3]{\item {#1} {\itshape #2}} {#3}}
```

9.5.7 Listing of definitions

One element of an ISO standard is the listing of definitions of terms.

`olddefinitions` The `olddefinitions` environment is for listing terms which have been defined in some other standard. It is defined in terms of the `itemize` environment.

```
954 \newenvironment{olddefinitions}{%
```

```
955   {\begin{itemize}}%
```

```
956   {\end{itemize}}
```

`\olddefinition` Within an `olddefinitions` environment each term is specified by the `\olddefinition{<phrase>}{<supplement>}` command.

```
957 \newcommand{\olddefinition}[2]{\item #1 #2}
```

`definitions` Terms being defined within the current document are listed within the `definitions` environment. ISO requires that each definition be sequentially numbered within the clause in which it is defined. This numbering is as though the definition formed a sub-clause.

`\c@cl@level` A counter for determining the current sectioning level.

```
958 \newcounter{cl@level}
```

`\@defcl` We use this internally for the `\definition` command. A default definition is supplied here as we are going to renew it, possibly several times.

```
959 \newcommand{\@defcl}[1]{}
```

Now we define the `definitions` environment.

```
960 \newenvironment{definitions}{%
```

First, set the `cl@level` according to the sectioning level within which the environment is called.

```
961   \setcounter{cl@level}{6}
```

```
962   \ifnum\value{sssssclause}=0 \setcounter{cl@level}{5} \fi
```

```
963   \ifnum\value{ssssclause}=0 \setcounter{cl@level}{4} \fi
```

```
964   \ifnum\value{sssclause}=0 \setcounter{cl@level}{3} \fi
```

```
965   \ifnum\value{ssclause}=0 \setcounter{cl@level}{2} \fi
```

```
966   \ifnum\value{sclause}=0 \setcounter{cl@level}{1} \fi
```

```
967   \ifnum\value{clause}=0 \setcounter{cl@level}{0} \fi
```

Now redefine an appropriate (s)clause definition to get a number on one line, followed by the heading on the next line with a bold normal font. A new paragraph is not started after the heading, and there is no entry in the ToC. As this is done within the group automatically set up by the environment, any original definitions will get restored afterwards.

```

968 \ifcase\value{cl@level} % 0, NOT YET IN A CLAUSE
969   \ClassWarning{iso}{Definitions started before the initial clause}
970   \renewcommand{\@defcl}[1]{\setcounter{note}{0}\setcounter{example}{0}
971     \par
972     \addvspace{\beforecskip}
973     \@afterindentfalse
974     \refstepcounter{clause}
975     {\raggedright\bfseries \theclause\ \ ##1\}}
    Do similar things for the other cases.
976 \or % 1, called in a clause
977   \renewcommand{\@defcl}[1]{\setcounter{note}{0}\setcounter{example}{0}
978     \par
979     \addvspace{\beforecskip}
980     \@afterindentfalse
981     \refstepcounter{sclause}
982     {\raggedright\bfseries \thesclause\ \ ##1\}}
983 \or % 2, called in an sclause
984   \renewcommand{\@defcl}[1]{\setcounter{note}{0}\setcounter{example}{0}
985     \par
986     \addvspace{\beforesscskip}
987     \@afterindentfalse
988     \refstepcounter{ssclause}
989     {\raggedright\bfseries \thessclause\ \ ##1\}}
990 \or % 3, called in an ssclause
991   \renewcommand{\@defcl}[1]{\setcounter{note}{0}\setcounter{example}{0}
992     \par
993     \addvspace{\beforesscskip}
994     \@afterindentfalse
995     \refstepcounter{ssssclause}
996     {\raggedright\bfseries \thesssclause\ \ ##1\}}
997 \or % 4, called in an ssssclause
998   \renewcommand{\@defcl}[1]{\setcounter{note}{0}\setcounter{example}{0}
999     \par
1000    \addvspace{\beforesscskip}
1001    \@afterindentfalse
1002    \refstepcounter{sssssclause}
1003    {\raggedright\bfseries \thesssssclause\ \ ##1\}}
1004 \or % 5, called in an sssssclause
1005   \renewcommand{\@defcl}[1]{\setcounter{note}{0}\setcounter{example}{0}
1006     \par
1007     \addvspace{\beforesscskip}
1008     \@afterindentfalse
1009     \refstepcounter{ssssssclause}
1010     {\raggedright\bfseries \thessssssclause\ \ ##1\}}
1011 \else % 5+, called in an sssssclause or lower
1012   \ClassWarning{iso}{Definitions too deeply nested}
1013   \renewcommand{\@defcl}[1]{
1014     \par
1015     \addvspace{\beforesscskip}

```

```

1016     \@afterindentfalse
1017     \refstepcounter{sssssclause}
1018     {\raggedright\bfseries \thesssssclause\ \ ##1\}}
1019 \fi}%
1020 {}

```

\definition Within a `definitions` environment the command `\definition{<phrase>}{<definition text>}` is used to specify and define each term. It uses the sectional heading definition stored in `\@defcl` set up by the environment.

```
1021 \newcommand{\definition}[2]{\@defcl{#1} #2}
```

9.5.8 Listing of symbols and abbreviations

Another possible element in a standard is the listing of symbols and abbreviations. This is similar to the original `definitions` listing, except that terms are not treated as clauses.

```

symbols
\sybollabel 1022 \newcommand{\sybollabel}[1]{#1 \hfill}
1023 \newenvironment{symbols}{\list{}}%
1024     {\itemindent 0em \leftmargin 8em
1025     \labelsep 1em \labelwidth 5em
1026     \let\makelabel\sybollabel}}%
1027     {\endlist}

```

\symboldef Within a `symbols` environment the command `\symboldef{<symbol>}{<meaning>}` is used to specify and explain each symbol or abbreviation.

```
1028 \newcommand{\symboldef}[2]{\item[#1] #2}
```

9.5.9 Listing of scope items

Another possible element in a standard is the listing of items that are within the scope; conversely, listing of items that are out of scope may also be useful.

inscope We define synonyms for the `itemize` list environment, and initiate the lists with some boilerplate. Use as, for example:

```

\begin{inscope}{international standard}
  \item ...
  \item ...
\end{inscope}

```

```

1029 \newenvironment{inscope}[1]{%
1030     \inscopename #1:
1031     \begin{itemize}}%
1032     {\end{itemize}}
1033 \newenvironment{outofscope}[1]{%
1034     \outofscopename #1:
1035     \begin{itemize}}%
1036     {\end{itemize}}

```

9.6 Setting parameters for existing environments

9.6.1 Array and tabular

`\arraycolsep` The columns in an array environment are separated by `2\arraycolsep`.
1037 `\setlength\arraycolsep{4\p@}`

`\tabcolsep` The columns in an tabular environment are separated by `2\tabcolsep`.
1038 `\setlength\tabcolsep{4\p@}`

`\arrayrulewidth` The width of rules in the array and tabular environments is given by `\arrayrulewidth`.
1039 `\setlength\arrayrulewidth{.4\p@}`

`\doublerulesep` The space between adjacent rules in the array and tabular environments is given by `\doublerulesep`.
1040 `\setlength\doublerulesep{2\p@}`

9.6.2 Tabbing

`\tabbingsep` This controls the space that the `\` command puts in. (See L^AT_EX manual for an explanation.)
1041 `\setlength\tabbingsep{\labelsep}`

9.6.3 Minipage

`\@minipagerestore` The macro `\@minipagerestore` is called upon entry to a minipage environment to set up things that are to be handled differently inside a minipage environment. 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.
1042 `\skip\@mpfootins = \skip\footins`

9.6.4 Framed boxes

`\fboxsep` The space left by `\fbox` and `\framebox` between the box and the text in it.

`\fboxrule` The width of the rules in the box made by `\fbox` and `\framebox`.
1043 `\setlength\fboxsep{3\p@}`
1044 `\setlength\fboxrule{.4\p@}`

9.6.5 Equation and eqnarray

`equation` and `eqnarray` counters are not required by ISO, and the equations are to be left-justified. The default is for the left-hand side of equations to be flushleft.

`\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.

```
1045 \renewcommand{\theequation}{\arabic{equation}}
```

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

```
1046 % \setlength\jot{3pt}
```

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

```
1047 % \def\@eqnnum{(\theequation)}
```

9.7 Floating objects

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

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

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

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

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

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

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

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

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

9.7.1 Figure

Here is the implementation of the figure environment.

```
\c@figure First we have to allocate a counter to number the figures. In this class figures are
numbered sequentially.
1048 \newcounter{figure}
1049 \renewcommand{\thefigure}{\@arabic\c@figure}

\fps@figure Here are the parameters for the floating objects of type ‘figure’.
\ftype@figure 1050 \def\fps@figure{tbp}
\ext@figure 1051 \def\ftype@figure{1}
\fnum@figure 1052 \def\ext@figure{lof}
1053 \def\fnum@figure{\figurename~\thefigure}

\iffigs We define a flag to tell whether the document contains any figures. Elsewhere a
flag, \ifinfloat, is defined to tell if we are in a float.
1054 \newif\iffigs\figsfalse

\@initisofig At the start of a figure environment we have to set a flag and do some work to
deal with the ISO requirements for the ToC, and also zero the floatnote counter.
1055 \newcommand{\@initisofig}{%
1056 \iffigs\else\figstrue
1057 \if@filesw \immediate\write\@mainaux{%
1058 \string\gdef\string\setfigs{%
1059 \string\floatlist{\listfigurename}{lof}}
1060 \fi
1061 \fi

Now deal with the possibility that the float may contain notes.
1062 \infloattrue\setcounter{floatnote}{0}
1063 }

figure This is the definition of the actual environment. The form with the * is used for
figure* double column figures.
1064 \newenvironment{figure}{%
1065 \@initisofig
1066 \@float{figure}}%

At the end of the environment we are no longer in a float.
1067 {\end@float\infloatfalse}

The starred version is similar.
1068 \newenvironment{figure*}{%
1069 \@initisofig
1070 \@dblfloat{figure}}%
1071 {\end@dblfloat\infloatfalse}
```


9.7.2 Table

Here is the implementation of the table environment. It is very much the same as the figure environment, the additional complication being that we have to flag that we are in a table, as well as being in a float.

```
\c@table First we have to allocate a counter to number the tables. In this class tables are
          numbered sequentially.
1072 \newcounter{table}
1073 \renewcommand{\thetable}{\@arabic\c@table}

\fps@table Here are the parameters for the floating objects of type ‘table’.
\ftype@table 1074 \def\fps@table{tbp}
\ext@table 1075 \def\ftype@table{2}
\fnum@table 1076 \def\ext@table{lot}
1077 \def\fnum@table{\tablename~\thetable}

\iftabs We define a flag to tell whether the document contains any tables. Elsewhere a
         flag, \ifinfloat, is defined to tell if we are in a float.
1078 \newif\iftabs\tabsfalse

\@initisotab Initial code at the start of a table environment.
1079 \newcommand{\@initisotab}{%
1080   \iftabs\else\tabstrue
1081     \if@filesw \immediate\write\@mainaux{%
1082       \string\gdef\string\settabs{%
1083         \string\floatlist{\listtablename}{lot}}
1084     \fi
1085     \fi
1086     \infloattrue\setcounter{floatnote}{0}
1087 }

table This is the definition of the actual environment. The form with the * is used for
table* double column tables.
1088 \newenvironment{table}{%
1089   \@initisotab
1090   \@float{table}}%
1091 {\end@float\infloatfalse}

The starred version is similar.
1092 \newenvironment{table*}{%
1093   \@initisotab
1094   \@dblfloat{table}}%
1095 {\end@dblfloat\infloatfalse}
```

9.7.3 A bottom float

We define an additional float environment. Unless something additional is done, this will not be listed in the table of contents.

```

\c@bottomfloat First we have to allocate a counter to number the float.
\thebottomfloat 1096 \newcounter{bottomfloat}
                  1097 \renewcommand{\thebottomfloat}{\@arabic\c@bottomfloat}

\fps@bottomfloat Here are the parameters for the floating objects of type ‘bottomfloat’.
\ftype@bottomfloat 1098 \def\fps@bottomfloat{b}
\ext@bottomfloat 1099 \def\ftype@bottomfloat{4}
\fnum@bottomfloat 1100 \def\ext@bottomfloat{lbf}
                  1101 \def\fnum@bottomfloat{\thebottomfloat}

bottomfloat This is the definition of the actual environment. The form with the * is used for
bottomfloat* double column floats.
                  1102 \newenvironment{bottomfloat}%
                  1103             {\@float{bottomfloat}}%
                  1104             {\end@float}
                  1105 \newenvironment{bottomfloat*}%
                  1106             {\@dblfloat{bottomfloat}}%
                  1107             {\end@dblfloat}

```

9.7.4 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.
                  1108 \newlength\abovecaptionskip
                  1109 \newlength\belowcaptionskip
                  1110 \setlength\abovecaptionskip{10\p@}
                  1111 \setlength\belowcaptionskip{10\p@}

                  The definition of this macro is \long in order to allow more than one paragraph
                  in a caption.
                  1112 \long\def\@makecaption#1#2{%
                  1113   \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.
                  1114   \sbox\@tempboxa{\captionsize\bfseries #1 -- #2}}%

                  We can the measure its width. It that is larger than the current \hsize we typeset
                  the caption as a centered paragraph.
                  1115   \ifdim \wd\@tempboxa >\hsize
                  1116     {\centering {\captionsize\bfseries #1 -- #2}\par}

```

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

```
1117 \else
1118   \global \@minipagefalse
1119   \hbox to\hsize{\hfil\box\@tempboxa\hfil}%
1120 \fi
1121 \vskip\belowcaptionskip}
```

`\contcaption` The `\contcaption` command can be used to put a ‘continuation’ caption into a float. It neither increments the float number nor makes any entry in the toc listings.

It is called as `\contcaption{<continued/concluded>}{<optional text>}`

```
1122 \newcommand{\contcaption}{\@contcaption\@capttype}
```

`\@contcaption` This does the work for us.

```
1123 \long\def\@contcaption#1#2{%
1124   \begingroup
1125     \@parboxrestore
1126     \normalsize
1127     \@makecaption{\csname fnum#1\endcsname}{\ignorespaces #2}\par
1128   \endgroup}
```

9.8 Font changing

Here we supply the declarative font changing commands that were common in \LaTeX version 2.09 and earlier. These commands work in text mode *and* in math mode. They are provided for compatibility, but one should start using the `\text...` and `\math...` commands instead. These commands are defined using `\DeclareTextFontCommand`, a command with three arguments: the user command to be defined; \LaTeX commands to execute in text mode and \LaTeX commands to execute in math mode.

`\rm` The commands to change the family. When in compatibility mode we select the `\tt` ‘default’ font first, to get \LaTeX 2.09 behaviour.

```
\sf 1129 \DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
1130 \DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
1131 \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.

```
1132 \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. However, we do warn the user that the selection will not have any effect. One should use `\upshape` to explicitly change back to the upright shape.

```

1133 \DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
1134 \DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
1135 \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’.

```

1136 \DeclareRobustCommand*\cal{\@fontswitch{\relax}{\mathcal}}
1137 \DeclareRobustCommand*\mit{\@fontswitch{\relax}{\mathnormal}}

```

9.9 Urls, etc

ISO uses its own format for typesetting urls. This is implemented here via the `url` package.

`\url` The `\url{<text>}` command is provided by the `url` package. It may be used for typesetting email addresses. The `\isourl{<text>}` command typesets `<text>` in the format required by ISO for an url; that is, the address is underlined and enclosed within (not-underlined) angle brackets.

NOTE: The underlining prohibits linebreaking in the url. I also tried the `ulem` package’s `\uline` command, but this also prevented any linebreaking, so we might as well stick to the \TeX `\underline`.

```

1138 %\newcommand{\isourl}[1]{\texttt{<}\underline{\url{#1}}\texttt{>}}
1139 \newcommand{\isourl}[1]{\texttt{<}\url{#1}\texttt{>}}

```

10 Cross Referencing

10.1 Label referencing

`\aref` Named references to labeled elements. `\bref{<label id>}` is a reference to a labeled informative bibliographic element (similar to the standard \LaTeX `\cite` command). `\cref` The others are to named elements of the document.

```

\aref 1140 \newcommand{\aref}[1]{\annexrefname~\ref{#1}}
\bref 1141 \newcommand{\bref}[1]{[\ref{#1}]}
\cref 1142 \newcommand{\cref}[1]{\clauserefname~\ref{#1}}
\trf 1143 \newcommand{\trf}[1]{\examplerefname~\ref{#1}}
\pref 1144 \newcommand{\pref}[1]{\figurerefname~\ref{#1}}
1145 \newcommand{\nref}[1]{\noterefname~\ref{#1}}
1146 \newcommand{\tref}[1]{\tablerefname~\ref{#1}}
1147 \newcommand{\pref}[1]{\pagerefname~\pageref{#1}}

```

10.2 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⟩ ≤ 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 ≥ \@pnumwidth`

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

```
1148 \newcommand{\@pnumwidth}{1.55em}
```

```
1149 \newcommand{\@tocrmarg}{2.55em}
```

```
1150 \newcommand{\@dotsep}{4.5}
```

`\tocentryskip` We define two lengths and a utility command.

```
\tocbaseline 1151 \newlength{\tocentryskip} \setlength{\tocentryskip}{1em}
```

```
\tocskip 1152 \newlength{\tocbaseline} \setlength{\tocbaseline}{20pt}
```

```
1153 \newcommand{\tocskip}[1]{%
```

```
1154 \addtocontents{toc}{\protect\vspace{#1}}}
```

10.2.1 Table of Contents

`\tableofcontents` This macro is used to request that L^AT_EX produces a table of contents. In this class the tables of contents, figures etc. are always set in single-column style.

```
1155 \newcommand{\tableofcontents}{%
```

```
1156 \if@twocolumn
```

```
1157 \@restonecoltrue\onecolumn
```

```

1158 \else
1159 \@restonecolfalse
1160 \fi

```

If the document is copyrighted, then the copyright notice is placed at the foot of page ii.

```

1161 %%% \setcounter{page}{2}
1162 %%% \thispagestyle{startpage}
1163 %%% \mbox{}
1164 %%% \ifcopyright\@copyrighttext\fi

```

Set the title for the toc, which must start on page (iii) of the document. The actual table of contents is made by calling `\starttoc{toc}`.

```

1165 %%% \cleardoublepage
1166 \setcounter{page}{3}
1167 \pagestyle{headings}
1168 \hbox to \textwidth{\Cfont \contentsname}\hfil\pagename}

```

Add a locator for a bookmark.

```

1169 \ifisohyper
1170 \pdfbookmark[1]{\contentsname}{isotoc}%
1171 \fi
1172 \begingroup
1173 \parskip\z@
1174 \starttoc{toc}
1175 \endgroup

```

Finish by restoring two column mode if necessary.

```

1176 \if@restonecol\twocolumn\fi}

```

Each sectioning command needs an additional macro to format its entry in the table of contents, as described above. In this class the formatting depends on whether or not the `sect` option is used.

```

\l@clause First the default specifications.
\l@sclause 1177 \newcommand{\l@clause}{\@dottedtocline{1}{0em}{2.3em}}
\l@ssclause 1178 \newcommand{\l@sclause}{\@dottedtocline{2}{1.5em}{3.2em}}
\l@sssclause 1179 \newcommand{\l@sssclause}{\@dottedtocline{3}{3em}{4.1em}}
\l@ssssclause 1180 \newcommand{\l@ssssclause}{\@dottedtocline{4}{4.5em}{5em}}
\l@sssssclause 1181 \newcommand{\l@sssssclause}{\@dottedtocline{5}{6em}{5.9em}}
\l@annex 1182 \newcommand{\l@sssssclause}{\@dottedtocline{6}{7.5em}{6.8em}}
1183 \newcommand{\l@annex}{\@dottedtocline{1}{0em}{11.0em}}

```

In this class lists of floats are made to appear as though they were an integral part of the table of contents. Further, headings are only printed if there is at least one float of the given kind in the body of the document.

`\floatlist` For print a heading for a list of floats.

```

1184 \newcommand{\floatlist}[2]{%
1185 \vspace{2\tocentryskip}

```

```

1186 \hbox to \textwidth{\bfseries #1\hfil}
1187 \vspace*{\tocentryskip}
1188 \nopagebreak
1189 \begingroup
1190     \parskip\z@
1191     \@starttoc{#2}
1192 \endgroup}

```

10.2.2 List of figures

`\iffigs` A flag for figure floats.

```
1193 \newif\iffigs\figsfalse
```

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

```
1194 \newcommand{\listoffigures}{%
1195     \ifx\undefined\setfigs\else\setfigs\fi}
```

`\loftnumberline` Used to add a dash after a figure/table number in the listing.

```
1196 \newcommand{\loftnumberline}[1]{#1 --- }
1197
```

`\l@figure` This macro produces an entry in the list of figures. Note that `FigureM.999L` is 6.15em.

```

1198 \newcommand{\l@figure}{\@dottedtocline{1}{0em}{7.5em}}
1199 \renewcommand{\l@figure}[2]{%
1200     \vskip \z@ \@plus.2\p@
1201     {%
1202         \leftskip 0em
1203         \rightskip \@tocrmarg
1204         \parfillskip -\rightskip
1205         \parindent 0em\@afterindenttrue
1206         \interlinepenalty\@M
1207         \leavevmode
1208         \@tempdima 3.15em
1209         \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
1210         {\let\numberline\loftnumberline \normalfont\figurename{} #1}\nobreak
1211         \loftfillnum{#2}}
1212 }
1213
1214 \newcommand{\loftfillnum}[1]{\normalfont%
1215     {\leaders\hbox{$\m@th\mkern 4.5mu\hbox{.}\mkern 4.5mu$}\hfill}\nobreak
1216     \hb@xt@\@pnumwidth{\hfil #1}\par}
1217
1218
```

10.2.3 List of tables

`\iftabs` A flag for table floats.

```
1219 \newif\iftabs\tabsfalse
```

`\listoftables` This macro is used to request that L^AT_EX produces a list of tables. It is very similar to `\listoffigures`. Note that `Table_M.999` is 5.75em.

```
1220 \newcommand{\listoftables}{%
1221     \ifx\undefined\settabs\else\settabs\fi}
```

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

```
1222 \newcommand{\l@table}{\dottedtocline{1}{0em}{6.5em}}
1223
1224 \renewcommand{\l@table}[2]{%
1225     \vskip \z@ \@plus.2\p@
1226     {%
1227         \leftskip 0em
1228         \rightskip \@tocrmarg
1229         \parfillskip -\rightskip
1230         \parindent 0em\@afterindenttrue
1231         \interlinepenalty\@M
1232         \leavevmode
1233         \@tempdima 2.75em
1234         \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
1235         {\let\numberline\loftnumberline \normalfont\tablename{} #1}\nobreak
1236         \loftfillnum{#2}}
1237 }
1238
```

`\@caption` This is a reimplementaion of the kernel `\@caption` macro (lfloat.dtx) to cater for the peculiarity of putting the float name before the number in the List of...

```
1239 \long\def\@isocaption#1[#2]#3{%
1240     \par
1241     \addcontentsline{\csname ext@#1\endcsname}{#1}%
1242         {\protect\numberline{\@nameuse{#1name}} {\@nameuse{the#1}} --- }%
1243         {\ignorespaces #2}}%
1244     \begingroup
1245         \@parboxrestore
1246         \if@minipage
1247             \@setminipage
1248         \fi
1249         \normalsize
1250         \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
1251     \endgroup}
1252
```

10.2.4 ToC and clause numbering

Commands are provided, based on the `tocvsec2` package, for changing the section numbering level and the ToC entry level.

`\if@knownclause` Helper macro to set a sectioning-related counter. Use as `\@setclcnt{<sec>}{<counter>}`
`\@setclcnt` to set `counter` to the level of `<sec>`.


```

1253 \newif\if@knownclause
1254 \newcommand{\@setclcnt}[2]{
1255   \@knownclausefalse
1256   \if\isostringsequal{#1}{none}
1257     \setcounter{#2}{-10}
1258     \@knownclausetrue
1259   \fi
1260   \if\isostringsequal{#1}{clause}
1261     \setcounter{#2}{1}
1262     \@knownclausetrue
1263   \fi
1264   \if\isostringsequal{#1}{sclause}
1265     \setcounter{#2}{2}
1266     \@knownclausetrue
1267   \fi
1268   \if\isostringsequal{#1}{ssclause}
1269     \setcounter{#2}{3}
1270     \@knownclausetrue
1271   \fi
1272   \if\isostringsequal{#1}{sssclause}
1273     \setcounter{#2}{4}
1274     \@knownclausetrue
1275   \fi
1276   \if\isostringsequal{#1}{ssssclause}
1277     \setcounter{#2}{5}
1278     \@knownclausetrue
1279   \fi
1280   \if\isostringsequal{#1}{sssssclause}
1281     \setcounter{#2}{6}
1282     \@knownclausetrue
1283   \fi
1284   \if\isostringsequal{#1}{all}
1285     \setcounter{#2}{50}
1286     \@knownclausetrue
1287   \fi
1288   \if@knownclause\else
1289     \ClassError{isov2}{%
1290       Unknown clause command name (#1)
1291     }{%
1292       I'll ignore it. Type \space <return> and I'll continue.\MessageBreak
1293       If you haven't mistyped the name then use \protect\setcounter\space instead.}
1294   \fi
1295 }

```

`\settocdepth` `\settocdepth{<sec>}` is the user command for setting *tocdepth* in the *.toc* file to the value corresponding to *<sec>*. It can only be used after the preamble.

```

1296 \newcommand{\settocdepth}[1]{%
1297   \@knownclausefalse
1298   \if\isostringsequal{#1}{none}
1299     \addtocontents{toc}{\protect\setcounter{tocdepth}{-10}}

```

```

1300     \@knownclausetrue
1301 \fi
1302 \if\isostringsequal{#1}{clause}
1303     \addtocontents{toc}{\protect\setcounter{tocdepth}{1}}
1304     \@knownclausetrue
1305 \fi
1306 \if\isostringsequal{#1}{sclause}
1307     \addtocontents{toc}{\protect\setcounter{tocdepth}{2}}
1308     \@knownclausetrue
1309 \fi
1310 \if\isostringsequal{#1}{ssclause}
1311     \addtocontents{toc}{\protect\setcounter{tocdepth}{3}}
1312     \@knownclausetrue
1313 \fi
1314 \if\isostringsequal{#1}{sssclause}
1315     \addtocontents{toc}{\protect\setcounter{tocdepth}{4}}
1316     \@knownclausetrue
1317 \fi
1318 \if\isostringsequal{#1}{ssssclause}
1319     \addtocontents{toc}{\protect\setcounter{tocdepth}{5}}
1320     \@knownclausetrue
1321 \fi
1322 \if\isostringsequal{#1}{sssssclause}
1323     \addtocontents{toc}{\protect\setcounter{tocdepth}{6}}
1324     \@knownclausetrue
1325 \fi
1326 \if\isostringsequal{#1}{all}
1327     \addtocontents{toc}{\protect\setcounter{tocdepth}{50}}
1328     \@knownclausetrue
1329 \fi
1330 \if@knownclause\else
1331     \ClassError{isov2}{%
1332         Unknown clause command name (#1)
1333     }{%
1334         I'll ignore it. Type \space <return> and I'll continue.}
1335 \fi
1336 }

```

`\maxtocdepth` `\maxtocdepth{<sec>}` can be used to initialise `tocdepth` to the value corresponding to `<sec>`. This can only be used between the end of the preamble and the `\tableofcontents` command.

```

1337 \newcommand{\maxtocdepth}[1]{%
1338     \@setclnt{#1}{tocdepth}
1339 }

```

`\setsecnumdepth` `\setsecnumdepth{<sec>}` is the user command for setting `secnumdepth` to the value for `<sec>`. It can only be used after the preamble.

```

1340 \newcommand{\setsecnumdepth}[1]{\leavevmode%
1341     \@setclnt{#1}{secnumdepth}
1342 }

```

```

\maxsecnumdepth \maxsecnumdepth{<sec>} can be used to initialise secnumdepth after the preamble
to the value corresponding to <sec>.
1343 \newcommand{\maxsecnumdepth}[1]{%
1344 \setclcnt{#1}{secnumdepth}
1345 }

```

10.3 Bibliography

This class does not implement a bibliography. The `references` environment is defined instead.

10.4 The index

`theindex` The environment ‘theindex’ can be used for indices. It makes an index with one column, 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.

ISO requires that an index, if present, must be the last element in the document.

```

1346 \newenvironment{theindex}%
1347   {\clearpage
1348    \typeout{Index}%
1349    \refstepcounter{clause}%
1350    \tocskip{\tocentryskip}%
1351    \addcontentsline{toc}{index}{\indexname}%
1352    \columnseprule \z@
1353    \onecolumn{\fibicl@use*{\indexname}}
1354    \parindent\z@
1355    \parskip\z@ \@plus .3\p@\relax
1356    \let\item\@idxitem}%
1357   {\clearpage}

```

`\l@index` Format the index entry in the table of contents.

```
1358 \newcommand{\l@index}{\@dottedtocline{1}{0em}{0pt}}
```

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

```

\subitem 1359 \newcommand{\@idxitem} {\par\hangindent 40\p@}
\subsubitem 1360 \newcommand{\subitem} {\par\hangindent 40\p@ \hspace*{20\p@}}
1361 \newcommand{\subsubitem}{\par\hangindent 40\p@ \hspace*{30\p@}}

```

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

```
1362 \newcommand{\indexspace}{\par \vskip 10\p@ \@plus5\p@ \@minus3\p@\relax}
```

The program GenIndex, written for processing ISO documents, takes an `.idx` file and converts it to a `theindex` format. The following are the formatting commands output by GenIndex.

`\indexfill` These define the format of leaders between the (sub-) topic and the page number.
`\sindexfill` ISO requires a dotted line between each index entry and the page number.
`\ssindexfill` 1363 `\newcommand{\indexfill}{\dotfill}`
1364 `\newcommand{\sindexfill}{\dotfill}`
1365 `\newcommand{\ssindexfill}{\dotfill}`

`\indexsee` These format entries of type ‘see ...’ and ‘see also ...’.
`\indexseealso` 1366 `\newcommand{\indexsee}[1]{\par \hspace*{2em} \emph{see} #1}`
1367 `\newcommand{\indexseealso}[1]{\par \hspace*{2em} \emph{see also} #1}`

`\alphaindexspace` These format the space between each alphabetic block of entries, and correspondingly for entries that begin with an analpahbetic character. ISO requires no additional spacing.

These commands take one parameter, intended to be the (letter) heading for the next block of entries. For example, we could have defined:

```
\newcommand{\alphaindexspace}[1]{\indexspace
{\bfseries #1}}
```

for printing a vertical space and a bold heading.

```
1368 \newcommand{\alphaindexspace}[1]{}
1369 \newcommand{\otherindexspace}[1]{}

```

For good measure we provide a style file for users of the MAKEINDEX program.

```
1370 </iso>
1371 <*ist>
1372   %%%iso.ist  Makeindex style file for ISO documents
1373 group_skip "\n\n"      % no vertical space between blocks
1374 headings_flag 0        % make sure headings are turned off
1375 delim_0 " \dotfill "  % dot leaders between entry and page numbers
1376 delim_1 " \dotfill "
1377 delim_2 " \dotfill "
1378
1379 </ist>
1380 <*iso>
```

10.5 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 begun this operation.

```
1381 \renewcommand{\footnoterule}{%
1382   \kern-3\p@
```

```

1383 \hrule width .4\columnwidth
1384 \kern 2.6\p@}

```

\c@footnote Footnotes are numbered sequentially throughout the document. ISO requires footnotes to be a superscripted arabic numeral with a right parenthesis. The counter is predefined.

```

1385 % \newcounter{footnote}
1386 \renewcommand{\thefootnote}{\arabic{footnote}}

```

\@makefntext The footnote mechanism of L^AT_EX calls the macro `\@makefntext` to produce the actual footnote. The macro gets the text of the footnote as its argument and should use `\@thefnmark` as the mark of the footnote. The macro `\@makefntextis` called when effectively inside a `\parbox` of width `\columnwidth` (i.e., with `\hsize = \columnwidth`).

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

```

\long\def\@makefntext#1{%
  \setpar{\@par
    \@tempdima = \hsize
    \advance\@tempdima-10pt
    \parshape \@ne 10pt \@tempdima}%
  \par
  \parindent 1em\noindent
  \hbox to \z@{\hss\@makefnmark}#1}

```

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

In this document class 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.

```

1387 \long\def\@makefntext#1{%
1388   \parindent 1em%
1389   \noindent
1390   \hbox to 1.8em{\hss\@makefnmark}#1}

```

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

```

1391 %\def\@makefnmark{\hbox{\$~{\@thefnmark}\m@th$}}

```

11 Version control tools

When preparing an international standard the document goes through several iterations. In particular it may change due to international ballot comments. The commands provided may be used to identify changes made to a document during its life cycle.

11.1 Print control

Members of the development group often need to see the changes between document versions, while the general public does not.

`\ifchangemarks` This controls the appearance of the version controls defined below.

```
1392 \newif\ifchangemarks\changemarksfalse
```

The version controls only work properly when the `draft` option is in effect. Also, the command `\changemarkstrue` must be put in the document preamble.

`\v@rid` This acts as an alias for `\marginpar` when both `changemarks` is true and the `draft` option is in effect, otherwise it throws away its two arguments.

```
1393 \newcommand{\v@rid}[2]{%
1394     \ifchangemarks
1395         \ifdr@ftd@c
1396             \marginpar[#1]{#2}%
1397     \fi\fi}
```

11.2 Change marking

The following commands flag changes in the typeset document. Each of the commands takes one parameter which is intended to be a ‘change number’ for tracking purposes. Some also take a text parameter which is the changed text.

`\editorial` `\editorial{<change id>}` Places the `<change id>` in the document to indicate an editorial change.

```
1398 \newcommand{\editorial}[1]{%
1399     \@bsphack
1400     \ifchangemarks
1401         \v@rid{\small\hfill$^{#1}$ED}%
1402         {\small ED$^{#1}$\hfill}%
1403     \fi\@esphack}
```

`\added` `\added{<text>}{<change id>}` Flags the additional `<text>` with the `<change id>`.

```
1404 \long\def\added#1#2{%
1405     \@bsphack
1406     \ifchangemarks
1407         \v@rid{\small\hfill$^{#2}\Rightarrow$}%
1408         {\small $\Leftarrow^{#2}$\hfill}%
1409         \emph{#1}%
1410     \else
```

```

1411     #1
1412     \fi\@esphack}

\deleted \deleted{<change id>} Places the <change id> in the document to indicate that
some text has been deleted.

1413 \newcommand{\deleted}[1]{%
1414     \@bsphack
1415     \ifchangemarks
1416         \v@rid{\small\hfill$^{#1}\Leftarrow$}%
1417         {\small $\Rightarrow^{#1}$\hfill}%
1418     \fi\@esphack}

\moved \moved{<text>}{<change id>} Flags the moved <text> with the <change id>.

1419 \long\def\moved#1#2{%
1420     \@bsphack
1421     \ifchangemarks
1422         \v@rid{\small\hfill$^{#2}\Leftrightarrow$}%
1423         {\small $\Leftrightarrow^{#2}$\hfill}%
1424         \emph{#1}%
1425     \else
1426         #1
1427     \fi\@esphack}

```

12 Structure and boilerplate

ISO standard documents have certain required elements and boilerplate.

12.1 Structural elements

foreword The `foreword` environment initializes the front matter for a standard and starts an unnumbered foreword clause. To ensure that the front matter is set in single column we use an environment.

```

1428 \newenvironment{foreword}%
1429     {\tableofcontents
1430     \listoffigures
1431     \listoftables
1432     \clearpage
1433     \if@twocolumn
1434         \@restonecoltrue\onecolumn
1435     \else
1436         \@restonecolfalse
1437     \fi
1438     \fibicl@use*{\forewordname}%
1439 %%     \tocskip{\tocenterskip}%
1440 %%     \addcontentsline{toc}{clause}{\forewordname}%
1441     \ifisohyper
1442         \pdfbookmark[1]{\forewordname}{isofwd}%
1443     \fi}%

```

```

1444  {\if@restonecol\twocolumn\fi}

\copyrighttext This command sets up the copyright notice on the first page of the table of con-
                tents. The text is set in a bottomfloat environment in a small size.
1445 \newcommand{\copyrighttext}{%
1446   \vfill
1447   %%    \begin{bottomfloat}[b]
1448         \begin{small}
1449         \copyrightnotice
1450         \end{small}
1451   %%    \end{bottomfloat}
1452 }

introduction Starts a new unnumbered introduction clause, the body of which is set in single
              column, so we use an environment.
1453 \newenvironment{introduction}%
1454   {\clearpage
1455    \if@twocolumn
1456      \@restonecoltrue\onecolumn
1457    \else
1458      \@restonecolfalse
1459    \fi
1460    \fibicl@use*{\introductionname}%
1461   %%    \tocskip{\tocentryskip}
1462   %%    \addcontentsline{toc}{clause}{\introductionname}%
1463    \ifisohyper
1464      \pdfbookmark[1]{\introductionname}{isointro}
1465    \fi}%
1466   {\if@restonecol\twocolumn\fi}

\scopeclause Starts a new numbered scope clause. This is given the label ;i1 as it is the first
              numbered clause.
1467 \newcommand{\scopeclause}{\clause{\scopename}\label{;i1}}

\normrefsclause Starts a new numbered normative references clause. This is given the label ;i2
                 as it is the second numbered clause.
1468 \newcommand{\normrefsclause}{\clause{\normrefsname}\label{;i2}}

\defclause These macros start new clauses for definitions, symbols and abbreviations. ISO
\symclause allows these to be grouped in various ways, depending on the amount of material
\abbclause in the respective categories. These are each given the label ;i3 as one should be
\defsymclause the third numbered clause.
\defabbclause 1469 \newcommand{\defclause}{\clause{\defname}\label{;i3}}
\symabbclause 1470 \newcommand{\symclause}{\clause{\symname}\label{;i3}}
\defsymabbclause 1471 \newcommand{\abbclause}{\clause{\abbname}\label{;i3}}
1472 \newcommand{\defsymclause}{\clause{\defsymname}\label{;i3}}
1473 \newcommand{\defabbclause}{\clause{\defabbname}\label{;i3}}
1474 \newcommand{\symabbclause}{\clause{\symabbname}\label{;i3}}
1475 \newcommand{\defsymabbclause}{\clause{\defsymabbname}\label{;i3}}

```


`\defsubclause` These macros start new sub-clauses for definitions, symbols and abbreviations.
`\symsubclause` ISO allows these to be grouped in various ways, depending on the amount of
`\abbsubclause` material in the respective categories.

```

\defsymsubclause 1476 \newcommand{\defsubclause}{\sclause{\defname}}
\defabbsubclause 1477 \newcommand{\symsubclause}{\sclause{\symname}}
\symabbsubclause 1478 \newcommand{\abbsubclause}{\sclause{\abbname}}
                  1479 \newcommand{\defsymsubclause}{\sclause{\defsymname}}
                  1480 \newcommand{\defabbsubclause}{\sclause{\defabbname}}
                  1481 \newcommand{\symabbsubclause}{\sclause{\symabbname}}

```

`\fcandaclause` This macro starts a clause ‘Fundamental concepts and assumptions’. The actual
title is given by the value of `\fcandaname`.

```

1482 \newcommand{\fcandaclause}{\clause{\fcandaname}}

```

`\bibannex` This macro starts a bibliography (which used to be an informative annex).

```

1483 \newcommand{\bibannex}{%
1484   \typeout{Bibliography}
1485   \clearpage
1486   \fibicl@use*{\bibname}
1487   \tocskip{\tocentryskip}
1488   \addcontentsline{toc}{index}{\bibname}
1489 }

```

12.2 Boilerplate

ISO defines the wording of certain textual elements within a standard.

This class has been prepared for standard documents in the English language.

The boilerplate text commands must be redefined for other languages.

`\copyrightnotice` The required English text of the copyright notice.

```

1490 \newcommand{\copyrightnotice}{%
1491 \copyright\quad \copyrightname\quad \thesyear\newline
1492 All rights reserved. Unless otherwise specified, no part of
1493 this publication may be reproduced or utilized in any form or
1494 by any means, electronic or mechanical, including photocopying
1495 and microfilm, without permission in writing from %% the publisher.
1496 %%\makebox[\textwidth][r]{%
1497 %%ISO/IEC Copyright Office $\bullet$ Case Postale 56 $\bullet$
1498 %%CH-1211 Gen{\‘e}ve 20 $\bullet$ Switzerland}
1499 %\vspace{\baselineskip}\newline
1500 %\hspace*{1em} International Organization for Standardization\newline
1501 %\hspace*{1em} Case Postale 56 $\bullet$ CH-2111 Gen{\‘e}ve 20 $\bullet$ Switzerland
1502 either ISO at the address below or ISO’s member body in the country
1503 of the requester.
1504 \par
1505 \noindent ISO copyright office \\\
1506 Case postale 56. CH-1211 Geneva 20 \\\
1507 Tel. +41 22 749 01 11 \\\

```

1508 Fax +41 22 734 10 79 \\
1509 E-mail \texttt{copyright@iso.ch} \\
1510 Web \texttt{www.iso.ch}

For an IS or a Tech Report, need a blank line and place of printing

1511 %\ifisstand \ifcopyright
1512 %% \vspace{\baselineskip}\newline\noindent
1513 %% Printed in Switzerland
1514 %\fi\fi
1515 %\iftchrep \ifcopyright
1516 %% \vspace{\baselineskip}\newline\noindent
1517 %% Printed in Switzerland
1518 %\fi\fi
1519 }

`\fwdbp` The prescribed text of the initial paragraphs in an ISO Standard Foreword.

1520 `\newcommand{\fwdbp}{\input{isofwdbp}}`

The following is the text contained in the file `isofwdbp.tex`.

1521 `</iso>`
1522 `<*fwd1>`
1523 `\ProvidesFile{isofwdbp.tex}[2001/08/29 Boilerplate for start of Foreword]`
1524
1525 ISO (the International Organization for Standardization) is a worldwide
1526 federation of national standards bodies (ISO member bodies). The work
1527 of preparing International Standards is normally carried out through
1528 ISO technical committees. Each member body interested in a subject for
1529 which a technical committee has been established has the right to be
1530 represented on that committee. International organizations,
1531 governmental and non-governmental, in liaison with ISO, also take part
1532 in the work. ISO collaborates closely with the International
1533 Electrotechnical Commission (IEC) on all matters of electrotechnical
1534 standardization.
1535
1536 International Standards are drafted in accordance with the rules given
1537 in the ISO/IEC Directives, Part 2.
1538
1539 The main task of technical committees is to prepare International Standards.
1540 Draft International Standards adopted by the technical committees are
1541 circulated to the member bodies for voting. Publication as an
1542 International Standard requires approval by at least 75% of the member
1543 bodies casting a vote.
1544 `\par`
1545
1546 `</fwd1>`
1547 `<*iso>`

`\tspasfwdp` The prescribed text of the initial paragraphs in an ISO Technical Specification or PAS Foreword.

1548 `\newcommand{\tspasfwdp}{\input{tspasfwdp}}`

The following is the text contained in the file `tspasfwd1.tex`.

```
1549 </iso>
1550 < *tspasfwd1 >
1551 \ProvidesFile{tspasfwd1.tex}[2001/07/06 Boilerplate for start of TS/PAS Foreword]
1552
1553 ISO (the International Organization for Standardization) is a worldwide
1554 federation of national standards bodies (ISO member bodies). The work
1555 of preparing International Standards is normally carried out through
1556 ISO technical committees. Each member body interested in a subject for
1557 which a technical committee has been established has the right to be
1558 represented on that committee. International organizations,
1559 governmental and non-governmental, in liaison with ISO, also take part
1560 in the work. ISO collaborates closely with the International
1561 Electrotechnical Commission (IEC) on all matters of electrotechnical
1562 standardization.
1563
1564 International Standards are drafted in accordance with the rules given
1565 in the ISO/IEC Directives, Part 2.
1566
1567     The main task of technical committees is to prepare International
1568 Standards.
1569 Draft International Standards adopted by the technical committees are
1570 circulated to the member bodies for voting. Publication as an
1571 International Standard requires approval by at least 75% of the member
1572 bodies casting a vote.
1573
1574     In other circumstances, particularly when there is an urgent market
1575 requirement for such documents, a technical committee may decide to
1576 publish other types of normative document:
1577 \begin{itemize}
1578 \item an ISO Publicly Available Specification (ISO/PAS) represents an
1579 agreement between technical experts in an ISO working group and is
1580 accepted for publication if it is approved by more than 50% of the
1581 members of the parent committee casting a vote;
1582
1583 \item an ISO Technical Specification (ISO/TS) represents an agreement
1584 between the members of a technical committee and is accepted for
1585 publication if it is approved by 2/3 of the members of the committee
1586 casting a vote.
1587 \end{itemize}
1588
1589     An ISO/PAS or ISO/TS is reviewed every three years with a view to
1590 deciding whether it can be transformed into an International Standard.
1591 \par
1592
1593 </ *tspasfwd1 >
```

The following is the text contained in the file `trfwd1.tex`.

```
1594 < *trfwd1 >
1595     %% trfwd1.tex Boilerplate for start of a tech rep Foreword clause
```

1596 %
1597
1598 ISO (the International Organization for Standardization) is a worldwide
1599 federation of national standards bodies (ISO member bodies). The work
1600 of preparing International Standards is normally carried out through
1601 ISO technical committees. Each member body interested in a subject for
1602 which a technical committee has been established has the right to be
1603 represented on that committee. International organizations,
1604 governmental and non-governmental, in liaison with ISO, also take part
1605 in the work. ISO collaborates closely with the International
1606 Electrotechnical Commission (IEC) on all matters of electrotechnical
1607 standardization.
1608
1609 International Standards are drafted in accordance with the rules
1610 given in the ISO/IEC Directives, Part 3.
1611
1612 The main task of technical committees is to prepare International
1613 Standards. Draft International Standards adopted by the technical
1614 committees are circulated to the member bodies for voting. Publication
1615 as an International Standard requires approval by at least 75\% of the
1616 member bodies casting a vote.
1617
1618 In other circumstances, particularly when there is an urgent market
1619 requirement for such documents, a technical committee may decide to
1620 publish other types of normative document:
1621 \begin{itemize}
1622 \item an ISO Publicly Available Specification (ISO/PAS) represents an
1623 agreement between technical experts in an ISO working group and is
1624 accepted for publication if it is approved by more than 50\% of the
1625 members of the parent committee casting a vote;
1626
1627 \item an ISO Technical Specification (ISO/TS) represents an agreement
1628 between the members of a technical committee and is accepted for
1629 publication if it is approved by 2/3 of the members of the committee
1630 casting a vote.
1631 \end{itemize}
1632
1633 An ISO/PAS or ISO/TS is reviewed every three years with a view to
1634 deciding whether it can be transformed into an International Standard.
1635 \par
1636
1637
1638 </trfwd1>
1639 <*iso>

\trfwdbpi Required texts for a technical report foreword. Use as: \trfwdbpii{<application
\trfwdbpii field>}.
1640 \newcommand{\trfwdbpi}{\input{trfwd1}}
1641 \newcommand{\trfwdbpii}[1]{%
1642 \ClassError{iso}{The \protect\trfwdbpii\space command has been removed}%

```

1643      {Type <return> to proceed, and change your source file before
1644      running LaTeX again.}
1645 }

```

`\intropatents` `\intropatents` is the boilerplate for the last Introduction paragraph dealing with potential additional patent rights.

```

1646 \newcommand{\intropatents}{\par
1647 Attention is drawn to the possibility that some of the elements of this
1648 document may be the subject of patent rights
1649 other than those mentioned above.
1650 ISO [and/or] IEC shall not be held responsible
1651 for identifying any or all such patent rights.\par}
1652

```

`\fwdnopatents` `\fwdnopatents` is the boilerplate for the Foreword paragraph dealing with potential patent rights.

```

1653 \newcommand{\fwdnopatents}{\par
1654 Attention is drawn to the possibility that some of the elements of this
1655 document may be the subject of patent rights.
1656 ISO shall not be held responsible
1657 for identifying any or all such patent rights.\par}
1658

```

`\normrefbp` The required text for the introduction of the normative references clause. Use as:

```

\normrefbp{<standard identifier>}

```

```

1659 \newcommand{\normrefbp}[1]{%
1660
1661 The following normative documents contain provisions which, through
1662 reference in this text, constitute provisions of this #1.
1663 For dated references, subsequent amendments to, or revisions of,
1664 any of these publications do not apply.
1665 However, parties
1666 to agreements based on this #1
1667 are encouraged to investigate the possibility of applying
1668 the most recent editions of the normative documents indicated below.
1669 For undated references, the latest edition of the normative
1670 document referred to applies.
1671 Members of ISO and IEC maintain registers of currently
1672 valid International Standards.
1673
1674 }

```

13 Initialization

13.1 Words and phrases

`\annexname` This document class is for documents prepared in the English language. To prepare a version for another language, various English words and phrases must be

```

\annexname
\bibname
\contentsname
\defname
\symname
\ablname
\defablname
\defsymname
\defsymablname
\fcandaname
\forewordname
\indexname

```

replaced. The English elements that require replacement are defined below in command names.

This list is for titles of document sections.

```

1675 \newcommand{\abbname}{Abbreviations}
1676 \newcommand{\annexname}{Annex}
1677 \newcommand{\bibname}{Bibliography}
1678 \newcommand{\contentsname}{Contents}
1679 \newcommand{\defname}{Terms and definitions}
1680 \newcommand{\defabname}{Terms, definitions, and abbreviations}
1681 \newcommand{\defsymname}{Terms, definitions, and symbols}
1682 \newcommand{\defsymabname}{Terms, definitions, abbreviations, and symbols}
1683 \newcommand{\fcandaname}{Fundamental concepts and assumptions}
1684 \newcommand{\forewordname}{Foreword}
1685 \newcommand{\indexname}{Index}
1686 \newcommand{\informativename}{informative}
1687 \newcommand{\introductionname}{Introduction}
1688 \newcommand{\normativename}{normative}
1689 \newcommand{\normrefsname}{Normative references}
1690 \newcommand{\scopename}{Scope}
1691 \newcommand{\sectionname}{Section}
1692 \newcommand{\symname}{Symbols}
1693 \newcommand{\symabname}{Symbols and abbreviations}

```

`\copyrightname` These are the names and phrases used for general elements.

```

\examplename 1694 \newcommand{\copyrightname}{ISO}
\figurename 1695 \newcommand{\examplename}{EXAMPLE}
\inscopename 1696 %%%\newcommand{\examplesname}{EXAMPLES}
\ISname 1697 \newcommand{\figurename}{Figure}
\listannexname 1698 \newcommand{\inscopename}{The following are within the scope of this }
\listfigurename 1699 \newcommand{\ISname}{INTERNATIONAL STANDARD}
\listtablename 1700 \ifdisstandard\renewcommand{\ISname}{FINAL DRAFT INTERNATIONAL STANDARD}\fi
\notename 1701 \ifdisstandard\renewcommand{\ISname}{DRAFT INTERNATIONAL STANDARD}\fi
\outofscopename 1702 \ifcdstandard\renewcommand{\ISname}{COMMITTEE DRAFT}\fi
\pagename 1703 \ifwdstandard\renewcommand{\ISname}{WORKING DRAFT}\fi
\tablename 1704 \iftechrep\renewcommand{\ISname}{TECHNICAL REPORT}\fi
\tbpname 1705 \iftechspec\renewcommand{\ISname}{TECHNICAL SPECIFICATION}\fi
1706 \ifpaspec\renewcommand{\ISname}{PUBLICLY AVAILABLE SPECIFICATION}\fi
1707 \ifotherdoc\renewcommand{\ISname}{}\fi
1708 \newcommand{\listannexname}{Annexes}
1709 \newcommand{\listfigurename}{Figures}
1710 \newcommand{\listtablename}{Tables}
1711 \newcommand{\notename}{NOTE}
1712 %%%\newcommand{\notesname}{NOTES}
1713 \newcommand{\outofscopename}{The following are outside the scope of this }
1714 \newcommand{\pagename}{Page}
1715 \newcommand{\tablename}{Table}
1716 \newcommand{\tbpname}{To be published.}

```

`\annexrefname` These are the names for referenced document elements. Except when starting

```

\clauserefname
\examplerefname
\figurerefname
\noterefname
\tablerefname
\pagerefname

```

a sentence or referring to a figure, references to document elements start with a lower case letter.

```
1717 \newcommand{\annexrefname}{annex}
1718 \newcommand{\clauserefname}{clause}
1719 \newcommand{\exemplerefname}{example}
1720 \newcommand{\figurerefname}{Figure}
1721 \newcommand{\noterefname}{note}
1722 \newcommand{\tablerefname}{Table}
1723 \newcommand{\pagerefname}{page}
```

`\abstractname` These names are used in the standard L^AT_EX classes but are not applicable in this class. We just make them null.

```
\chaptername 1724 \newcommand{\abstractname}{}
\partname 1725 \newcommand{\appendixname}{}
\refname 1726 \newcommand{\chaptername}{}
1727 \newcommand{\partname}{}
1728 \newcommand{\refname}{}

```

13.2 Date

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

```
1729 \newcommand{\today}{\ifcase\month\or
1730   January\or February\or March\or April\or May\or June\or
1731   July\or August\or September\or October\or November\or December\fi
1732   \space\number\day, \number\year}
```

13.3 Two column mode

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

```
1733 \setlength\columnsep{10\p@}
```

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

```
1734 \setlength\columnseprule{0\p@}
```

13.4 The page style

We use the page style *headings* by default and start with roman numbering for the front matter, this being reset to arabic by the title or first main matter section/clause.

```
1735 \pagestyle{headings}
1736 \pagenumbering{roman}
```

We set the sectional counters to zero and the `tocdepth` to one (clauses only listed).

```
1737 \setcounter{clause}{0}
1738 \setcounter{annex}{0}
1739 \setcounter{tocdepth}{1}
```

13.5 Single or double sided printing

We do not try to make each page as long as all the others, even though it is two-side printing.

```
1740 \@twosidetrue
1741 \raggedbottom
```

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.

```
1742 \if@twocolumn
1743   \twocolumn
1744   \sloppy
1745   \flushbottom
```

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

```
1746 \else
1747   \onecolumn
1748 \fi
```

The end of the class definitions.

```
1749 </iso>
```

14 The askinc package

This package provides an interactive ‘include’ facility. It was developed by Phil Spiby of CADDETC, Leeds, United Kingdom in the late eighties.

```
1750 <*inc>
```

\infile The `\infile{<file name>}` command is a cross between the `\input` and `\include` commands. When this package is used, at runtime the user is asked to interactively specify a comma-separated list of the names of `\infiled` files that are to be processed. In this sense it acts like the `\include` and `\includeonly` pair of commands. If no list is entered at the terminal (by hitting the `<RETURN>` key) then all `\infiled` files are processed. In this sense it acts like the `\input` command. However, like the `\include` command, an `\infiled` file cannot contain any other `\infiled` file.

temp Define a counter `temp` for general use within the include files. This is required to ensure that the contents of `\incfiles` is used and not `\incfiles` the string.

```
1751 \newcounter{temp}
```

Now for the rest of the definition.

```
1752 \def\readinclude#1\endread{\gdef\myincludeonly{\includeonly{#1}}}
1753 \long\def\stripsspace#1 \nextspace{#1}
1754 \typeout{Which files do you want processing ?}
1755 \message{enter names (separated by commas) or <RET> for all.}
1756 \message{}\global\read-1 to\incfiles
```



```

1757 \if\incfiles\par\let\infile\input
1758 \else\let\infile\include
1759 \edef\incfiles{\expandafter\stripspace\incfiles\nextspace}
1760 \expandafter\readinclude\incfiles\endread\myincludeonly\fi

```

The end of the askinc package.

```
1761 </inc>
```

References

- [GMS94] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Addison-Wesley Publishing Company, 1994.
- [ISO97] ISO/IEC Directives Part 3. *Drafting and presentation of International Standards*, Third edition, 1997.
- [ISO01] ISO/IEC Directives Part 2. *Rules for the structure and drafting of International Standards*, Fourth edition, 2001.
- [Wil96] Peter R. Wilson. *LaTeX for standards: The LaTeX package files user manual*. NIST Report NISTIR, June 1996.

Index

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

Symbols	<code>\dblfpbot</code> <u>494</u>	520, 525, 530, 549
<code>\%</code> 1542, 1571,	<code>\dblfpsep</code> <u>494</u>	<code>\fontswitch</code> 1136, 1137
1580, 1615, 1624	<code>\dblfpstop</code> <u>494</u>	<code>\footnotemark</code> 936, 940
<code>\@afterindentfalse</code> .	<code>\defcl</code> <u>959</u> , 970, 977,	<code>\fpbot</code> <u>491</u>
. 706, 973,	984, 991, 998,	<code>\fpsep</code> <u>491</u>
980, 987, 994,	1005, 1013, 1021	<code>\fpstop</code> <u>491</u>
1001, 1008, 1016	<code>\dotsep</code> <u>1148</u>	<code>\hangfrom</code> <u>626</u>
<code>\@afterindenttrue</code> .	<code>\dottedtocline</code> . . .	<code>\highpenalty</code> <u>425</u>
. 1205, 1230	. . . 1177–1183,	<code>\idxitem</code> . . 1356, <u>1359</u>
<code>\@beginparpenalty</code> . <u>768</u>	1198, 1222, 1358	<code>\ifdefinable</code> 82
<code>\@bsphack</code> 1399,	<code>\endparpenalty</code> . . . <u>768</u>	<code>\ifpackageloaded</code> . 93
1405, 1414, 1420	<code>\enumctr</code> . 831, 835, 836	<code>\infannex</code> <u>722</u>
<code>\@caption</code> <u>1239</u>	<code>\enumdepth</code> . . . 829–831	<code>\initisofig</code>
<code>\@capttype</code> 1122	<code>\eqnum</code> <u>1047</u>	. . . <u>1055</u> , 1065, 1069
<code>\@contcaption</code> 1122, <u>1123</u>	<code>\esphack</code> 1403,	<code>\initisotab</code>
<code>\@copyrighttext</code> . . .	1412, 1418, 1427	. . . <u>1079</u> , 1089, 1093
. . . 583, 1164, <u>1445</u>	<code>\evenfoot</code> <u>497</u> , 522,	<code>\is@str@ngsequali</code> .
<code>\dblfloat</code>	527, 532, 551, 553 73, 75
. . . 1070, 1094, 1106	<code>\evenhead</code> . . . <u>497</u> ,	

<code>\@is@str@ngsequalii</code>	<code>\@parboxrestore</code> . . .	<code>_</code> 11
. 74, 75 1125, 1245	
<code>\@isocaption</code> 1239	<code>\@pnumwidth</code> . 1148, 1216	
<code>\@itemdepth</code> . . . 847–849	<code>\@ptsize</code> . . 69, 106–	A
<code>\@itemitem</code> . . . 849, 853	109, 222, 223, 225	<code>\abbclause</code> 1469
<code>\@itempenalty</code> 768	<code>\@rc@ifdefinable</code> . . 82	<code>\abbname</code> 1471, 1478, 1675
<code>\@knownclausefalse</code> .	<code>\@repannex</code> 722	<code>\abbsubclause</code> 1476
. 1255, 1297	<code>\@restonecolfalse</code> .	<code>\abovecaptionskip</code> .
<code>\@knownclausetrue</code> 576, 1108, 1113
. 1258,	1159, 1436, 1458	<code>\abovedisplayshortskip</code>
1262, 1266,	<code>\@restonecoltrue</code> 574, 233, 239,
1270, 1274,	1157, 1434, 1456	245, 264, 276,
1278, 1282,	<code>\@runninghead</code>	288, 303, 315, 327
1286, 1300,	. 498, 519, 520,	<code>\abovedisplayskip</code> .
1304, 1308,	524, 525, 538, 542 232, 238,
1312, 1316,	<code>\@setclcnt</code> . . . 1253,	244, 248, 263,
1320, 1324, 1328	1338, 1341, 1344	275, 287, 297,
<code>\@listI</code> 249, 781	<code>\@setfontsize</code> . 231,	302, 314, 326, 336
<code>\@listi</code> 249,	237, 243, 262,	<code>\abstractname</code> 1724
266, 278, 290,	274, 286, 301,	<code>\addannextotoc</code>
305, 317, 329, 781	313, 325, 339–	. 717, 725, 735, 745
<code>\@listii</code> 790	345, 348–354,	<code>\addcontentsline</code> . .
<code>\@listiii</code> 790	357–363, 370, 378	719, 1241, 1351,
<code>\@listiv</code> 790	<code>\@setitparams</code> . . .	1440, 1462, 1488
<code>\@listv</code> 790	. 771, 787, 793,	<code>\added</code> 1404
<code>\@listvi</code> 790	798, 803, 808, 813	<code>\addtocontents</code>
<code>\@lowpenalty</code>	<code>\@setminipage</code> . . . 1247	. . . 1154, 1299,
. . . . 425, 768–770	<code>\@setnoteparams</code> 882,	1303, 1307,
<code>\@mainaux</code> . . 1057, 1081	894, 907, 916, 923	1311, 1315,
<code>\@makecaption</code>	<code>\@smidgeon</code>	1319, 1323, 1327
. . 1108, 1127, 1250	. 251, 394, 398,	<code>\addtocounter</code>
<code>\@makefnmark</code> 1390, 1391	402, 406, 410, 414 629, 631, 938
<code>\@makefnmark</code> 1387	<code>\@star@or@long</code> 80	<code>\afterskip</code> 393, 636, 643
<code>\@maxdepth</code> 443	<code>\@startsection</code>	<code>\afterscscskip</code> . . 393, 650
<code>\@medpenalty</code> 425	. 633, 640, 647,	<code>\aftersscscskip</code> . 393,
<code>\@minipagefalse</code> . . 1118	653, 659, 665, 671	656, 662, 668, 674
<code>\@minipagerestore</code> 1042	<code>\@starttoc</code> . 1174, 1191	<code>\alphaindexspace</code> . 1368
<code>\@mparswitchtrue</code> . . 110	<code>\@thefnmark</code> . . 939, 1391	<code>anexample</code> (environ-
<code>\@mpfootins</code> 1042	<code>\@tocrmarg</code>	ment) 913
<code>\@nameuse</code> 1242	. . 1148, 1203, 1228	<code>\annexname</code> 710, 720, 1675
<code>\@normalsize</code> 227	<code>\@toodeep</code> 829, 847	<code>\annexrefname</code> 1140, 1717
<code>\@normannex</code> 722	<code>\@topnum</code> 705	<code>anote</code> (environment) . 891
<code>\@oddfn</code> . 497, 521,	<code>\@twocolumnfalse</code> . . 111	<code>\appendixname</code> 1724
526, 531, 550, 552	<code>\@twocolumntrue</code> . . . 112	<code>\aref</code> 1140
<code>\@oddhead</code> 497,	<code>\@twosidetrue</code> 110, 1740	<code>\arraycolsep</code> 1037
519, 524, 529, 548	<code>\@usfalse</code> 98	<code>\arrayrulewidth</code> . . 1039
<code>\@onelineskip</code>	<code>\@ustrue</code> 105	<code>\AtBeginDocument</code> . . 92
. 251, 395, 399,	<code>\'</code> 1498, 1501	
403, 407, 411, 415		B
		<code>\baselineskip</code>

. 534, 539, 541,	\c@pyrightopttrue . 115	\contentsname
546, 568, 570,	\c@pyrighttrue 220	. . 1168, 1170, <u>1675</u>
715, 778, 784,	\c@sclause <u>600</u>	\copyright
1499, 1512, 1516	\c@secnumdepth <u>596</u> , 719	511, 536, 544, 1491
\baselinestretch . . <u>423</u>	\c@sscclause <u>600</u>	\copyrighthead <u>510</u> ,
\beforecskip	\c@ssclause <u>600</u>	521, 522, 552, 553
. <u>393</u> , 635, 642, 972	\c@ssssclause <u>600</u>	\copyrightname
\beforecscskip	\c@sssssclause <u>600</u>	. 511, 536, 537,
. <u>393</u> , 649, 979	\c@table <u>1072</u>	544, 545, 1491, <u>1694</u>
\beforesscscskip <u>393</u> ,	\c@topnumber <u>476</u>	\copyrightnotice
655, 661, 667,	\c@totalnumber <u>480</u> 1449, <u>1490</u>
673, 986, 993,	\c@yextra <u>605</u>	cover (environment) . <u>572</u>
1000, 1007, 1015	\cal <u>1136</u>	\cref <u>1140</u>
\belowcaptionskip	\captionsize	\csname . . . 835, 853,
. <u>1108</u> , 1121 <u>420</u> , 1114, 1116	1127, 1241, 1250
\belowdisplaysshortskip	\cdstandardfalse	D
. 234, 240, 120, 129,	\d@isfalse 933
246, 265, 277,	138, 147, 166,	\d@istrue 942, 944
289, 304, 316, 328	176, 185, 194, 203	\dblfloatpagefraction
\belowdisplayskip	\cdstandardtrue . . . 156 <u>485</u>
. 248, 297, 336	\centering 1116	\dblfloatsep <u>489</u>
\bf <u>1132</u>	\centerline 714	\dbltextfloatsep <u>489</u>
\bibannex <u>1483</u>	\Cfont 365,	\dbltopfraction <u>484</u>
\bibname 1486, 1488, <u>1675</u>	637, 638, 644,	\DeclareOldFontCommand
bottomfloat (environ- <u>1102</u>	645, 710, 714, 1168 1129–1135
ment) <u>1102</u>	\changemarksfalse 1392	\DeclareRobustCommand
bottomfloat* (environ- <u>1102</u>	\chaptername <u>1724</u> 1136, 1137
ment) <u>1102</u>	\ClassError	\defabbclosure <u>1469</u>
\bottomfraction <u>479</u> 1289, 1331, 1642	\defabbnname
\bref <u>1140</u>	\ClassWarning 1473, 1480, <u>1675</u>
\brokenpenalty <u>434</u> 693, 969, 1012	\defabbsubclause <u>1476</u>
C	\clause <u>628</u> ,	\defclause <u>1469</u>
\c@annex <u>597</u>	692, 694, 695,	\definition <u>1021</u>
\c@bottomfloat <u>1096</u>	1467–1475, 1482	definitions (environ- <u>958</u>
\c@bottomnumber <u>478</u>	\clausemark 676	ment) <u>958</u>
\c@cl@level <u>958</u>	\clauserefname	\defname 1469, 1476, <u>1675</u>
\c@clause <u>597</u> 1142, <u>1717</u>	\defsubclause <u>1476</u>
\c@dbltopnumber <u>483</u>	\cleardoublepage	\defsymabbclosure <u>1469</u>
\c@example <u>910</u> 559, 1165	\defsymabbnname
\c@fibiclosure <u>597</u>	\clearpage 580, 1475, <u>1675</u>
\c@figure <u>1048</u>	683, 1347, 1357,	\defsymclause <u>1469</u>
\c@floatnote <u>874</u>	1432, 1454, 1485	\defsymname
\c@footnote <u>1385</u>	\clubpenalty <u>428</u> 1472, 1479, <u>1675</u>
\c@infrefctr <u>945</u>	\columnsep <u>445</u> , <u>1733</u>	\defsymsubclause <u>1476</u>
\c@note <u>874</u>	\columnseprule	\deleted <u>1413</u>
\c@pyrightfalse 113, 1352, <u>1734</u>	description (environ- <u>857</u>
162, 172, 209, 218	\columnwidth 1383	ment) <u>857</u>
\c@pyrightoptfalse	\complement <u>554</u>	\descriptionlabel
. 114, 116	\contcaption <u>1122</u> 859, <u>861</u>

<code>\displaywidowpenalty</code>	quotation 862	<code>\floatsep</code> 486
. 430	quote 869	<code>\flushbottom</code> 1745
<code>\disref</code> 933	references 945	<code>\fnum@bottomfloat</code> 1098
<code>\disstandardfalse</code> .	symbols 1022	<code>\fnum@figure</code> 1050
. 119 , 128 ,	table* 1088	<code>\fnum@table</code> 1074
137 , 155 , 165 ,	table 1088	<code>\footins</code> 475 , 1042
175 , 184 , 193 , 202	theindex 1346	<code>\footnoterule</code> 1381
<code>\disstandardtrue</code> . . 146	<code>\eref</code> 1140	<code>\footnotesep</code> 474
<code>\dotfill</code> 1363 – 1365	<code>\evensidemargin</code> . . . 449	<code>\footnotesize</code> 299
<code>\doublerulesep</code> . . . 1040	example (environment) 918	<code>\footnotetext</code> 941
<code>\dr@ftd@cfalse</code> 210 , 213	<code>\examplename</code>	<code>\footskip</code> 442
<code>\dr@ftd@ctrue</code> 215 917 , 924 , 1694	foreword (environ-
	<code>\exemplerefname</code>	ment) 1428
 1143 , 1717	<code>\forewordname</code> 1438 ,
E	examples (environ-	1440 , 1442 , 1675
<code>\edef</code> 73 ,	ment) 913	<code>\fps@bottomfloat</code> 1098
74 , 831 , 849 , 1759	<code>\examplesname</code> 1696	<code>\fps@figure</code> 1050
<code>\editorial</code> 1398	<code>\ext@bottomfloat</code> . . . 1098	<code>\fps@table</code> 1074
<code>\efont</code> 365	<code>\ext@figure</code> 1050	<code>\fref</code> 1140
<code>\emph</code> 1366 ,	<code>\ext@table</code> 1074	<code>\ftype@bottomfloat</code> 1098
1367 , 1409 , 1424	<code>\extrahead</code> 516 ,	<code>\ftype@figure</code> 1050
<code>\endcsname</code> 835 , 853 ,	519 , 520 , 524 , 525	<code>\ftype@table</code> 1074
1127 , 1241 , 1250		<code>\fwdbp</code> 1520
<code>\endenumerate</code> 838	F	<code>\fwdnopatents</code> 1653
<code>\enditemize</code> 856	<code>\fboxrule</code> 1043	G
<code>\endread</code> 1752 , 1760	<code>\fboxsep</code> 1043	<code>\Gfont</code> 365
<code>\enumerate</code> 828	<code>\fcandaclause</code> 1482	
enumerate (environ-	<code>\fcandaname</code> 1482 , 1675	H
ment) 828	<code>\fdisstandardfalse</code> .	<code>\hb@xt@</code> 1216
environments: 118 , 127 ,	<code>\headheight</code> 435
<code>anexample</code> 913	145 , 154 , 164 ,	<code>\headsep</code> 435
<code>anote</code> 891	174 , 183 , 192 , 201	<code>\hrulefill</code> 79
<code>bottomfloat*</code> . . . 1102	<code>\fdisstandardtrue</code> . . 136	<code>\Huge</code> 338
<code>bottomfloat</code> . . . 1102	<code>\fibicl@use</code> 639 , 1353 ,	<code>\huge</code> 338 , 369 , 377
<code>cover</code> 572	1438 , 1460 , 1486	<code>\hyperpage</code> 95
<code>definitions</code> 958	<code>\figsfalse</code> 1054 , 1193	
<code>description</code> 857	<code>\figstrue</code> 1056	I
<code>enumerate</code> 828	figure (environment) 1064	<code>\if</code> 78 , 555 ,
<code>examples</code> 913	figure* (environment)	557 , 1256 , 1260 ,
<code>example</code> 918 1064	1264 , 1268 ,
<code>figure*</code> 1064	<code>\figurename</code>	1272 , 1276 ,
<code>figure</code> 1064 1053 , 1210 , 1694	1280 , 1284 ,
<code>foreword</code> 1428	<code>\figurerefname</code>	1298 , 1302 ,
<code>inscope</code> 1029 1144 , 1717	1306 , 1310 ,
<code>introduction</code> . . . 1453	<code>\file</code> 48	1314 , 1318 ,
<code>itemize</code> 846	<code>\fillline</code> 79 ,	1322 , 1326 , 1757
<code>notes</code> 882 534 , 539 , 541 , 546	<code>\if@filesw</code> 1057 , 1081
<code>note</code> 897	<code>\floatlist</code>	<code>\if@knownclause</code>
<code>nreferences</code> 926 1059 , 1083 , 1184 1253 , 1330
<code>olddefinitions</code> . . 954	<code>\floatpagefraction</code> . 482	
<code>outofscope</code> 1029		

<code>\if@minipage</code>	1246	<code>\indexname</code>		<code>\item</code> .	867, 871, 895,
<code>\if@mparswitch</code>	<u>110</u>	. .	1351, 1353, <u>1675</u>		908, 917, 924,
<code>\if@restonecol</code>		<code>\indexsee</code>	<u>1366</u>		932, 936, 940,
.	<u>70</u> , 585,	<code>\indexseealso</code>	<u>1366</u>		953, 957, 1028,
	1176, 1444, 1466	<code>\indexspace</code>	<u>1362</u>		1356, 1578,
<code>\if@twocolumn</code> .	<u>111</u> ,	<code>\infannex</code>	<u>722</u>		1583, 1622, 1627
	567, 573, 1156,	<code>\infile</code>	<u>1751</u>	<code>\itemindent</code> . . .	270,
	1433, 1455, 1742	<code>\infloatfalse</code>			282, 294, 309,
<code>\if@twoside</code>	<u>110</u>	873, 1067,		321, 333, <u>759</u> ,
<code>\if@us</code>	<u>98</u> , 460	1071, 1091, 1095		777, 782, 783,
<code>\ifc@copyright</code>		<code>\infloattrue</code> .	1062, 1086		791, 792, 796,
.	113, <u>218</u> , 510,	<code>\informativename</code> . .			797, 801, 802,
	536, 544, 583,	723, 725, <u>1675</u>		806, 807, 811,
	1164, 1511, 1515	<code>\init@nncx</code>			812, 858, 864,
<code>\ifc@copyrightopt</code> <u>113</u> ,	219	682, 728, 738, 748		886, 928, 950, 1024
<code>\ifcdstandard</code> <u>117</u> ,	1702	<code>inscope</code> (environment)		<code>\itemize</code>	846
<code>\ifchangemarks</code> <u>1392</u> ,		<u>1029</u>	<code>itemize</code> (environment)	<u>846</u>
	1394, 1400,	<code>\in scopename</code> .	1030, <u>1694</u>	<code>\itemsep</code>	269,
	1406, 1415, 1421	<code>\interlinepenalty</code> .			281, 293, 308,
<code>\ifd@is</code>	<u>933</u> , 935	433, 1206, 1231		320, 332, 780, 786
<code>\ifdisstandard</code> <u>117</u> ,	1701	<code>\intextsep</code>	<u>486</u>	<code>\itshape</code>	932,
<code>\ifdr@ftd@c</code>		<code>introduction</code> (environ-			936, 940, 953, 1133
.	210, 456, 1395	ment)	<u>1453</u>		
<code>\iffdisstandard</code> . . .		<code>\introductionname</code> .			J
.	<u>117</u> , 1700	1460,	<code>\jot</code>	<u>1046</u>
<code>\iffigs</code> <u>1054</u> , <u>1056</u> ,	<u>1193</u>	1462, 1464, <u>1675</u>		L
<code>\ifinfloat</code> <u>873</u> ,	892,	<code>\introelement</code>	<u>554</u>	<code>\l@annex</code>	<u>1177</u>
	899, 905, 914, 921	<code>\intropatents</code>	<u>1646</u>	<code>\l@clause</code>	<u>1177</u>
<code>\ifisohyper</code> .	<u>90</u> , 700,	<code>\ISname</code> . .	535, 543, <u>1694</u>	<code>\l@figure</code>	<u>1198</u>
	1169, 1441, 1463	<code>\isoemptystring</code> . . .		<code>\l@index</code>	<u>1358</u>
<code>\ifisstand</code>	1511	<u>71</u> , 555, 557	<code>\l@sclosure</code>	<u>1177</u>
<code>\ifisstandard</code>	<u>117</u>	<code>\isohyperfalse</code>	91	<code>\l@ssclosure</code>	<u>1177</u>
<code>\ifotherdoc</code>		<code>\isohypertrue</code>	94	<code>\l@sssclosure</code>	<u>1177</u>
.	<u>117</u> , 504, 561, 1707	<code>\isostringsequal</code> . .		<code>\l@ssssclosure</code>	<u>1177</u>
<code>\ifpaspec</code>	<u>124</u> , 1706	<u>71</u> , 1256,	<code>\l@sssssclosure</code>	<u>1177</u>
<code>\ifpdf</code>	<u>84</u>	1260, 1264,	<code>\l@sssssclosure</code>	<u>1177</u>
<code>\ift@chrep</code>	1515	1268, 1272,	<code>\l@table</code>	<u>1222</u>
<code>\iftabs</code> <u>1078</u> ,	1080, <u>1219</u>	1276, 1280,	<code>\label</code>	1467–1475
<code>\iftechrep</code> . .	<u>117</u> , 1704	1284, 1298,	<code>\labelenumi</code>	<u>821</u>
<code>\iftechspec</code> . .	<u>124</u> , 1705	1302, 1306,	<code>\labelenumii</code>	<u>821</u>
<code>\ifwdstandard</code> <u>117</u> ,	1703	1310, 1314,	<code>\labelenumiii</code>	<u>821</u>
<code>\ignorespaces</code>	1318, 1322, 1326	<code>\labelenumiv</code>	<u>821</u>
.	1127, 1243, 1250	<code>\isourl</code>	<u>1138</u>	<code>\labelinhref</code> . .	945, 949
<code>\immediate</code> .	1057, 1081	<code>\isref</code>	<u>932</u>	<code>\labelitemi</code>	<u>839</u>
<code>\incfiles</code>	1756,	<code>\isstandfalse</code> . .		<code>\labelitemii</code>	<u>839</u>
	1757, 1759, 1760	117, 135,	<code>\labelitemiii</code>	<u>839</u>
<code>\include</code>	1758	144, 153, 163,	<code>\labelitemiv</code> . .	842, 845
<code>\includeonly</code>	1752	173, 182, 191, 200	<code>\labelsep</code>	
<code>\indexfill</code>	<u>1363</u>	<code>\isstandtrue</code> . . .	126	<u>760</u> , 774, 776,
		<code>\it</code>	<u>1133</u>		782, 791, 796,

801, 806, 811,	<code>\loftfillnum</code>	1078, 1193,
858, 885, 886,	.. 1211, 1214, 1236	1219, 1253, 1392
929, 951, 1025, 1041	<code>\loftnumberline</code> . . .	<code>\newline</code> 555,
<code>\labelwidth</code> . . . 760,	.. 1196, 1210, 1235	557, 1491, 1499,
775, 776, 858,	<code>\Lopt</code> 47	1500, 1512, 1516
887, 929, 951, 1025	<code>\Lpack</code> 52	<code>\nextspace</code> . 1753, 1759
<code>\languageofedition</code> . 498		<code>\Nfont</code> 365,
<code>\LARGE</code> 338, 385, 386		895, 908, 917, 924
<code>\Large</code> 338,	M	<code>\nopagebreak</code> 1188
371, 379, 387, 710	<code>\mainelement</code> 554	<code>\normalfont</code> . . . 839,
<code>\large</code> 338, 372, 380, 388	<code>\make@command</code> 80	861, 1129–1135,
<code>\Lcount</code> 49	<code>\makeannexhead</code>	1210, 1214, 1235
<code>\leaders</code> 1215	.. 713, 724, 734, 744	<code>\normallineskip</code> . . . 421
<code>\Leftarrow</code> . 1408, 1416	<code>\makebox</code> 1496	<code>\normalsize</code> . . . 227,
<code>\leftmargin</code> 266, 278,	<code>\makecommand</code> 80	365, 373, 381,
290, 305, 317,	<code>\makelabel</code> . . . 836,	389, 420, 1126, 1249
329, 752, 781,	854, 859, 930, 1026	<code>\normannex</code> 722
790, 795, 800,	<code>\makepreannexhead</code>	<code>\normativename</code>
805, 810, 865,	.. 708, 723, 733, 743	.. 733, 735, 1675
870, 889, 892,	<code>\marginpar</code> 1396	<code>\normrefbp</code> 1659
893, 905, 906,	<code>\marginparpush</code> 449	<code>\normrefsclause</code> . . 1468
914, 915, 921,	<code>\marginparsep</code> 449	<code>\normrefsname</code> 1468, 1675
922, 928, 950, 1024	<code>\marginparwidth</code> . . . 449	<code>note</code> (environment) . . 897
<code>\leftmargini</code>	<code>\mathbf</code> 1132	<code>\notelabel</code> 881
.. 266, 278, 290,	<code>\mathcal</code> 1136	<code>\notename</code> 895, 908, 1694
305, 317, 329,	<code>\mathit</code> 1133	<code>\noterefname</code> 1145, 1717
752, 761, 775, 781	<code>\mathnormal</code> 1137	<code>notes</code> (environment) . 882
<code>\leftmarginii</code> . 752, 790	<code>\mathrm</code> 1129	<code>\notesname</code> 1712
<code>\leftmarginiii</code> 752, 795	<code>\mathsf</code> 1130	<code>\nref</code> 1140
<code>\leftmarginiv</code> . 752, 800	<code>\mathtt</code> 1131	<code>\nreferencelabel</code> . .
<code>\leftmarginv</code> . . 752, 805	<code>\maxdepth</code> 443	.. 926, 930
<code>\leftmarginvi</code> . 752, 810	<code>\maxsecnumdepth</code> . . 1343	<code>nreferences</code> (environ-
<code>\Leftrightarrow</code>	<code>\maxtocdepth</code> 1337	ment) 926
.. 1422, 1423	<code>\mbox</code> 79, 511, 513, 516,	
<code>\leftskip</code> 1202,	536, 544, 582, 1163	O
1209, 1227, 1234	<code>\message</code> . . . 1755, 1756	<code>\oddsidemargin</code> 449
<code>\Lenv</code> 51	<code>\MessageBreak</code> 1292	<code>\olddefinition</code> 957
<code>\linebreak</code>	<code>\mit</code> 1136	<code>olddefinitions</code> (envi-
.. 534, 539, 541, 546	<code>\mkern</code> 1215	ronment) . . . 954
<code>\lineskip</code> 421	<code>\moved</code> 1419	<code>\onecolumn</code>
<code>\listannexname</code> . . . 1694	<code>\myincludeonly</code>	574, 1157, 1353,
<code>\listfigurename</code> 1752, 1760	1434, 1456, 1747
.. 1059, 1694		<code>\otherdocfalse</code>
<code>\listoffigures</code>	N	.. 123, 134,
.. 1194, 1430	<code>\n</code> 1373	143, 152, 161,
<code>\listoftables</code> 1220, 1431	<code>\new@command</code> 83	171, 181, 190, 199
<code>\listparindent</code>	<code>\newif</code> 70, 84,	<code>\otherdoctrue</code> 208
.. 863, 864, 888	90, 98, 113, 114,	<code>\otherindexspace</code> . 1368
<code>\listtablename</code>	117–125, 210,	<code>outofscope</code> (environ-
.. 1083, 1694	873, 933, 1054,	ment) 1029

<code>\outofscopename</code>	<code>\pstyle</code> 50	<code>\SSCfont</code> 365,
. 1034, <u>1694</u>		657, 663, 669, 675
<code>\overfullrule</code>		<code>\ssclause</code> <u>646</u>
. 211, 212, 214		<code>\ssclausemark</code> <u>678</u>
	Q	<code>\ssindexfill</code> <u>1363</u>
P	<code>\quad</code> 1491	<code>\sssclause</code> <u>646</u>
<code>\p@enumii</code> <u>825</u>	quotation (environ-	<code>\sssclausemark</code> 679
<code>\p@enumiii</code> <u>825</u>	ment) <u>862</u>	<code>\sssssclause</code> <u>646</u>
<code>\p@enumiv</code> <u>825</u>	quote (environment) . <u>869</u>	<code>\sssssclausemark</code> 680
<code>\p@inrefctr</code> <u>945</u>		<code>\ssssclause</code> <u>646</u>
<code>\pagename</code> 1168, <u>1694</u>	R	<code>\ssssclausemark</code> 681
<code>\pagenumbering</code>	<code>\read</code> 1756	<code>\ssssclausemark</code> 681
. 559, 578, 1736	<code>\readinclude</code> 1752, 1760	<code>\standard</code> <u>498</u>
<code>\pageref</code> 1147	<code>\RecordChanges</code> 45	<code>\stepcounter</code> 898, 919
<code>\pagerefname</code> 1147, <u>1717</u>	<code>\rectoisotitlehead</code> . <u>533</u>	<code>\string</code> 1058,
<code>\paperheight</code> 100, 103	<code>\ref</code> 1140–1146	1059, 1082, 1083
<code>\paperwidth</code> 101, 104	<code>\reference</code> <u>953</u>	<code>\stripspace</code> 1753, 1759
<code>\parbox</code> 548, 549	references (environ-	<code>\subitem</code> <u>1359</u>
<code>\parindent</code> <u>424</u> , 1205,	ment) <u>945</u>	<code>\subsubitem</code> <u>1359</u>
1230, 1354, 1388	<code>\refname</code> <u>1724</u>	<code>\symabbclosure</code> <u>1469</u>
<code>\parsep</code> 268, 269,	<code>\repannex</code> <u>722</u>	<code>\symablname</code>
280, 281, 292,	<code>\RequirePackage</code> 68 1474, 1481, <u>1675</u>
293, 307, 308,	<code>\Rightarrow</code> 1407, 1417	<code>\symabbsubclosure</code> <u>1476</u>
319, 320, 331,	<code>\rm</code> <u>1129</u>	<code>\symboldef</code> <u>1028</u>
332, 778, 784, 866	<code>\Roman</code> 820	<code>\symbollabel</code> <u>1022</u> , 1026
<code>\parskip</code> <u>251</u> ,	<code>\romannumeral</code> 831, 849	symbols (environment)
396, 400, 404,	S <u>1022</u>
408, 412, 416,	<code>\sbox</code> 1114	<code>\symclause</code> <u>1469</u>
1173, 1190, 1355	<code>\sc</code> <u>1133</u>	<code>\symname</code> 1470, 1477, <u>1675</u>
<code>\partname</code> <u>1724</u>	<code>\SCfont</code> 365, 651	<code>\symsubclause</code> <u>1476</u>
<code>\partopsep</code> <u>763</u> , 883	<code>\sclause</code> <u>646</u> , 1476–1481	
<code>\paspecfalse</code> 125, 133,	<code>\sclausemark</code> 677	T
142, 151, 160,	<code>\scopeclause</code> <u>1467</u>	<code>\tabbingsep</code> <u>1041</u>
170, 180, 189, 207	<code>\scopename</code> 1467, <u>1675</u>	<code>\tabcolsep</code> <u>1038</u>
<code>\paspectrue</code> 198	<code>\scriptsize</code> <u>338</u>	<code>table</code> (environment) <u>1088</u>
<code>\pdfbookmark</code>	<code>\scshape</code> 536,	<code>table*</code> (environment) <u>1088</u>
. 1170, 1442, 1464	537, 544, 545, 1135	<code>\tablename</code>
<code>\pdffalse</code> 86	<code>\sectionname</code> <u>1675</u> 1077, 1235, <u>1694</u>
<code>\pdfoutput</code> 85	<code>\setfigs</code> 1058, 1195	<code>\tableofcontents</code>
<code>\pdftrue</code> 88	<code>\setsecnumdepth</code> <u>1340</u> <u>1155</u> , 1429
<code>\postdisplaypenalty</code> <u>430</u>	<code>\settabs</code> 1082, 1221	<code>\tablerefname</code> 1146, <u>1717</u>
<code>\predisplaypenalty</code> . <u>430</u>	<code>\settocdepth</code> <u>1296</u>	<code>\tabsfalse</code> 1078, 1219
<code>\pref</code> <u>1140</u>	<code>\sf</code> <u>1129</u>	<code>\tabstrue</code> 1080
<code>\ProvidesFile</code> 61–63,	<code>\sffamily</code> 1130	<code>\tbpname</code> 941, <u>1694</u>
65, 66, 1523, 1551	<code>\sindexfill</code> <u>1363</u>	<code>\techrepfalse</code> 122,
<code>\ProvidesPackage</code> 64	<code>\sl</code> <u>1133</u>	131, 140, 149,
<code>\ps@headings</code> <u>518</u>	<code>\small</code> 49, <u>260</u> ,	158, 168, 187, 205
<code>\ps@isotitlehead</code> <u>533</u>	366, 367, 1401,	<code>\techreptrue</code> 178, 196
<code>\ps@nohead</code> <u>528</u>	1402, 1407,	<code>\techspecfalse</code>
<code>\ps@startpage</code> <u>523</u>	1408, 1416, 124, 132,
	1417, 1422, 1423	

141, 150, 159,	<code>\theHssc</code>	616	<code>\topfraction</code>	477
169, 179, 197, 206	<code>\theHsssc</code>	616	<code>\topmargin</code>	449
<code>\techspectrue</code>	<code>\theHssssc</code>	616	<code>\topsep</code>	267, 279,
<code>\temp</code>	<code>\theHsssssc</code>	616	291, 306, 318,	
<code>\textasteriskcentered</code>	<code>\theHtable</code>	701	330, 779, 785, 884	
841	<code>theindex</code> (environ-		<code>\topskip</code>	435, 443
<code>\textbullet</code>	ment)	1346	<code>\tref</code>	1140
<code>\textemdash</code>	<code>\theinfrefctr</code>	945	<code>\trfwdbpi</code>	1640
<code>\textfloatsep</code>	<code>\thenote</code>		<code>\trfwdpii</code>	1640
<code>\textfloatsep</code>	. 874, 878, 901, 908		<code>\trfwdpbii</code>	1642
<code>\textfraction</code>	<code>\thepage</code>	521, 522,	<code>\tspasfwdpb</code>	1548
<code>\textheight</code>	526, 527, 550–553		<code>\tt</code>	1129
<code>\textperiodcentered</code>	<code>\thesclause</code>	610, 696, 982	<code>\ttfamily</code>	1131
<code>\textsc</code>	<code>\theslanguage</code>	498	<code>\twocolumn</code>	
<code>\textsf</code>	<code>\theslanguage</code>	498	568, 585, 1176,	
<code>\textsl</code>	<code>\theslanguage</code>	498	1444, 1466, 1743	
<code>\texttt</code>	<code>\theslanguage</code>	498		
48, 51, 1138,	<code>\theslanguage</code>	498	U	
1139, 1509, 1510	<code>\theslanguage</code>	498	<code>\underline</code>	1138
<code>\textwidth</code>	<code>\theslanguage</code>	498	<code>\url</code>	1138
. 445, 548, 549,	<code>\theslanguage</code>	498		
1168, 1186, 1496	<code>\theslanguage</code>	498	V	
<code>\Tfont</code>	<code>\theslanguage</code>	498	<code>\v@rid</code>	1393, 1401,
<code>\theannex</code>	<code>\theslanguage</code>	498	1407, 1416, 1422	
696–698, 710, 720	<code>\theslanguage</code>	498	<code>\value</code>	689–
<code>\thebottomfloat</code>	<code>\theslanguage</code>	498	691, 941, 962–968	
1101	<code>\theslanguage</code>	498	<code>\vbox</code>	568, 570
<code>\theclause</code>	<code>\theslanguage</code>	498	<code>\versoisotitlehead</code>	533
. 607, 610, 630, 975	<code>\theslanguage</code>	498		
<code>\theenumi</code>	<code>\theslanguage</code>	498	W	
<code>\theenumii</code>	<code>\theslanguage</code>	498	<code>\wdstandardfalse</code>	
<code>\theenumiii</code>	<code>\theslanguage</code>	498 121, 130,	
<code>\theenumiv</code>	<code>\theslanguage</code>	498	139, 148, 157,	
<code>\theequation</code>	<code>\theslanguage</code>	498	177, 186, 195, 204	
<code>\theexample</code>	<code>\theslanguage</code>	498	<code>\wdstandardtrue</code>	167
<code>\thefibicl@use</code>	<code>\theslanguage</code>	498	<code>\widowpenalty</code>	428
<code>\thefigure</code>	<code>\theslanguage</code>	498	<code>\write</code>	1057, 1081
. . . 698, 1049, 1053	<code>\theslanguage</code>	498		
<code>\thefloatnote</code>	<code>\theslanguage</code>	498	X	
. . . . 874, 879, 901	<code>\theslanguage</code>	498	<code>\xdef</code>	939
<code>\thefootnote</code>	<code>\theslanguage</code>	498		
. 939, 1386	<code>\theslanguage</code>	498	Y	
<code>\theHannex</code>	<code>\theslanguage</code>	498	<code>\yearofedition</code>	498
. 616, 699, 701, 702	<code>\theslanguage</code>	498		
<code>\theHclause</code>	<code>\theslanguage</code>	498	Z	
<code>\theHexample</code>	<code>\theslanguage</code>	498	<code>\zerocounters</code>	624,
<code>\theHfigure</code>	<code>\theslanguage</code>	498	628, 646, 652,	
<code>\theHfloatnote</code>	<code>\theslanguage</code>	498	658, 664, 670, 687	
<code>\theHnote</code>	<code>\theslanguage</code>	498		
<code>\theHsclause</code>	<code>\theslanguage</code>	498		