

# The minitoc package\*

Jean-Pierre F. Drucbert  
jean-pierre.drucbert@onera.fr

16th September 2005

---

\*This document corresponds to minitoc v43, dated 2005/09/16.

# Summary

<b>Contents</b>	<b>3</b>
<b>List of Figures</b>	<b>12</b>
<b>List of Tables</b>	<b>13</b>
<b>I User's Manual</b>	<b>14</b>
1 The <code>minitoc</code> package	16
2 Frequently Asked Questions	42
3 Installation	54
4 Memento	58
Acknowledgments	65
<b>II Implementation</b>	<b>66</b>
5 Commented code of <code>minitoc.sty</code>	68
6 Commented code of <code>mtcoeff.sty</code>	210
7 Patch for the <code>memoir</code> class	222
8 The <code>minitoc</code> language definition ( <code>.mld</code> ) and object ( <code>.mlo</code> ) files	224
<b>Complements</b>	<b>285</b>
Bibliography	285
Change History	290
Index	299

# Contents

<b>Contents</b>	<b>3</b>
<b>List of Figures</b>	<b>12</b>
<b>List of Tables</b>	<b>13</b>
<b>I User's Manual</b>	<b>14</b>
<b>1 The minitoc package</b>	<b>16</b>
1.1 Introduction	17
1.1.1 License	18
1.2 How to use the minitoc package	18
1.2.1 Loading the package and creating the mini-tables	18
1.2.2 Preparing the mini-tables	19
1.2.3 Placing the mini-tables	20
1.2.4 Starred chapters, parts and sections	23
1.3 Typesetting of the mini-tables	24
1.3.1 Chapter-level mini-tables	24
1.3.2 Titles for chapter-level mini-tables	25
1.3.3 Part-level mini-tables	26
1.3.4 Titles for part-level mini-tables	26
1.3.5 Section-level mini-tables	28
1.3.6 Titles for section-level mini-tables	28
1.3.7 Position of the titles	28
1.3.7.1 For mini-tables at the part level	28
1.3.7.2 For mini-tables at the chapter level	28
1.3.7.3 For mini-tables at the section level	29
1.3.7.4 Summary of the positioning of titles	29
1.3.8 Line spacing in the mini-tables	29
1.3.9 Simplified commands for fonts	30
1.3.10 Simplified command for mini-table titles	30
1.3.11 Simplified command for mini-table depths	31
1.3.12 Languages for the titles	31
1.3.13 Altering the layout of the mini-tables ( <i>experimental feature</i> )	31
1.4 Special Features	33
1.4.1 Horizontal Rules	33
1.4.2 Page Numbers, Leaders	34

1.4.3	Features for parttocs and other mini-tables	34
1.4.3.1	Remark about page styles	36
1.4.4	The “Chapter 0” Problem (solved)	36
1.4.4.1	Before version #23 (1994/11/08)	36
1.4.4.2	Since version #23 (1994/11/08)	37
1.4.5	Special Entries for TOC, LOF, LOT, Bibliography and Index	37
1.4.6	The notoccite option	38
1.5	The hints option	38
1.6	Usage with MS-DOS	40
1.7	Why several L <sup>A</sup> T <sub>E</sub> X runs are required ?	40
1.8	The mtcoff package	41
<b>2</b>	<b>Frequently Asked Questions</b>	<b>42</b>
2.1	How to avoid a page break near the rules before and after the mini-table ?	43
2.2	How about implementing others layouts for a mini-table ?	43
2.3	Two consecutive backslashes in a contents line make an error	43
2.4	Reordering chapters makes havoc	43
2.5	Extensions for the names of auxiliary files	44
2.6	Playing with the chapter number	44
2.7	Supported document classes	45
2.8	Compatibility with L <sup>A</sup> T <sub>E</sub> X versions	45
2.9	Other mini-tables	45
2.10	Why so many auxiliary files ?	46
2.11	How to do mini-tables at levels other than chapter ?	46
2.12	Incompatibility with L <sup>A</sup> T <sub>E</sub> X2.09	47
2.13	Documents resetting the chapter number at each part	47
2.14	The mini-tables have too much spaced lines	47
2.15	The secttocs are wrong	47
2.16	How to remove the lines of dots ?	47
2.17	How to use the hyperref package with minitoc ?	48
2.18	Problem while upgrading minitoc	48
2.19	A local table of contents for the set of appendices	48
2.20	Use with the appendix package	49
2.21	Use with the tocloft package	49
2.22	Use with the memoir class	50
2.23	There are too many commands for fonts, titles, and depths	50
2.24	Compatibility with the A <sub>M</sub> S document classes	50
2.25	How to hide some entries from the main table of contents	51
2.26	How to define your own .mld file ?	51
2.27	Use with the abstract package	52
2.28	Use with the sectsty package	52
2.29	Strange alignment in the minitocs	52
<b>3</b>	<b>Installation</b>	<b>54</b>
<b>4</b>	<b>Memento</b>	<b>58</b>
	<b>Acknowledgments</b>	<b>65</b>

## II Implementation 66

<b>5</b>	<b>Commented code of <code>minitoc.sty</code></b>	<b>68</b>
5.1	Introduction . . . . .	71
5.2	Identification code . . . . .	71
5.3	A file descriptor to write . . . . .	72
5.4	Indentation and skip . . . . .	72
5.5	Tests and flags . . . . .	72
5.5.1	Use of section-level mini-lists of floats. . . . .	72
5.5.2	Presence of some packages and classes. . . . .	73
5.5.3	Presence or absence of some sectioning commands . . . . .	73
5.5.4	Flags to check to see if some commands are used . . . . .	74
5.6	Preparation for the <code>notoccite</code> option . . . . .	75
5.7	Preparation for the <code>tight</code> and <code>k-tight</code> options . . . . .	75
5.8	Preparation to work with <code>hyperref</code> . . . . .	75
5.9	Check if the <code>sectsty</code> package is loaded, and when . . . . .	76
5.10	Is the <code>memoir</code> class loaded? . . . . .	76
5.11	Testing the emptiness of a file . . . . .	77
5.12	Internal macros to decrement <code>minitoc</code> counters . . . . .	78
5.13	Patching the <code>\part</code> command . . . . .	79
5.14	Adding an entry in the TOC for a starred part . . . . .	79
5.15	Section level macros . . . . .	80
5.16	Corrections for numbering . . . . .	80
5.17	Patching the <code>\section</code> command . . . . .	81
5.18	Adding an entry in the TOC for a starred section . . . . .	81
5.19	Chapter level macros . . . . .	82
5.20	Patching the <code>\chapter</code> command . . . . .	82
5.21	Adding an entry in the TOC for a starred chapter . . . . .	82
5.22	Miscellaneous declarations . . . . .	83
5.23	Autoconfiguration of extensions . . . . .	83
5.24	Detecting obsolete versions of $\text{\LaTeX}$ . . . . .	85
5.25	A macro to make a TOC entry without leaders nor page numbers . . . . .	85
5.26	Default values for the page-number customization macros . . . . .	86
5.27	“Features” for the mini-tables . . . . .	88
5.28	Fake tables of contents . . . . .	90
5.29	Depth counters for <code>minilofs</code> and <code>minilots</code> . . . . .	91
5.30	Chapter level commands . . . . .	91
5.31	Starred parts, chapters or sections . . . . .	91
5.32	Font commands for the mini-tables . . . . .	93
5.33	Internal commands to position the mini-table titles . . . . .	93
5.34	The <code>mtc@verse</code> environment . . . . .	94
5.35	The <code>\minitoc</code> , <code>\minilof</code> , and <code>\minilot</code> commands . . . . .	94
5.35.1	The <code>\minitoc</code> command . . . . .	94
5.35.2	The <code>\minilof</code> command . . . . .	97
5.35.3	The <code>\minilot</code> command . . . . .	100
5.36	Patching the <code>\chapter</code> command, continued . . . . .	103
5.37	The <code>\addstarred...</code> commands . . . . .	104
5.38	TOC entries without leaders . . . . .	104
5.39	Mini-tables with or without leaders . . . . .	105
5.40	The <code>\dominitoc</code> command and its siblings . . . . .	106

5.40.1	Analysis and splitting of the TOC file . . . . .	108
5.41	Mini-lists of figures . . . . .	112
5.41.1	Analysis and splitting of the list of figures file . . . . .	112
5.42	Mini-lists of tables . . . . .	114
5.42.1	Analysis and splitting of the list of tables file . . . . .	114
5.43	Macro to write a contents line . . . . .	117
5.44	Depth counters for partlofs and partlots . . . . .	118
5.45	Part level commands . . . . .	118
5.46	Fonts for the parttocs . . . . .	119
5.47	Default titles for part-level mini-tables . . . . .	120
5.48	The ptc@verse environment . . . . .	122
5.49	The part level mini-tables : \parttoc, \partlof, and \partlot . . . . .	122
5.49.1	The \parttoc command . . . . .	122
5.49.2	The \partlof command . . . . .	125
5.49.3	The \partlot command . . . . .	128
5.50	Auxiliary commands for printing parttocs . . . . .	131
5.51	Patching the \part command, continued . . . . .	131
5.52	The \doparttoc command and its siblings . . . . .	132
5.52.1	Processing macros for the parttocs . . . . .	133
5.52.2	Processing macros for the partlofs . . . . .	137
5.52.3	Processing macros for the partlots . . . . .	139
5.53	Depth counters for sectlofs and sectlots . . . . .	141
5.54	Section-level commands . . . . .	142
5.55	Fonts commands for secttocs and co. . . . .	142
5.56	Internal macros for title positioning . . . . .	143
5.57	The stc@verse environment . . . . .	143
5.58	The \secttoc, \sectlof, and \sectlot commands . . . . .	144
5.58.1	The \secttoc command . . . . .	144
5.58.2	The \sectlof command . . . . .	146
5.58.3	The \sectlot command . . . . .	149
5.59	Auxiliary internal commands, section level . . . . .	152
5.60	Patching the \section command (continued) . . . . .	152
5.61	The \dosecttoc command and siblings . . . . .	153
5.62	End of section-level commands . . . . .	162
5.63	Necessary \l@... commands . . . . .	162
5.64	The horizontal rules and their default values . . . . .	162
5.65	The \mtcset... commands . . . . .	163
5.65.1	Keywords for the \mtcset... commands . . . . .	163
5.65.2	The \mtcsetfont command . . . . .	165
5.65.3	The \mtcsetttitlefont command . . . . .	167
5.65.4	The \mtcsetttitle command . . . . .	169
5.65.5	The \mtcsetformat command . . . . .	170
5.65.6	The \mtcsetpagenumbers command . . . . .	174
5.65.7	The \mtcsetrules command . . . . .	175
5.65.8	The \mtcsetfeature command . . . . .	177
5.65.9	The \mtcsetdepth command . . . . .	179
5.66	The mtchideinmaintoc environment . . . . .	180
5.67	Fixing the “Index” entry in the TOC . . . . .	182
5.68	Fixing the “Glossary” entry in the TOC . . . . .	183
5.69	The \mtcselectlanguage command . . . . .	185

5.70	The <code>\mtcloadmlo</code> internal command	185
5.71	The “coffee breaks”	186
5.72	Initialization of counters	186
5.73	Declarations for simple options	186
5.73.1	Options <code>tight</code> and <code>loose</code> , <code>k-tight</code> and <code>k-loose</code>	186
5.73.2	Options <code>checkfiles</code> and <code>nocheckfiles</code>	187
5.73.3	Options <code>dotted</code> and <code>undotted</code>	187
5.73.4	Option <code>notoccite</code>	187
5.73.5	Option <code>shorttext</code>	187
5.74	The <code>insection</code> option	187
5.75	Language options	188
5.76	The <code>hints</code> option	190
5.76.1	First part : <code>\mtc@hints@begindoc</code>	191
5.76.1.1	Hint about the <code>appendix</code> package	191
5.76.1.2	Hint about the <code>tocbibind</code> package	191
5.76.1.3	Hint about the <code>tocloft</code> package	192
5.76.1.4	Hint about the <code>titletoc</code> package	192
5.76.1.5	Hint about the <code>placeins</code> package	192
5.76.1.6	Hint about the <code>memoir</code> class	193
5.76.1.7	Hint about the <code>amsart</code> and <code>amsproc</code> classes	193
5.76.1.8	Hint about the <code>amsbook</code> class	194
5.76.1.9	Hint about the <code>abstract</code> package	194
5.76.1.10	Hint about the alteration of the sectionning commands	194
5.76.1.10.1	Alteration of <code>\part</code>	194
5.76.1.10.2	Alteration of <code>\chapter</code>	195
5.76.1.10.3	Alteration of <code>\section</code>	195
5.76.1.11	Hint about the coherence of the calling sequences of some commands	196
5.76.2	Final part : <code>\mtc@hints@enddoc</code>	197
5.76.2.1	Hint about <code>\sect-lof lot</code> and the <code>insection</code> option	197
5.76.2.2	Final part of the coherence tests	198
5.76.2.3	Check if the main tables have been prepared (first part)	199
5.76.2.4	Check if the main tables have been prepared (second part)	201
5.76.2.5	Check the number of mini-tables, in case of short extensions	203
5.76.2.6	Final part of the hint about the <code>sectsty</code> package	204
5.76.2.7	Check if empty mini-tables have been detected	204
5.76.2.8	Check if obsolete commands have been used	206
5.76.2.9	Check if some hints have been written	207
5.77	Processing of options	207
5.77.1	Processing the <code>insection</code> option	208
5.77.2	Processing the <code>notoccite</code> option	208
5.77.3	Processing the <code>hints</code> option	208
5.77.4	Saving the sectionning commands	208
5.78	The <code>fminitoc.dtx</code> file	209
<b>6</b>	<b>Commented code of <code>mtcoeff.sty</code></b>	<b>210</b>
6.1	Why <code>mtcoeff.sty</code> ?	210
6.2	Identification of the package	211

6.3	Faking counters and dimensions . . . . .	211
6.4	Faking simple user commands . . . . .	212
6.5	Faking user commands with one optional argument . . . . .	212
6.6	Disabling the internal commands . . . . .	213
6.7	Disabling the font commands . . . . .	214
6.8	Disabling the <code>\mtcset...</code> commands . . . . .	215
6.9	Disabling the new <code>\l@...</code> commands . . . . .	215
6.10	Ignore the obsolete commands . . . . .	215
6.11	Disabling the <code>\mtcselectlanguage</code> command . . . . .	216
6.12	Disabling the <code>\mtcloadmlo</code> command . . . . .	216
6.13	Disabling the commands for the horizontal rules . . . . .	216
6.14	Disabling the commands for the page numbers . . . . .	217
6.15	Disabling the mini-table features commands . . . . .	217
6.16	Disabling miscellaneous flags and commands . . . . .	218
6.17	Caution for some commands . . . . .	218
6.18	Disabling commands for “coffee” . . . . .	219
6.19	Disabling the <code>mtchideinmain...</code> environments . . . . .	220
6.20	Disabling the <code>\mtcfixindex</code> command . . . . .	220
6.21	Disabling the <code>\mtcfixglossary</code> command . . . . .	220
6.22	Disabling the <code>\addstarred...</code> commands . . . . .	220
<b>7</b>	<b>Patch for the <code>memoir</code> class</b>	<b>222</b>
<b>8</b>	<b>The minitoc language definition (<code>.mld</code>) and object (<code>.mlo</code>) files</b>	<b>224</b>
8.1	“Acadian” language : <code>acadian.mld</code> . . . . .	227
8.2	“Acadien” language : <code>acadien.mld</code> . . . . .	228
8.3	“Afrikaan” language : <code>afrikaan.mld</code> . . . . .	228
8.4	“Afrikaans” language : <code>afrikaans.mld</code> . . . . .	228
8.5	“American” language : <code>american.mld</code> . . . . .	229
8.6	“Arab” language : <code>arab.mld</code> . . . . .	229
8.7	“Arabic” language : <code>arabic.mld</code> . . . . .	229
8.8	“Armenian” language : <code>armenian.mld</code> . . . . .	230
8.9	“Austrian” language : <code>austrian.mld</code> . . . . .	230
8.10	“Bahasa” language : <code>bahasa.mld</code> . . . . .	230
8.11	“Bangla” language : <code>bangla.mld</code> . . . . .	231
8.12	“Basque” language : <code>basque.mld</code> . . . . .	231
8.13	“Bicig” language : <code>bicig.mld</code> . . . . .	232
8.14	“Brazil” language : <code>brazil.mld</code> . . . . .	232
8.15	“brazilian” language : <code>brazilian.mld</code> . . . . .	233
8.16	“Breton” language : <code>breton.mld</code> . . . . .	233
8.17	“british” language : <code>british.mld</code> . . . . .	234
8.18	“Bulgarian” language : <code>bulgarian.mld</code> . . . . .	234
8.19	“Bulgarianb” language : <code>bulgarianb.mld</code> . . . . .	235
8.20	“Buryat” language : <code>buryat.mld</code> . . . . .	235
8.21	“Canadian” language : <code>canadian.mld</code> . . . . .	236
8.22	“Canadien” language : <code>canadien.mld</code> . . . . .	236
8.23	“Castillan” language : <code>castillan.mld</code> . . . . .	236
8.24	“Castillian” language : <code>castillian.mld</code> . . . . .	237
8.25	“Catalan” language : <code>catalan.mld</code> . . . . .	237
8.26	“Chinese1” language : <code>chinese1.mld</code> and <code>chinese1.mlo</code> . . . . .	237



8.27	“Chinese2” language : chinese2.mld and chinese2.mlo . . . . .	238
8.28	“Croatian” language : croatian.mld . . . . .	238
8.29	“Czech” language : czech.mld . . . . .	239
8.30	“Danish” language : danish.mld . . . . .	239
8.31	“Dutch” language : dutch.mld . . . . .	240
8.32	“English” language : english.mld . . . . .	240
8.33	“Esperant” language : esperant.mld . . . . .	241
8.34	“Esperanto” language : esperanto.mld . . . . .	241
8.35	“Estonian” language : estonian.mld . . . . .	241
8.36	“Ethiopia” language : ethiopia.mld . . . . .	242
8.37	“Ethiopian” language : ethiopian.mld . . . . .	243
8.38	“Farsi1” language : farsi1.mld and farsi1.mlo . . . . .	243
8.39	“Farsi2” language : farsi2.mld and farsi2.mlo . . . . .	243
8.40	“Finnish” language : finnish.mld . . . . .	244
8.41	“Finnish2” language : finnish2.mld . . . . .	244
8.42	“Francais” language : francais.mld . . . . .	245
8.43	“French” language : french.mld . . . . .	245
8.44	“Frenchb” language : frenchb.mld . . . . .	245
8.45	“Frenchle” language : frenchle.mld . . . . .	246
8.46	“Frenchpro” language : frenchpro.mld . . . . .	246
8.47	“Galician” language : galician.mld . . . . .	246
8.48	“German” language : german.mld . . . . .	247
8.49	“Germanb” language : germanb.mld . . . . .	247
8.50	“Greek” language : greek.mld . . . . .	248
8.51	“Greek-mono” language : greek-mono.mld . . . . .	248
8.52	“Greek-polydemo” language : greek-polydemo.mld . . . . .	249
8.53	“Greek-polykatha” language : greek-polykatha.mld . . . . .	250
8.54	“Guarani” language : guarani.mld . . . . .	251
8.55	“Hangul1” language : hangul1.mld and hangul1.mlo . . . . .	252
8.56	“Hangul2” language : hangul2.mld and hangul2.mlo . . . . .	252
8.57	“Hangul3” language : hangul3.mld and hangul3.mlo . . . . .	253
8.58	“Hangul4” language : hangul4.mld and hangul4.mlo . . . . .	253
8.59	“Hanja1” language : hanja1.mld and hanja1.mlo . . . . .	254
8.60	“Hanja2” language : hanja2.mld and hanja2.mlo . . . . .	255
8.61	“Hebrew” language : hebrew.mld . . . . .	255
8.62	“Hungarian” language : hungarian.mld . . . . .	256
8.63	“Icelandic” language : icelandic.mld . . . . .	256
8.64	“Interlingua” language : interlingua.mld . . . . .	257
8.65	“Irish” language : irish.mld . . . . .	257
8.66	“Italian” language : italian.mld . . . . .	257
8.67	“Japanese” language : japanese.mld and japanese.mlo . . . . .	258
8.68	“Japanese2” language : japanese2.mld and japanese2.mlo . . . . .	258
8.69	“Japanese3” language : japanese3.mld and japanese3.mlo . . . . .	259
8.70	“Japanese4” language : japanese4.mld and japanese4.mlo . . . . .	260
8.71	“Japanese5” language : japanese5.mld and japanese.mlo . . . . .	260
8.72	“Latin” language : latin.mld . . . . .	261
8.73	“Latin2” language : latin2.mld . . . . .	261
8.74	“Latvian” language : latvian.mld . . . . .	262
8.75	“Letton” language : letton.mld . . . . .	262
8.76	“Lithuanian” language : lithuanian.mld . . . . .	262

8.77	“Lsorbian” language : lsorbian.mld . . . . .	263
8.78	“Magyar” language : magyar.mld . . . . .	263
8.79	“Magyar2” language : magyar2.mld . . . . .	264
8.80	“Malayalam-keli” language : malayalam-keli.mld . . . . .	264
8.81	“Malayalam-rachana” language : malayalam-rachana.mld . . . . .	265
8.82	“Malayalam-rachana2” language : malayalam-rachana2.mld . . . . .	265
8.83	“Mongol” language : mongol.mld . . . . .	266
8.84	“Naustrian” language : naustrian.mld . . . . .	266
8.85	“Ngerman” language : ngerman.mld . . . . .	267
8.86	“Ngermanb” language : ngermanb.mld . . . . .	267
8.87	“Norsk” language : norsk.mld . . . . .	268
8.88	“Nynorsk” language : nynorsk.mld . . . . .	268
8.89	“Polish” language : polish.mld . . . . .	269
8.90	“Polish2” language : polish2.mld . . . . .	269
8.91	“Portuges” language : portuges.mld . . . . .	270
8.92	“Portuguese” language : portuguese.mld . . . . .	270
8.93	“Romanian” language : romanian.mld . . . . .	270
8.94	“Russian” language : russian.mld . . . . .	271
8.95	“Russian2m” language : russian2m.mld . . . . .	271
8.96	“Russian2o” language : russian2o.mld . . . . .	272
8.97	“Russianb” language : russianb.mld . . . . .	273
8.98	“Russianc” language : russianc.mld . . . . .	274
8.99	“Samin” language : samin.mld . . . . .	275
8.100	“Scottish” language : scottish.mld . . . . .	275
8.101	“Serbian” language : serbian.mld . . . . .	276
8.102	“Serbianc” language : serbianc.mld . . . . .	276
8.103	“Slovak” language : slovak.mld . . . . .	277
8.104	“Slovene” language : slovene.mld . . . . .	277
8.105	“Spanish” language : spanish.mld . . . . .	278
8.106	“Spanish2” language : spanish2.mld . . . . .	278
8.107	“Spanish3” language : spanish3.mld . . . . .	279
8.108	“Swedish” language : swedish.mld . . . . .	279
8.109	“Thai” language : thai.mld and thai.mlo . . . . .	280
8.110	“Turkish” language : turkish.mld . . . . .	280
8.111	“UKenglish” language : UKenglish.mld . . . . .	280
8.112	“Ukraineb” language : ukraineb.mld . . . . .	281
8.113	“USenglish” language : USenglish.mld . . . . .	281
8.114	“Usorbian” language : lsorbian.mld . . . . .	282
8.115	“Vietnam” language : vietnam.mld . . . . .	282
8.116	“Vietnamese” language : vietnamese.mld . . . . .	283
8.117	“Welsh” language : welsh.mld . . . . .	283

<b>Complements</b>	<b>285</b>
--------------------	------------

<b>Bibliography</b>	<b>285</b>
---------------------	------------

<b>Change History</b>	<b>290</b>
-----------------------	------------

*CONTENTS*

11

**Index**

**299**

# List of Figures

1.1	Layout of a ToC (LoF, LoT) entry . . . . .	32
2.1	Three compilations for <code>mini toc</code> . . . . .	44
5.1	Hack to detect the limitation to short extensions. . . . .	84

# List of Tables

1.1	Commands for a <code>minitoc</code>	19
1.2	Commands for a <code>secttoc</code>	21
1.3	Commands for a <code>parttoc</code>	22
1.4	Commands to add an entry in the table of contents for a starred part, chapter, or section.	24
1.5	Fonts and titles for the mini-table commands.	25
1.6	Fonts for the mini-table entries.	26
1.7	Available languages	27
1.8	Horizontal rules	33
1.9	Page numbers	33
1.10	Features for <code>parttoc-s</code> and other mini-tables	35
1.11	Extensions of the auxiliary files	40
3.1	List of files ( <code>minitoc.1</code> )	57
4.1	Package options	58
4.2	General commands	59
4.3	Commands, part level	60
4.4	Commands, chapter level	61
4.5	Commands, section level	62
4.6	Commands for horizontal rules	62
4.7	Commands for page numbers	63
4.8	Specific commands for <code>parttocs</code> and other mini-tables features	63
4.9	Preparation and insertion commands	64
4.10	Adjustment commands	64
4.11	Obsolete commands	64

FIX INDEX? FIX INDEX?

## **Part I**

# **User's Manual**

# Table of Contents

<b>1</b>	<b>The minitoc package</b>	<b>16</b>
<b>2</b>	<b>Frequently Asked Questions</b>	<b>42</b>
<b>3</b>	<b>Installation</b>	<b>54</b>
<b>4</b>	<b>Memento</b>	<b>58</b>
	<b>Acknowledgments</b>	<b>65</b>

# Chapter 1

## The minitoc package

### Contents

---

<b>1.1</b>	<b>Introduction</b>	<b>17</b>
1.1.1	License	18
<b>1.2</b>	<b>How to use the minitoc package</b>	<b>18</b>
1.2.1	Loading the package and creating the mini-tables	18
1.2.2	Preparing the mini-tables	19
1.2.3	Placing the mini-tables	20
1.2.4	Starred chapters, parts and sections	23
<b>1.3</b>	<b>Typesetting of the mini-tables</b>	<b>24</b>
1.3.1	Chapter-level mini-tables	24
1.3.2	Titles for chapter-level mini-tables	25
1.3.3	Part-level mini-tables	26
1.3.4	Titles for part-level mini-tables	26
1.3.5	Section-level mini-tables	28
1.3.6	Titles for section-level mini-tables	28
1.3.7	Position of the titles	28
1.3.7.1	For mini-tables at the part level	28
1.3.7.2	For mini-tables at the chapter level	28
1.3.7.3	For mini-tables at the section level	29
1.3.7.4	Summary of the positioning of titles	29
1.3.8	Line spacing in the mini-tables	29
1.3.9	Simplified commands for fonts	30
1.3.10	Simplified command for mini-table titles	30
1.3.11	Simplified command for mini-table depths	31
1.3.12	Languages for the titles	31
1.3.13	Altering the layout of the mini-tables ( <i>experimental feature</i> )	31
<b>1.4</b>	<b>Special Features</b>	<b>33</b>
1.4.1	Horizontal Rules	33
1.4.2	Page Numbers, Leaders	34
1.4.3	Features for parttocs and other mini-tables	34
1.4.3.1	Remark about page styles	36
1.4.4	The “Chapter 0” Problem (solved)	36
1.4.4.1	Before version #23 (1994/11/08)	36
1.4.4.2	Since version #23 (1994/11/08)	37



1.4.5	Special Entries for TOC, LOF, LOT, Bibliography and Index	37
1.4.6	The <code>notoccite</code> option	38
<b>1.5</b>	<b>The <code>hints</code> option</b>	<b>38</b>
<b>1.6</b>	<b>Usage with MS-DOS</b>	<b>40</b>
<b>1.7</b>	<b>Why several <math>\LaTeX</math> runs are required?</b>	<b>40</b>
<b>1.8</b>	<b>The <code>mtcoff</code> package</b>	<b>41</b>

## Figures

1.1	Layout of a ToC (LoF, LoT) entry	32
-----	----------------------------------	----

## Tables

1.1	Commands for a <code>minitoc</code>	19
1.2	Commands for a <code>secttoc</code>	21
1.3	Commands for a <code>parttoc</code>	22
1.4	Commands to add an entry in the table of contents for a starred part, chapter, or section.	24
1.5	Fonts and titles for the mini-table commands.	25
1.6	Fonts for the mini-table entries.	26
1.7	Available languages	27
1.8	Horizontal rules	33
1.9	Page numbers	33
1.10	Features for <code>parttoc</code> -s and other mini-tables	35
1.11	Extensions of the auxiliary files	40

## 1.1 Introduction

The `minitoc` package, initially written by Nigel WARD and Dan JURAFSKY, has been almost completely redesigned by Jean-Pierre F. DRUCBERT (ONERA/Centre de Toulouse). This package creates a mini-table of contents (a “minitoc”<sup>1</sup>) at the beginning of each chapter of a document. It is also possible to have a mini-list of figures (a “minilof”) and a mini-list of tables (a “minilot”). The document class should of course define chapters (classes like `book` or `report`) or sections (classes like `article`<sup>2</sup>). Thus, this package should not be used with document classes without standard sectionning commands (like `letter`). When the document class defines a “part” sectionning level (*i.e.* classes like `book`, `report` and `article`), you can create a “partial” table of contents (a “parttoc”) at the beginning of each part of a document. It is also possible to have a partial list of figures (a “partlof”) and a partial list of tables (a “partlot”). When the document class has no `\chapter` command but has a `\section` command, you may use section level tables of contents (“secttocs”) at the beginning of each section; and you can also have section level lists of figures (“sectlofs”) or of tables (“sectlots”).

<sup>1</sup>The `minitoc` package introduces its own jargon, explained in this document. It should not be too difficult, however, to learn and use; it will be used here, of course.

<sup>2</sup>As the standard `proc` class loads the standard `article` class, this class will be just considered as a variant of the `article` class.

All these tables (“minitocs”, “partlots”, “sectlofs”, etc.) are collectively referenced as “mini-tables” (or sometimes “mini-lists”).



**Note:** you cannot use chapter level and section level mini-tables in the same document. This restriction is intended to avoid documents with full of local tables of contents, lists of figures and tables at every sectionning level.



**Note:** the commands relative to the part level are defined only if the document class defines `\part`. The commands relative to the section level are defined only if the document class defines `\section` but does not define `\chapter`.

The current version of this package is #43.

### 1.1.1 License

This package must be distributed and/or may be modified under the conditions of the **L<sup>A</sup>T<sub>E</sub>X Project Public License**, either version 1.3 of this license or (as convenient) any later version. The latest version of this license is in

<http://www.latex-project.org/lppl.txt>

and version 1.3 or later is part of all distributions of L<sup>A</sup>T<sub>E</sub>X version 2003/12/01 or later.

But please don’t bother me about hacked versions; they will not be supported.

## 1.2 How to use the minitoc package

### 1.2.1 Loading the package and creating the mini-tables

`\usepackage` To use the minitoc package, you must introduce a command:  
`\minitoc`

```
\usepackage[...options...]{minitoc}
```

in the preamble of the document<sup>3</sup>. The mini-table of contents will be in the chapter, after the `\chapter` command, at the point of the `\minitoc` command. The `\minitoc` command may occur *almost anywhere*<sup>4</sup> inside a chapter. Of course, it is better to put

<sup>3</sup>This command must be placed *after* any modification done on the sectionning commands; if you modify sectionning commands after loading the minitoc package, this one might not work properly.

<sup>4</sup>“Almost anywhere” means “in a normal place”, like between two paragraphs of normal text, or in a (wide enough) minipage, but not in a too strange position (like a marginal note or a footnote). Even a multicolumn or a floating environment can be used, but with care. But note that a minitoc can be rather long, if the chapter is complex and if you ask details with a high value for `minitocdepth`.

Table 1.1: Commands for a minitoc

\documentclass[...]{book}		
\usepackage[...options...]{minitoc}		
...		
\setlength{\mtcindent}{24pt}	default	
\setlength{\mtcskipamount}{\bigskipamount}	default	
...		
\setcounter{minitocdepth}{2}	default	
\renewcommand{\mtcfont}{\small\rmfamily\upshape\mdseries}		default
\renewcommand{\mtcSfont}{\small\rmfamily\upshape\bfseries}		default
or:		
\mtcsetdepth{minitoc}{2}		default
\mtcsetfont{minitoc}{*}{\small\rmfamily\upshape\mdseries}		default
\mtcsetfont{minitoc}{section}{\small\rmfamily\upshape\bfseries}		default
...		
\begin{document}		
...		
\dominitoc		
\dominilof		
\dominilot		
\tableofcontents	or \faketableofcontents	
\listoffigures	or \fakelistoffigures	
\listoftables	or \fakelistoftables	
...		
\chapter{...}		
\minitoc	if you want one	
\mtcskip		
\minilof	if you want one	
\mtcskip		
\minilot	if you want one	
...		


it at the beginning of the chapter, eventually after some introductory material. But you can also decide to put it at the end of the chapter. You should use the same conventions in all chapters. If you want to add the mini-table of contents for a chapter, you must use the sequence given in table 1.1. For each mini-table of contents, an auxiliary file will be created with a name of the form  $\langle document \rangle.mtc\langle N \rangle$ , where  $\langle N \rangle$  is the absolute chapter number. “Absolute” means that this number is unique, and always increasing from the first chapter<sup>5</sup>. The suffix is  $.mlf\langle N \rangle$  for mini-lists of figures and is  $.mlt\langle N \rangle$  for mini-lists of tables. (If under MS-DOS or any operating system with short extensions to filenames, see section 1.6 on page 40 and section 2.5 on page 44). There are similar commands for mini-tables at the part or section level, depending of the document class.

## 1.2.2 Preparing the mini-tables


\dominitoc    The commands<sup>6</sup> \dominitoc, \dominilof, and \dominilot (for mini-tables at the  
\dominilof  
\dominilot

<sup>5</sup>The concept of an “absolute” counter for the mini-tables has solved some obscure problems, and also made obsolete some commands, like \firstpartis, \firstchapteris, and \firstsectionis.

chapter level), take the  $\langle document \rangle.toc$ ,  $\langle document \rangle.lof$ , and  $\langle document \rangle.lot$  files, respectively, and cut slices from them to create the  $\langle document \rangle.mtc\langle N \rangle$ ,  $\langle document \rangle.mlf\langle N \rangle$ , and  $\langle document \rangle.mlt\langle N \rangle$  files.

$\backslash dosecttoc$	The commands $\backslash dosecttoc$ , $\backslash dosectlof$ , and $\backslash dosectlot$ (for mini-tables at the section level) and $\backslash doparttoc$ , $\backslash dopartlof$ , and $\backslash dopartlot$ (for mini-tables at the part level) are analog.
$\backslash dosectlof$	
$\backslash dosectlot$	
$\backslash doparttoc$	
$\backslash dopartlof$	To obtain a satisfactory result ( <i>i.e.</i> non empty), please note that all these commands must <i>imperatively be put before</i> any command analog to the $\backslash tableofcontents$ , $\backslash listoffigures$ , and $\backslash listoftables$ commands, or their $\backslash fake\dots$ siblings.
$\backslash dopartlot$	
$\backslash tableofcontents$	
$\backslash listoffigures$	
$\backslash listoftables$	It is also <i>strongly</i> recommended to put these commands <i>before</i> any sectionning command producing an entry in the table of contents (for the $\backslash do\dots toc$ commands), and <i>before</i> any $\backslash caption$ -like command producing an entry in the list of figure (for the $\backslash do\dots lof$ commands) or in the list of tables (for the $\backslash do\dots lot$ commands); else disorder in the mini-tables may result.
	

### 1.2.3 Placing the mini-tables

$\backslash mtcskip$	The $\backslash mtcskip$ command may be used to add a vertical skip between two mini-tables. Its height is $\backslash mtcskipamount$ (equal to $\backslash bigskipamount$ by default). $\backslash mtcskip$ eliminates any immediate previous vertical skip, to not accumulate vertical space when a mini-table is empty and skipped by the <code>checkfiles</code> option.
$\backslash mtcskipamount$	
$\backslash bigskipamount$	
$\backslash secttoc$	The section-level table of contents will be in the section, after the $\backslash section$ command, at the point of the $\backslash secttoc$ command. The $\backslash secttoc$ command may occur <i>almost anywhere</i> inside a section. Of course, it is better to put it at the beginning of the section, or after some short introductory material. You should use the same conventions in all sections. If you want to add the section-level table of contents for a section, you must use the sequence given in Table 1.2 on the next page. For each section-level table of contents, an auxiliary file will be created with a name of the form $\langle document \rangle.stc\langle N \rangle$ , where $\langle N \rangle$ is the absolute section number. The suffix is $.slf\langle N \rangle$ for section-level lists of figures and is $.slt\langle N \rangle$ for section-level lists of tables. (If under MS-DOS or any operating system with short extensions to filenames, see section 1.6 on page 40 and section 2.5 on page 44).
$\backslash section$	
$\backslash usepackage$	As floats (figures and tables) could drift somewhere outside the printing area of the text of the section, the sectlofs and sectlots can be rather strange. In order to have a better behaviour of these mini-tables, it may be useful to add the <code>insection</code> option in the $\backslash usepackage$ command:
$\backslash FloatBarrier$	
	

$\backslash usepackage[insection]{minitoc}$

if you want more coherent sectlofs and sectlots. Sometimes, it might be necessary to use the  $\backslash FloatBarrier$  command of this package to correctly place the figure or ta-

<sup>6</sup>The code of these  $\backslash do\dots$  commands is directly derived from that of the `xr` package [12], by David CARLISLE, with his permission.

Table 1.2: Commands for a secttoc

\documentclass[...]{article}		
\usepackage[...options...]{minitoc}		
...		
\setlength{\stcindent}{24pt}	default	
...		
\setcounter{secttocdepth}{2}	default	
\renewcommand{\stcfont}{\small\rmfamily\upshape\mdseries}		default
\renewcommand{\stcSSfont}{\small\rmfamily\upshape\bfseries}		default
or:		
\mtcsetdepth{secttoc}{2}		default
\mtcsetfont{secttoc}{*}{\small\rmfamily\upshape\mdseries}		default
\mtcsetfont{secttoc}{subsection}{\small\rmfamily\upshape\bfseries}		default
...		
\begin{document}		
...		
\dosecttoc		
\dosectlof		
\dosectlot		
\tableofcontents	or \faketableofcontents	
\listoffigures	or \fakelistoffigures	
\listoftables	or \fakelistoftables	
...		
\section{...}		
\secttoc	if you want one	
\sectlof	if you want one	
\sectlot	if you want one	
...		

ble and have a correct mini-table. The `insection` option loads the `placeins` package [3] with its `verbose`, `section` and `bottom` options. The `placeins` package, by Donald ARSENAU, is available on CTAN archives; `placeins.sty` contains its own documentation, with a copy in `placeins.txt`. You need a version whose date is at least 2005/04/18.

If you want to add the partial table of contents for a part, you must use the sequence given in Table 1.3 on the following page. For each partial table of contents, an auxiliary file will be created with a name of the form `<document>.ptc<N>`, where `<N>` is the absolute part number. The suffix is `.plf<N>` for partial lists of figures and is `.plt<N>` for partial lists of tables. (If under MS-DOS or any operating system with short extensions to filenames, see section 1.6 on page 40 and section 2.5 on page 44).



**Note:** the user is responsible of asking or not asking a mini-table (mini-toc, -lof or -lot) for some chapter. Asking a `minilof` for a chapter without any figure would result in an empty and ugly mini-list of figures (*i.e.* the title and two horizontal rules). He is also responsible of requiring or not requiring a partial toc (lof or lot) for some part. Asking a `partlof` for a part without any figure would result in an empty and ugly part list of figures (*i.e.* the title alone on a page). Analogous remarks apply to section-level mini-tables (`secttoc`, `sectlof`, and `sectlot`) and to the part-level mini-tables (`parttoc`, `partlof`, and `partlot`).

Table 1.3: Commands for a parttoc

\documentclass[...]{book}		
\usepackage[...options...]{minitoc}		
...		
\setlength{\ptcindent}{0pt}	default	
...		
\setcounter{parttocdepth}{2}	default	
\renewcommand{\ptcfont}{\normalsize\rmfamily\upshape\mdseries}		default
\renewcommand{\ptcCfont}{\normalsize\rmfamily\upshape\bfseries}		default
\renewcommand{\ptcSfont}{\normalsize\rmfamily\upshape\mdseries}		default
or:		
\mtcsetdepth{parttoc}{2}		default
\mtcsetfont{parttoc}{*}{\normalsize\rmfamily\upshape\mdseries}		default
\mtcsetfont{parttoc}{chapter}{\normalsize\rmfamily\upshape\bfseries}		default
\mtcsetfont{parttoc}{section}{\normalsize\rmfamily\upshape\mdseries}		default
...		
\begin{document}		
...		
\doparttoc		
\dopartlof		
\dopartlot		
\tableofcontents	or \faketableofcontents	
\listoffigures	or \fakelistoffigures	
\listoftables	or \fakelistoftables	
...		
\part{...}		
\parttoc	if you want one	
\partlof	if you want one	
\partlot	if you want one	
...		

But since version #35, empty mini-tables are just ignored and this problem should disappear in normal circumstances. Nevertheless, it is recommended to put no `\minitoc` command in a chapter without sections and no `\minilof` or `\minilot` command in a chapter without figures or tables. The `checkfiles` (see section 1.2.3 on page 20) package option (default) skips empty mini-tables (with a note in the `.log` file); the `nocheckfiles` package option restores the old behaviour (empty mini-tables are displayed).

By default, the mini-tables and partial tables of contents contain only references higher and to sections and subsections. The counters `parttocdepth`, `minitocdepth` and `secttocdepth`, similar to `tocdepth`, allow the user to modify this behaviour. Mini or partial lists of figures or tables are not affected by the value of these counters, but if there are depth counters for these lists (`lofdepth` and `lotdepth`), as done by the `subfig` package [13] from Steven Douglas COCHRAN, new depth counters are created if necessary, with obvious names like `partlofdepth`, `partlotdepth`, `minilofdepth`, `minilotdepth`, `sectlofdepth`, and `sectlotdepth`.

## 1.2.4 Starred chapters, parts and sections

`\addstarredpart`  
`\addstarredchapter`  
`\addstarredsection`

**NOTE:** if using `\chapter*` and a

`\addcontentsline{toc}{chapter}{...}`



command to add something in the table of contents, the numbering of the minitoc auxiliary files would be altered. To avoid that problem, say:

```
\addstarredpart{...}
\addstarredchapter{...}
\addstarredsection{...}
```

`\addcontentsline`

These commands apply only for the level of a part-, mini- or sect-toc; for lower levels, use, as usual:

`\addcontentsline{toc}{section}{...}`

for example, to add a section-level entry in the global toc and in the minitoc of a starred chapter:

```
\chapter*{Title of chapter}
\addstarredchapter{Title of chapter}
\minitoc
\section*{First section}
\addcontentsline{toc}{section}{First section}
\section*{Second section}
\addcontentsline{toc}{section}{Second section}
```

`\adjustptc`  
`\adjustmtc`  
`\adjuststc`

There is sometimes a problem with mini-tables when you use `\chapter*` (or `\section*`): the minitocs appear in the wrong chapter. You can add a `\adjustmtc` (or `\adjuststc` or `\adjustptc`) command at the end of the starred chapter (or section or part) to increment the corresponding counter. Do not use commands like `\stepcounter{mtc}` or `\addtocounter{mtc}{...}` (which should work, but it is cheating), because the `mtcoff` package (see section 1.8 on page 41) knows what to do about `\adjustmtc` (and others), but can do nothing about `\stepcounter` or `\addtocounter`, as they are a standard basic commands of  $\text{\LaTeX}$ , not minitoc specific commands. Syntax:

```
\adjustptc[n]
\adjustmtc[n]
\adjuststc[n]
```

where  $n$  is the increment (default: 1).

Table 1.4: Commands to add an entry in the table of contents for a starred part, chapter, or section.

Level	With title
part	<code>\mtcaddpart[⟨title⟩]</code>
chapter	<code>\mtcaddchapter[⟨title⟩]</code>
section	<code>\mtcaddsection[⟨title⟩]</code>

`\decrementptc` There are similar commands to *decrement* or *increment* by 1 these counters: `\decrementptc`, `\decrementmtc`, `\decrementstc`, `\incrementptc`, `\incrementmtc`, and `\incrementstc`; the same remarks as above apply.

`\mtcaddpart` A more clever way to solve this problem is to use commands similar to:  
`\mtcaddchapter`  
`\mtcaddsection`

`\mtcaddchapter[⟨title⟩]`

This command adds an entry in the table of contents (and adjusts the counter, because it calls `\adjustmtc`). The table 1.4 summarizes these commands, that you put *after* `\chapter*`, etc. If the optional argument is omitted or empty or blank, no entry will be visible in the table of contents nor in the minitocs. If the optional argument is something invisible (like `~`, `\space` or `\quad`), the result will be strange but still logically correct.

## 1.3 Typesetting of the mini-tables

The mini-tables are typeset in a verse-like environment, and can be split over pages.

### 1.3.1 Chapter-level mini-tables

`\mtcfont` The mini-table of contents is typeset in the `\mtcfont` font, which is  
`\mtcSfont` `\small\rmfamily` by default. Section entries are typeset in the `\mtcSfont` font,  
`\mtcSSfont` which is `\small\bfseries` by default. For subsections, subsubsections, paragraphs  
`\mtcSSSfont` and subparagraphs, the commands `\mtcSSfont`, `\mtcSSSfont`, `\mtcPfont` and  
`\mtcPfont` `\mtcSPfont` are available (by default, `\small\rmfamily`) to enable the use of various  
`\mtcSPfont` fonts. Mini lists of figures and tables are typeset in the fonts `\mlffont` and `\mltfont`,  
`\mlffont` which are `\small\rmfamily` by default. Tables 1.5 on the following page and 1.6 on  
`\mltfont` page 26 summarize these many commands<sup>7</sup>.



Note that the default choice of fonts is certainly not perfect and hence is not definitive. A symptom of this imperfection is the presence of poor alignments in the mini-tables, if

<sup>7</sup>Thanks to Stefan ULRICH, who contributed these tables initially.



Table 1.5: Fonts and titles for the mini-table commands.

Command	Font default setting	Title string default setting	Title font default setting
For the <code>\part...</code> commands:			
<code>\parttoc</code>	<code>\ptcfont</code> <code>\normalsize\rmfamily*</code> <code>\small\rmfamily**</code>	<code>\ptctitle</code> Table of Contents <sup>†</sup>	<code>\ptifont</code> <code>\LARGE\bfseries*</code> <code>\Large\bfseries**</code>
<code>\partlof</code>	<code>\plffont</code> <code>\normalsize\rmfamily*</code> <code>\small\rmfamily**</code>	<code>\plftitle</code> List of Figures <sup>†</sup>	<code>\ptifont</code> <code>\LARGE\bfseries*</code> <code>\Large\bfseries**</code>
<code>\partlot</code>	<code>\pltfont</code> <code>\normalsize\rmfamily*</code> <code>\small\rmfamily**</code>	<code>\plttitle</code> List of Tables <sup>†</sup>	<code>\ptifont</code> <code>\LARGE\bfseries*</code> <code>\Large\bfseries**</code>
For the <code>\mini...</code> commands:*			
<code>\minitoc</code>	<code>\mtcfont</code> <code>\small\rmfamily</code>	<code>\mtctitle</code> Contents <sup>†</sup>	<code>\mtifont</code> <code>\large\bfseries</code>
<code>\minilof</code>	<code>\mlffont</code> <code>\small\rmfamily</code>	<code>\mlftitle</code> Figures <sup>†</sup>	<code>\mtifont</code> <code>\large\bfseries</code>
<code>\minilot</code>	<code>\mltfont</code> <code>\small\rmfamily</code>	<code>\plttitle</code> Tables <sup>†</sup>	<code>\mtifont</code> <code>\large\bfseries</code>
For the <code>\sect...</code> commands:**			
<code>\secttoc</code>	<code>\stcfont</code> <code>\small\rmfamily</code>	<code>\stctitle</code> Contents <sup>†</sup>	<code>\stifont</code> <code>\Large\bfseries</code>
<code>\sectlof</code>	<code>\slffont</code> <code>\small\rmfamily</code>	<code>\mlftitle</code> Figures <sup>†</sup>	<code>\stifont</code> <code>\Large\bfseries</code>
<code>\sectlot</code>	<code>\sltfont</code> <code>\small\rmfamily</code>	<code>\plttitle</code> Tables <sup>†</sup>	<code>\stifont</code> <code>\Large\bfseries</code>

\*for document classes with `\chapter` level (e.g. book, report).\*\*for document classes with no `\chapter` level (e.g. article).<sup>†</sup>default for english; changed by the language definition files or `\renewcommand`All these fonts use `\rmfamily`, `\upshape`, and `\mdseries` by default.

bold and non-bold font are mixed<sup>8</sup> (the true length of 1em is not the same for the fonts). This can eventually be adjusted by changing some fonts.

### 1.3.2 Titles for chapter-level mini-tables

`\mtifont` Titles are typeset in the `\mtifont` (`\large\bfseries` by default) font and the text  
`\mtctitle` strings of the titles are defined by `\mtctitle`, `\mlftitle` and `\mlttitle`, which  
`\mlftitle` are the strings “Contents”, “Figures” and “Tables” by default. These title commands  
`\mlttitle` should be redefined by `\renewcommand` or `\mtcsetttitle` for languages other than  
`\mtcsetttitle` english. The language option files like `french.mld` and `english.mld` (the suffix  
`\mtcselectlanguage` .mld means “minitoc language definition (file)”) (and others, see table 1.7 on page 27  
and section 1.3.12 on page 31) are available. You can easily prepare a similar file for  
a preferred language (see section 2.26 on page 51). You can change the language of  
these titles by using the `\mtcselectlanguage{language}` macro.

<sup>8</sup>This appears e.g. if you are using the Computer Modern Roman (CMR) fonts. The symptom disappears if you do not use bold CMR fonts or if you use the TX fonts (txfonts package [40]), by example, like in this document. See also section 2.29 on page 52.

Table 1.6: Fonts for the mini-table entries.

Level	Font	default setting
For the <code>\parttoc</code> entries:		
Chapter*	<code>\ptcCfont*</code>	<code>\normalsize\bfseries*</code>
Section	<code>\ptcSfont</code>	<code>\normalsize\rmfamily*</code> <code>\small\bfseries**</code>
Subsection	<code>\ptcSSfont</code>	<i>(like \ptcfont)</i>
Subsubsection	<code>\ptcSSfont</code>	<i>(like \ptcfont)</i>
Paragraph	<code>\ptcPfont</code>	<i>(like \ptcfont)</i>
Subparagraph	<code>\ptcSPfont</code>	<i>(like \ptcfont)</i>
For the <code>\minitoc</code> entries:*		
Section	<code>\mtcSfont</code>	<code>\small\bfseries</code>
Subsection	<code>\mtcSSfont</code>	<i>(like \mtcfont)</i>
Subsubsection	<code>\mtcSSfont</code>	<i>(like \mtcfont)</i>
Paragraph	<code>\mtcPfont</code>	<i>(like \mtcfont)</i>
Subparagraph	<code>\mtcSPfont</code>	<i>(like \mtcfont)</i>
For the <code>\secttoc</code> entries:**		
Subsection	<code>\stcSSfont</code>	<code>\normalsize\bfseries</code>
Subsubsection	<code>\stcSSfont</code>	<i>(like \stcfont)</i>
Paragraph	<code>\stcPfont</code>	<i>(like \stcfont)</i>
Subparagraph	<code>\stcSPfont</code>	<i>(like \stcfont)</i>

\*for document classes with `\chapter` level (e.g. book, report).  
\*\*for document classes with no `\chapter` level (e.g. article).

### 1.3.3 Part-level mini-tables

<code>\ptcfont</code>	The partial table of contents is typeset in the <code>\ptcfont</code> font, which is defined as
<code>\ptcCfont</code>	<code>\normalsize\rmfamily</code> by default. Chapter entries are typeset in the <code>\ptcCfont</code>
<code>\ptcSfont</code>	font, which is <code>\normalsize\bfseries</code> by default. Section entries are typeset in the
<code>\ptcSSfont</code>	<code>\ptcSfont</code> font, which is <code>\normalsize\rmfamily</code> by default. For subsections, sub-
<code>\ptcSSSfont</code>	subsections, paragraphs and subparagraphs, the commands <code>\ptcSSfont</code> , <code>\ptcSSSfont</code> ,
<code>\ptcPfont</code>	<code>\ptcPfont</code> , and <code>\ptcSPfont</code> are available (by default, <code>\normalsize\rmfamily</code> ) if
<code>\ptcSPfont</code>	you want to use various fonts. Partial lists of figures and tables are typeset in the fonts
<code>\plffont</code>	<code>\plffont</code> and <code>\pltfont</code> , which are <code>\normalsize\rmfamily</code> by default.
<code>\pltfont</code>	

### 1.3.4 Titles for part-level mini-tables

<code>\ptifont</code>	Titles are typeset in the <code>\ptifont</code> ( <code>\LARGE\bfseries</code> by default) font and the text
<code>\ptctitle</code>	strings of the titles are defined by <code>\ptctitle</code> , <code>\plftitle</code> and <code>\plttitle</code> , which
<code>\plftitle</code>	are the strings “Table of Contents”, “List of Figures” and “List of Tables” by default.
<code>\plttitle</code>	These title commands should be redefined by <code>\renewcommand</code> or <code>\mtcsettitle</code>
<code>\mtcsettitle</code>	for languages other than english. The language definition files like <code>french.mld</code>
<code>\mtcselectlanguage</code>	and <code>english.mld</code> (and many others, see table 1.7 on the following page and

Table 1.7: Available languages

1. afrikaan (afrikaans)	23. ethiopia (ethiopian) <sup>c</sup>	44. icelandic <sup>f</sup>	67. polish
2. arab (arabic) <sup>a,c</sup>	24. farsi1 <sup>c,f,g</sup>	45. interlingua	68. polish2 <sup>c,e</sup>
3. armenian <sup>c</sup>	25. farsi2 <sup>c,f,g</sup>	46. irish	69. portuguese (portuges)
4. bahasa <sup>c</sup>	26. finnish	47. italian	70. romanian
5. bangla <sup>c</sup>	27. finnish2	48. japanese <sup>c,d,g</sup>	71. russian <sup>b,c</sup>
6. basque	28. french (frenchb, frenchle, frenchpro, francais, acadien, canadien)	49. japanese2 <sup>c,d,g</sup>	72. russianb <sup>b,c</sup>
7. bicig <sup>c</sup>		50. japanese3 <sup>c,d,g</sup>	73. russianc <sup>b,c</sup>
8. brazil (brazilian)		51. japanese4 <sup>c,d,g</sup>	74. russian2m <sup>c,e</sup>
9. breton		52. japanese5 <sup>c,d,g</sup>	75. russian2o <sup>c,e</sup>
10. bulgarian <sup>c</sup>	29. galician	53. latin	76. samin
11. bulgarianb <sup>c</sup>	30. german (austrian)	54. latin2	77. scottish
12. buryat <sup>c</sup>	31. germanb	55. latvian (letton) <sup>e</sup>	78. serbian
13. catalan	32. greek <sup>c</sup>	56. lithuanian	79. serbianc <sup>c</sup>
14. chinese1 <sup>c,g</sup>	33. greek-mono <sup>c,e</sup>	57. lsorbian	80. slovak
15. chinese2 <sup>c,g</sup>	34. greek- polydemo <sup>c,e</sup>	58. magyar (hungarian)	81. slovene
16. croatian	35. greek- polykatha <sup>c,e</sup>	59. magyar2	82. spanish (castillan, castillian)
17. czech	36. guarani <sup>h</sup>	60. malayalam- keli <sup>c</sup>	83. spanish2
18. danish	37. hangul1 <sup>c,d,g</sup>	61. malayalam- rachana <sup>c</sup>	84. spanish3 <sup>e,f</sup>
19. dutch	38. hangul2 <sup>c,d,g</sup>	62. malayalam- rachana2 <sup>c</sup>	85. swedish
20. english (american, british, canadian, UKenglish, USenglish)	39. hangul3 <sup>c,d,g</sup>	63. mongol <sup>c</sup>	86. thai <sup>c,d,f,g</sup>
21. esperant (esperanto)	40. hangul4 <sup>c,d,g</sup>	64. ngermanb (ngerman, naustrian)	87. turkish
22. estonian	41. hanja1 <sup>c,d,g</sup>	65. norsk	88. ukraineb <sup>b,c</sup>
	42. hanja2 <sup>c,d,g</sup>	66. nynorsk	89. usorbian
	43. hebrew <sup>c,d</sup>		90. vietnam (vietnamese) <sup>c,d</sup>
			91. welsh

<sup>a</sup> The arab(ic) language requires the use of ArabTeX [25, 26].<sup>b</sup> The russian language is not yet supported by babel [10], but russianb is supported if you use babel-3.6 or higher; russianc is an extra.<sup>c</sup> Some languages may require specific fonts.<sup>d</sup> Requires the CJK package [29].<sup>e</sup> Requires Lambda, the version of L<sup>A</sup>T<sub>E</sub>X for Omega.<sup>f</sup> Requires a 8-bit input encoding.<sup>g</sup> Uses also a .mlo file.<sup>h</sup> Requires a specific input encoding.

section 1.3.12 on page 31) are available. You can easily prepare a similar file for a preferred language (see section 2.26 on page 51). You can change the language of these titles by using the `\mtcselectlanguage{language}` macro.

### 1.3.5 Section-level mini-tables

<code>\stcfont</code> <code>\stcSSfont</code> <code>\stcSSSfont</code> <code>\stcPfont</code> <code>\stcPSfont</code> <code>\slffont</code> <code>\sltfont</code>	<p>The section-level table of contents is typeset in the <code>\stcfont</code> font, which is defined as <code>\normalsize\rmfamily</code> by default. Subsection entries are typeset in the <code>\stcSSfont</code> font, which is <code>\normalsize\bfseries</code> by default. Subsubsection entries are typeset in the <code>\stcSSSfont</code> font, which is <code>\normalsize\rmfamily</code> by default. For paragraphs and subparagraphs, the commands <code>\stcPfont</code> and <code>\stcSPfont</code> are available (by default, <code>\normalsize\rmfamily</code>) if you want to use various fonts. Section-level lists of figures and tables are typeset in the fonts <code>\slffont</code> and <code>\sltfont</code>, which are defined as <code>\normalsize\rmfamily</code> by default.</p>
---	---

### 1.3.6 Titles for section-level mini-tables

<code>\stifont</code> <code>\stctitle</code> <code>\slftitle</code> <code>\slttitle</code> <code>\mtcsettitle</code> <code>\mtcselectlanguage</code>	<p>Titles are typeset in the <code>\stifont</code> (<code>\normalsize\bfseries</code> by default) font and the text strings of the titles are defined by <code>\stctitle</code>, <code>\slftitle</code> and <code>\slttitle</code>, which are the strings “Contents”, “Figures” and “Tables” by default. These title commands should be redefined by <code>\renewcommand</code> or <code>\mtcsettitle</code> for languages other than english. The language option files like <code>french.mld</code> and <code>english.mld</code> (and also many others, as listed in table 1.7 on the page before and explained in section 1.3.12 on page 31) are available. You can easily prepare a similar file for your preferred language (see section 2.26 on page 51). You can change the language of these titles by using the <code>\mtcselectlanguage{language}</code> macro.</p>
---	---

### 1.3.7 Position of the titles

#### 1.3.7.1 For mini-tables at the part level

<code>\doparttoc</code> <code>\dopartlof</code> <code>\dopartlot</code> <code>\parttoc</code> <code>\partlof</code> <code>\partlot</code>	<p>By default, titles are on the left. The preparation commands <code>\doparttoc</code>, <code>\dopartlof</code> and <code>\dopartlot</code> accept an optional argument to change the default position of the corresponding title: [l] for left (default), [c] for center, [r] for right, or [e] (or [n]) for empty (no title). The change is global for all the document. If you want to change the position of the title for only one <code>parttoc</code> (or <code>partlof</code> or <code>partlot</code>), just use such an optional argument with the command <code>\parttoc</code> (or <code>\partlof</code> or <code>\partlot</code>).</p>
--	---

#### 1.3.7.2 For mini-tables at the chapter level

<code>\dominitoc</code> <code>\dominilof</code> <code>\dominilot</code> <code>\minitoc</code> <code>\minilof</code> <code>\minilot</code>	<p>By default, titles are on the left. The preparation commands <code>\dominitoc</code>, <code>\dominilof</code> and <code>\dominilot</code> accept an optional argument to change the default position of the corresponding title: [l] for left (default), [c] for center, [r] for right, or [e] (or [n]) for “empty” (“no” title). The change is global for all the document. If you want to change the position of the title for only one <code>minitoc</code> (or <code>minilof</code> or <code>minilot</code>), just use such an optional argument with the command <code>\minitoc</code> (or <code>\minilof</code> or <code>\minilot</code>).</p>
--	---

### 1.3.7.3 For mini-tables at the section level

`\dosecttoc` By default, titles are on the left. The preparation commands `\dosecttoc`, `\dosectlof` and `\dosectlot` accept an optional argument to change the default position of the corresponding title: `[l]` for left (default), `[c]` for center, `[r]` for right, or `[e]` (or `[n]`) for empty (no title). The change is global for all the document. If you want to change the position of the title for only one secttoc (or sectlof or sectlot), just use such an optional argument with the command `\secttoc` (or `\sectlof` or `\sectlot`).

### 1.3.7.4 Summary of the positionning of titles

`\doparttoc` To summarize: by default, all titles are on the left. However, each one of the following preparation commands:

`\dopartlof`

`\dopartlot`

`\dominitoc` `\doparttoc`, `\dopartlof`, `\dopartlot`,

`\dominilof` `\dominitoc`, `\dominilof`, `\dominilot`,

`\dominilot` `\dosecttoc`, `\dosectlof`, `\dosectlot`

`\dosecttoc`

`\dosectlof` accepts an optional argument to change the positionning of the title: `[l]` for left (default), `[c]` for center, `[r]` for right, `[e]` or `[n]` for empty (no title), for all the corresponding mini-tables. The following insertions commands:

`\dosectlot`

`\parttoc` `\parttoc`, `\partlof`, `\partlot`,

`\partlof` `\parttoc`, `\partlof`, `\partlot`,

`\partlot` `\parttoc`, `\partlof`, `\partlot`,

`\minitoc` `\minitoc`, `\minilof`, `\minilot`,

`\minilof` `\secttoc`, `\sectlof`, `\sectlot`

`\minilot`

`\secttoc` accept the same optional arguments, but these options change the positionning only for the title of the current mini-table.

`\sectlof`

`\sectlot`

## 1.3.8 Line spacing in the mini-tables

`\iftightmtc` With the commands `\tightmtctrue` (or the `tight` package option) and `\tightmtcfalse` (or the `loose` package option, which is the default), the mini-tables will have less (tight) or more (loose) space between contents lines.

But with the Koma-script classes (`scrartcls`, `scrbook` and `scrreprt`), it may sometimes be necessary to use the following options or commands, because we need to set `\parskip` to zero in place of `\parsep` to tighten the mini-table. The efficiency of the following options depends of the options given to these Koma-script classes (`parindent` option, `parskip` option and variants).

`\iftightmtc` For the Koma-script classes, with the commands `\ktightmtctrue` (or the `k-tight` package option) and `\ktightmtcfalse` (or the `k-loose` package option, which is the default), the mini-tables will have less (tight) or more (loose) space between contents lines.

### 1.3.9 Simplified commands for fonts

`\mtcsetfont`  
`\mtcsettitlefont` To simplify the redefinition of the fonts for mini-tables, there are two useful commands:

```
\mtcsetfont{mini-table}{sectionning-level}{commands}
\mtcsettitlefont{mini-table}{commands}
```

By example,

```
\mtcsetfont{minitoc}{subsection}%
{\small\rmfamily\upshape\bfseries}
```

will redefine `\mtcSSfont` with the given font commands.

Note that `\mtcsetfont{parttoc}{*}{...}` allows also to redefine `\ptcfont`, etc.

Moreover,

```
\mtcsettitlefont{parttoc}{\Large\rmfamily\itshape\mdseries}
```

will redefine `\pti font` (for titles in the parttocs, partlofs and partlofs) with the given font commands.

### 1.3.10 Simplified command for mini-table titles

`\mtcsettitle` To simplify the redefinition of the titles for mini-tables, the `\mtcsettitle` command is also available:

```
\mtcsettitle{mini-table}{title string}
```

By example,

```
\mtcsettitle{minitoc}{Description of contents}
```

will redefine `\mtctitle` with the given string. This command checks that you redefine a title for a mini-table type available in your document class.

### 1.3.11 Simplified command for mini-table depths

`\mtcsetdepth` To simplify the redefinition of the depths for mini-tables, the `\mtcsetdepth` command is also available:

```
\mtcsetdepth{mini-table}{depth}
```

By example,

```
\mtcsetdepth{minitoc}{4}
```

will set the counter `minitocdepth` with the given value. This command checks that you set a depth for a mini-table type available in your document class (and that it is possible to change its depth).

### 1.3.12 Languages for the titles

Most of the strings defined in the language option files (`.mld`) were taken from the superb `babel` [10] package by Johannes BRAAMS, some were adapted, others were made available by gentle users or taken from specific packages, like `ArabTeX` [25, 26], `ArmTeX` (armenian) [16], `BangTeX` (bangla) [36], `ethiop` [6], `guarani` [7], `Malayalam` [1], `Montex` (mongol) [14, 15], `CJK` (chinese, corean-hangul/hanja, japanese, thai) [29], `FarsiTeX` (farsi or iranian), or `vietnam.sty` — `latvian` (letton), `greek-mono`, `greek-polydemo`, `greek-polykatha`, `polish2`, `russian2m`, `russian2o` and `spanish3` need `Lambda`, *i.e.* the Omega version of `LATeX`, (see [24]), or even found by searching on the Web (`bulgarianb.mld` for upper cyrillic bulgarian, `japanese.mld` for japanese, `serbianc.mld` for cyrillic serbian). Other languages are welcome. See table 1.7 on page 27.



But for some oriental languages<sup>9</sup>, the sources of the titles use some exotic encodings which are difficult to manipulate in a `.dtx` file, hence the `.mld` file is then just a wrapper which loads a special file, nicknamed a `.mlo` file<sup>10</sup>, not generated by the `.dtx` file in the current version of `minitoc` package, but via `filecontents` environments in the `minitoc.ins` file, and playing with the “catcode” of the “delete” character.

### 1.3.13 Altering the layout of the mini-tables (*experimental feature*)

The layout of a mini-table is described in the figure 1.1 on the next page (this figure is adapted from [44]), which defines some internal commands (these are not *dimensions*, but `LATeX` commands, created by `\newcommand`, modifiable via `\renewcommand`).

<sup>9</sup>Mainly for chinese, farsi, hangul (korean), hanja (korean), japanese and thai variants.

<sup>10</sup>The suffix `.mlo` means *minitoc language object*.

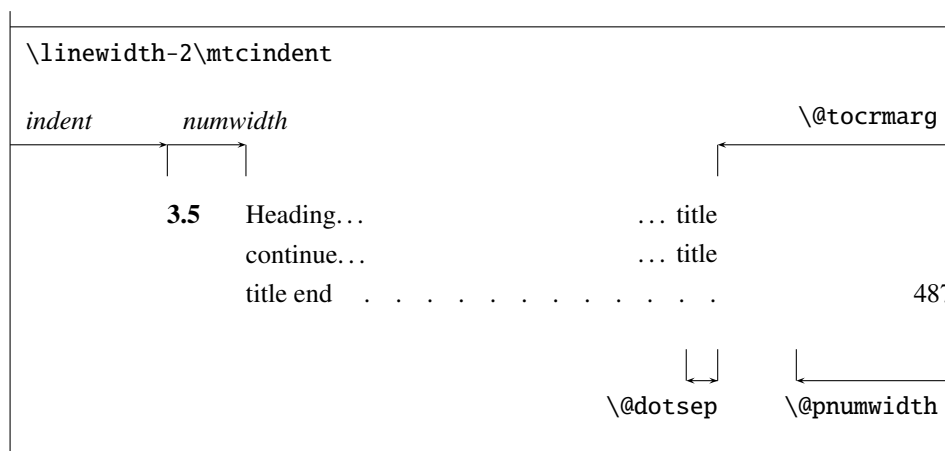


Figure 1.1: Layout of a ToC (LoF, LoT) entry

- $\text{\@dotsep}$ , which is the separation between the dots in the dotted line. It is a pure number expressing *math units*; 18 math units make 1em (one quad), which is about the width of a “m” in the current font.
- $\text{\@pnumwidth}$ , is the width of the space reserved for the page number. It is a  $\text{\LaTeX}$  command containing the representation of a length (e.g. 1.55em).
- $\text{\@tocrmarg}$ , is the distance (margin) between the right border of the table and the end of the dotted line. It should be larger than  $\text{\@pnumwidth}$ , and can be a rubber length (*i.e.* contains some glue, like 2.55em plus 1fil); if you specify the “... plus 1fil” portion, the text of the entry will be ragged on right; it is useful if you have long entries, and it can avoid most hyphenations.

$\text{\mtcsetformat}$  As these commands are internal (their names contain the “@” character) and must have a local effect only on specific mini-tables, you should alter them indirectly via the  $\text{\mtcsetformat}$  command:

```
 $\text{\mtcsetformat}\{mini-table\}\{parameter\}\{value\}$ 
```

where *mini-table* is one of the parttoc, partlof, partlot, minitoc, minilof, minilot, secttoc, sectlof or sectlot keywords; *parameter* is one of the dotinterval (for  $\text{\@dotsep}$ ), pagenumwidth (for  $\text{\@pnumwidth}$ ), or tocrightrightmargin (for  $\text{\@tocrmarg}$ ) keywords; so:

```
 $\text{\mtcsetformat}\{partlof\}\{tocrightmargin\}\{2.55em\ plus\ 1fil\}$ 
```

will set the right margin to 2.55em plus 1fil in the lists of tables at the part level. The elasticity (plus 1fil) is useful if the table captions are long.

Note that the tocrightrightmargin (for  $\text{\@tocrmarg}$ ) parameter should obviously be greater than the pagenumwidth parameter (this appears in the figure 1.1).



Table 1.8: Horizontal rules

				defaults for		
	rules in		no rules in	book	report	article
<code>\ptcrule</code>	parttocs	<code>\noptcrule</code>	parttocs	N	N	Y
<code>\plfrule</code>	parttocs	<code>\noplfrule</code>	parttocs	N	N	Y
<code>\plrulerule</code>	parttocs	<code>\noplrulerule</code>	parttocs	N	N	Y
<code>\mtcrule</code>	minitocs	<code>\nomtcrule</code>	minitocs	Y	Y	N-A
<code>\mlfrule</code>	minitocs	<code>\nomlfrule</code>	minitocs	Y	Y	N-A
<code>\mltrulerule</code>	minitocs	<code>\nomltrulerule</code>	minitocs	Y	Y	N-A
<code>\stcrule</code>	secttocs	<code>\nostcrule</code>	secttocs	N-A	N-A	Y
<code>\slfrule</code>	secttocs	<code>\noslfrule</code>	secttocs	N-A	N-A	Y
<code>\sltrulerule</code>	secttocs	<code>\nosltrulerule</code>	secttocs	N-A	N-A	Y

Table 1.9: Page numbers

Type	Page numbers (Default)	No page numbers
parttoc	<code>\ptcpagenumbers</code>	<code>\noptcpagenumbers</code>
minitoc	<code>\mtcpagenumbers</code>	<code>\nomtcpagenumbers</code>
secttoc	<code>\stcpagenumbers</code>	<code>\nostcpagenumbers</code>
partlof	<code>\plfpagenumbers</code>	<code>\noplfpagenumbers</code>
minilof	<code>\mlfpagenumbers</code>	<code>\nomlfpagenumbers</code>
sectlof	<code>\slfpagenumbers</code>	<code>\noslfpagenumbers</code>
partlot	<code>\pltpagenumbers</code>	<code>\nopltpagenumbers</code>
minilot	<code>\mltpagenumbers</code>	<code>\nomltpagenumbers</code>
sectlot	<code>\sltpagenumbers</code>	<code>\nosltpagenumbers</code>

If the `dotinterval` parameter (for `\@dotsep`) is very large (try 450, then increase or decrease), the dots of leaders will be so much spread out that they will disappear.

## 1.4 Special Features

### 1.4.1 Horizontal Rules

`\mtcsetrules` By default, most of mini-tables have horizontal rules after their titles and at their ends. The exception is the “parttoc” in a book- or report-like document (*i.e.* when `\chapter` is defined). To activate or deactivate these rules, the commands of the table 1.8 are available. But you can also use the following command, which is simpler:

```
\mtcsetrules{mini-table}{on|off}
```

where *mini-table* is one of the `parttoc`, `partlof`, `partlot`, `minitoc`, `minilof`, `minilot`, `secttoc`, `sectlof`, or `sectlot` keywords; the keywords `on` and `off` have the following synonyms<sup>11</sup> :

- `on`, `ON`, `yes`, `YES`, `y`, `Y`, `true`, `TRUE`, `t`, `T`, `vrai`, `VRAI`, `v`, `V`, `oui`, `OUI`, `o`, `O`, and `1`;
- `off`, `OFF`, `no`, `NO`, `n`, `N`, `false`, `FALSE`, `faux`, `FAUX`, `f`, `F`, `non`, `NON`, and `0`.

## 1.4.2 Page Numbers, Leaders

`\mtcsetpagenumbers` By default, the page numbers are listed in each `minitoc`, `minilof`, etc. Some authors want only the section titles (with the section numbers), but without page numbers. Hence the obvious declarations of table 1.9 on the preceding page are available. But you can also use the following command:

```
\mtcsetpagenumbers{mini-table}{on|off}
```

where *mini-table* is one of the `parttoc`, `partlof`, `partlot`, `minitoc`, `minilof`, `minilot`, `secttoc`, `sectlof`, or `sectlot` keywords; the keywords `on` and `off` have the following synonyms<sup>11</sup> :

- `on`, `ON`, `yes`, `YES`, `y`, `Y`, `true`, `TRUE`, `t`, `T`, `vrai`, `VRAI`, `v`, `V`, `oui`, `OUI`, `o`, `O`, and `1`;
- `off`, `OFF`, `no`, `NO`, `n`, `N`, `false`, `FALSE`, `faux`, `FAUX`, `f`, `F`, `non`, `NON` and `0`.

In the mini-tables, they are leaders of dots between the section titles and the page numbers. The `undotted` package option removes these dots. The `dotted` package option is the default. See also section 1.3.13 on page 31.

## 1.4.3 Features for parttocs and other mini-tables

By default, a `parttoc` (or a `partlof` or a `partlot`) is preceded and followed by a `\cleardoublepage`, and has a page style of `empty`. Since version #32, you can modify this behaviour by redefining the commands of table 1.10 on the following page, whose meaning is obvious.

`\mtcsetfeature` The command:

```
\mtcsetfeature{mini-table}{keyword}{commands}
```

---

<sup>11</sup>`O` and `o` are the letter `O`, `0` is the zero digit.

Table 1.10: Features for parttoc-s and other mini-tables

Type	Command	Default
parttoc	<code>\beforeparttoc</code>	<code>\cleardoublepage</code>
parttoc	<code>\afterparttoc</code>	<code>\cleardoublepage</code>
parttoc	<code>\thispageparttocstyle</code>	<code>\thispagestyle{empty}</code>
partlof	<code>\beforepartlof</code>	<code>\cleardoublepage</code>
partlof	<code>\afterpartlof</code>	<code>\cleardoublepage</code>
partlof	<code>\thispagepartlofstyle</code>	<code>\thispagestyle{empty}</code>
partlot	<code>\beforepartlot</code>	<code>\cleardoublepage</code>
partlot	<code>\afterpartlot</code>	<code>\cleardoublepage</code>
partlot	<code>\thispagepartlotstyle</code>	<code>\thispagestyle{empty}</code>
minitoc	<code>\beforeminitoc</code>	<code>\empty</code>
minitoc	<code>\afterminitoc</code>	<code>\empty</code>
minitoc	<code>\thispageminitocstyle</code>	<code>\empty</code>
minilof	<code>\beforeminilof</code>	<code>\empty</code>
minilof	<code>\afterminilof</code>	<code>\empty</code>
minilof	<code>\thispageminilofstyle</code>	<code>\empty</code>
minilot	<code>\beforeminilot</code>	<code>\empty</code>
minilot	<code>\afterminilot</code>	<code>\empty</code>
minilot	<code>\thispageminilotstyle</code>	<code>\empty</code>
secttoc	<code>\beforesecttoc</code>	<code>\empty</code>
secttoc	<code>\aftersecttoc</code>	<code>\empty</code>
secttoc	<code>\thispagesecttocstyle</code>	<code>\empty</code>
sectlof	<code>\beforesectlof</code>	<code>\empty</code>
sectlof	<code>\aftersectlof</code>	<code>\empty</code>
sectlof	<code>\thispagesectlofstyle</code>	<code>\empty</code>
sectlot	<code>\beforesectlot</code>	<code>\empty</code>
sectlot	<code>\aftersectlot</code>	<code>\empty</code>
sectlot	<code>\thispagesectlotstyle</code>	<code>\empty</code>
<code>\mtcsetfeature{mini-table}{before after pagestyle}{command}</code> Modifies the features for a mini-table.		

allows you to redefine any of these commands. *mini-table* is one of the mini-table names: `parttoc...` `sectlot`. *keyword* is one of the followings: `before`, `after` or `pagestyle`. *commands* is either a sequence of commands like `\clearpage`, `\cleardoublepage`, `\thispagestyle{...}`, etc., either `\empty` (does nothing).

#### 1.4.3.1 Remark about page styles<sup>12</sup>



The default commands for part-level mini-tables page styles are defined as being `\thispagestyle{empty}`, because in document classes defining the `\chapter` command, the part-level mini-tables are on their own pages. If the document is printed recto-verso, the first page is recto. Usually, these pages are not numbered and have no header and no footer. This behaviour comes from the default definitions of the commands of table 1.10 on the page before. If you want an other behaviour, you can change these definitions. Note that, by default, only the *first* page of these mini-tables are in the `empty` page style. You can set the style of this first page by using `\thispagestyle` and set the style of the following pages by using `\pagestyle`, but you must not forget to reset the normal style after the mini-table. Look at this short example<sup>13</sup>:

```
\mtcsetfeature{parttoc}{before}%
  {\cleardoublepage}
\mtcsetfeature{parttoc}{thispagestyle}%
  {\thispagestyle{empty}\pagestyle{myheadings}}
\mtcsetfeature{parttoc}{after}%
  {\cleardoublepage\pagestyle{headings}}
```

where we add a `\cleardoublepage` before each `parttoc`, then we set the `empty` page style for the first page of the `parttocs`, the `myheadings` page style for the following pages of the `parttocs`, and set `headings` page style for the pages after the mini-table, after a `\cleardoublepage`.

### 1.4.4 The “Chapter 0” Problem (solved)

Some documents do not begin with chapter number one, but with chapter number zero (or even a weirder number).

#### 1.4.4.1 Before version #23 (1994/11/08)

```
\firstpartis
\firstchapteris
\firstsectionis
```

To make the `minitoc` package work with such documents, you must insert the command:

<sup>12</sup>This remark is taken and adapted from a draft of the second edition of the JMPL [4], by Benjamin BAYART, where he comments the `minitoc` package.

<sup>13</sup>This example shows that the third argument can be a *sequence* of commands: we set the style of the current page and the style of the following pages.

`\firstchapteris{⟨N⟩}`

before the `\dominitoc` and analogous commands.  $\langle N \rangle$  is the number of the first chapter. This command *does not* modify the numbering of chapters, you must use a `\addtocounter{chapter}{-1}` command to get a first chapter numbered 0. The `\firstpartis` and `\firstsectionis` commands are similar for parts and sections with a non standard numbering.

#### 1.4.4.2 Since version #23 (1994/11/08)



These commands are now obsolete, as this problem has been solved (via the “absolute” numbering of the mini-table auxiliary files). Thus they just produce harmless warnings.

### 1.4.5 Special Entries for TOC, LOF, LOT, Bibliography and Index



If you want to add entries in the Table of Contents for objects like the Table of Contents itself, the List of Figures, the List of Tables, the Bibliography or the Index, you should use the `tocbibind` [47] package by Peter R. WILSON (this package is available from the CTAN archives).

`\dominitoc` But these entries are considered as chapters (or sections in an article class document) when the `.toc` file is scanned to prepare the minitocs (the `\dominitoc` phase).

`\mtcaddchapter` So you must add an `\mtcaddchapter` command, *without argument*, after each of the  
`\tableofcontents` involved commands `\tableofcontents`, `\listoffigures`, and `\listoftables`.

`\listoffigures`  
`\listoftables` For the bibliography, you should add a `\adjustmtc` command after the `\bibliography`  
`\adjustmtc` command.

`\bibliography`  
`\printindex` For the index, it is a bit more complicated, you add the following commands just after the  
`\addcontentsline` `\printindex` command:

`\mtcaddchapter`  
`\mtcfixindex`

```

\addcontentsline{lof}{xchapter}{}
\addcontentsline{lot}{xchapter}{}
\mtcaddchapter

```

But this can be done by:

```
\mtcfixindex[chapter|section|part]
```

where the optional argument is the level for the index entry in the TOC. By default, if `\chapter` is defined, the `chapter` level is used, else the `section` level. If neither `\chapter` or `\section` are defined, the `part` level will be used if `\part` is defined; else



an error is reported. It is *recommended* that you check the result and, if necessary, you adjust the optional argument.

`\printglossary` For the glossary, it is like for the index, you add the following commands just after the  
`\addcontentsline` `\printglossary` command:  
`\mtcaddchapter`  
`\mtcfixglossary`

```

\addcontentsline{lof}{xchapter}{}
\addcontentsline{lot}{xchapter}{}
\mtcaddchapter

```

But this can be done by:

```
\mtcfixglossary[chapter|section|part]
```



where the optional argument is the level for the glossary entry in the TOC. By default, if `\chapter` is defined, the `chapter` level is used, else the `section` level. If neither `\chapter` or `\section` are defined, the `part` level will be used if `\part` is defined; else an error is reported. It is *recommended* that you check the result and, if necessary, you adjust the optional argument.

Of course, in documents where the TOC, LOF, LOT, bibliography and/or index (or glossary) are processed as starred sections, you must modify these additions to use section level commands.

And proceed with care, tracking in the `.log` file the insertion of `.mtc<N>` files (and siblings). They are some examples in the `add.tex` file distributed with `minitoc`.

### 1.4.6 The notoccite option

`\cite` This option loads the `notoccite` [2] package (by Donald ARSENEAU). It avoids problems with `\cite` commands in sectioning commands or captions: if you then run `LaTeX` using the `unsorted` style, or a similar style, these citations get numbered starting from the page in the table of contents where is the parasite citation, not the number they should have in the main text. The `notoccite` package prevents this. As `minitoc` prints TOCs, it is subject to the same problem.

## 1.5 The hints option

This package option detects some actions and the loading of some packages and classes known as interacting with `minitoc`, and also some frequent misuses and errors. This list of interacting packages and classes is, of course, not closed. If a known package is loaded, this option writes some hints in the `.log` file and emits a warning. The hints written in the `.log` file may suggest you to consult the present document or the `minitoc.bug` file.



*Your advice about this option will be welcome.* This option is activated by default. You can inhibit it via the `nohints` option. The following problems and potential problems are currently detected:

<code>\part</code>	• Alteration of some of the following commands <sup>14</sup> : <code>\part</code> , <code>\@part</code> , <code>\@spart</code> , <code>\chapter</code> , <code>\@chapter</code> , <code>\@schapter</code> , <code>\section</code> , <code>\@sect</code> , and <code>\@ssect</code> . Note that the <code>hyperref</code> (see section 2.17 on page 48) package alters these commands at <code>\begin{document}</code> , hence this problem might be reported if you use this package, but these alterations seem harmless.
<code>\@part</code>	
<code>\spart</code>	
<code>\@spart</code>	
<code>\chapter</code>	
<code>\@chapter</code>	
<code>\schapter</code>	• Presence of the following packages or classes, which need some precautions: <code>amsbook</code> (class), <code>memoir</code> (class), <code>appendix</code> , <code>placeins</code> (beware to its options and its release date (2005/04/18 at least)), <code>tocbibind</code> , and <code>tocloft</code> .
<code>\@schapter</code>	
<code>\section</code>	
<code>\@sect</code>	• Presence of the following packages or classes, which are incompatible with the <code>minitoc</code> package: <code>amsart</code> (class), <code>amsproc</code> (class), <code>titlesec</code> , and <code>titletoc</code> <sup>15</sup> .
<code>\@ssect</code>	
<code>\parttoc</code>	• Usage of <code>\parttoc</code> without calling <code>\doparttoc</code> , ... , usage of <code>\sectlot</code> without calling <code>\dosectlot</code> . Or the reverse.
<code>\doparttoc</code>	
<code>\sectlot</code>	• Usage of <code>\parttoc</code> without calling <code>\[fake]tableofcontents</code> , ... , usage of <code>\sectlot</code> without calling <code>\[fake]listoftables</code> .
<code>\dosectlot</code>	
<code>\tableofcontents</code>	• Usage of <code>\sectlof</code> and/or <code>\sectlot</code> without using the <code>insection</code> package option of <code>minitoc</code> (or the <code>placeins</code> package without its <code>section</code> and below options).
<code>\listoftables</code>	
<code>\faketableofcontents</code>	• If you are using short extensions (because of your operating system or the <code>shorttext</code> package option, see section 1.6 on the next page) and go beyond the limit of 99 parts, chapters or sections, the <code>hints</code> package option displays a warning.
<code>\fakelistoftables</code>	
<code>\sectlof</code>	
<code>\sectlot</code>	
<code>abstract</code>	• If the <code>abstract</code> package [45] (by Peter R. WILSON), is used with its <code>addtotoc</code> option, a “Abstract” entry is added to the table of contents, as a starred chapter if the document class defines <code>\chapter</code> , else as a starred section. This is detected and you should add a <code>\mtcaddchapter[]</code> or a <code>\mtcaddsection[]</code> command after your <code>abstract</code> environment.
<code>\mtcaddchapter</code>	
<code>\mtcaddsection</code>	
	• If the <code>sectsty</code> package [32] (by Rowland McDONNELL) is used, it must be loaded <i>before</i> the <code>minitoc</code> package.
	• If you try to insert empty mini-tables, the <code>hints</code> option gives a global warning (except if you used also the <code>nocheckfiles</code> option, see section 1.2.3 on page 20).
<code>\firstpartis</code>	• If you use one of the obsolete commands ( <code>\firstpartis</code> , <code>\firstchapteris</code> , or <code>\firstsectionis</code> ), a warning is issued for each use, of course, but also a global hint as reminder.
<code>\firstchapteris</code>	
<code>\firstsectionis</code>	

<sup>14</sup>The commands containing the “@” character in their names are internal commands of L<sup>A</sup>T<sub>E</sub>X, of a package or of a class; they are sometimes altered by another packages; reconsider then the loading order of the packages.

<sup>15</sup>The `titlesec` package redefines the sectionnning commands in a way completely alien to the standard L<sup>A</sup>T<sub>E</sub>X way; hence `minitoc` and `titlesec-titletoc` are fundamentally incompatible, and it is very sad.

Table 1.11: Extensions of the auxiliary files

mini-table	long extensions (UNIX, etc.)	short extensions (MS-DOS, etc.)
parttoc	.ptc $\langle N \rangle$	.P $\langle N \rangle$
partlof	.plf $\langle N \rangle$	.G $\langle N \rangle$
partlot	.plf $\langle N \rangle$	.U $\langle N \rangle$
minitoc	.mtc $\langle N \rangle$	.M $\langle N \rangle$
minilof	.mlf $\langle N \rangle$	.F $\langle N \rangle$
minilot	.mlf $\langle N \rangle$	.T $\langle N \rangle$
secttoc	.stc $\langle N \rangle$	.S $\langle N \rangle$
sectlof	.slf $\langle N \rangle$	.H $\langle N \rangle$
sectlot	.slf $\langle N \rangle$	.V $\langle N \rangle$

## 1.6 Usage with MS-DOS



Under MS-DOS (and other PC oriented old operating systems), the filename extensions are limited to 3 characters. The `minitoc` package determines dynamically the type of extensions available and will use it. All other modifications will be done automatically. The `.mtc $\langle N \rangle$`  extensions will become `.M $\langle N \rangle$` , where  $\langle N \rangle$  is the absolute chapter number. The extensions `.mlf $\langle N \rangle$`  and `.mlt $\langle N \rangle$`  become `.F $\langle N \rangle$`  and `.T $\langle N \rangle$` . The `.ptc $\langle N \rangle$`  extensions become `.P $\langle N \rangle$` , where  $\langle N \rangle$  is the absolute part number. The extensions `.plf $\langle N \rangle$`  and `.plt $\langle N \rangle$`  become `.G $\langle N \rangle$`  and `.U $\langle N \rangle$` . The `.stc $\langle N \rangle$`  extensions become `.S $\langle N \rangle$` , where  $\langle N \rangle$  is the absolute section number. The extensions `.slf $\langle N \rangle$`  and `.slt $\langle N \rangle$`  become `.H $\langle N \rangle$`  and `.V $\langle N \rangle$` . All these extensions are listed in table 1.11. Of course, this implies a limit of 99 chapters in a document, but do you really need so many chapters (or sections in an article)? The limit of 99 parts does not seem too serious for most documents, but for sections, it could be tragical. The `hints` option (section 1.5 on page 38) will report such situations. See also section 2.5 on page 44.

## 1.7 Why several L<sup>A</sup>T<sub>E</sub>X runs are required?

The mini-tables, at part, chapter and section levels, are using some space on the first pages on each chapter, part or section, thus the page numbers are altered. After the first L<sup>A</sup>T<sub>E</sub>X run, the mini-tables and lists, partial tables and lists and section-level tables and lists will be empty (in fact skipped since version #35); after the second run, they appear (if not empty), but because they modify the page numbering, page numbers are wrong; after the third L<sup>A</sup>T<sub>E</sub>X run, the mini, part- and section-level tables and lists should be correct (see figure 2.1 on page 44).



## 1.8 The `mtcoff` package

If a document has been prepared with the `minitoc` package, it contains many `minitoc` specific commands, most of them being `\dominitoc`, `\faketableofcontents`, and `\minitoc` commands (and their equivalents for lists of figures and tables). If you want to typeset this document without any mini-table, you have just to replace the `minitoc` package by the `mtcoff` package (without option), and all these commands will be ignored, eventually writing warning messages in the `.log` file. At least two  $\text{\LaTeX}$  runs will be necessary to get a correct page numbering and cross references. It also sanitizes the `.aux`, `.toc`, `.lof`, and `.lot` files from `minitoc` specific spurious commands.

## Chapter 2

# Frequently Asked Questions

### Contents

---

2.1	How to avoid a page break near the rules before and after the mini-table? . . . . .	43
2.2	How about implementing others layouts for a mini-table? . . . .	43
2.3	Two consecutive backslashes in a contents line make an error . .	43
2.4	Reordering chapters makes havoc . . . . .	43
2.5	Extensions for the names of auxiliary files . . . . .	44
2.6	Playing with the chapter number . . . . .	44
2.7	Supported document classes . . . . .	45
2.8	Compatibility with L <sup>A</sup> T <sub>E</sub> X versions . . . . .	45
2.9	Other mini-tables . . . . .	45
2.10	Why so many auxiliary files? . . . . .	46
2.11	How to do mini-tables at levels other than chapter? . . . . .	46
2.12	Incompatibility with L <sup>A</sup> T <sub>E</sub> X2.09 . . . . .	47
2.13	Documents resetting the chapter number at each part . . . . .	47
2.14	The mini-tables have too much spaced lines . . . . .	47
2.15	The secttocs are wrong . . . . .	47
2.16	How to remove the lines of dots? . . . . .	47
2.17	How to use the hyperref package with minitoc? . . . . .	48
2.18	Problem while upgrading minitoc . . . . .	48
2.19	A local table of contents for the set of appendices . . . . .	48
2.20	Use with the appendix package . . . . .	49
2.21	Use with the tocloft package . . . . .	49
2.22	Use with the memoir class . . . . .	50
2.23	There are too many commands for fonts, titles, and depths . . . .	50
2.24	Compatibility with the $\mathcal{A}\mathcal{M}\mathcal{S}$ document classes . . . . .	50
2.25	How to hide some entries from the main table of contents . . . .	51
2.26	How to define your own .mld file? . . . . .	51
2.27	Use with the abstract package . . . . .	52
2.28	Use with the sectsty package . . . . .	52
2.29	Strange alignment in the minitocs . . . . .	52

---

## Figures

---

2.1 Three compilations for <code>minitoc</code> . . . . .	44
---	----

---

Here is a list of problems and frequently asked questions about `minitoc.sty`. If the version has a number less than 43, please upgrade to version #43. This list is also given in the `minitoc.bug` file, in pure text form.

If a problem arises, it is often wise to: a) use the `hints` option (see section 1.5 on page 38), which is activated by default, and b) read the `<document>.log` file, which may contain pertinent messages.

### 2.1 How to avoid a page break near the rules before and after the mini-table?

`\enlargethispage` This problem seemed solved since version #8, but version #12 adds better fixes. You may have to make some final tuning with `\enlargethispage`. See the  $\text{\LaTeX}$  manual [27]. The `needspace` [43] package may also be useful.

### 2.2 How about implementing others layouts for a mini-table?

Suggestions are welcome, but look at the section 1.3.13 on page 31.

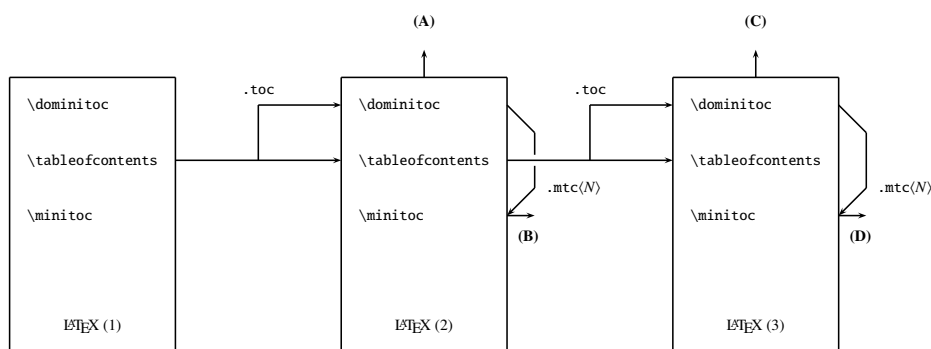
### 2.3 Two consecutive backslashes in a contents line make an error

Use `\protect\linebreak`.

### 2.4 Reordering chapters makes havoc

*If you reorder chapters, havoc follows... mini-tables going in wrong chapters.*

The best way seems to make one run with the `mtcoff` package replacing the `minitoc` package, then restore the `minitoc` package and re-execute  $\text{\LaTeX}$  three times (yes, it is time consuming...). See figure 2.1 on the next page. Running with the `mtcoff` package



(A) `\tableofcontents` produces a table of contents, which is likely inaccurate.

(B) `\minitoc` produces minitocs, which are likely inaccurate.

(C) `\tableofcontents` produces a table of contents, which is accurate.

(D) `\minitoc` produces minitocs, which are accurate.

Figure 2.1: Three compilations for `minitoc`

ensures that the standard auxiliary files are cleared from “spurious” commands introduced by `minitoc`. A more radical solution is to delete the `.aux`, `.toc`, `.lof` and `.lot` files relative to the document, then re-execute  $\LaTeX$  three times.

## 2.5 Extensions for the names of auxiliary files

*This package creates auxiliary files with extensions like `.mtc(N)`. Some operating systems allow only 3 characters extensions. What to do?*

No modification is needed: all became automatic since version #28! If you insist to use 3 characters extensions, even on operating systems allowing more, just use the package option `shorttext`. Then you will get first the auto-configuration messages, then a message saying that you will use short extensions. But be careful to not have more than 99 mini-tables of the same kind (even empty)!

## 2.6 Playing with the chapter number



*Do not cheat* with the “chapter” counter, *i.e.* do not write ugly things like

```
\setcounter{chapter}{6}
```

The mechanism would break. It is better to add `\chapter` commands, to create empty (but numbered in a legal way) chapters. Since version #10, the `minitoc` package works with appendices. Version #19 allows to begin with a chapter other than number 1. And

look at “Special Entries for TOC, LOF, LOT, Bibliography and Index”, section 1.4.5 on page 37.

The same remark applies to the `part` and `section` counters.

## 2.7 Supported document classes

The `minitoc` package is restricted to document classes which define chapters in the standard way, like ‘`book`’ and ‘`report`’, or sections in the standard way, like ‘`article`’. There are “parttocs” if the document class defines the `\part` command. Note that classes like “`letter`”, which have not the classical sectionning structure, cannot be supported. Classes using sectionning commands with other names are not supported<sup>1</sup>. See also section 2.24 on page 50.

## 2.8 Compatibility with L<sup>A</sup>T<sub>E</sub>X versions

Some users have failed to make `minitoc` to work. They got a message like:

```
Undefined command ... \@inputcheck ...
Your version of latex.tex is obsolete. Trying to continue...
```

or:

```
Undefined command ... \reset@font ...
Your version of latex.tex is very obsolete.
Trying to continue... crossing fingers.
```

The `\reset@font` command has been added to `latex.tex` on September 29th, 1991 and the `\@inputcheck` command on March 18th, 1992 and this version of `latex.tex` has been released on March 25th, 1992. If you get this message, you have an old version of `latex.tex`. Get a recent one from the archives and regenerate a `latex.fmt` format via `initex` (or your configuration tool).

## 2.9 Other mini-tables

Some demanding users want to have `minilof`, `minilot` and `minibbl`. First, “`minibbl`” is another problem, strongly related to the B<sup>B</sup>T<sub>E</sub>X’s dealing with `.aux` files. Look at

---

<sup>1</sup>This would be very difficult: any user can create new sectionning commands (often with the help from some packages) with standard or new names; this is only limited by the imagination. The `minitoc` package relies on the names of the standard sectionning commands and on the syntax of these commands.

the `chapterbib`, `bibunits`, `multibib`, and `bibtopic` packages. Version #13 has implemented basic `minilofs` and `minilots`. `Minibbls` are not the aim of this package (see <http://www.tex.ac.uk/cgi-bin/texfaq2html?label=multibib>).

## 2.10 Why so many auxiliary files?

This package creates a lot of auxiliary files and some users have argued that it is too many. A deep redesign would be necessary to avoid that. Using only one big auxiliary file (or one for all `minitocs`, one for all `minilofs`, ...) would make the reading of such file very slow, as it would be read for each `\miniXXX` macro! Moreover, this would make the `checkfiles` (see section 1.2.3 on page 20) package option impractical to implement. Note that the many files `*.mtc*`, etc., may be deleted after the  $\text{\LaTeX}$  run. They are rebuilt by the `\dominitoc` command (and siblings). But, since version #35, `minitoc` is able to detect and skip empty `*.mtc*` files (and siblings) to avoid ugly titles with just two thin rules. It would not be easy to do with one big auxiliary file.

These files contain the mini-tables extracted from the `.toc`, `.lof`, and `.lot` files. They are no more useful after the  $\text{\LaTeX}$  run. If you run  $\text{\LaTeX}$  via a script or a “makefile”, it may be useful to add to it a cleaning feature (which should be optional, to allow debugging). The table 1.11 on page 40 gives the list of the extensions for these files (note that a `\document\}.mtc` auxiliary file is also created as a scratch file).

As an example, you can look at the `rubber` script [5] (written in Python) by Emmanuel BEFFARA:

<http://rubber.sourceforge.net/>  
<http://www.pps.jussieu.fr/~beffara/soft/rubber/>

## 2.11 How to do mini-tables at levels other than chapter?

Here also, some redesign was needed. From version #15, there are `parttocs`, `partlofs` and `partlots` for the part level in `book|report`-like and `article`-like documents, `secttocs`, `sectlofs` and `sectlots` for the section level in `article`-like documents. Note that you can not have `minitocs` features at chapter and section level in the same document, because doing so would make an unreadable monster. The user must choose the main class of the document according to the size of it (e.g. do not write an article of more than 100 sections: this is a report, or even a book!).

	part	chapter	section
book	*	*	
report	*	*	
article	*		*

## 2.12 Incompatibility with L<sup>A</sup>T<sub>E</sub>X 2.09

`\protect`  
`\contentsline` The more recent version of L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> adds `\protect` before `\contentsline` in the `.toc`, `.lof` and `.lof` files. The version #17 of `minitoc` attempts to be compatible with L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> and L<sup>A</sup>T<sub>E</sub>X 2.09. This will be the *last* version usable with L<sup>A</sup>T<sub>E</sub>X 2.09. Versions #18 and later are L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> specific, and no more compatible with L<sup>A</sup>T<sub>E</sub>X 2.09, which is completely obsolete.

## 2.13 Documents resetting the chapter number at each part

Since version #23, `minitoc` works with document classes resetting chapter (or section) number at each part (or chapter). This is possible because the auxiliary files for the mini-tables have now an *absolute* number.

## 2.14 The mini-tables have too much spaced lines

From version #29, you can have tight mini-tables with the `tight` option, and with the `k-tight` option for the Koma-script classes (since version #43).

## 2.15 The secttocs are wrong

Secttocs did not work: corrected (version #38).

## 2.16 How to remove the lines of dots?

The lines of dots (leaders) between section titles and page numbers are removed by the `undotted` option (#29). See also section 1.3.13 on page 31.

## 2.17 How to use the hyperref package with minitoc?

Since version #31, minitoc works correctly with the powerful hyperref package [39], thanks to Heiko OBERDIEK, who used the work of Bernd JAEHNE and Didier VERNA. If you add the loading of the hyperref package to a document yet using minitoc, you will get error message about spurious closing braces. Just let finish the L<sup>A</sup>T<sub>E</sub>X run, then re-L<sup>A</sup>T<sub>E</sub>X the document. There will be no problem if you remove the loading of hyperref and add it again: the problem occurs only when upgrading from minitoc #30 to minitoc #31 (or higher) with a document already processed and adding hyperref at the same time! It seems better to process the document with minitoc #31 (or higher) without hyperref, then with hyperref, because some internal commands written into the auxiliary files have been modified. If used, the hyperref package must be loaded *before* minitoc. Note that (f)minitoc.dtx show a (not so) basic example of the use of the hyperref package with minitoc.

## 2.18 Problem while upgrading minitoc

If upgrading from version #30 or lower to version #31 or higher, you should delete the .aux, .toc, .lof, .lot files of the document, else the first L<sup>A</sup>T<sub>E</sub>X run with version #31 or higher will produce a lot of errors (the next run should be ok). See also the section 2.17.

## 2.19 A local table of contents for the set of appendices

<code>\doparttoc</code> <code>\tableofcontents</code> <code>\appendix</code> <code>\part</code> <code>\parttoc</code> <code>\addtocontents</code> <code>\protect</code> <code>\setcounter</code> <code>\chapter</code> <code>\partbegin</code>	<p>Some users need a table of contents for the appendices, but without putting the entries of it into the main table of contents. The solution is to put the appendices in a <code>\part</code> subdivision of the document and ask for a table of contents at the <code>\part</code> level:</p> <pre> \doparttoc           % after \begin{document} . . . \tableofcontents . . . \appendix \part{Appendices}    % create a part level subdivision \parttoc             % create a local table of contents % To suppress the appendix part in the main toc \addtocontents{toc}{\protect\setcounter{tocdepth}{-1}} \chapter{First appendix} . . . % Add this at the end of appendices if there is something % after the appendices (like an index or a bibliography) % to put a bound to the contents of \parttoc \addtocontents{toc}{\protect\partbegin} </pre>
---	---

See also section 2.25 on page 51.



## 2.20 Use with the `appendix` package

`appendices`    If you use the `appendix` package [46] (by Peter R. WILSON), you will observe a serious problem with `minitocs` in the `appendices` environment (and after it): they do not match with their appendix. In fact, the environment opening `\begin{appendices}` hides a `\addcontentsline` command for a chapter or a section, putting trouble in the numbering of `minitocs` or `secttocs`. Two solutions are available. The first one is to add a `\adjustmtc` or `\adjuststc` command (depending if the `appendices` are at the chapter or section level) after *each* `\begin{appendices}` command. The other solution is to add the following commands in the preamble *after* the loading of the `appendix` package:

```
\let\oldappendices\appendices
\def\appendices{\oldappendices\adjustmtc}
```

if `appendices` are at the chapter level, OR:

```
\let\oldappendices\appendices
\def\appendices{\oldappendices\adjuststc}
```

if `appendices` are at the section level.

## 2.21 Use with the `tocloft` package

`\mtcsetfont`    (This answer is given in the documentation of the `tocloft` package [44].) The `tocloft` (by Peter R. WILSON) and `minitoc` packages have an unfortunate interaction<sup>2</sup>, which fortunately can be fixed. In the normal course of events, when `minitoc` is used in a chaptered document it will typeset section entries in the `minitocs` in bold font. If `tocloft` is used in conjunction with `minitoc`, then the `minitoc` section entries are typeset in the normal font, except for the page numbers which are in bold font, while the ToC section entries are all in normal font.

One cure, if you want the `minitoc` section entries to be all in normal small font, is to put:

```
\renewcommand{\mtcSfont}{\normalfont\small}
```

or:

```
\mtcsetfont{minitoc}{section}{\normalfont\small}
```

---

<sup>2</sup>Discovered by Lyndon DUDGING.

in the preamble.

Otherwise, the cure is the following incantation:

```
\renewcommand{\cftsecfont}{\bfseries}
\renewcommand{\cftsecleader}{\bfseries\cftdotfill{\cftdotsep}}
\renewcommand{\cftsecpagefont}{\bfseries}
```

To have the section entries in both the ToC and the minitocs in bold then put the incantation in the preamble. To have only the minitoc section entries in bold while the ToC entries are in the normal font, put the incantation between the `\tableofcontents` command and the first `\chapter` command.

As `tocloft` is a very powerful and useful package, these cures are worth to be added if you need the benefits of this package.

## 2.22 Use with the `memoir` class

The `memoir` class [48] offers basically the functionalities of the `appendix`, `tocbibind` and `tocloft` packages (this class and these packages have the same author, Peter R. WILSON), hence it has the same problems; see above the available solutions. If your version of the `memoir` class is recent, the syntax of the `\chapter` command is different and the `memoir` class is *no more compatible* with the `minitoc` package, but a patch is inserted to fix the problem.

## 2.23 There are too many commands for fonts, titles, and depths

<code>\mtcsetfont</code>	Since version #41, the <code>\mtcsetfont</code> and <code>\mtcsettitlefont</code> commands are available.
<code>\mtcsettitlefont</code>	You do not need anymore to know <code>\mtcSSSfont</code> , <code>\ptifont</code> , etc.
<code>\mtcsettitle</code>	Since version #42, the <code>\mtcsettitle</code> command is available. You do not need anymore to know <code>\mtctitle</code> , <code>\slttitle</code> , etc.
<code>\mtcsetdepth</code>	Since version #43, the <code>\mtcsetdepth</code> command is available. You do not need anymore to know the counters <code>minitocdepth</code> , <code>seclotdepth</code> , etc.

## 2.24 Compatibility with the $\mathcal{AMS}$ document classes

<code>\mtcaddchapter</code>	The <code>amsart.cls</code> and <code>amsproc.cls</code> document classes are incompatible with <code>minitoc</code> .
-----------------------------	--

The `amsbook.cls` document class requires the insertion of commands if you want a list of figures and/or a list of tables:

```
\listoffigures
\mtcaddchapter % added
\listoftables
\mtcaddchapter % added
```

## 2.25 How to hide some entries from the main table of contents

`mtchideinmaintoc` It is a problem similar to that of section 2.19 on page 48. An example is having a local table of contents for a chapter (`\minitoc`) whose entries should not appear in the main table of contents. Just use the `mtchideinmaintoc` environment:

```
\chapter{Title}
\begin{mtchideinmaintoc}[level]
\minitoc
\section{sub-title}
...
\end{mtchideinmaintoc}
```

This environment accepts an optional numeric argument, which is the depth of hiding in the main toc (default: -1, complete hiding).

Of course, the environments `mtchideinmainlof` and `mtchideinmainlot` are also available, to hide some entries in the main list of figures or of tables.

## 2.26 How to define your own .mld file?

`\mtcsettitle` First, you should not directly modify one of the distributed `.mld` files. The simplest way to alter some title is to redefine the corresponding command via `\renewcommand` or better via `\mtcsettitle`. If you really want to have your own `.mld` file, you copy an existing `.mld` file into one with a new name (not the name of a distributed `.mld` file). Then you modify this new `.mld` file and you can use it via `\mtcselectlanguage`. You can always contact me to add this new `.mld` file to the distribution. These remarks apply also to the `<language>[.mld-.mlo]` pairs of language definition files.

## 2.27 Use with the abstract package

`abstract`  
`\mtcaddchapter`  
`\mtcaddsection`  
`\chapter`

If the `abstract` package [45] (by Peter R. WILSON), is used with its `addtotoc` option, a “Abstract” entry is added to the table of contents, as a starred chapter if the document class defines `\chapter`, else as a starred section. This problem is detected by the `hints` option and you should add a `\mtcaddchapter[]` or a `\mtcaddsection[]` command after your `abstract` environment.

## 2.28 Use with the sectsty package

If the `sectsty` package [32] (by Rowland McDONNELL) is used, it must be loaded *before* the `minitoc` package, because it alters (redefines) the sectioning commands. Of course, `hints` option detects this problem.

## 2.29 Strange alignment in the minitocs

*In minitocs, subsections titles are not aligned with sections, as they are in the main table of contents.*

The entries of a table of contents are formatted via internal commands like `\l@part`, `\l@chapter`, `\l@section`, etc.

The “part” and “chapter” levels (and “section” for an article) use specific commands which are somewhat complex for a more elaborated formatting. For the “section” (in the `report` and `book` classes) and lower levels, these commands are (book class, `book.cls`) by default:

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

which will be applied in the main table of contents and in the minitocs. The arguments of `\@dottedtocline` are

- 1) the logical depth (which will be compared to `tocdepth` or `minitocdepth`).
- 2) the indentation.
- 3) the width reserved for the page number.

In the standard `book`, `report` and `article` classes, the dimensions (second and third arguments) are given in “em” units, and this unit depends of the current font. In the main table of contents, the section and subsection entries are written in the *same* font, hence usually the alignment is correct. But in the `minitocs`, the section entries are written in a bold font while the subsection entries are written in a non bold font (see table 1.6 on page 26), hence one “em” has different sizes in these two fonts and the alignment is changed.

There are several solutions:

- Redefine the `\l@section ... \l@subparagraph` commands to use font independent units (pt, mm, pc, etc.). This redefinition must be performed between `\makeatletter` and `\makeatother`, because these commands have a `@` in their names; you must use `\renewcommand*` to redefine these commands.
- Use the `tocloft` [44] package to change the indentation, with font independent units. But then see section 2.21 on page 49.
- Use the same font for the section and subsection entries in the `minitocs`, using the `\mtcsetfont` command (see section 1.3.9 on page 30) or redefining the `\mtcSfont`, `\mtcSSfont`, `\mtcSSSfont`, `\mtcPfont` and `\mtcSPfont` commands (see table 1.6 on page 26), or similar.

## Chapter 3

# Installation

### Tables

3.1 List of files ( <code>minitoc.l</code> ) . . . . .	57
--	----

Installation of the `minitoc` package (version #43).

This package contains a lot of files. The list of all files is given in `minitoc.l`. See table 3.1 on page 57.

The files are sorted into “classes” below (a file can appear in more than one class). Each class specifies the function and the placement of its files.

(0) The files `minitoc.ins` and `minitoc.dtx` are the basic source files of this package. The file `fminitoc.dtx` loads `minitoc.dtx` but selects the french documentation. The language selection is done by using `\ifcase ... \or ... \fi` constructs.



(1) The files `minitoc.sty`, `mtcoeff.sty`, and *all* `*.mld` and `*.mlo` files are the package itself<sup>1</sup>. `mtcpatchmem.sty` is a temporary fix for compatibility with the `memoir` class. They must be *all* installed in a directory where  $\text{\LaTeX} 2_{\epsilon}$  finds the `.sty` files.

(2) Informative text files:

- `INSTALL` is a file describing the installation of the package. You are (almost) reading it (but it is shorter).
- `minitoc.l` contains the list of all files of the `minitoc` distribution. See table 3.1 on page 57.
- `README` is a file describing briefly the `minitoc` package, plus some useful infos.

<sup>1</sup>The large number `*.mld` files is (partially) a consequence of the fact that some languages have aliases (or dialects) and hence one `*.mld` file for each name (a `*.mld` file can load another one); the english and french languages are evident examples. For some languages, the multiplicity of the `*.mld` files corresponds to a multiplicity of fonts and/or encodings (chinese, greek, japanese, korean, malayalam, polish, russian, serbian), or even for spelling reforms (german, greek, norsk).

- `catalog` contains basic infos about the `minitoc` package (abstract, date, author, version, licence).
  - `TODO` lists some suggested developments of the package, not yet implemented. Comments and suggestions are welcome.
- (3) `minitoc-ex.tex`, `mini-art.tex`, `add.tex`, `add.bib` are example files, to play with.
  - (4) `minitoc.bug`, `minitoc.sum` are plain text documentation: list of problems (faq, see chapter 2 on page 42) and summary of commands (see chapter 4 on page 58).
  - (5) `minitoc.ins`, `minitoc.ist`, `mtcglo.ist`, `minitoc.lan`, `minitoc.dtx`, and `minitoc.bib` are the source of the documentation in (non perfect) english.
  - (6) `minitoc.dtx`, `fminitoc.dtx`, `fminitoc.bib`, `fminitoc.ist`, `fminitoc.lan`, `franc.sty`, `frbib.sty`, `frnew.sty`, `frplain1.bst` are the source (and tools) of the documentation in french<sup>2</sup>.
  - (7) `minitoc.pdf`, `minitoc.ps` are the documentation in (non perfect) english, in PDF and PostScript formats.
  - (8) `fminitoc.pdf`, `fminitoc.ps` are the documentation in french, in PDF and PostScript formats. The french documentation and source files must not be omitted.
  - (9) `pmk` is a shell script<sup>3</sup> to prepare the package and its documentation; you should adapt it to your needs. There are also four partial scripts (to be adapted):
    - `imk`, which prepares the package from `minitoc.ins` and `minitoc.dtx`;
    - `emk`, which prepares the english documentation from `minitoc.dtx`;
    - `fmk`, which prepares the french documentation from `fminitoc.dtx` and `minitoc.dtx`;
    - `rmk`, which sorts the files into classes (one directory for each class).

These scripts are currently written in C-shell, but there are very simple, and should be easy to convert in another classic shell.

- The files of (0) must be installed in a directory where  $\text{\LaTeX} 2_{\epsilon}$  finds `.dtx` and `.ins` files.
- The files of (1) must be installed in a directory where  $\text{\LaTeX} 2_{\epsilon}$  finds `.sty` files.
- The files of (2), (3), (4), (5), (6) and (9) must be installed in a separate directory, but must not be omitted.
- The files of (7) and (8) must be installed as on-line documentation. Note that the PostScript files of the distribution can be printed recto-verso on some printers.

<sup>2</sup>This seems rather strange. In fact, the english and french documentations are both in the `minitoc.dtx` file. `fminitoc.dtx` sets a flag then loads `minitoc.dtx`; hence the file `fminitoc.dtx` is much smaller than `minitoc.dtx`. Thus, `minitoc.ins` contains also some utility files which are automatically created (some `.sty` files, `minitoc.ist`, `mtcglo.ist`, `fminitoc.ist`, `minitoc.lan`, `fminitoc.lan`). The english and french versions are not word-by-word translations, but they are in parallel in the file `minitoc.dtx`, and this helps the maintenance. Note that the `fminitoc.ins` file does not exist.

<sup>3</sup>You can sip a cappuccino while this script is running!

Note that `minitoc.dtx` and `fminitoc.dtx` are (not so trivial) examples of using `minitoc` (with `hyperref`).



Table 3.1: List of files (minitoc.1)

**class (0) :**

-minitoc.ins  
-minitoc.dtx  
-fminitoc.dtx

**class (1) :**

-minitoc.sty  
-mtcoeff.sty  
-mtcpatchmem.sty  
-acadian.mld  
-acadien.mld  
-afrikaan.mld  
-afrikaans.mld  
-american.mld  
-arab.mld  
-arabic.mld  
-armenian.mld  
-austrian.mld  
-bahasa.mld  
-bangla.mld  
-basque.mld  
-bicig.mld  
-brazil.mld  
-brazilian.mld  
-breton.mld  
-british.mld  
-bulgarian.mld  
-bulgarianb.mld  
-buryat.mld  
-canadian.mld  
-canadien.mld  
-castillan.mld  
-castillian.mld  
-catalan.mld  
-chinese1.mld, and chinese1.mlo  
-chinese2.mld, and chinese2.mlo  
-croatian.mld  
-czech.mld  
-danish.mld  
-dutch.mld  
-english.mld  
-esperant.mld  
-esperanto.mld  
-estonian.mld  
-ethiopia.mld  
-ethiopian.mld  
-farsi1.mld, and farsi1.mlo  
-farsi2.mld, and farsi2.mlo  
-finnish.mld  
-finnish2.mld  
-francais.mld  
-french.mld  
-frenchb.mld  
-frenchle.mld  
-frenchpro.mld  
-galician.mld  
-german.mld  
-germanb.mld  
-greek.mld  
-greek-mono.mld

-greek-polydemo.mld  
-greek-polykatha.mld  
-guarani.mld  
-hangul1.mld, and hangul1.mlo  
-hangul2.mld, and hangul2.mlo  
-hangul3.mld, and hangul3.mlo  
-hangul4.mld, and hangul4.mlo  
-hanja1.mld, and hanja1.mlo  
-hanja2.mld, and hanja2.mlo  
-hebrew.mld  
-hungarian.mld  
-icelandic.mld  
-interlingua.mld  
-irish.mld  
-italian.mld  
-japanese.mld, and japanese.mlo  
-japanese2.mld, and japanese2.mlo  
-japanese3.mld, and japanese3.mlo  
-japanese4.mld, and japanese4.mlo  
-japanese5.mld, and japanese5.mlo  
-latin.mld  
-latin2.mld  
-latvian.mld  
-letton.mld  
-lithuanian.mld  
-lsorbian.mld  
-magyar.mld  
-magyar2.mld  
-malayalam-keli.mld  
-malayalam-rachana.mld  
-malayalam-rachana2.mld  
-mongol.mld  
-naustrian.mld  
-ngerman.mld  
-ngermanb.mld  
-norsk.mld  
-nynorsk.mld  
-polish.mld  
-polish2.mld  
-portuges.mld  
-portuguese.mld  
-romanian.mld  
-russian.mld  
-russianb.mld  
-russianc.mld  
-russian2m.mld  
-russian2o.mld  
-samin.mld  
-scottish.mld  
-serbian.mld  
-serbianc.mld  
-slovak.mld  
-slovene.mld  
-spanish.mld  
-spanish2.mld  
-spanish3.mld  
-swedish.mld  
-thai.mld, and thai.mlo  
-turkish.mld  
-UKenglish.mld  
-ukraine.mld

-USenglish.mld  
-usorbian.mld  
-vietnam.mld  
-vietnamese.mld  
-welsh.mld

**class (2) :**

-INSTALL  
-minitoc.l  
-README  
-catalog  
-TODO

**class (3) :**

-minitoc-ex.tex  
-mini-art.tex  
-add.tex  
-add.bib

**class (4) :**

-minitoc.bug  
-minitoc.sum

**class (5) :**

-minitoc.ins  
-minitoc.dtx  
-minitoc.bib  
-minitoc.ist  
-mtcglo.ist  
-minitoc.lan

**class (6) :**

-minitoc.dtx  
-fminitoc.dtx  
-fminitoc.bib  
-fminitoc.ist  
-fminitoc.lan  
-franc.sty  
-frbib.sty  
-frnew.sty  
-frplain1.bst

**class (7) :**

-minitoc.pdf  
-minitoc.ps

**class (8) :**

-fminitoc.pdf  
-fminitoc.ps

**class (9) :**

-pmk  
-imk  
-emk  
-fmk  
-rmk

# Chapter 4

## Memento

Tables	
4.1	Package options . . . . . 58
4.2	General commands . . . . . 59
4.3	Commands, part level . . . . . 60
4.4	Commands, chapter level . . . . . 61
4.5	Commands, section level . . . . . 62
4.6	Commands for horizontal rules . . . . . 62
4.7	Commands for page numbers . . . . . 63
4.8	Specific commands for parttocs and other mini-tables features . . . . 63
4.9	Preparation and insertion commands . . . . . 64
4.10	Adjustment commands . . . . . 64
4.11	Obsolete commands . . . . . 64

Table 4.1: Package options

Options	Default	Meaning
shorttext	*NO*	To use short extensions for auxiliary files.
loose, tight	loose	Spacing of lines in mini-tables.
k-loose, k-tight	k-loose	Spacing of lines in mini-tables (Koma-script classes).
dotted, undotted	dotted	Dotted lines from entries to page numbers.
insection	*NO*	Keeps floats (figures and tables) from drifting outside of their section. Useful if you use sectlofs/sectlots.
notoccite	*NO*	Useful if you have \cite commands in sectionning titles and use an unsorted bibliographic style.
hints, nohints	hints	Adds hints in the .log file. Useful to detect some problems.

Language options are listed in table 1.7 on page 27. Default: english.

Table 4.2: General commands

Command	Meaning
<code>\faketableofcontents</code>	Replaces <code>\tableofcontents</code> if you want mini-tables of contents but no main table of contents.
<code>\fakelistoffigures</code>	Replaces <code>\listoffigures</code> if you want mini-lists of figures but no main list of figures.
<code>\fakelistoftables</code>	Replaces <code>\listoftables</code> if you want mini-lists of tables but no main list of tables.
<code>\mtcselectlanguage{language}</code>	Loads <code>language.mld</code> to select a language for mini-tables titles.
<code>\mtcsetdepth{mini-table}{depth}</code>	Changes the depth for some mini-tables.
<code>\mtcsetfeature{mini-table}{before after pagestyle}{commands}</code>	Modifies the features for a mini-table.
<code>\mtcsetfont{mini-table}{sectioning-level}{font commands}</code>	Redefines a minitoc font command.
<code>\mtcsetformat{mini-table}{parameter}{value}</code>	Changes the layout of some mini-tables.
<code>\mtcsetpagenumbers{mini-table}{on off}</code>	Activates/inhibits page numbers in some mini-tables.
<code>\mtcsetrules{mini-table}{on off}</code>	Activates/inhibits horizontal rules in some mini-tables.
<code>\mtcsettitle{mini-table}{title string}</code>	Changes the title for some mini-tables.
<code>\mtcsettitlefont{mini-table}{font commands}</code>	Changes the font of the title for some mini-tables.
<code>\mtcskip</code>	To add a vertical skip between the mini-tables.
<code>\mtcskipamount</code>	Length of <code>\mtcskip</code> . Default: <code>\bigskipamount</code> .
<code>\tightmtcfalse</code>	Loose mini-tables. Default.
<code>\tightmtctrue</code>	Tight mini-tables.
<code>\ktightmtcfalse</code>	Loose mini-tables. Default. (Koma-script classes).
<code>\ktightmtctrue</code>	Tight mini-tables. (Koma-script classes).
<code>\undottedmtcfalse</code>	Dotted lines in mini-tables (from entry to page number). Default.
<code>\undottedmtctrue</code>	No dotted lines in mini-tables (from entry to page number).

Table 4.3: Commands, part level

Command	Meaning
<code>\doparttoc[x]</code>	Before <code>\[fake]tableofcontents</code> if you use <code>\parttoc*</code> .
<code>\dopartlof[x]</code>	Before <code>\[fake]listoffigures</code> if you use <code>\partlof*</code> .
<code>\dopartlot[x]</code>	Before <code>\[fake]listoftables</code> if you use <code>\partlot*</code> .
<code>\parttoc[x]</code>	After each <code>\part</code> command for which a parttoc is needed*.
<code>\partlof[x]</code>	After each <code>\part</code> command for which a partlof is needed*.
<code>\partlot[x]</code>	After each <code>\part</code> command for which a partlot is needed*.
<code>\setcounter{parttocdepth}{depth}</code>	Depth of the following parttocs. Analog to <code>tocdepth</code> . Default: 2. Has no action on partlofs and partlots.
<i>or:</i>	
<code>\mtcsetdepth{parttoc partlof partlot}{depth}</code>	Idem, but can also act on partlofs and partlots.
<code>\ptcindent</code>	Left/right indentation of a partial table. Default: 24pt.
<code>\ptcfont</code>	Font command for parttoc. Default: <code>\small\rmfamily\upshape\mdseries</code> (article) or: <code>\normalsize\rmfamily\upshape\mdseries</code> (book, report).
<code>\ptcCfont</code>	Font command for parttoc, chapter entries. Default: <code>\normalsize\rmfamily\upshape\bfseries</code> .
<code>\ptcSfont</code>	Font command for parttoc, section entries. Default: <code>\small\rmfamily\upshape\bfseries</code> (article) or: <code>\small\rmfamily\upshape\bfseries</code> (book, report).
<code>\ptcSSfont</code>	Font command for parttoc, subsection entries**.
<code>\ptcSSSfont</code>	Font command for parttoc, subsubsection entries**.
<code>\ptcPfont</code>	Font command for parttoc, paragraph entries**.
<code>\ptcSPfont</code>	Font command for parttoc, subparagraph entries**.
<code>\plffont</code>	Font for partlof. Default: <code>\small\rmfamily\upshape\mdseries</code> .
<code>\pltfont</code>	Font for partlot. Default: <code>\small\rmfamily\upshape\mdseries</code> .
<code>\ptctitle</code>	Title of parttocs. Default: Table of Contents.
<code>\plftitle</code>	Title of partlofs. Default: List of Figures.
<code>\plttitle</code>	Title of partlots. Default: List of Tables.
<code>\ptifont</code>	Font for partXXX titles. Default: <code>\Large\rmfamily\upshape\bfseries</code> (article) or: <code>\LARGE\rmfamily\upshape\bfseries</code> (book, report).

\*: `[x]` is an optional argument to set the position of the title; the setting is local for the `\partXXX` commands, global for the `\dopartXXX` commands. The values of `x` are: l for left (default), c for centered, r for right, n or e for no title.

\*\* : defaults like `\ptcfont`.

Table 4.4: Commands, chapter level

Command	Meaning
<code>\dominitoc[x]</code>	Before <code>\[fake]tableofcontents</code> if you use <code>\minitoc*</code> .
<code>\dominilof[x]</code>	Before <code>\[fake]listoffigures</code> if you use <code>\minilof*</code> .
<code>\dominilot[x]</code>	Before <code>\[fake]listoftables</code> if you use <code>\minilot*</code> .
<code>\minitoc[x]</code>	After each <code>\chapter</code> command for which a minitoc is needed*.
<code>\minilof[x]</code>	After each <code>\chapter</code> command for which a minilof is needed*.
<code>\minilot[x]</code>	After each <code>\chapter</code> command for which a minilot is needed*.
<code>\setcounter{minitocdepth}{depth}</code>	Depth of the following minitocs. Analog to <code>tocdepth</code> . Default: 2. Has no action on minilofs and minilots.
<i>or:</i>	
<code>\mtcsetdepth{minitoc minilof minilot}{depth}</code>	Idem, but can also act on minilofs and minilots.
<code>\mtcindent</code>	Left/right indentation of a mini-table. Default: 24pt.
<code>\mtcfont</code>	Font command for minitoc. Default: <code>\small\rmfamily\upshape\mdseries</code> .
<code>\mtcSfont</code>	Font command for minitoc, section entries. Default: <code>\small\rmfamily\upshape\bfseries</code> .
<code>\mtcSSfont</code>	Font command for minitoc, subsection entries**.
<code>\mtcSSSfont</code>	Font command for minitoc, subsubsection entries**.
<code>\mtcPfont</code>	Font command for minitoc, paragraph entries**.
<code>\mtcSPfont</code>	Font command for minitoc, subparagraph entries**.
<code>\mlffont</code>	Font for minilof. Default: <code>\small\rmfamily\upshape\mdseries</code> .
<code>\mltfont</code>	Font for minilot. Default: <code>\small\rmfamily\upshape\mdseries</code> .
<code>\mtctitle</code>	Title of minitocs. Default: Contents.
<code>\mlftitle</code>	Title of minilofs. Default: Figures.
<code>\mltttitle</code>	Title of minilots. Default: Tables.
<code>\mtifont</code>	Font for miniXXX titles. Default: <code>\large\rmfamily\upshape\bfseries</code> .

\*: `[x]` is an optional argument to set the position of the title; the setting is local for the `\miniXXX` commands, global for the `\dominiXXX` commands. The values of `x` are: l for left (default), c for centered, r for right, n or e for no title.

\*\* : defaults like `\mtcfont`.

Table 4.5: Commands, section level

Command	Meaning
<code>\dosecttoc[x]</code>	Before <code>\[fake]tableofcontents</code> if you use <code>\secttoc*</code> .
<code>\dosectlof[x]</code>	Before <code>\[fake]listoffigures</code> if you use <code>\sectlof*</code> .
<code>\dosectlot[x]</code>	Before <code>\[fake]listoftables</code> if you use <code>\sectlot*</code> .
<code>\secttoc[x]</code>	After each <code>\section</code> command for which a <code>secttoc</code> is needed*.
<code>\sectlof[x]</code>	After each <code>\section</code> command for which a <code>sectlof</code> is needed*.
<code>\sectlot[x]</code>	After each <code>\section</code> command for which a <code>sectlot</code> is needed*.
<code>\setcounter{secttocdepth}{depth}</code>	Depth of the following <code>secttocs</code> . Analog to <code>tocdepth</code> . Default: 2. Has no action on <code>sectlofs</code> and <code>sectlots</code> .
<i>or:</i>	
<code>\mtcsetdepth{secttoc sectlof sectlot}{depth}</code>	Idem, but can also act on <code>sectlofs</code> and <code>sectlots</code> .
<code>\stcindent</code>	Left/right indentation of a mini-table. Default: 24pt.
<code>\stcfont</code>	Font command for <code>secttoc</code> . Default: <code>\small\rmfamily\upshape\mdseries</code> .
<code>\stcSSfont</code>	Font command for <code>secttoc</code> , subsection entries**.
<code>\stcSSSfont</code>	Font command for <code>secttoc</code> , subsubsection entries**.
<code>\stcPfont</code>	Font command for <code>secttoc</code> , paragraph entries**.
<code>\mtcSPfont</code>	Font command for <code>secttoc</code> , subparagraph entries**.
<code>\slffont</code>	Font for <code>sectlof</code> . Default: <code>\small\rmfamily\upshape\mdseries</code> .
<code>\sltfont</code>	Font for <code>sectlot</code> . Default: <code>\small\rmfamily\upshape\mdseries</code> .
<code>\stctitle</code>	Title of <code>secttocs</code> . Default: Contents.
<code>\slftitle</code>	Title of <code>sectlofs</code> . Default: Figures.
<code>\slttitle</code>	Title of <code>sectlots</code> . Default: Tables.
<code>\stifont</code>	Font for <code>sectXXX</code> titles. Default: <code>\large\rmfamily\upshape\bfseries</code> .

\*: `[x]` is an optional argument to set the position of the title; the setting is local for the `\sectXXX` commands, global for the `\dosectXXX` commands. The values of `x` are: l for left (default), c for centered, r for right, n or e for no title.

\*\* : defaults like `\stcfont`.

Table 4.6: Commands for horizontal rules

Command	Meaning
<code>\[no]ptcrule</code>	Activates or inhibits rules in <code>parttocs</code> .
<code>\[no]mtcrule</code>	Activates or inhibits rules in <code>minitocs</code> .
<code>\[no]stcrule</code>	Activates or inhibits rules in <code>secttocs</code> .
<code>\[no]plfrule</code>	Activates or inhibits rules in <code>partlofs</code> .
<code>\[no]mlfrule</code>	Activates or inhibits rules in <code>minilofs</code> .
<code>\[no]slfrule</code>	Activates or inhibits rules in <code>sectlofs</code> .
<code>\[no]pltrule</code>	Activates or inhibits rules in <code>partlots</code> .
<code>\[no]mltrule</code>	Activates or inhibits rules in <code>minilots</code> .
<code>\[no]sltrule</code>	Activates or inhibits rules in <code>sectlots</code> .
<code>\mtcsetrules{mini-table}{on off}</code>	Activates/inhibits horizontal rules in some mini-tables.

By default, `parttocs` have no rules; `minitocs` and `secttocs` have rules. In articles, `parttocs` have rules.

Table 4.7: Commands for page numbers

Command	Meaning
<code>\[no]ptcpagenumbers</code>	Activates or inhibits page numbers in parttocs.
<code>\[no]plfpagenumbers</code>	Activates or inhibits page numbers in partlofs.
<code>\[no]pltpagenumbers</code>	Activates or inhibits page numbers in partlots.
<code>\[no]mtcpagenumbers</code>	Activates or inhibits page numbers in minitocs.
<code>\[no]mlfpagenumbers</code>	Activates or inhibits page numbers in minilofs.
<code>\[no]mltpagenumbers</code>	Activates or inhibits page numbers in minilots.
<code>\[no]stcpagenumbers</code>	Activates or inhibits page numbers in secttocs.
<code>\[no]slfpagenumbers</code>	Activates or inhibits page numbers in sectlofs.
<code>\[no]sltpagenumbers</code>	Activates or inhibits page numbers in sectlots.
<code>\mtcsetpagenumbers{mini-table}{on off}</code>	Activates/inhibits page numbers in some mini-tables.

By default, the page numbers are present.

Table 4.8: Specific commands for parttocs and other mini-tables features

Command	Default	Meaning
<code>\beforeparttoc</code>	<code>\cleardoublepage</code>	Action before a parttoc.
<code>\beforepartlof</code>	<code>\cleardoublepage</code>	Action before a partlof.
<code>\beforepartlot</code>	<code>\cleardoublepage</code>	Action before a partlot.
<code>\afterparttoc</code>	<code>\cleardoublepage</code>	Action after a parttoc.
<code>\afterpartlof</code>	<code>\cleardoublepage</code>	Action after a partlof.
<code>\afterpartlot</code>	<code>\cleardoublepage</code>	Action after a partlot.
<code>\thispageparttocstyle</code>	<code>\thispagestyle{empty}</code>	Page style for a parttoc.
<code>\thispagepartlofstyle</code>	<code>\thispagestyle{empty}</code>	Page style for a partlof.
<code>\thispagepartlotstyle</code>	<code>\thispagestyle{empty}</code>	Page style for a partlot.
<code>\beforeminitoc</code>	<code>\empty</code>	Action before a minitoc.
<code>\beforeminilof</code>	<code>\empty</code>	Action before a minilof.
<code>\beforeminilot</code>	<code>\empty</code>	Action before a minilot.
<code>\afterminitoc</code>	<code>\empty</code>	Action after a minitoc.
<code>\afterminilof</code>	<code>\empty</code>	Action after a minilof.
<code>\afterminilot</code>	<code>\empty</code>	Action after a minilot.
<code>\thispageminitocstyle</code>	<code>\empty</code>	Page style for a minitoc.
<code>\thispageminilofstyle</code>	<code>\empty</code>	Page style for a minilof.
<code>\thispageminilotstyle</code>	<code>\empty</code>	Page style for a minilot.
<code>\beforesecttoc</code>	<code>\empty</code>	Action before a secttoc.
<code>\beforesectlof</code>	<code>\empty</code>	Action before a sectlof.
<code>\beforesectlot</code>	<code>\empty</code>	Action before a sectlot.
<code>\aftersecttoc</code>	<code>\empty</code>	Action after a secttoc.
<code>\aftersectlof</code>	<code>\empty</code>	Action after a sectlof.
<code>\aftersectlot</code>	<code>\empty</code>	Action after a sectlot.
<code>\thispagesecttocstyle</code>	<code>\empty</code>	Page style for a secttoc.
<code>\thispagesectlofstyle</code>	<code>\empty</code>	Page style for a sectlof.
<code>\thispagesectlotstyle</code>	<code>\empty</code>	Page style for a sectlot.
<code>\mtcsetfeature{mini-table}{before after pagestyle}{commands}</code>	Modifies the features for a mini-table.	

Table 4.9: Preparation and insertion commands

Type	Phase	Level		
		part	chapter	section
table of contents	preparation	<code>\doparttoc[p]</code>	<code>\dominitoc[p]</code>	<code>\dosecttoc[p]</code>
	insertion	<code>\parttoc[p]</code>	<code>\minitoc[p]</code>	<code>\secttoc[p]</code>
list of figures	preparation	<code>\dopartlof[p]</code>	<code>\dominiloof[p]</code>	<code>\dosectlof[p]</code>
	insertion	<code>\partlof[p]</code>	<code>\miniloof[p]</code>	<code>\sectlof[p]</code>
list of tables	preparation	<code>\dopartlot[p]</code>	<code>\dominilot[p]</code>	<code>\dosectlot[p]</code>
	insertion	<code>\partlot[p]</code>	<code>\minilot[p]</code>	<code>\sectlot[p]</code>

Each of these commands accepts one optional argument  $p$ , which specifies the position of the title of the mini-table. This argument  $p$  has a global effect for the preparation commands, but local for the insertion commands. It is a letter: [l] for left aligned (default), [c] for centered, [r] for right aligned, [e] or [n] for empty (no title).

Table 4.10: Adjustment commands

Command	Meaning
<code>\adjustptc[n]</code>	Adjusts (increments) the parttoc counter <code>ptc</code> by $n$ .
<code>\adjustmtc[n]</code>	Adjusts (increments) the minitoc counter <code>mtc</code> by $n$ .
<code>\adjuststc[n]</code>	Adjusts (increments) the secttoc counter <code>stc</code> by $n$ .
<code>\decrementptc</code>	Adjusts (decrements by 1) the parttoc counter <code>ptc</code> .
<code>\decrementmtc</code>	Adjusts (decrements by 1) the minitoc counter <code>mtc</code> .
<code>\decrementstc</code>	Adjusts (decrements by 1) the secttoc counter <code>stc</code> .
<code>\incrementptc</code>	Adjusts (increments by 1) the parttoc counter <code>ptc</code> .
<code>\incrementmtc</code>	Adjusts (increments by 1) the minitoc counter <code>mtc</code> .
<code>\incrementstc</code>	Adjusts (increments by 1) the secttoc counter <code>stc</code> .
<code>\mtcaddpart[title]</code>	Adds the title of a <code>\part*</code> in the ToC.
<code>\mtcaddchapter[title]</code>	Adds the title of a <code>\chapter*</code> in the ToC.
<code>\mtcaddsection[title]</code>	Adds the title of a <code>\section*</code> in the ToC.
<code>\mtcfixindex[chapter section part]</code>	Adjusts the entry for index in the ToC.
<code>\mtcfixglossary[chapter section part]</code>	Adjusts the entry for glossary in the ToC.
<code>\begin{mtchideinmaintoc}[depth] ... \end{mtchideinmaintoc}</code>	Environment to hide entries in the main table of contents.
<code>\begin{mtchideinmainlof}[depth] ... \end{mtchideinmainlof}</code>	Environment to hide entries in the main list of figures.
<code>\begin{mtchideinmainlot}[depth] ... \end{mtchideinmainlot}</code>	Environment to hide entries in the main list of tables.

Table 4.11: Obsolete commands

Command	Meaning
<code>\firstpartis{N}</code>	$N$ is the number of the first part.
<code>\firstchapteris{N}</code>	$N$ is the number of the first chapter.
<code>\firstsectionis{N}</code>	$N$ is the number of the first section.



# Acknowledgments

I need to thank the following peoples<sup>1</sup>, for their help, their questions, their interventions in the news groups<sup>2</sup>, and/or the packages and classes they wrote:

Hassan ABOLHASSANI, Alex AJ, Achod André ARADIAN, Donald ARSENEAU, David ASPINALL, Philipp BACHMANN, Marin BALGARENSKY, Benjamin BAYART, Stephan P. VON BECHTOLSHEIM, Emmanuel BEFFARA, Javier BEZOS, Laurent BLOCH, Johannes BRAAMS, Daniel BOURBONNAIS, David CARLISLE, Steven Douglas COCHRAN, David B. COOK, Oliver CORFF, Prakash COUNTCHAM, Serguei DACHIAN, Adrian DAERR, Ben DE RYDT, Lyndon DUDDING, Marko ÈEHAJA, Victor EIJKHOUT, Thomas ESSER, Karl F. EVERITT, Robin FAIRBAIRNS, Ulrike FISCHER, Mohammad GHODSI, Michel GOOSSENS, Boumediene HAMZI, Danny HEAP, Thorsten HEIN, Florence HENRY, Morten HØGHOLM, Don HOSEK, Yufan HU, Dmitry IVANOV, Bernd JAEHNE, Radwan JALAM, Dan JURAFSKY, Bil KLEB, Thankmar KRONZUCKER, Alexej M. KRYUKOV, Toshiki KUMAZAWA, Frank KÜSTER, Fabio LANARI, Robert LANGE, Dag LANGMYHR, Jean-Marc LASGOUTTES, Claire LAUVERNET, Werner LEMBERG, Thomas LEONHARDT, Stéphane LEPOLOZEC, Adam LEWENBERG, Knut LICKERT, Dan LUECKING, Anders LYHNE, Marcus MARR, Rowland McDONNELL, Yanick MICHOU, Frank MITTELBACH, Lapo Filippo MORI, Michael MORRISON, Cuong NGUYEN, Julien NICOLAS, Heiko OBERDIEK, Piet VAN OOSTRUM, Palash Baran PAL, Roozbeh POURNADER, Sebastian RAHTZ, Tony ROBERTS, Denis ROEGEL, Jan Michel RYNNING, David SAMSOEN, Uwe SCHNEIDER, Dung TA QUANG, Laurent TORDELLA, Stefan ULRICH, Jari VAARIO, Vincent VAQUIN, Didier VERNÀ, Sylvain VESCO, Nigel WARD, Staszek WAWRYKIEWICZ, Stephan WEBANCK, Peter R. WILSON, and Tim WRIGHT.

---

<sup>1</sup>And I apologize to whose I forgot.

<sup>2</sup>Mainly, `fr.comp.text.tex` and `comp.text.tex`.

## **Part II**

# **Implementation**

## Table of Contents

<b>5</b>	<b>Commented code of <code>minitoc.sty</code></b>	<b>68</b>
<b>6</b>	<b>Commented code of <code>mtcoff.sty</code></b>	<b>210</b>
<b>7</b>	<b>Patch for the <code>memoir</code> class</b>	<b>222</b>
<b>8</b>	<b>The <code>minitoc</code> language definition (<code>.mld</code>) and object (<code>.mlo</code>) files</b>	<b>224</b>

## Chapter 5

# Commented code of `minitoc.sty`

### Contents

---

<b>5.1</b>	<b>Introduction . . . . .</b>	<b>71</b>
<b>5.2</b>	<b>Identification code . . . . .</b>	<b>71</b>
<b>5.3</b>	<b>A file descriptor to write . . . . .</b>	<b>72</b>
<b>5.4</b>	<b>Indentation and skip . . . . .</b>	<b>72</b>
<b>5.5</b>	<b>Tests and flags . . . . .</b>	<b>72</b>
5.5.1	Use of section-level mini-lists of floats. . . . .	72
5.5.2	Presence of some packages and classes. . . . .	73
5.5.3	Presence or absence of some sectionning commands . . . .	73
5.5.4	Flags to check to see if some commands are used . . . . .	74
<b>5.6</b>	<b>Preparation for the <code>notoccite</code> option . . . . .</b>	<b>75</b>
<b>5.7</b>	<b>Preparation for the <code>tight</code> and <code>k-tight</code> options . . . . .</b>	<b>75</b>
<b>5.8</b>	<b>Preparation to work with <code>hyperref</code> . . . . .</b>	<b>75</b>
<b>5.9</b>	<b>Check if the <code>sectsty</code> package is loaded, and when . . . . .</b>	<b>76</b>
<b>5.10</b>	<b>Is the <code>memoir</code> class loaded? . . . . .</b>	<b>76</b>
<b>5.11</b>	<b>Testing the emptiness of a file . . . . .</b>	<b>77</b>
<b>5.12</b>	<b>Internal macros to decrement <code>minitoc</code> counters . . . . .</b>	<b>78</b>
<b>5.13</b>	<b>Patching the <code>\part</code> command . . . . .</b>	<b>79</b>
<b>5.14</b>	<b>Adding an entry in the TOC for a starred part . . . . .</b>	<b>79</b>
<b>5.15</b>	<b>Section level macros . . . . .</b>	<b>80</b>
<b>5.16</b>	<b>Corrections for numbering . . . . .</b>	<b>80</b>
<b>5.17</b>	<b>Patching the <code>\section</code> command . . . . .</b>	<b>81</b>
<b>5.18</b>	<b>Adding an entry in the TOC for a starred section . . . . .</b>	<b>81</b>
<b>5.19</b>	<b>Chapter level macros . . . . .</b>	<b>82</b>
<b>5.20</b>	<b>Patching the <code>\chapter</code> command . . . . .</b>	<b>82</b>
<b>5.21</b>	<b>Adding an entry in the TOC for a starred chapter . . . . .</b>	<b>82</b>
<b>5.22</b>	<b>Miscellaneous declarations . . . . .</b>	<b>83</b>
<b>5.23</b>	<b>Autoconfiguration of extensions . . . . .</b>	<b>83</b>
<b>5.24</b>	<b>Detecting obsolete versions of <code>LaTeX</code> . . . . .</b>	<b>85</b>
<b>5.25</b>	<b>A macro to make a TOC entry without leaders nor page numbers</b>	<b>85</b>
<b>5.26</b>	<b>Default values for the page-number customization macros . . . .</b>	<b>86</b>
<b>5.27</b>	<b>“Features” for the mini-tables . . . . .</b>	<b>88</b>

<b>5.28</b>	<b>Fake tables of contents</b>	<b>90</b>
<b>5.29</b>	<b>Depth counters for minilofs and minilots</b>	<b>91</b>
<b>5.30</b>	<b>Chapter level commands</b>	<b>91</b>
<b>5.31</b>	<b>Starred parts, chapters or sections</b>	<b>91</b>
<b>5.32</b>	<b>Font commands for the mini-tables</b>	<b>93</b>
<b>5.33</b>	<b>Internal commands to position the mini-table titles</b>	<b>93</b>
<b>5.34</b>	<b>The mtc@verse environment</b>	<b>94</b>
<b>5.35</b>	<b>The \minitoc, \minilof, and \minilot commands</b>	<b>94</b>
5.35.1	The \minitoc command	94
5.35.2	The \minilof command	97
5.35.3	The \minilot command	100
<b>5.36</b>	<b>Patching the \chapter command, continued</b>	<b>103</b>
<b>5.37</b>	<b>The \addstarred... commands</b>	<b>104</b>
<b>5.38</b>	<b>TOC entries without leaders</b>	<b>104</b>
<b>5.39</b>	<b>Mini-tables with or without leaders</b>	<b>105</b>
<b>5.40</b>	<b>The \dominitoc command and its siblings</b>	<b>106</b>
5.40.1	Analysis and splitting of the TOC file	108
<b>5.41</b>	<b>Mini-lists of figures</b>	<b>112</b>
5.41.1	Analysis and splitting of the list of figures file	112
<b>5.42</b>	<b>Mini-lists of tables</b>	<b>114</b>
5.42.1	Analysis and splitting of the list of tables file	114
<b>5.43</b>	<b>Macro to write a contents line</b>	<b>117</b>
<b>5.44</b>	<b>Depth counters for partlofs and partlots</b>	<b>118</b>
<b>5.45</b>	<b>Part level commands</b>	<b>118</b>
<b>5.46</b>	<b>Fonts for the parttocs</b>	<b>119</b>
<b>5.47</b>	<b>Default titles for part-level mini-tables</b>	<b>120</b>
<b>5.48</b>	<b>The ptc@verse environment</b>	<b>122</b>
<b>5.49</b>	<b>The part level mini-tables: \parttoc, \partlof, and \partlot</b>	<b>122</b>
5.49.1	The \parttoc command	122
5.49.2	The \partlof command	125
5.49.3	The \partlot command	128
<b>5.50</b>	<b>Auxiliary commands for printing parttocs</b>	<b>131</b>
<b>5.51</b>	<b>Patching the \part command, continued</b>	<b>131</b>
<b>5.52</b>	<b>The \doparttoc command and its siblings</b>	<b>132</b>
5.52.1	Processing macros for the parttocs	133
5.52.2	Processing macros for the partlofs	137
5.52.3	Processing macros for the partlots	139
<b>5.53</b>	<b>Depth counters for sectlofs and sectlots</b>	<b>141</b>
<b>5.54</b>	<b>Section-level commands</b>	<b>142</b>
<b>5.55</b>	<b>Fonts commands for secttocs and co.</b>	<b>142</b>
<b>5.56</b>	<b>Internal macros for title positionning</b>	<b>143</b>
<b>5.57</b>	<b>The stc@verse environment</b>	<b>143</b>
<b>5.58</b>	<b>The \secttoc, \sectlof, and \sectlot commands</b>	<b>144</b>
5.58.1	The \secttoc command	144
5.58.2	The \sectlof command	146
5.58.3	The \sectlot command	149
<b>5.59</b>	<b>Auxiliary internal commands, section level</b>	<b>152</b>
<b>5.60</b>	<b>Patching the \section command (continued)</b>	<b>152</b>
<b>5.61</b>	<b>The \dosecttoc command and siblings</b>	<b>153</b>
<b>5.62</b>	<b>End of section-level commands</b>	<b>162</b>
<b>5.63</b>	<b>Necessary \l@... commands</b>	<b>162</b>
<b>5.64</b>	<b>The horizontal rules and their default values</b>	<b>162</b>
<b>5.65</b>	<b>The \mtcset... commands</b>	<b>163</b>
5.65.1	Keywords for the \mtcset... commands	163

5.65.2	The <code>\mtcsetfont</code> command	165
5.65.3	The <code>\mtcsettitlefont</code> command	167
5.65.4	The <code>\mtcsettitle</code> command	169
5.65.5	The <code>\mtcsetformat</code> command	170
5.65.6	The <code>\mtcsetpagenumbers</code> command	174
5.65.7	The <code>\mtcsetrules</code> command	175
5.65.8	The <code>\mtcsetfeature</code> command	177
5.65.9	The <code>\mtcsetdepth</code> command	179
<b>5.66</b>	<b>The <code>mtchideinmaintoc</code> environment</b>	<b>180</b>
<b>5.67</b>	<b>Fixing the “Index” entry in the TOC</b>	<b>182</b>
<b>5.68</b>	<b>Fixing the “Glossary” entry in the TOC</b>	<b>183</b>
<b>5.69</b>	<b>The <code>\mtcselectlanguage</code> command</b>	<b>185</b>
<b>5.70</b>	<b>The <code>\mtcloadmlo</code> internal command</b>	<b>185</b>
<b>5.71</b>	<b>The “coffee breaks”</b>	<b>186</b>
<b>5.72</b>	<b>Initialization of counters</b>	<b>186</b>
<b>5.73</b>	<b>Declarations for simple options</b>	<b>186</b>
5.73.1	Options <code>tight</code> and <code>loose</code> , <code>k-tight</code> and <code>k-loose</code>	186
5.73.2	Options <code>checkfiles</code> and <code>nocheckfiles</code>	187
5.73.3	Options <code>dotted</code> and <code>undotted</code>	187
5.73.4	Option <code>notoccite</code>	187
5.73.5	Option <code>shorttext</code>	187
<b>5.74</b>	<b>The <code>insection</code> option</b>	<b>187</b>
<b>5.75</b>	<b>Language options</b>	<b>188</b>
<b>5.76</b>	<b>The <code>hints</code> option</b>	<b>190</b>
5.76.1	First part: <code>\mtc@hints@begindoc</code>	191
5.76.1.1	Hint about the <code>appendix</code> package	191
5.76.1.2	Hint about the <code>tocbibind</code> package	191
5.76.1.3	Hint about the <code>tocloft</code> package	192
5.76.1.4	Hint about the <code>titletoc</code> package	192
5.76.1.5	Hint about the <code>placeins</code> package	192
5.76.1.6	Hint about the <code>memoir</code> class	193
5.76.1.7	Hint about the <code>amsart</code> and <code>amsproc</code> classes	193
5.76.1.8	Hint about the <code>amsbook</code> class	194
5.76.1.9	Hint about the <code>abstract</code> package	194
5.76.1.10	Hint about the alteration of the sectioning commands	194
5.76.1.10.1	Alteration of <code>\part</code>	194
5.76.1.10.2	Alteration of <code>\chapter</code>	195
5.76.1.10.3	Alteration of <code>\section</code>	195
5.76.1.11	Hint about the coherence of the calling sequences of some commands	196
5.76.2	Final part: <code>\mtc@hints@enddoc</code>	197
5.76.2.1	Hint about <code>\sect-lof lot</code> and the <code>insection</code> option	197
5.76.2.2	Final part of the coherence tests	198
5.76.2.3	Check if the main tables have been prepared (first part)	199
5.76.2.4	Check if the main tables have been prepared (second part)	201
5.76.2.5	Check the number of mini-tables, in case of short extensions	203
5.76.2.6	Final part of the hint about the <code>sectsty</code> package	204
5.76.2.7	Check if empty mini-tables have been detected	204
5.76.2.8	Check if obsolete commands have been used	206

5.76.2.9	Check if some hints have been written . . . . .	207
<b>5.77</b>	<b>Processing of options . . . . .</b>	<b>207</b>
5.77.1	Processing the <code>insection</code> option . . . . .	208
5.77.2	Processing the <code>notoccite</code> option . . . . .	208
5.77.3	Processing the <code>hints</code> option . . . . .	208
5.77.4	Saving the sectionning commands . . . . .	208
<b>5.78</b>	<b>The <code>fminitoc.dtx</code> file . . . . .</b>	<b>209</b>

## Figures

5.1	Hack to detect the limitation to short extensions. . . . .	84
-----	--	----

## 5.1 Introduction

This long chapter presents the code of the `minitoc` package and attempts to explain it. Some comments of the original source file <sup>1</sup> are skipped, like the history, because they do not need further examination (they will be put in the change history).

The code is split in sections to make the reading easier, and the sections are sometimes reordered to make the reading easier.

Most of the `minitoc` external commands have `mtc`, `ptc`, `stc`, or one of the mini-table names (`parttoc`, ..., `sectlot`) in their names. Most of the `minitoc` internal commands have `@mtc`, `@ptc`, `@stc`, or `parttoc@`, ..., `sectlot@` in their names, or a similar convention. The few exceptions should be explicit enough to not conflict with other packages.

## 5.2 Identification code

The code of `minitoc.sty` starts here:

```

1 <*minitoc>

\NeedsTeXFormat{LaTeX2e}[1996/06/01]%
\ProvidesPackage{minitoc}%
\PackageInfo{minitoc}%
2 \NeedsTeXFormat{LaTeX2e}[1996/06/01]%
3 \ProvidesPackage{minitoc}%
4   [2005/09/16 v43 Package minitoc]
5 \PackageInfo{minitoc}%
6   {*** minitoc package, version 43 ***\@gobble}

```

<sup>1</sup>The source file of version #42. Version #43 includes the conversion of the package to `.dtx`-`.ins` format. Version #42 has not been distributed because of that.

### 5.3 A file descriptor to write

`\tf@mtc` A file descriptor is needed to write the files containing the mini-tables, it is `\tf@mtc`.  
`\newwrite`  
 7 `\newwrite\tf@mtc`

### 5.4 Indentation and skip

`\mtcindent` We define the indentation `\mtcindent` (both sides) of the mini-tables and the command  
`\mtcskip` `\mtcskip` to make a vertical skip before a mini-table, its value is `\mtcskipamount`  
`\mtcskipamount` (default: `\bigskipamount`).  
`\parskip`  
 8 `\newlength\mtcindent`  
 9 `\newskip\mtcskipamount`  
 10 `\setlength{\mtcskipamount}{\bigskipamount}`  
 11 `\def\mtcskip{\unskip\removeatlastskip{\parskip=\z@\addvspace{\mtcskipamount}}}`

Note that `\mtcskip` uses a local group to avoid the influence of `\parskip`.

### 5.5 Tests and flags

We need to declare some flags<sup>2</sup> (via `\newif`) to detect the loading of some packages or classes and the availability of some commands (this will be used by the `hints` option (section 5.76 on page 190) or to allow the definition of some `minitoc` commands).

#### 5.5.1 Use of section-level mini-lists of floats.

`\if@mtc@sect@floats@` We will check if the commands `\dosectlof` and `\dosectlot` are used:  
`\dosectlof`  
`\dosectlot` 12 `\newif\if@mtc@sect@floats@ \@mtc@sect@floats@false`

---

<sup>2</sup>Not so many years ago, some authors had a preference for using counters rather than flags, because a flag costs 3 control sequences (`\iffoo`, `\foofalse` and `\footrue`), which use memory. But the number of count registers is limited to 256 in the native  $\text{\TeX}$  engine (much more with  $\epsilon\text{-\TeX}$ , but still limited in number), while memory has become rather cheap today. And a code programmed with flags (`\iffoo ... \else ... \fi`) is easier to structure than a code programmed with counters, IMHO.



### 5.5.2 Presence of some packages and classes.

`\if@mtc@placeinsLoaded@` We will check if the `placeins` package is loaded, then if the `memoir` is loaded (and if it is a recent version), then if the `sectsty` package is loaded (before or after `minitoc`).

`\if@mtc@memoirLoaded@`

`\if@mtc@memoirnew@`

`\if@mtc@sectstyLoaded@` 13 `\newif\if@mtc@placeinsLoaded@ \@mtc@placeinsLoaded@false`

`\if@mtc@sectstyLoaded@a@` 14 `\newif\if@mtc@memoirLoaded@ \@mtc@memoirLoaded@false`

15 `\newif\if@mtc@memoirnew@ \@mtc@memoirnew@false`

16 `\newif\if@mtc@sectstyLoaded@ \@mtc@sectstyLoaded@false`

17 `\newif\if@mtc@sectstyLoaded@a@ \@mtc@sectstyLoaded@a@false`

`\if@mtc@empty@parttoc@` We will check if you have attempted to insert some empty mini-tables:

`\if@mtc@empty@partlof@`

`\if@mtc@empty@partlot@` 18 `\newif\if@mtc@empty@parttoc@ \@mtc@empty@parttoc@false`

`\if@mtc@empty@minitoc@` 19 `\newif\if@mtc@empty@partlof@ \@mtc@empty@partlof@false`

`\if@mtc@empty@minitoc@` 20 `\newif\if@mtc@empty@partlot@ \@mtc@empty@partlot@false`

`\if@mtc@empty@minilof@` 21 `\newif\if@mtc@empty@minitoc@ \@mtc@empty@minitoc@false`

`\if@mtc@empty@minilof@` 22 `\newif\if@mtc@empty@minilof@ \@mtc@empty@minilof@false`

`\if@mtc@empty@sectlot@` 23 `\newif\if@mtc@empty@minilof@ \@mtc@empty@minilof@false`

`\if@mtc@empty@sectlof@` 24 `\newif\if@mtc@empty@minilof@ \@mtc@empty@minilof@false`

`\if@mtc@empty@sectlof@` 25 `\newif\if@mtc@empty@sectlof@ \@mtc@empty@sectlof@false`

`\if@mtc@empty@sectlot@` 26 `\newif\if@mtc@empty@sectlot@ \@mtc@empty@sectlot@false`

### 5.5.3 Presence or absence of some sectionning commands

We define and set flags about the presence of the sectionning commands (in fact, the counters associated with these commands).

`\if@mtc@part@def@` The part counter:

27 `\newif\if@mtc@part@def@ \@mtc@part@def@false`

28 `\@ifundefined{part}{\@mtc@part@def@false}{\@mtc@part@def@true}`

`\if@mtc@chapter@def@` The chapter counter:

29 `\newif\if@mtc@chapter@def@ \@mtc@chapter@def@false`

30 `\@ifundefined{chapter}{\@mtc@chapter@def@false}{\@mtc@chapter@def@true}`

`\if@mtc@section@def@` The section counter:

31 `\newif\if@mtc@section@def@ \@mtc@section@def@false`

32 `\@ifundefined{section}{\@mtc@section@def@false}{\@mtc@section@def@true}`

We define and set flags about the absence of the sectionning commands.

`\if@mtc@part@undef@` The part counter:

```
33 \newif\if@mtc@part@undef@ \@mtc@part@undef@true
34 \ifundefined{part}{\@mtc@part@undef@true}{\@mtc@part@undef@false}
```

`\if@mtc@chapter@undef@` The chapter counter:

```
35 \newif\if@mtc@chapter@undef@ \@mtc@chapter@undef@true
36 \ifundefined{chapter}{\@mtc@chapter@undef@true}{\@mtc@chapter@undef@false}
```

`\if@mtc@section@undef@` The section counter:

```
37 \newif\if@mtc@section@undef@ \@mtc@section@undef@true
38 \ifundefined{section}{\@mtc@section@undef@true}{\@mtc@section@undef@false}
```

## 5.5.4 Flags to check to see if some commands are used

We define a pair of flags for each mini-table type: one for the command itself and one for the preparation command (`\do...`). These flags will be used by the hints package option (section 5.76 on page 190).

`\if@parttoc@used@` For the part level:

```
\if@partlof@used@
\if@partlot@used@ 39 \newif\if@parttoc@used@ \global\@parttoc@used@false
\if@doparttoc@used@ 40 \newif\if@partlof@used@ \global\@partlof@used@false
\if@dopartlof@used@ 41 \newif\if@partlot@used@ \global\@partlot@used@false
\if@dopartlot@used@ 42 \newif\if@doparttoc@used@ \global\@doparttoc@used@false
43 \newif\if@dopartlof@used@ \global\@dopartlof@used@false
44 \newif\if@dopartlot@used@ \global\@dopartlot@used@false
```

`\if@minitoc@used@` For the chapter level:

```
\if@minilof@used@
\if@minilot@used@ 45 \newif\if@minitoc@used@ \global\@minitoc@used@false
\if@dominitoc@used@ 46 \newif\if@minilof@used@ \global\@minilof@used@false
\if@dominilof@used@ 47 \newif\if@minilot@used@ \global\@minilot@used@false
\if@dominilot@used@ 48 \newif\if@dominitoc@used@ \global\@dominitoc@used@false
49 \newif\if@dominilof@used@ \global\@dominilof@used@false
50 \newif\if@dominilot@used@ \global\@dominilot@used@false
```

```

\if@secttoc@used@ For the section level:
\if@sectlof@used@
\if@sectlot@used@ 51 \newif\if@secttoc@used@ \global\@secttoc@used@false
\if@dosecttoc@used@ 52 \newif\if@sectlof@used@ \global\@sectlof@used@false
\if@dosectlof@used@ 53 \newif\if@sectlot@used@ \global\@sectlot@used@false
\if@dosectlot@used@ 54 \newif\if@dosecttoc@used@ \global\@dosecttoc@used@false
55 \newif\if@dosectlof@used@ \global\@dosectlof@used@false
56 \newif\if@dosectlot@used@ \global\@dosectlot@used@false

\if@firstpartis@used@ We also detect the use of some obsolete commands:
\if@firstchapteris@used@
\if@firstsectionis@used@ 57 \newif\if@firstpartis@used@ \global\@firstpartis@used@false
58 \newif\if@firstchapteris@used@ \global\@firstchapteris@used@false
59 \newif\if@firstsectionis@used@ \global\@firstsectionis@used@false

```

## 5.6 Preparation for the notoccite option

`\mtc@hook@beforeinputfile` We declare a flag for the presence of this option and the new internal command `\mtc@hook@beforeinputfile`, which is a “hook” command (redefinable command) used by this option (requested by Donald ARSENEAU for his notoccite package [2]). See section 1.4.6 on page 38.

```

60 \newif\if@mtc@notoccite@ \@mtc@notoccite@false
61 \@ifundefined{mtc@hook@beforeinputfile}%
62   {\let\mtc@hook@beforeinputfile\relax}{}

```

## 5.7 Preparation for the tight and k-tight options

`\iftightmtc` We just declare a flag for each of these options; they are set false by default (loose and `\ifktightmtc` k-loose options):

```

63 \newif\iftightmtc \tightmtcfalse
64 \newif\ifktightmtc \ktightmtcfalse

```

## 5.8 Preparation to work with hyperref

`\AtBeginDocument` This code prepares the interface with the hyperref package [39]. A flag is defined, `\if@mtc@hyper@used@` then this preparation is performed in `\AtBeginDocument` if this package is loaded. This action defines some commands for the hyperref package.

```

65 \PackageInfo{minitoc}{*** compatible with hyperref ***\@gobble}
66 \newif\if@mtc@hyper@used@ \global\@mtc@hyper@used@false
67 \AtBeginDocument{%
68   \@ifpackageloaded{hyperref}{%
69     \global\@mtc@hyper@used@true
70     \def\toclevel@xpart{1000}%
71     \def\toclevel@xchapter{1000}%
72     \def\toclevel@xsect{1000}%
73     \let\toclevel@starpert\toclevel@part
74     \let\toclevel@starchapter\toclevel@chapter
75     \let\toclevel@starsection\toclevel@section
76     \let\toclevel@starsubsection\toclevel@subsection
77     \let\toclevel@starsubsubsection\toclevel@subsubsection
78     \let\toclevel@starparagraph\toclevel@paragraph
79     \let\toclevel@starsubparagraph\toclevel@subparagraph
80   }{}%
81 }

```

## 5.9 Check if the sectsty package is loaded, and when

\AtBeginDocument We must test if the sectsty package [32] is loaded before or after minitoc, so we test  
 \if@mtc@sectstyLoaded@ when minitoc is loaded and also in a \AtBeginDocument block, when all packages  
 \if@mtc@sectstyLoaded@a@ have been loaded. See section 5.76.2.6 on page 204.

```

82 \@ifpackageloaded{sectsty}{\@mtc@sectstyLoaded@true}{}
83 \AtBeginDocument{\@ifpackageloaded{sectsty}{\@mtc@sectstyLoaded@a@true}{} }

```

## 5.10 Is the memoir class loaded?

\if@mtc@memoirLoaded@ We test if the memoir [48] class is loaded. This class needs some compatibility adjust-  
 \if@mtc@memoirnew@ ments or may be incompatible if too recent. In the later case, a patch is inserted (see  
 \if@mtcpatchmemoir@ chapter 7 on page 222).

```

84 \newif\if@mtcpatchmemoir@ \@mtcpatchmemoir@false
85 \ifclassloaded{memoir}%
86   {\@mtc@memoirLoaded@true\relax%
87   \PackageInfo{minitoc}%
88     {*** the memoir class is loaded: compatibility attempted ***\@gobble}}%
89   {\@mtc@memoirLoaded@false}
90 \if@mtc@memoirLoaded@
91   \@ifundefined{@m@chapter}%
92     {\@mtc@memoirnew@false\PackageInfo{minitoc}%
93       {*** old version of the memoir class ***\@gobble}}
94     {\@mtc@memoirnew@true\PackageInfo{minitoc}%
95       {*** recent version of the memoir class ***\@gobble}}

```

```

96     \PackageInfo{minitoc}{*** This version of the memoir class uses \MessageBreak
97                                     a version of \string\chapter\space which is \MessageBreak
98                                     incompatible with the minitoc package. \MessageBreak
99 We try to patch ***\@gobble}%
100 \@mtcpatchmemoir@true}
101 \fi

```

\if@mtcpatchmemoir@ And now the patch:

```

102 \if@mtcpatchmemoir@
103 \InputIfFileExists{mtcpatchmem.sty}{}{%
104     \PackageError{minitoc}
105         {*** Unable to patch the memoir class ***}%
106         {So it remains incompatible. Sorry.}}
107 \fi

```

## 5.11 Testing the emptiness of a file

\mtc@ifmtarg Some macros for testing if an argument of a macro is empty (taken from the package  
\mtc@xifmtarg ifmtarg [49], by Peter R. WILSON and Donald ARSENEAU, and from while.tip, by  
\mtc@EndWhile Stephan P. von BECHTOLSHEIM [42]). The group is necessary to keep local the catcode  
\mtc@WhilePreCondition change of “Q”, hence a \gdef is needed for \mtc@ifmtarg.

```

\mtc@WhileCondition
\mtc@WhileBody
\mtc@While
\mtc@WhileNext
108 \begingroup
109 \catcode'\Q=3
110 \long\gdef\mtc@ifmtarg#1{%
111 \mtc@xifmtarg#1Q\@secondoftwo\@firstoftwo\@nil}
112 \long\gdef\mtc@xifmtarg#1#2Q#3#4#5\@nil{#4}
113 \endgroup
114 \let\mtc@EndWhile = \fi
115 \def\mtc@While #1#2#3\mtc@EndWhile{%
116     \def\mtc@WhilePreCondition{#1}%
117     \def\mtc@WhileCondition{#2}%
118     \def\mtc@WhileBody{#3}%
119     \mtc@@While
120 }
121 \def\mtc@@While{%
122     \mtc@WhilePreCondition
123     \mtc@WhileCondition
124     \def\mtc@WhileNext{%
125         \mtc@WhileBody
126         \mtc@@While
127     }%
128     \else
129     \def\mtc@WhileNext{}%
130     \fi
131     \mtc@WhileNext
132 }

```

```

\if@mtc@checkfiles Some macros to test if a file is empty or not: \mtc@CkFile{file} returns
\if@mtc@FE \@mtc@FEtrue if the file is empty, \@mtc@FEfalse if the file is not empty. An
\if@mtc@LI inexistent file is empty. A file full of white space (space, tabulation, newline) is empty.
\mtc@While Comments are empty.
\mtc@Body
\mtc@EndWhile Note: on a big empty file, the \mtc@While loop may be time consuming, but not an
\mtc@CkFile eternity (33 s for 106 lines on my computer), and the first non-empty line stops the loop.
\mtc@CkStr \jobname.mtc is used as scratch file. It is erased after use.
\mtc@Rline
\tf@mtc
\@inputcheck 133 \newif\if@mtc@LI\@mtc@LItrue
134 \newif\if@mtc@FE\@mtc@FEtrue
135 \newif\if@mtc@checkfiles\@mtc@checkfilestrue
136 \def\mtc@Body{\immediate\read\@inputcheck to
137 \mtc@Rline\relax
138 \ifeof\@inputcheck\relax\@mtc@LIfalse\fi
139 \expandafter\ifx\mtc@Rline\par\relax
140 \def\mtc@Rline{}
141 \else
142 \ifeof\@inputcheck\relax\global\@mtc@LIfalse\fi
143 \mtc@ifmtarg{\mtc@Rline}{\relax}%
144 {\@mtc@FEfalse\@mtc@LIfalse}
145 \fi
146 }
147 \def\mtc@CkFile#1{%
148 \@mtc@LItrue\@mtc@FEtrue
149 \if@mtc@checkfiles
150 \IfFileExists{#1}{%
151 \immediate\openin\@inputcheck #1\relax
152 \mtc@While{}{\if@mtc@LI\relax}%
153 {\mtc@Body}%
154 \mtc@EndWhile}%
155 {\@mtc@FEtrue}%
156 \else
157 \@mtc@FEfalse%
158 \fi}
159 \closein\@inputcheck\relax
160 \def\mtc@CkStr#1{%
161 \immediate\openout\tf@mtc \jobname.mtc
162 \immediate\write\tf@mtc{#1}%
163 \immediate\closeout\tf@mtc
164 \mtc@CkFile{\jobname.mtc}%
165 \immediate\openout\tf@mtc \jobname.mtc
166 \immediate\closeout\tf@mtc}

```

## 5.12 Internal macros to decrement mini toc counters

```

\mtc@onebackpart It is sometimes necessary to decrement a minitoc counter (ptc, mtc or stc) by one.
\mtc@onebackchapter These macros are:
\mtc@onebacksection
\addtocounter

```

```

167 \def\mtc@onebackpart{\addtocounter{ptc}{-1}}
168 \def\mtc@onebackchap{\addtocounter{mtc}{-1}}
169 \def\mtc@onebacksect{\addtocounter{stc}{-1}}

```

### 5.13 Patching the \part command

`\part` If the `\part` command is not defined (by the document class, usually), we cannot patch it  
`\mtc@svspart` and a warning is displayed<sup>3</sup>. Else, we patch its two branches, `\@part` (for the unstarred  
`\mtc@svpart` version) or `\@spart` (for the starred version, `\part*`): we add `\stepcounter{ptc}` to  
`\@spart` increment the parttoc counter ptc. See also section 5.51 on page 131.  
`\@part`  
`\stepcounter` The code of the next section (section 5.14) is also skipped if `\part` is not defined.

```

170 \@ifundefined{part}{%
171   \PackageWarningNoLine{minitoc}%
172   {*** part level macros NOT available ***}
173 }{%% else undefined part (\part defined)
174   \PackageInfo{minitoc}%
175   {*** part level macros available ***\@gobble}
176   \let\mtc@svspart\@spart
177   \def\@spart{\stepcounter{ptc}\mtc@svspart}
178   \let\mtc@svpart\@part
179   \def\@part{\stepcounter{ptc}\mtc@svpart}

```

### 5.14 Adding an entry in the TOC for a starred part

`\mtcaddpart` To add an entry in the TOC for a starred part, we need the `\mtcaddpart` macro, which  
`\mtc@ifmtarg` has an optional argument, the title of the part as it should appear in the TOC.  
`\contentsline`  
`\addcontentsline` By default, this argument is empty. If it is empty (tested via `\mtc@ifmtarg`) or omitted,  
`\adjustptc` we add a `\contentsline{xpart}{...}` line in the .toc file. If it is not empty, we  
`\l@xpart` add a `\contentsline{part}{title...}` line in the .toc file. We always add a  
`\l@part` `\contentsline{xpart}{...}` line in the .lof and .lot files. Then we increment the  
 ptc counter, via `\adjustptc` (defined in section 5.45 on page 118). Using `xpart` as first  
 argument of `\contentsline` means that `\l@xpart` will be invoked in place of `\l@part`  
 to print the entry in the TOC, but `\l@xpart` uses a huge depth (10000) for this entry,  
 hence it will never be really printed (except if you cheat).

```

180 \newcommand{\mtcaddpart}[1][{}]{%
181   \mtc@ifmtarg{#1}{\addcontentsline{toc}{xpart}{}}{%
182     \addcontentsline{toc}{part}{#1}}%

```

<sup>3</sup>Document classes with sectioning commands but no `\part` command are likely non standard, hence the warning displayed on the terminal.

```

183 \addcontentsline{lof}{xpart}{}%
184 \addcontentsline{lot}{xpart}{}%
185 \adjustptc}

```

This code terminates (temporarily) the part level commands.

```

186 }%

```

## 5.15 Section level macros

`\chapter` The section level macros are defined if `\chapter` is not defined and `\section` defined, i.e. in document classes like `article`, but not in document classes like `book` or `report`.  
`\section` So we test if `\chapter` is defined and if `\section` is defined, with adequate warnings. If neither are defined, you are in big trouble to use the `minitoc` package with the class of your document.

```

187 \@ifundefined{chapter}{\PackageInfo{minitoc}%
188     {*** chapter level macros NOT available ***\@gobble}%
189 \@ifundefined{section}{\PackageInfo{minitoc}%
190     {*** section level macros NOT available ***\@gobble}%
191 \PackageWarningNoLine{minitoc}%
192     {*** no section or chapter level macros available ***
193     \MessageBreak
194     *** PLEASE VERIFY YOUR MAIN DOCUMENT CLASS ***}}%
195 {\PackageInfo{minitoc}%
196     {*** section level macros available ***\@gobble}%

```

## 5.16 Corrections for numbering

`\mtc@onebacksect` As the TOC, the LOF and the LOT are considered as (starred) sections, we must  
`\tableofcontents` decrement the secttoc counter (stc) via `\mtc@onebacksect` when the corresponding  
`\listoffigures` commands are executed. Hence we patch these commands.  
`\listoftables`

```

\mtcsv@tableofcontents 197 \let\mtcsv@tableofcontents\tableofcontents
\mtcsv@listoffigures 198 \let\mtcsv@listoffigures\listoffigures
\mtcsv@listoftables 199 \let\mtcsv@listoftables\listoftables
200 \def\tableofcontents{\mtcsv@tableofcontents\mtc@onebacksect}
201 \def\listoffigures{\mtcsv@listoffigures\mtc@onebacksect}
202 \def\listoftables{\mtcsv@listoftables\mtc@onebacksect}

```



## 5.17 Patching the `\section` command

```

\mtc@svsection  If the \section command is not defined (by the document class, usually), we can-
\mtc@svss       not patch it and a warning is displayed. Else, we patch its two branches, \@sect
\@ssect        (for the unstarred version) or \@ssect (for the starred version, \section*): we add
\@sect         \stepcounter{stc} to increment the secttoc counter stc, only in the unstarred case
\section       (the version #25 has removed a spurious decrementation of this counter).
\stepcounter

203     \let\mtc@svsection\section
204     \def\section{\stepcounter{stc}\mtc@svsection}
205     \let\mtc@svss\@ssect

```

## 5.18 Adding an entry in the TOC for a starred section

```

\mtcaddsection  To add an entry in the TOC for a starred section, we need the \mtcaddsection macro,
\mtc@ifmtarg    which has an optional argument, the title of the section as it should appear in the
\contentsline   TOC. By default, this argument is empty. If it is empty (tested via \mtc@ifmtarg)
\adjuststc     or omitted, we add a \contentsline{xsection}{}... line in the .toc file. If
\l@xsection    it is not empty, we add a \contentsline{section}{title...}... line in the
\l@section     .toc file. We always add a \contentsline{xsection}{}... line in the .lof
               and .lot files. Then we increment the stc counter, via \adjuststc (defined in
               section 5.54 on page 142). Using xsection as first argument of \contentsline
               means that \l@xsection will be invoked in place of \l@section to print the entry
               in the TOC, but \l@xsection uses a huge depth (10000) for this entry, hence it will
               never be really printed (except if you cheat).

```

```

206 \newcommand{\mtcaddsection}[1][{}]{%
207     \mtc@ifmtarg{#1}{\addcontentsline{toc}{xsection}{}}{%
208         {\addcontentsline{toc}{section}{#1}}%
209     \addcontentsline{lof}{xsection}{}%
210     \addcontentsline{lot}{xsection}{}%
211     \adjuststc}

```

This code terminates (temporarily) the section level commands, and we continue with chapter level macros.

```

212 }}%
213 {%

```

## 5.19 Chapter level macros

`\chapter` The chapter level macros are defined if `\chapter` is defined, *i.e.* in document classes like `book` or `report`. So we test if `\chapter` is defined, with adequate warnings. The test is already done above, we are in the “else” branch of `\@ifundefined{chapter}`.

```
214 \PackageInfo{minitoc}{*** chapter level macros available ***\@gobble}
```

## 5.20 Patching the `\chapter` command

`\chapter` The `\chapter` command is defined (by the document class, usually). We patch its two  
`\@chapter` branches, `\@chapter` (for the unstarred version) or `\@schapter` (for the starred version,  
`\mtc@svchapter` `\chapter*`): we add call to `\stepcounter{mtc}` to increment the minitoc counter `mtc`.  
`\stepcounter` Only the unstarred branch (`\@chapter`) is patched here. The other branch is patched later  
(section 5.36 on page 103).

```
215 \let\mtc@svchapter\@chapter
216 \def\@chapter{\stepcounter{mtc}\mtc@svchapter}
```

## 5.21 Adding an entry in the TOC for a starred chapter

`\mtcaddchapter` To add an entry in the TOC for a starred chapter, we need the `\mtcaddchapter`  
`\mtc@ifmtarg` macro, which has an optional argument, the title of the chapter as it should appear in  
`\contentsline` the TOC. By default, this argument is empty. If it is empty (tested via `\mtc@ifmtarg`)  
`\adjustmtc` or omitted, we add a `\contentsline{xchapter}{...}` line in the `.toc` file. If  
`\l@xchapter` it is not empty, we add a `\contentsline{chapter}{title...}` line in the  
`\l@chapter` `.toc` file. We always add a `\contentsline{xchapter}{...}` line in the `.lof`  
and `.lot` files. Then we increment the `mtc` counter, via `\adjustmtc` (defined in  
section 5.31 on page 91). Using `xchapter` as first argument of `\contentsline`  
means that `\l@xchapter` will be invoked in place of `\l@chapter` to print the entry  
in the TOC, but `\l@xchapter` uses a huge depth (10000) for this entry, hence it will  
never be really printed (except if you cheat).

```
217 \newcommand{\mtcaddchapter}[1][{}]{%
218   \mtc@ifmtarg{#1}{\addcontentsline{toc}{xchapter}{}}%
219   {\addcontentsline{toc}{chapter}{#1}}%
220   \addcontentsline{lof}{xchapter}{}%
221   \addcontentsline{lot}{xchapter}{}%
222   \adjustmtc}
```

This code terminates (temporarily) the chapter level commands, *i.e.* terminates the `\ifundefined{chapter}` at the beginning of section 5.15 on page 80.

```
223 }%
```

## 5.22 Miscellaneous declarations

```
\newread The \newread command must be redeclared as being \outer (as Donald ARSENEAU
\mtc@toks told me). We need a token register (\mtc@toks), a temporary string (\mtc@string),
\mtc@string struts (two kinds, each one using a box containing an invisible vertical rule), a rule with
\mtc@strut all dimensions equal to zero (\mtc@zrule) and a command discouraging page breaks
\mtc@strutbox (\mtc@BBR, for “bad break”). For the struts, which are boxes containing an invisible
\mtc@hstrut vertical rule, we use “ex” units, to follow the current font.
\mtc@hstrutbox
\mtc@v 224 \def\newread{\alloc@6\read\chardef\sixt@@n}
\mtc@zrule 225 \newtoks\mtc@toks
\mtc@BBR 226 \def\mtc@string{\relax}
227 \newbox\mtc@strutbox
228 \setbox\mtc@strutbox=\hbox{\rule[1.8ex]{\z@}{2.5ex}}
229 \def\mtc@strut{\relax\ifmmode\copy\mtc@strutbox
230 \else\unhcopy\mtc@strutbox\fi}
231 \newbox\mtc@hstrutbox
232 \setbox\mtc@hstrutbox=\hbox{\rule[1.ex]{\z@}{1.ex}}
233 \def\mtc@hstrut{\relax\ifmmode\copy\mtc@hstrutbox
234 \else\unhcopy\mtc@hstrutbox\fi}
235 \def\mtc@v{\leavevmode\mtc@strut}
236 \def\mtc@zrule{\rule[\z@]{\z@}{\z@}}
237 \def\mtc@BBR{\unpenalty\nopagebreak[4]}
```

## 5.23 Autoconfiguration of extensions

`\tf@mtc` This code is a hack to determine if the operating system is able or unable to use long extensions ( > 3 characters) in file names. We define a file descriptor (`\tf@mtc`) to write files<sup>4</sup>. This code is verbose if long extensions cannot be used, else the messages are only

<sup>4</sup> Note that it is the *only* new file descriptor created by the `minitoc` package. All files written by `minitoc` use this descriptor, or one of the standard descriptors, e.g. for the `.log` file. In fact, `minitoc` writes also in the `.toc`, `.lof` and `.lot` files, but via file descriptors already used by standard commands like `\tableofcontents`, `\listoffigures` and `\listoftables`. We can conclude that `minitoc` itself uses only one file descriptor (or write stream). Some other attempts to make per chapter TOCs have failed by quickly leading to exhaustion of file descriptors (TeX offers only 16 file descriptors for writing), because they called the standard internal `\starttoc` macro, which invokes `\newwrite`, for each mini-table. As `minitoc` writes into only one file at a time (and in the `.log` file, and in the standard contents files, of course), we can reuse the same file descriptor and avoid this serious problem. The `minitoc` package writes in the contents files when it encounters a major sectioning command (`\part`, `\chapter`, or `\section`), if necessary. It writes into the mini-table auxiliary files only via the mini-table preparing commands (`\doparttoc`, ..., `\dosectlot`), once at a time. You do not need a new hammer for each nail.

Phase (time from left to right):		1	2	3
OS with long extensions	\jobname.mtc1	TRUE	TRUE	*
	\jobname.mtc		FALSE	
OS with short extensions	\jobname.mtc(1)	TRUE	FALSE	*

Figure 5.1: Hack to detect the limitation to short extensions.

written in the .log file. The sequencing of these operations is vital. The figure 5.1 shows this sequence. A star (\*) denotes which file is read in phase 3.

```

\if@longextensions@ (0) First, a message and a new flag:
  \tf@mtc
  \openout 238 \PackageInfo{minitoc}%
  \write 239 {*** Autoconfiguration of extensions ***\@gobble}
\closeout 240 \newif\if@longextensions@\@longextensions@false
  \input (1) We write “\@longextensions@true” in \jobname.mtc1. But if the OS has short
  \jobname extensions, the real name of the file will be truncated to \jobname.mtc.

241 \immediate\openout\tf@mtc \jobname.mtc1
242 \immediate\write\tf@mtc{\string\@longextensions@true}
243 \immediate\closeout\tf@mtc

(2) We write “\@longextensions@false” in \jobname.mtc.

244 \immediate\openout\tf@mtc \jobname.mtc
245 \immediate\write\tf@mtc{\string\@longextensions@false}
246 \immediate\closeout\tf@mtc

(3) We read \jobname.mtc1. But if the OS has short extensions, the real name of the
file will be truncated to \jobname.mtc.

247 \input{\jobname.mtc1}

(4) Hence, the flag is true if we read really from \jobname.mtc1, but false if we read
from \jobname.mtc. The text and the severity of the messages are different.

248 \if@longextensions@
249 \PackageInfo{minitoc}%
250 {*** Long extensions (Unix-like) will be used ***\@gobble}
251 \PackageInfo{minitoc}%
252 {==> this version is configured for UNIX-like \MessageBreak
253 \space\space\space\space(long extensions) file names\@gobble}%
254 \else
255 \PackageWarningNoLine{minitoc}%
256 {*** Short extensions (MSDOS-like) will be used ***sight***}
257 \PackageWarningNoLine{minitoc}%
258 {==> this version is configured for MSDOS-like
259 \MessageBreak \space\space\space\space(8+3) file names}
260 \fi

```

- (5) We erase the contents of the two files (because `\jobname.mtc` is also used later as a scratch file, see section 5.11 on page 77).

```

261 \immediate\openout\tf@mtc \jobname.mtc
262 \immediate\closeout\tf@mtc
263 \immediate\openout\tf@mtc \jobname.mtc1
264 \immediate\closeout\tf@mtc

```

## 5.24 Detecting obsolete versions of L<sup>A</sup>T<sub>E</sub>X

`\@inputcheck` This code detects old versions of the L<sup>A</sup>T<sub>E</sub>X kernel that are no more supported and with which the minitoc package can hardly work. The trick is to detect the absence of some internal L<sup>A</sup>T<sub>E</sub>X commands, `\@inputcheck` and `\reset@font`. If you get one of these messages, you are in bad luck and should *urgently* update your L<sup>A</sup>T<sub>E</sub>X installation, which is rusting since... many years!



```

265 \@ifundefined{@inputcheck}%
266   {\PackageWarningNoLine{minitoc}%
267     {Your version of latex.tex is obsolete.\MessageBreak
268     Trying to continue..}\newread\@inputcheck\relax}{}}
269 \@ifundefined{reset@font}%
270   {\PackageWarningNoLine{minitoc}%
271     {Your version of latex.tex is very obsolete.\MessageBreak
272     Trying to continue... crossing fingers}%
273   \let\reset@font\relax}{}

```

## 5.25 A macro to make a TOC entry without leaders nor page numbers

`\@undottedtocline` The (internal) macro `\@undottedtocline` is a modified version of the standard command `\@dottedtocline`. It will be used in customization macros.

```

274 \def\@undottedtocline#1#2#3#4#5{%
275   \ifnum #1>\c@tocdepth \else
276     \vskip \z@ plus.2\p@
277     {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
278     \parindent #2\relax\@afterindenttrue
279     \interlinepenalty\@M
280     \leavevmode
281     \@tempdima #3\relax \advance\leftskip \@tempdima \hbox{}}%
282     \hskip -\leftskip
283     #4\nobreak\hfill \nobreak
284     \null\par}%
285   \fi}

```

## 5.26 Default values for the page-number customization macros

`\if@mtc@memoirLoaded@` This section defines some customization macros for the presence or absence of page numbers in the mini-tables. But if the memoir class [48] is loaded, it does the job. So, we test first `\if@mtc@memoirLoaded@` to use the commands of memoir when they are available.

286 `\if@mtc@memoirLoaded@`

`\mtcpagenumbers` For entries in minitocs:  
`\nomtcpagenumbers`

```
287 \def\mtcpagenumbers{%
288   \cftpagenumber{section}
289   \cftpagenumber{subsection}
290   \cftpagenumber{subsubsection}
291   \cftpagenumber{paragraph}
292   \cftpagenumber{subparagraph}}
293 \def\nomtcpagenumbers{%
294   \cftpagenumberoff{section}
295   \cftpagenumberoff{subsection}
296   \cftpagenumberoff{subsubsection}
297   \cftpagenumberoff{paragraph}
298   \cftpagenumberoff{subparagraph}}
```

`\stcpagenumbers` For entries in secttocs:  
`\nostcpagenumbers`

```
299 \def\stcpagenumbers{%
300   \cftpagenumber{subsection}
301   \cftpagenumber{subsubsection}
302   \cftpagenumber{paragraph}
303   \cftpagenumber{subparagraph}}
304 \def\nostcpagenumbers{%
305   \cftpagenumberoff{subsection}
306   \cftpagenumberoff{subsubsection}
307   \cftpagenumberoff{paragraph}
308   \cftpagenumberoff{subparagraph}}
```

`\ptcpagenumbers` For entries in parttocs:  
`\noptcpagenumbers`

```
309 \def\ptcpagenumbers{%
310   \cftpagenumber{chapter}
311   \cftpagenumber{section}
312   \cftpagenumber{subsection}
313   \cftpagenumber{subsubsection}
314   \cftpagenumber{paragraph}
315   \cftpagenumber{subparagraph}}
316 \def\noptcpagenumbers{%
317   \cftpagenumberoff{chapter}}
```

```

318 \cftpagenumbersoff{section}
319 \cftpagenumbersoff{subsection}
320 \cftpagenumbersoff{subsubsection}
321 \cftpagenumbersoff{paragraph}
322 \cftpagenumbersoff{subparagraph}}

```

```

\mlfpagenumbers For entries in minilofs, sectlofs, and partlofs:
\nomlfpagenumbers
\slfpagenumbers 323 \def\mlfpagenumbers{\cftpagenumberon{figure}}
\noslfpagenumbers 324 \def\nomlfpagenumbers{\cftpagenumbersoff{figure}}
\plfpagenumbers 325 \def\slfpagenumbers{\cftpagenumberon{figure}}
\noplfpagenumbers 326 \def\noslfpagenumbers{\cftpagenumbersoff{figure}}
327 \def\plfpagenumbers{\cftpagenumberon{figure}}
328 \def\noplfpagenumbers{\cftpagenumbersoff{figure}}

```

```

\mltpagenumbers For entries in minilots, sectlots, and partlots:
\nomltpagenumbers
\sltpagenumbers 329 \def\mltpagenumbers{\cftpagenumberon{table}}
\nosltpagenumbers 330 \def\nomltpagenumbers{\cftpagenumbersoff{table}}
\pltpagenumbers 331 \def\sltpagenumbers{\cftpagenumberon{table}}
\nopltpagenumbers 332 \def\nosltpagenumbers{\cftpagenumbersoff{table}}
333 \def\pltpagenumbers{\cftpagenumberon{table}}
334 \def\nopltpagenumbers{\cftpagenumbersoff{table}}

```

Else, minitoc will use its own commands.

```

335 \else

```

```

\mtcpagenumbers First, for minitocs, secttocs and parttocs:
\nomtcpagenumbers
\mlfpagenumbers 336 \def\mtcpagenumbers{\let\mtc@pgno\null}
\nomlfpagenumbers 337 \def\nomtcpagenumbers{\let\mtc@pgno\relax}
\mltpagenumbers 338 \def\stcpagenumbers{\let\stc@pgno\null}
\nomltpagenumbers 339 \def\nostcpagenumbers{\let\stc@pgno\relax}
340 \def\ptcpagenumbers{\let\ptc@pgno\null}
341 \def\noptcpagenumbers{\let\ptc@pgno\relax}

```

```

\mlfpagenumbers Then, for minilofs, sectlofs and partlofs:
\nomlfpagenumbers
\mlfpagenumbers 342 \def\mlfpagenumbers{\let\mlf@pgno\null}
\nomlfpagenumbers 343 \def\nomlfpagenumbers{\let\mlf@pgno\relax}
\mltpagenumbers 344 \def\slfpagenumbers{\let\slf@pgno\null}
\nomltpagenumbers 345 \def\noslfpagenumbers{\let\slf@pgno\relax}
346 \def\plfpagenumbers{\let\plf@pgno\null}
347 \def\noplfpagenumbers{\let\plf@pgno\relax}

```

```

\mltpagenumbers Then, for minilots, sectlots and partlots:
\nomltpagenumbers
\mltpagenumbers 348 \def\mltpagenumbers{\let\mlt@pgno\null}
\nomltpagenumbers 349 \def\nomltpagenumbers{\let\mlt@pgno\relax}
\mltpagenumbers 350 \def\sltpagenumbers{\let\slt@pgno\null}
\nomltpagenumbers 351 \def\nosltpagenumbers{\let\slt@pgno\relax}
352 \def\pltpagenumbers{\let\plt@pgno\null}
353 \def\nopltpagenumbers{\let\plt@pgno\relax}
354 \fi

\ptcpagenumbers Then the default values are set; page numbers are present:
\plfpagenumbers
\pltpagenumbers 355 \ptcpagenumbers
\mtcpagenumbers 356 \plfpagenumbers
\mlfpagenumbers 357 \pltpagenumbers
\mltpagenumbers 358 \mtcpagenumbers
\stcpagenumbers 359 \mlfpagenumbers
\slfpagenumbers 360 \mltpagenumbers
\sltpagenumbers 361 \stcpagenumbers
362 \slfpagenumbers
363 \sltpagenumbers

```

## 5.27 “Features” for the mini-tables

Each kind of mini-table has three “features”: a “before” feature, an “after” feature and a “thispagestyle” feature.

A “before” feature is defined by a macro like `\beforeparttoc` which contains code to be executed before any mini-table of a given type: `\beforeparttoc` is executed before each parttoc. Usually such features contains only trivial commands like `\clear[double]page`, or `\empty`.

An “after” feature is analog but its code is executed after each mini-table of a given type.

A “thispagestyle” feature is defined by a macro like `\thispagestyleparttoc` which contains code to define the page style implied by mini-tables of a given type: the command `\thispagestyleparttoc` can be defined as `\thispagestyle{...}`. Usually, the “thispagestyle” feature is only defined for part-level mini-tables, which use page breaks in their before and after features. For chapter- and section-level mini-tables, the “thispagestyle” feature is usually defined as `\empty`.

We set the default values for the part-level features depending on the presence of the `\chapter` command, as article-like documents are different from the book- or report-like documents for the layout of part-level mini-tables.



`\chapter` If `\chapter` is not defined, the part level mini-tables have no “before” feature (by default):

```
\beforeparttoc
\beforepartlof
\beforepartlot
364 \@ifundefined{chapter}{%
365   \let\beforeparttoc\empty
366   \let\beforepartlof\empty
367   \let\beforepartlot\empty}%
```

`\cleardoublepage` But if `\chapter` is defined, they have a `\cleardoublepage` as default “before” feature:

```
368   {\let\beforeparttoc\cleardoublepage
369    \let\beforepartlof\cleardoublepage
370    \let\beforepartlot\cleardoublepage}
```

`\beforeinitoc` Chapter level mini-tables have no “before” feature (by default):

```
\beforeminilof
\beforeminilot
371 \let\beforeinitoc\empty
372 \let\beforeminilof\empty
373 \let\beforeminilot\empty
```

`\beforesecttoc` Section level mini-tables have no “before” feature (by default):

```
\beforesectlof
\beforesectlot
374 \let\beforesecttoc\empty
375 \let\beforesectlof\empty
376 \let\beforesectlot\empty
```

`\chapter` If `\chapter` is not defined, the part level mini-tables have no “after” feature (by default):

```
\afterparttoc
\afterpartlof
\afterpartlot
377 \@ifundefined{chapter}{%
378   \let\afterparttoc\empty
379   \let\afterpartlof\empty
380   \let\afterpartlot\empty}%
```

`\cleardoublepage` But if `\chapter` is defined, they have a `\cleardoublepage` as default “after” feature:

```
381   {\let\afterparttoc\cleardoublepage
382    \let\afterpartlof\cleardoublepage
383    \let\afterpartlot\cleardoublepage}
```

`\afterinitoc` Chapter level mini-tables have no “after” feature (by default):

```
\afterminilof
\afterminilot
384 \let\afterinitoc\empty
385 \let\afterminilof\empty
386 \let\afterminilot\empty
```

`\aftersecttoc` Section level mini-tables have no “after” feature (by default):

`\aftersectlof`

`\aftersectlot` 387 `\let\aftersecttoc\empty`

388 `\let\aftersectlof\empty`

389 `\let\aftersectlot\empty`

`\thispagestyle` By default, all the “thispagestyle” features (at part level) use the empty page style. It affects only the first page of the mini-table.

`\thispageparttocstyle`

`\thispagepartlofstyle`

`\thispagepartlotstyle`

390 `\def\thispageparttocstyle{\thispagestyle{empty}}`

`\thispageminitocstyle`

391 `\def\thispagepartlofstyle{\thispagestyle{empty}}`

`\thispageminilofstyle`

392 `\def\thispagepartlotstyle{\thispagestyle{empty}}`

`\thispageminilotstyle`

393 `\def\thispageminitocstyle{\empty}`

`\thispagesecttocstyle`

394 `\def\thispageminilofstyle{\empty}`

`\thispagesectlofstyle`

395 `\def\thispageminilotstyle{\empty}`

`\thispagesectlotstyle`

396 `\def\thispagesecttocstyle{\empty}`

397 `\def\thispagesectlofstyle{\empty}`

398 `\def\thispagesectlotstyle{\empty}`

`\mtcsetfeature` In section 5.65.8 on page 177, we will define the `\mtcsetfeature` macro which is much easier user interface to set the mini-tables “features”.

## 5.28 Fake tables of contents

`\faketableofcontents` If you don’t want a table of contents, but want minitocs, you need to create the .toc file, without inserting it into your document. This `\faketableofcontents` command is a stripped off version of the standard command `\tableofcontents`. We define in the same way the commands `\fakelistoffigures` and `\fakelistoftables`, using in fact a stripped version `\fake@starttoc` of `\@starttoc`. But it is nice to reset to zero the ptc, mtc, and stc counters now, if they are defined<sup>5</sup>.

```
399 \def\faketableofcontents{\fake@starttoc{toc}%
400   \ifundefined{c@ptc}{\setcounter{ptc}{0}}%
401   \ifundefined{c@mtc}{\setcounter{mtc}{0}}%
402   \ifundefined{c@stc}{\setcounter{stc}{0}}%
403   }
404 \def\fakelistoffigures{\fake@starttoc{lof}}
405 \def\fakelistoftables{\fake@starttoc{lot}}
406 \def\fake@starttoc#1{\begingroup
407   \makeatletter
408   \if@filesw \expandafter\newwrite\csname tf@#1\endcsname
409   \immediate\openout \csname tf@#1\endcsname
410   \jobname.#1\relax
411   \fi
412   \global\@nobeckfalse \endgroup}
```

<sup>5</sup>Remember the infamous “stc0” bug.

This code uses the same file descriptors (for writing) than the original commands.

## 5.29 Depth counters for minilofs and minilots

`\AtBeginDocument` If the counters `lofdepth` and `lotdepth` are defined, we create the necessary new counters: `minilofdepth` and `minilotdepth`. These counters are initialized to 2. This is done after the loading of the packages, in an `\AtBeginDocument` block:

```
413 \AtBeginDocument{%
414   \ifundefined{c@lofdepth}{}%
415     {\newcounter{minilofdepth}\setcounter{minilofdepth}{2}}%
416   \ifundefined{c@lotdepth}{}%
417     {\newcounter{minilotdepth}\setcounter{minilotdepth}{2}}%
418 }
```

## 5.30 Chapter level commands

From here, we define the chapter-level commands.

`\mtc@markboth` First, we memorize the marks (not used today, but...):  
`\@mkboth`

```
419 \global\let\mtc@markboth\markboth
420 \global\let\@mkboth\markboth
```

## 5.31 Starred parts, chapters or sections

`\addst@rred` We define commands to manage the starred sectionning commands: `\part*`, `\chapter*`  
`\addcontentsline` and `\section*`. The section-level is different depending if `\chapter` is defined or not.  
`\stepcounter` Eventually, a counter is incremented. A contents line is added in the `.toc` file, with the  
`\c@ptc` right depth to print it (see `\l@star...` later, in section 5.63 on page 162).

```
\c@mtc
\c@stc
421 \def\addst@rred#1#2{%
422   \addcontentsline{toc}{star#1}{#2}%
423   \ifundefined{c@ptc}{}%
424     \expandafter\ifx\csname #1\endcsname\part\relax
425       \stepcounter{ptc}%
426     \fi
427   }%
428   \ifundefined{c@mtc}{}%
429     \expandafter\ifx\csname #1\endcsname\chapter\relax
```

```

430     \stepcounter{mtc}%
431     \fi
432 }%
433 \@ifundefined{c@stc}{}{%
434     \expandafter\ifx\csname #1\endcsname\section\relax
435 %%%     \@ifundefined{chapter}{\stepcounter{stc}}{}%
436     \stepcounter{stc}%
437     \fi
438 }%
439 }%

```

\addstarredsection If \chapter is not defined, we just define \addstarredsection:

```

\chapter
\addst@rred 440 \@ifundefined{chapter}{}%
441 \gdef\addstarredsection#1{\addst@rred{section}{#1}}
442 }%

```

else we begin to define the stuff for chapter-level commands (the “else” branch of \@ifundefined{chapter}):

```

443 {%

```

```

\The@mtc We define now: the internal format of the mtc counter (\The@mtc), the obsolete com-
\firstchapteris mand \firstchapteris (it just emits a harmless warning), the mtc counter (initialized
\if@firstchapteris@used@ to 0), the \adjustmtc command (increments mtc, by 1 by default), the \decrementmtc
\adjustmtc command (decrements mtc by 1), the \incrementmtc command (increments mtc by 1),
\decrementmtc the format of the mtc counter (\themtc), the counter minitocdepth, initialized to 2,
\incrementmtc for the depth of a minitoc (analog to the standard tocdepth counter), the horizontal
\themtc rules to draw before and after minitocs (\mtc@rule), and we copy that definition into
\mtc@rule analog macros for other kinds of mini-tables. We also set the default value (24pt) of
\mlf@rule \mtcindent, the indentation for minitocs (both sides). The rules are 0.4pt thick. They
\mlt@rule are defined via \hrule to stay in vertical mode for the final \kern.
\plf@rule
\plt@rule
\slf@rule 444 \def\The@mtc{\arabic{mtc}}
445 \def\firstchapteris#1%
\slt@rule {\PackageWarning{minitoc}%
\mtcindent 447 {*** \string\firstchapteris \space is an obsolete command ***}}%
448 \@firstchapteris@used@true}
449 \newcounter{mtc}
450 \setcounter{mtc}{0}
451 \newcommand{\adjustmtc}[1][1]{\addtocounter{mtc}{#1}}
452 \def\decrementmtc{\addtocounter{mtc}{-1}}
453 \def\incrementmtc{\addtocounter{mtc}{+1}}
454 \gdef\themtc{\arabic{mtc}}
455 \newcounter{minitocdepth}
456 \setcounter{minitocdepth}{2}
457 \def\mtc@rule{\kern-3\p@ \hrule \@width\columnwidth \kern2.6\p@}
458 \let\mlf@rule\mtc@rule
459 \let\mlt@rule\mtc@rule

```

```

460 \let\plf@rule\mtc@rule
461 \let\plt@rule\mtc@rule
462 \let\slf@rule\mtc@rule
463 \let\slt@rule\mtc@rule
464 \mtcindent=24\p@

```

## 5.32 Font commands for the mini-tables

```

\mtcfont We define these commands with full NFSS [28] descriptions. These definitions are ef-
\mtcSfont fective if \chapter is defined. The fonts for titles are also defined here. See also the
\mtcSSfont \mtcsetfont macro (section 5.65.2 on page 165) and the \mtcsetttitlefont macro
\mtcSSSfont later (section 5.65.3 on page 167).
\mtcPfont
\mtcSPfont 465 \def\mtcfont{\small\rmfamily\upshape\mdseries}
\mlffont 466 \def\mtcSfont{\small\rmfamily\upshape\bfseries}
\mltfont 467 \let\mtcSSfont\mtcfont
\mtifont 468 \let\mtcSSSfont\mtcfont
469 \let\mtcPfont\mtcfont
470 \let\mtcSPfont\mtcfont
471 \let\mlffont\mtcfont
472 \let\mltfont\mtcfont
473 \def\mtifont{\large\rmfamily\upshape\bfseries}

\coffeeont And \coffeeont is used for “coffee breaks” in the minutes package [30].

474 \def\coffeeont{\small\rmfamily\slshape\mdseries}

```

## 5.33 Internal commands to position the mini-table titles

```

\df@mtitc The commands \miniXXX and \dominiXXX accept an optional argument to left justify,
\df@mtilf center, right justify or omit the title of the chapter-level mini-tables. By default, these
\df@mtilt titles are left justified. The choice made in a \dominiXXX command is global and mem-
\do@mtitc orized in \df@mtitc, \df@mtilf or \df@mtilt; the choice made in a \miniXXX com-
\do@mtilf mand is local and stored in \do@mtitc, \do@mtilf or \do@mtilt. See the \minitoc@
\do@mtilt macro later (section 5.35.1 on the next page). An empty title needs a vertical correction
(Frank MITTELBACH).

\c@mti Centering, flushleft, flushright or empty titles:
\l@mti
\r@mti 475 \def\c@mti#1{\null\hfill #1\hfill\null}
\e@mti 476 \def\l@mti#1{\null #1\hfill\null}
\n@mti 477 \def\r@mti#1{\null\hfill #1\hfill\null}

```

```

478 \def\l@mti#1{\vspace{-\baselineskip}}
479 \def\n@mti#1{\vspace{-\baselineskip}}

```

```

\l@mti Default: titles on left:
\do@mtitc
\df@mtitc 480 \let\do@mtitc\l@mti
\do@mtilf 481 \let\df@mtitc\l@mti
\df@mtilf 482 \let\do@mtilf\l@mti
\do@mtilt 483 \let\df@mtilf\l@mti
\df@mtilt 484 \let\do@mtilt\l@mti
485 \let\df@mtilt\l@mti

```

### 5.34 The mtc@verse environment

```

mtc@verse Each minitoc is placed inside a mtc@verse environment. This environment is analog to
\iftightmtc the standard verse environment and hence defined via two commands: \mtc@verse and
\ifktightmtc \endmtc@verse. As it is a list environment, we first define (in a local way) \, then call
\list{} and set some dimensions like \itemsep, \itemindent, \listparindent,
\list \topsep. \parsep is set to zero if the tight option is active (to reduce the spacing of
\itemsep the lines). Both margins are set to \mtcindent. \endmtc@verse terminates the list and
\itemindent discourages a page break.
\listparindent
\topsep 486 \def\mtc@verse{\let\=\@centercr
\parsep 487 \list{}\itemsep\z@
\mtcindent 488 \itemindent \z@
489 \listparindent \itemindent
490 \partopsep\z@
491 \iftightmtc \parsep\z@ \fi
492 \ifktightmtc \parskip\z@ \fi
493 \topsep=lex
494 \leftmargin\mtcindent
495 \rightmargin\leftmargin\item[]
496 \def\endmtc@verse{\nolpagebreak[4]\endlist}

```

### 5.35 The \minitoc, \minilof, and \minilot commands

These three commands are very similar, with only cosmetic differences.

#### 5.35.1 The \minitoc command

```

\minitoc The \minitoc command must be used after \chapter if you need a minitoc (no
\chapter automatic minitoc).

```

`\dominitoc` This command accepts an optional argument, whose default value has eventually been set earlier by a `\dominitoc` command. The letter “d” represents this default value. `\dominitoc` has itself an optional argument which sets the default value of the optional argument of `\minitoc`. The default value of the optional argument of the `\dominitoc` command is “1”. It seems tortuous, but it is simple to use: we have a default behaviour (1) which can be altered globally via the optional argument of `\dominitoc`, or locally via the optional argument of `\minitoc`.

`\minitoc` So we define `\minitoc` with an optional argument and its (current) default value, and `\minitoc@` call the true code in the `\minitoc@` macro (which has one delimited argument); we use `\@ifnextchar` the `\@ifnextchar` trick to detect a left bracket for the optional argument:

```
497 \def\minitoc{\@ifnextchar[{ \minitoc@}]{\minitoc@[d]}}
```

The real code of `\minitoc` is in `\minitoc@`, which has a mandatory argument (delimited by brackets) specifying the position of the title.

`\if@minitoc@used@` First, we set the global flag `\@minitoc@used@true` to note that `\minitoc` has been called (this will be used by a hint later, section 5.76.2.2 on page 198).

```
498 \def\minitoc@[#1]{%
499 \global\@minitoc@used@true
```

`\@tocfile` The name of the file containing the minitoc is constructed from `\jobname` and a suffix `\@tocfile`, which is `mtc` (long extensions) or `M` (short extensions) followed by the absolute number of the minitoc.

```
500 \if@longextensions@
501 \def\@tocfile{mtc\The@mtc}%
502 \else
503 \def\@tocfile{M\The@mtc}%
504 \fi
```

`\mtc@CkFile` Then we test (via `\mtc@CkFile`) the emptiness of this file. A warning is given if the file `\if@mtc@FE` is empty and a flag is set (a hint will signal that an empty minitoc has been requested). `\if@mtc@empty@minitoc@`

```
505 \mtc@CkFile{\jobname.\@tocfile}
506 \if@mtc@FE
507 \PackageInfo{minitoc}%
508 {\jobname.\@tocfile\space is empty}
509 \@mtc@empty@minitoc@true
510 \else
```

`\thispagemininitocstyle` We call `\thispagemininitocstyle` to set the page style (by default, this does nothing because, by default, there is no page break before a minitoc). The marks are not treated, because usually there is no new page for a minitoc.

```

511      \thispageminitocstyle
512 %%      \mtc@markboth{\uppercase{\mtctitle}}{\uppercase{\mtctitle}}%

\beforeminitoc We call \beforeminitoc, then begin a samepage environment (to try to discourage page
samepage breaks in a minitoc) and look at the position of the title. If the title is empty, the layout
\do@mtitc is corrected. We print the title with its font (\mtifont), then the top rule of the minitoc
\e@mti (if rules are present), using a tabular environment (to inhibit a page break between the
\n@mti title and the top rule). The font is set to \mtcfont.
\c@mti
\l@mti 513      \beforeminitoc
\r@mti 514      \relax\begin{samepage}%
\df@mtic 515      \if #1e\let\do@mtitc\e@mti
\mtc@CkStr 516      \else\if #1n\let\do@mtitc\n@mti
\mtctitle 517      \else\if #1c\let\do@mtitc\c@mti
\if@mtc@FE 518      \else\if #1l\let\do@mtitc\l@mti
\mtcfont 519      \else\if #1r\let\do@mtitc\r@mti
\mtifont 520      \else\if #1d\let\do@mtitc\df@mtitc
\mtc@rule 521      \fi\fi\fi\fi\fi\fi
tabular 522      \mtc@CkStr{\mtctitle}\if@mtc@FE \let\do@mtitc\e@mti\relax\fi
523      \raggedright
524      \parskip=\z@%
525      \reset@font\mtcfont%
526      \parindent=\z@%
527      \nopagebreak[4]%
528      \kern-0.8\baselineskip\nopagebreak[4]%
529      \par\noindent %
530      \ifx\mtc@rule\relax
531      \begin{tabular}{@{}p{\columnwidth}@{}}
532      \reset@font\mtifont\do@mtitc{\mtc@v\mtctitle}\\
533      \end{tabular}%
534      \else
535      \begin{tabular}{@{}p{\columnwidth}@{}}
536      \reset@font\mtifont\do@mtitc{\mtc@v\mtctitle}\\ \hline
537      \end{tabular}%
538      \fi

\mtc@zrule We forbid a page break after the title and the top rule, then set some layout parameters
\mtc@BBR and begin an mtc@verse environment:
\mtcindent
mtc@verse 539      \nopagebreak[4]\null\leavevmode\mtc@zrule\\ \mtc@BBR
540      \leftmargin\mtcindent
541      \rightmargin\mtcindent
542      \itemindent=\z@\labelwidth=\z@%
543      \labelsep=\z@\listparindent=\z@%
544      \begin{mtc@verse}%

\c@tocdepth We force the effective depth of the mini-table (\c@tocdepth) to the required depth
\c@minitocdepth (\c@minitocdepth), so the printing is done inside the mtc@verse environment, where
\\ tocdepth has been forced to minitocdepth, to print only the entries whose level is low
\mtc@BBR

```



enough. then inhibit a page break. The blank line is necessary to avoid a parasite negative indentation.

```
545      \c@tocdepth=\c@minitocdepth
546      \leavevmode\\mtc@BBR
547      \vskip -.5\baselineskip
```

```
\mtc@pgno We test the presence of leaders and page numbers, then print the minitoc by inputting the
\@dottedtocline minitoc file. But before reading the minitoc file, we must call the hook macro (asked for
\@undottedtocline by Donald ARSENEAU for his notoccite package [2]) \mtc@hook@beforeinputfile
\mtc@hook@beforeinputfile and the macro \mtc@setform which adjusts some layout parameters (defined by the user
\mtc@setform via \mtcsetformat). We work in a group to keep local some macro redefinitions.
\mtcsetformat
\mtc@strut 548 \beginngroup
549   \makeatletter
550   \ifundefined{mtc@pgno}%
551   {\let\@dottedtocline\@undottedtocline}{}
552   \@fileswfalse\mtc@hook@beforeinputfile
553   \mtc@setform%
554   \@input{\jobname.\@tocfile}
555   \vspace{-1ex} \vspace{-.5\baselineskip}
556   \leavevmode\mtc@strut
557   \global\@nobreakfalse\endgroup
```

```
mtc@verse We close the mtc@verse environment, add the bottomrule (while preventing a page
\mtc@bottom@rule break), then close the samepage environment, and call \afterminitoc. The blank line
\\ (\\) is essential.
samepage
\afterminitoc 558   \end{mtc@verse}%
559   \kern0.\baselineskip%
560   \nopagebreak[4]\mtc@bottom@rule\null\leavevmode\\%
561   \vskip-1.0\baselineskip\mtc@zrule
562   \end{samepage}%
563   \par\pagebreak[1]\vspace*{-1ex}\afterminitoc\fi}
```

```
\mtc@bottom@rule And we define the bottom rule for a minitoc, with some space under the minitoc:
\mtc@rule
```

```
564 \def\mtc@bottom@rule{%
565   \ifx\mtc@rule\relax\relax\else
566     \vskip -2.5ex
567     \rule[2.4\p@]{\columnwidth}{.4\p@}\vspace*{2.6\p@}\fi}
```

### 5.35.2 The \minilof command

\minilof The \minilof command is very similar to the \minitoc command.

`\minilof` The `\minilof` command must be used after `\chapter` if you need a minilof (no `\chapter` automatic minilof).

`\dominilof` This command accepts an optional argument, whose default value has eventually been  
`\minilof` set earlier by a `\dominilof` command. The letter “d” represents this default value. `\dominilof` has itself an optional argument which sets the default value of the optional argument of `\minilof`. The default value of the optional argument of the `\dominilof` command is “1”. It seems tortuous, but it is simple to use: we have a default behaviour (1) which can be altered globally via the optional argument of `\dominilof`, or locally via the optional argument of `\minilof`.

`\minilof` So we define `\minilof` with an optional argument and its (current) default value, and  
`\minilof@` call the true code in the `\minilof@` macro (which has one delimited argument); we use  
`\@ifnextchar` the `\@ifnextchar` trick to detect a left bracket for the optional argument:

```
568 \def\minilof{\@ifnextchar[{\minilof@}{\minilof@d}}
```

The real code of `\minilof` is in `\minilof@`, which has a mandatory argument (delimited by brackets) specifying the position of the title.

`\if@minilof@used@` First, we set the global flag `\@minilof@used@true` to note that `\minilof` has been called (this will be used by a hint later, section 5.76.2.2 on page 198).

```
569 \def\minilof@[#1]{%
570 \global\@minilof@used@true
```

`\@tocfile` The name of the file containing the minilof is constructed from `\jobname` and a suf-  
`\if@longextensions@` fix `\@tocfile`, which is mlf (long extensions) or F (short extensions) followed by the absolute number of the minilof.

```
571 \if@longextensions@%
572 \def\@tocfile{mlf\The@mtc}%
573 \else
574 \def\@tocfile{F\The@mtc}%
575 \fi
```

`\mtc@CkFile` Then we test (via `\mtc@CkFile`) the emptiness of this file. A warning is given if the file  
`\if@mtc@FE` is empty and a flag is set (a hint will signal that and empty minilof has been requested).  
`\if@mtc@empty@minilof@`

```
576 \mtc@CkFile{\jobname.\@tocfile}
577 \if@mtc@FE
578 \PackageInfo{minitoc}%
579 {\jobname.\@tocfile\space is empty}
580 \@mtc@empty@minilof@true
581 \else
```

`\thispageminilofstyle` We call `\thispageminilofstyle` to set the page style (by default, this does nothing because, by default, there is no page break before a minilof). The marks are not treated, because usually there is no new page for a minilof.

```
582      \thispageminilofstyle
583 %%      \mtc@markboth{\uppercase{\mlftitle}}{\uppercase{\mlftitle}}%
```

`\beforeminilof` We call `\beforeminilof`, then begin a `samepage` environment (to try to discourage page breaks in a minilof) and look at the position of the title. If the title is empty, the layout is corrected. We print the title with its font (`\mtifont`), then the top rule of the minilof (if rules are present), using a `tabular` environment (to inhibit a page break between the title and the top rule). The font is set to `\mlffont`.

```
584      \beforeminilof
585      \relax\begin{samepage}%
586      \if #1e\let\do@mtilf\e@mti
587      \else\if #1n\let\do@mtilf\n@mti
588      \else\if #1c\let\do@mtilf\c@mti
589      \else\if #1l\let\do@mtilf\l@mti
590      \else\if #1r\let\do@mtilf\r@mti
591      \else\if #1d\let\do@mtilf\df@mtilf
592      \fi\fi\fi\fi\fi\fi
593      \mtc@CkStr{\mlftitle}\if@mtc@FE \let\do@mtilf\e@mti\relax\fi
594      \raggedright
595      \parskip=\z@
596      \reset@font\mlffont
597      \parindent=\z@
598      \nopagebreak[4]%
599      \kern-0.8\baselineskip\nopagebreak[4]%
600      \par\noindent
601      \ifx\mlf@rule\relax
602      \begin{tabular}{@{}p{\columnwidth}@{}}
603      \reset@font\mtifont\do@mtilf{\mtc@v\mlftitle}\\
604      \end{tabular}%
605      \else
606      \begin{tabular}{@{}p{\columnwidth}@{}}
607      \reset@font\mtifont\do@mtilf{\mtc@v\mlftitle}\\ \hline
608      \end{tabular}%
609      \fi
```

`\mtc@zrule` We forbid a page break after the title and the top rule, then set some layout parameters and begin an `mtc@verse` environment:

```
\mtc@BBR
\mtcindent
mtc@verse 610      \nopagebreak[4]\null\leavevmode\mtc@zrule\\\mtc@BBR
611      \leftmargin\mtcindent
612      \rightmargin\mtcindent
613      \itemindent=\z@\labelwidth=\z@%
614      \labelsep=\z@\listparindent=\z@%
615      \begin{mtc@verse}%
```

`\c@tocdepth` We force the effective depth of the mini-table (`\c@tocdepth`) to the required depth  
`\c@minilofdepth` (`\c@minilofdepth`), so the printing is done inside the `mtc@verse` environment, where  
`\` `tocdepth` has been forced to `minilofdepth`, to print only the entries whose level is low  
`\mtc@BBR` enough. then inhibit a page break. The blank line is necessary to avoid a parasite negative  
indentation.

```
616      \ifundefined{c@lofdepth}{\c@tocdepth=\c@minilofdepth}
617      \ifnum\c@tocdepth<1\relax\c@tocdepth=1\fi
618      \leavevmode\mtc@BBR
619      \vskip -.5\baselineskip
```

`\mtc@pgno` We test the presence of leaders and page numbers, then print the minilof by inputting the  
`\@dottedtocline` minilof file. But before reading the minilof file, we must call the hook macro (asked for  
`\@undottedtocline` by Donald ARSENEAU for his notoccite package [2]) `\mtc@hook@beforeinputfile`  
`\mtc@hook@beforeinputfile` and the macro `\mlf@setform` which adjusts some layout parameters (defined by the user  
`\mlf@setform` via `\mtcsetformat`). We work in a group to keep local some macro redefinitions.

```
\mtcsetformat
\mtc@strut 620 \begingroup
621   \makeatletter
622   \ifundefined{mlf@pgno}%
623   {\let\@dottedtocline\@undottedtocline}{}
624   \@fileswfalse\mtc@hook@beforeinputfile
625   \mlf@setform
626   \@input{\jobname.\@tocfile}
627   \vspace{-1ex} \vspace{-.5\baselineskip}
628   \leavevmode\mtc@strut
629   \global\@nbreakfalse\endgroup
```

`mtc@verse` We close the `mtc@verse` environment, add the bottomrule (while preventing a page  
`\mtc@bottom@rule` break), then close the `samepage` environment, and call `\afterminilof`. The blank line  
`\` (`\`) is essential.

```
samepage
\afterminilof 630      \end{mtc@verse}%
631      \kern-1.\baselineskip%
632      \nolinebreak[4]\mlf@rule\null\leavevmode\%
633      \vskip-1.0\baselineskip\mtc@zrule\end{samepage}%
634      \par\pagebreak[1]\vspace*{-1ex}\afterminilof\fi%
```

### 5.35.3 The `\minilot` command

`\minilot` The `\minilot` command is absolutely similar to the `\minilof` command:

`\minilot` The `\minilot` command must be used after `\chapter` if you need a minilot (no  
`\chapter` automatic minilot).

`\dominilot` This command accepts an optional argument, whose default value has eventually been set earlier by a `\dominilot` command. The letter “d” represents this default value. `\dominilot` has itself an optional argument which sets the default value of the optional argument of `\minilot`. The default value of the optional argument of the `\dominilot` command is “1”. It seems tortuous, but it is simple to use: we have a default behaviour (1) which can be altered globally via the optional argument of `\dominilot`, or locally via the optional argument of `\minilot`.

`\minilot` So we define `\minilot` with an optional argument and its (current) default value, and `\minilot@` call the true code in the `\minilot@` macro (which has one delimited argument); we use `\@ifnextchar` the `\@ifnextchar` trick to detect a left bracket for the optional argument:

```
635 \def\minilot{\@ifnextchar[{ \minilot@}{\minilot@d}}
```

The real code of `\minilot` is in `\minilot@`, which has a mandatory argument (delimited by brackets) specifying the position of the title.

`\if@minilot@used@` First, we set the global flag `\@minilot@used@true` to note that `\minilot` has been called (this will be used by a hint later, section 5.76.2.2 on page 198).

```
636 \def\minilot#[#1]{%
637 \global\@minilot@used@true
```

`\@tocfile` The name of the file containing the minilot is constructed from `\jobname` and a suffix `\@tocfile`, which is `mlt` (long extensions) or `T` (short extensions) followed by the absolute number of the minilot.

```
638 \if@longextensions@%
639 \def\@tocfile{mlt\The@mtc}%
640 \else
641 \def\@tocfile{T\The@mtc}%
642 \fi
```

`\mtc@CkFile` Then we test (via `\mtc@CkFile`) the emptiness of this file. A warning is given if the file `\if@mtc@FE` is empty and a flag is set (a hint will signal that an empty minilot has been requested). `\if@mtc@empty@minilot@`

```
643 \mtc@CkFile{\jobname.\@tocfile}
644 \if@mtc@FE
645 \PackageInfo{minitoc}%
646 {\jobname.\@tocfile\space is empty}
647 \@mtc@empty@minilot@true
648 \else
```

`\thispageminilotstyle` We call `\thispageminilotstyle` to set the page style (by default, this does nothing because, by default, there is no page break before a minilot). The marks are not treated, because usually there is no new page for a minilot.

```

649      \thispageminilotstyle
650 %%      \mtc@markboth{\uppercase{\mltttitle}}{\uppercase{\mltttitle}}%

\beforeminilot We call \beforeminilot, then begin a samepage environment (to try to discourage page
samepage breaks in a minilot) and look at the position of the title. If the title is empty, the layout
\do@mtitc is corrected. We print the title with its font (\mtifont), then the top rule of the minilot
\e@mti (if rules are present), using a tabular environment (to inhibit a page break between the
\n@mti title and the top rule). The font is set to \mltfont.
\c@mti
\l@mti 651      \beforeminilot
\r@mti 652      \relax\begin{samepage}%
\df@mtic 653      \if #1e\let\do@mtilt\e@mti
\mtc@CkStr 654      \else\if #1n\let\do@mtilt\n@mti
\mtctitle 655      \else\if #1c\let\do@mtilt\c@mti
\if@mtc@FE 656      \else\if #1l\let\do@mtilt\l@mti
\mltfont 657      \else\if #1r\let\do@mtilt\r@mti
\mtifont 658      \else\if #1d\let\do@mtilt\df@mtilt
\mtc@rule 659      \fi\fi\fi\fi\fi\fi
tabular 660      \mtc@CkStr{\mltttitle}\if@mtc@FE \let\do@mtilt\e@mti\relax\fi
661      \raggedright
662      \parskip=\z@%
663      \reset@font\mltfont%
664      \parindent=\z@%
665      \nopagebreak[4]%
666      \kern-0.8\baselineskip\nopagebreak[4]%
667      \par\noindent
668      \ifx\mlt@rule\relax
669      \begin{tabular}{@{}p{\columnwidth}@{}}
670      \reset@font\mtifont\do@mtilt{\mtc@v\mltttitle}\\
671      \end{tabular}%
672      \else
673      \begin{tabular}{@{}p{\columnwidth}@{}}
674      \reset@font\mtifont\do@mtilt{\mtc@v\mltttitle}\\\hline
675      \end{tabular}%
676      \fi

\mtc@zrule We forbid a page break after the title and the top rule, then set some layout parameters
\mtc@BBR and begin an mtc@verse environment:
\mtcindent
mtc@verse 677      \nopagebreak[4]\null\leavevmode\mtc@zrule\\\mtc@BBR
678      \leftmargin\mtcindent
679      \rightmargin\mtcindent
680      \itemindent=\z@\labelwidth=\z@%
681      \labelsep=\z@\listparindent=\z@%
682      \begin{mtc@verse}%

\c@tocdepth We force the effective depth of the mini-table (\c@tocdepth) to the required depth
\c@minilotdepth (\c@minilotdepth), so the printing is done inside the mtc@verse environment, where
\\ tocdepth has been forced to minilotdepth, to print only the entries whose level is low
\mtc@BBR

```

enough. then inhibit a page break. The blank line is necessary to avoid a parasite negative indentation.

```
683      \ifundefined{c@lotdepth}{}{\c@tocdepth=\c@minilotdepth}
684      \ifnum\c@tocdepth<1\relax\c@tocdepth=1\fi
685      \leavevmode\\mtc@BBR
686      \vskip -.5\baselineskip
```

```
\mtc@pgno We test the presence of leaders and page numbers, then print the minilot by inputting the
\@dottedtocline minilot file. But before reading the minilot file, we must call the hook macro (asked for
\@undottedtocline by Donald ARSENEAU for his notoccite package [2]) \mtc@hook@beforeinputfile
\mtc@hook@beforeinputfile and the macro \mlt@setform which adjusts some layout parameters (defined by the user
\mlt@setform via \mtcsetformat). We work in a group to keep local some macro redefinitions.
\mtcsetformat
\mtc@strut 687 \beginngroup
688 \makeatletter
689 \ifundefined{mlt@pgno}%
690 {\let\@dottedtocline\@undottedtocline}{}
691 \@filesfalse\mtc@hook@beforeinputfile
692 \mlt@setform
693 \@input{\jobname.\@tocfile}
694 \vspace{-1ex} \vspace{-.5\baselineskip}
695 \leavevmode\mtc@strut
696 \global\@nobreakfalse\endgroup
```

```
mtc@verse We close the mtc@verse environment, add the bottomrule (while preventing a page
\mtc@bottom@rule break), then close the samepage environment, and call \afterminilot. The blank line
\\ (\) is essential.
samepage
\afterminilot 697 \end{mtc@verse}%
698 \kern-1.\baselineskip%
699 \nopagebreak[4]\mlt@rule\null\leavevmode\\%
700 \vskip-1.0\baselineskip\mtc@zrule\end{samepage}%
701 \par\pagebreak[1]\vspace*{-1ex}\afterminilot\fi%
```

### 5.36 Patching the \chapter command, continued

```
\l@xchapter First, we define \l@xchapter which is like \l@chapter, but with a huge depth, to
\@dottedtocline inhibit its printing (except if you cheat):
\l@chapter
\chapter 702 \def\l@xchapter{\@dottedtocline{\@M}{1em}{2.3em}}
703 \def\xchapter{xchapter}
```

`\@chapter` Then we patch `\@chapter` (the non-starred branch of `\chapter`) to add pseudo-chapter entries in the LOF and the LOT (these entries will be used by the `\dominiXXX` commands to split the LOF and the LOT into slices).

`\sv@chapter`

`\addcontentsline`

`\ignorespaces`

```

704 \let\sv@chapter\@chapter
705 \def\@chapter[#1]#2{\sv@chapter[#1]{#2}\relax%
706   \addcontentsline{lof}{xchapter}{#1}%
707   \addcontentsline{lot}{xchapter}{#1}%
708   \ignorespaces}

```

`\mtc@schapter` We also patch `\@schapter` (the starred branch of `\chapter`) to add marks in the TOC to delimit chapters; these marks will be used by the `\dominiXXX` commands to take slices from the LOF and the LOT; as they are defined as `\relax`, they should not perturbate other packages.

`\@schapter`

`\addtocontents`

`\chapterbegin`

`\chapterend`

```

709 \let\mtc@schapter\@schapter
710 \def\@schapter{\addtocontents{toc}{\protect\chapterend}\mtc@schapter}
711 \def\@schapter{\addtocontents{@@@}{\protect\chapterbegin}\mtc@schapter}
712 \let\chapterbegin\relax
713 \let\chapterend\relax

```

### 5.37 The `\addstarred...` commands

`\addstarredsection` If the command `\chapter` is undefined, we define the command `\addstarredsection` (only if `\section` is defined). If the command `\chapter` is defined, we define the

`\addstarredchapter` command `\addstarredchapter`. If the command `\part` is defined, we define the

`\addstarredpart` command `\addstarredpart`. We use the utility command `\addst@rred` defined in

`\chapter` section 5.31 on page 91.

`\section`

`\part`

`\addst@rred`

```

714 \@ifundefined{chapter}%
715   {\@ifundefined{section}%
716     {}{\def\addstarredsection#1{\addst@rred{section}{#1}}}%
717   {\def\addstarredchapter#1{\addst@rred{chapter}{#1}}}
718 \@ifundefined{part}%
719   {}{\def\addstarredpart#1{\addst@rred{part}{#1}}}

```

### 5.38 TOC entries without leaders

`\@Undottedtocline` We define two internal macros to format TOC entries without leaders. The macro

`\coffeeont` `\@Undottedtocline` prints no page number, but `\@Undottedtoclinep` prints it.

```

720 \def\@Undottedtocline#1#2#3#4#5{%
721   \ifnum #1>\c@tocdepth \else

```



```

722 \vskip \z@ \@plus.2\p@
723 {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
724 \parindent #2\relax\@afterindenttrue
725 \interlinepenalty\@M
726 \leavevmode
727 \@tempdima #3\relax
728 \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
729 {\coffee font #4}\nobreak
730 \nobreak\null
731 \par}%
732 \fi}

```

`\@Undottedtoclinep` The same but with the page number:  
`\coffee font`

```

733 \def\@Undottedtoclinep#1#2#3#4#5{%
734 \ifnum #1>\c@tocdepth \else
735 \vskip \z@ \@plus.2\p@
736 {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
737 \parindent #2\relax\@afterindenttrue
738 \interlinepenalty\@M
739 \leavevmode
740 \@tempdima #3\relax
741 \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
742 {#4}\nobreak
743 \hfill
744 \nobreak\null
745 \hb@xt@\@pnumwidth{\hfil\normalfont \normalcolor #5}%
746 \par}%
747 \fi}

```

### 5.39 Mini-tables with or without leaders

`\minitoc@` This code creates the flag, then patches each mini-table command (its internal  
`\minilof@` part). We alter the commands `\minitoc@`, `\minilof@`, etc., to test the flag  
`\minilot@` `\ifundottedmtc` and, if true, replace locally `\@dottedtocline` by its dotless version  
`\@Undottedtoclinep`. Of course, we must also test the availability of the `\chapter`,  
`\@dottedtocline` `\part` and `\section` commands, to avoid to define many unnecessary commands.  
`\@Undottedtoclinep`

```

\sv@minitoc@ 748 \newif\ifundottedmtc\undottedmtcfalse
\sv@minilof@ 749 \@ifundefined{chapter}{}{}%
\sv@minilot@ 750 \let\sv@minitoc@\minitoc@
751 \def\minitoc@[#1]{{\ifundottedmtc\let\@dottedtocline\@Undottedtoclinep\fi
752 \sv@minitoc@[#1]}}%
753 \let\sv@minilof@\minilof@
754 \def\minilof@[#1]{{\ifundottedmtc\let\@dottedtocline\@Undottedtoclinep\fi
755 \sv@minilof@[#1]}}%
756 \let\sv@minilot@\minilot@

```

```

757 \def\minilot[#1]{\ifundottedmtc\let\@dottedtocline\@Undottedtocline\fi
758 \sv\minilot[#1]}}

\sv@parttoc@ For the part level:
\sv@partlof@
\sv@partlot@ 759 \@ifundefined{part}{}{%
\ifundottedmtc 760 \let\sv@parttoc@\parttoc@
\parttoc@ 761 \def\parttoc[#1]{\ifundottedmtc\let\@dottedtocline\@Undottedtocline\fi
\partlof@ 762 \sv@parttoc[#1]}}%
\partlot@ 763 \let\sv@partlof@\partlof@
764 \def\partlof[#1]{\ifundottedmtc\let\@dottedtocline\@Undottedtocline\fi
765 \sv@partlof[#1]}}%
766 \let\sv@partlot@\partlot@
767 \def\partlot[#1]{\ifundottedmtc\let\@dottedtocline\@Undottedtocline\fi
768 \sv@partlot[#1]}}

\sv@secttoc@ For the section level:
\sv@sectlof@
\sv@sectlot@ 769 \@ifundefined{chapter}{}%
\ifundottedmtc 770 \@ifundefined{section}{}{%
\secttoc@ 771 \let\sv@secttoc@\secttoc@
\sectlof@ 772 \def\secttoc[#1]{\ifundottedmtc\let\@dottedtocline\@Undottedtocline\fi
\sectlot@ 773 \sv@secttoc[#1]}}%
774 \let\sv@sectlof@\sectlof@
775 \def\sectlof[#1]{\ifundottedmtc\let\@dottedtocline\@Undottedtocline\fi
776 \sv@sectlof[#1]}}%
777 \let\sv@sectlot@\sectlot@
778 \def\sectlot[#1]{\ifundottedmtc\let\@dottedtocline\@Undottedtocline\fi
779 \sv@sectlot[#1]}}}}

```

## 5.40 The \dominitoc command and its siblings

\dominitoc The three commands \dominitoc, \dominilof and \dominilot are, of course,  
\dominilof very similar. They take the \jobname.toc file (resp. the \jobname.lof and  
\dominilot \jobname.lot files) produced by the previous L<sup>A</sup>T<sub>E</sub>X run and cut it in slices  
\contentsline (one slice per chapter or starred chapter) into the \jobname.mtc<N> files (resp.  
\chapbegin the \jobname.mlf<N> and \jobname.mlt<N> files), using specific lines in the  
\starchapter \jobname.toc (resp. \jobname.lof and \jobname.lot) file. These lines  
are essentially chapter-level entry commands (like \contentsline{chapter}{...},  
\contentsline{xchapter}{...}, \contentsline{starchapter}{...}, \chapbegin  
delimiting chapters in the TOC (or in the LOF or the LOT). Analog part-level lines  
delimit parts, hence also chapters.

\dominitoc As \dominitoc has an optional argument, whose default value is “l” (left), it calls  
\dominitoc@ \dominitoc@ with a argument delimited by brackets.  
@@dominitoc  
\if@dominitoc@used@

\dominitoc \@dominitoc \MTC@next	The \dominitoc command extracts information from the .toc file and create the minitocs files, with the adequate extension.
--	--

\dominilof The \dominilof command extracts information from the .lof file and create the  
\@dominilof minilofs files, with the adequate extension.  
\MLF@next

<code>\dominilot</code>	The <code>\dominilot</code> command extracts information from the <code>.lot</code> file and create the
<code>\@dominilot</code>	minilots files, with the adequate extension.
<code>\MLT@next</code>	

`\if@dominitoc@used@` Some code to flag the use of the command and manage the position of the minitoc title:

```

\df@m%ti%tc
    \e@m%ti% 796 \def\dominitoc@[#1]{%
    \n@m%ti% 797 \global\@dominitoc@used@true
    \c@m%ti% 798 \if #1e\let\df@m%ti%tc\e@m%ti%
    \l@m%ti% 799 \else\if #1n\let\df@m%ti%tc\n@m%ti%
    \r@m%ti% 800 \else\if #1c\let\df@m%ti%tc\c@m%ti%
    \@dominitoc 801 \else\if #1l\let\df@m%ti%tc\l@m%ti%

```

```

802 \else\if #1r\let\df@mtitc\r@mti%
803 \fi\fi\fi\fi\fi%
804 \@@dominitoc}

```

`\if@dominilof@used@` Some code to flag the use of the command and manage the position of the minilof title:

```

\df@mtilf
\@mti 805 \def\dominilof@[#1]{%
\@nmti 806 \global\@dominilof@used@true
\@cmti 807 \if #1e\let\df@mtilf\@mti%
\@lmti 808 \else\if #1n\let\df@mtilf\@nmti%
\@rmti 809 \else\if #1c\let\df@mtilf\@cmti%
\@@dominilof 810 \else\if #1l\let\df@mtilf\@lmti%
811 \else\if #1r\let\df@mtilf\@rmti%
812 \fi\fi\fi\fi\fi%
813 \@@dominilof}

```

`\if@dominilot@used@` Some code to flag the use of the command and manage the position of the minilot title:

```

\df@mtilt
\@mti 814 \def\dominilot@[#1]{%
\@nmti 815 \global\@dominilot@used@true
\@cmti 816 \if #1e\let\df@mtilt\@mti%
\@lmti 817 \else\if #1n\let\df@mtilt\@nmti%
\@rmti 818 \else\if #1c\let\df@mtilt\@cmti%
\@@dominilot 819 \else\if #1l\let\df@mtilt\@lmti%
820 \else\if #1r\let\df@mtilt\@rmti%
821 \fi\fi\fi\fi\fi%
822 \@@dominilot}

```

`\@@dominitoc` These macros invoke the `\@domini...` macros to create the mini-table file, then close the file descriptor.

```

\@@dominilof
\@@dominilot
\tf@mtc 823 \def\@@dominitoc{\@dominitoc{\jobname}\immediate\closeout\tf@mtc}
824 \def\@@dominilof{\@dominilof{\jobname}\immediate\closeout\tf@mtc}
825 \def\@@dominilot{\@dominilot{\jobname}\immediate\closeout\tf@mtc}

```

### 5.40.1 Analysis and splitting of the TOC file

This is done via a loop managed by the following macros <sup>6</sup>:

`\MTC@next` Processes the next entry in the list and removes it from the head of the list:  
`\MTC@list`  
`\MTC@loop`

---

<sup>6</sup>This code is derived from the `xr` package [12], by David CARLISLE, with his permission. The strings “HO”, “DV”, and “BJ” in the comments denote modifications made by Heiko OBERDIEK, Didier VERNA, and Bernd JAEHNE for the support of `hyperref`, essentially by adding an argument to some macros, to use the `hyperlink` argument in the contents lines.

```

826 \def\MTC@next#1\relax#2\{\%
827   \edef\MTC@list{#2}%
828   \MTC@loop{#1}%
829 }

```

```

\MTC@toc    Check if the list is empty:
\MTC@list
\MTC@explist 830 \def\MTC@toc{%
831   \ifx\MTC@list\@empty\else\expandafter\MTC@explist\fi
832 }

```

```

\MTC@contentsline The macro \MTC@contentsline analyses the lines read from the TOC file and detects
\arabic           interesting keywords. If \chapter is found, the mtc counter (which simulates the chapter
\chapter          counter, but is absolute) is incremented and a new minitoc file is created.
\themtc
\tf@mtc 833 \def\MTC@contentsline#1#2#3#4{% %%H0/BJ: 4 instead of 3 parameters
834   \gdef\themtc{\arabic{mtc}}% %%H0: space removed
835   \expandafter\ifx\csname #1\endcsname\chapter
836     \stepcounter{mtc}%

```

```

\if@longextensions@ We test if long or short extensions are used, to build the name of the mini-table file, then
\themtc             open it (after closing the file descriptor):
\mtcname
\tf@mtc 837   \if@longextensions@%
\closeout 838     \PackageInfo{minitoc}%
\openout 839     {Writing\space\jobname.mtc\themtc\@gobble}%
840     \def\mtcname{\jobname.mtc\themtc}%
841   \else
842     \PackageInfo{minitoc}%
843     {Writing\space\jobname.M\themtc\@gobble}%
844     \def\mtcname{\jobname.M\themtc}%
845   \fi
846   \immediate\closeout\tf@mtc
847   \immediate\openout\tf@mtc=\mtcname
848   \fi

```

```

\mtc@toks Now, we filter the relevant contents lines, the token register \mtc@toks is used as a
          verbatim memory.

```

```

849   \mtc@toks{\noexpand\leavevmode #2}%

```

```

\MTC@WriteContentsline Each interesting contents line is copied, with a font command added before it. We begin
\section              with the standatd sectionning commands, below \chapter:
\subsection
\subsubsection 850   \expandafter\ifx\csname #1\endcsname\section
\paragraph      851     \MTC@WriteContentsline{#1}{mtcS}{#3}{#4}%
\subparagraph

```

```

852 \fi
853 \expandafter\ifx\csname #1\endcsname\subsection
854 \MTC@WriteContentsline{#1}{mtcSS}{#3}{#4}%
855 \fi
856 \expandafter\ifx\csname #1\endcsname\subsubsection
857 \MTC@WriteContentsline{#1}{mtcSSS}{#3}{#4}%
858 \fi
859 \expandafter\ifx\csname #1\endcsname\paragraph
860 \MTC@WriteContentsline{#1}{mtcP}{#3}{#4}%
861 \fi
862 \expandafter\ifx\csname #1\endcsname\subparagraph
863 \MTC@WriteContentsline{#1}{mtcSP}{#3}{#4}%
864 \fi

```

\coffee A coffee break contents line is written for \coffee:  
\MTC@WriteCoffeeline

```

865 \expandafter\ifx\csname #1\endcsname\coffee
866 \MTC@WriteCoffeeline{#1}{#3}%
867 \fi

```

\starchapter If it is \starchapter (for a starred chapter), we increment the mtc counter, build a new  
\stepcounter minitoc file name, close the file descriptor and open it with this new file.  
@if@longextensions@

```

\mtcname 868 \expandafter\ifx\csname #1\endcsname\starchapter
\themtc 869 \stepcounter{mtc}%
\tf@mtc 870 \if@longextensions@
\closeout 871 \PackageInfo{minitoc}%
\openout 872 {Writing\space\jobname.mtc\themtc\@gobble}%
873 \def\mtcname{\jobname.mtc\themtc}%
874 \else
875 \PackageInfo{minitoc}%
876 {Writing\space\jobname.M\themtc\@gobble}%
877 \def\mtcname{\jobname.M\themtc}%
878 \fi
879 \immediate\closeout\tf@mtc
880 \immediate\openout\tf@mtc=\mtcname
881 \fi

```

\starsection For starred sectionning commands lower than \chapter, a contents line is written into  
\MTC@WriteContentsline the minitoc file, with a font command added:

```

\starsubsection
\starsubsubsection 882 \expandafter\ifx\csname #1\endcsname\starsection
\starparagraph 883 \MTC@WriteContentsline{#1}{mtcS}{#3}{#4}%
\starsubparagraph 884 \fi
885 \expandafter\ifx\csname #1\endcsname\starsubsection
886 \MTC@WriteContentsline{#1}{mtcSS}{#3}{#4}%
887 \fi
888 \expandafter\ifx\csname #1\endcsname\starsubsubsection
889 \MTC@WriteContentsline{#1}{mtcSSS}{#3}{#4}%
890 \fi

```

```

891 \expandafter\ifx\csname #1\endcsname\starparagraph
892 \MTC@WriteContentsline{#1}{mtcP}{#3}{#4}%
893 \fi
894 \expandafter\ifx\csname #1\endcsname\starsubparagraph
895 \MTC@WriteContentsline{#1}{mtcSP}{#3}{#4}%
896 \fi
897 }

```

\MTC@explist The loop to read the lines of the TOC file; it expands the list of entries and call \MTC@next to process the first one:

\MTC@list

```

898 \def\MTC@explist{\expandafter\MTC@next\MTC@list\}

```

\MTC@loop If an entry is found, loop through line by line, looking for interesting entries. Otherwise, process the next entry in the list.

\@inputcheck

\MTC@toc

\MTC@read

```

899 \def\MTC@loop#1{\openin\@inputcheck#1\relax
900 \ifeof\@inputcheck
901 \PackageWarning{minitoc}%
902 {No file #1.\MessageBreak MINITOCs NOT PREPARED}%
903 \expandafter\MTC@toc
904 \else
905 \PackageInfo{minitoc}{PREPARING MINITOCs FROM #1}%
906 \expandafter\MTC@read
907 \fi
908 }

```

\MTC@read

Read the next entry of the .toc file.

\read

\@inputcheck

\MTC@line

```

909 \def\MTC@read{%
910 \read\@inputcheck to\MTC@line

```

\MTC@test

The . . . . . make sure that \MTC@test has sufficient arguments:

\MTC@line

\MTC@

```

911 \expandafter\MTC@test\MTC@line. . . . \MTC% %%H0: . added
912 }%

```

\MTC@test

The \MTC@test macro finds the “interesting” commands in the TOC file, mainly to delimit chapters:

\contentsline

\MTC@contentsline

\mtc@string

\@input

\MTC@list

\chapterend

\closeout

\tf@mtc

\openout

\chapterbegin

\addtocounter

\MTC@toc

\MTC@read

```

913 %%H0/BJ: now patch \MTC@test,
914 %%H0/BJ: call \MTC@contentsline with 4 instead of 3 parameters

```

Look at the first token of the line. If it is an interesting entry, process it. If it is `\@input`, add the file to the list. Otherwise ignore. Go around the loop if not at end of file. Finally process the next file in the list.

```

915 \long\def\MTC@test#1#2#3#4#5#6\MTC@{% %H0: #6 added
916   \ifx#1\contentsline
917     \let\mtc@string\string
918     \MTC@contentsline{#2}{#3}{#4}{#5}%
919     %%H0/BJ: 4. parameter added by Tony Roberts
920     \let\mtc@string\relax
921   \else\ifx#1\@input
922     \edef\MTC@list{\MTC@list#2\relax}%
923   \else\ifx#1\chapterend
924     \immediate\closeout\tf@mtc
925     \immediate\openout\tf@mtc=\jobname.mtc
926   \else\ifx#1\chapterbegin
927     \addtocounter{mtc}{-1}%
928   \fi\fi\fi\fi
929   \ifeof\@inputcheck
930     \expandafter\MTC@toc
931   \else
932     \expandafter\MTC@read
933   \fi
934 }%
```

## 5.41 Mini-lists of figures

The code is similar to the code for mini-tables of contents, but with less commands to recognize.

### 5.41.1 Analysis and splitting of the list of figures file

`\MLF@next` This is done via a loop managed by the following macros:

`\MLF@list`

`\MLF@loop` Processes the next entry in the list and removes it from the head of the list:

```

935 \def\MLF@next#1\relax#2\{\%
936   \edef\MLF@list{#2}%
937   \MLF@loop{#1}}
```

`\MLF@lof` Checks if the list is empty:

`\MLF@list`

```

\MLF@explist 938 \def\MLF@lof{%
939   \ifx\MLF@list\@empty\else\expandafter\MLF@explist\fi}
```



`\MLF@contentsline` The macro `\MLF@contentsline` analyses the lines read from the LOF file and detects interesting keywords. If `\xchapter` is found, `mtc` is incremented and a new minilof file is created.

```
940 \def\MLF@contentsline#1#2#3#4{% %%H0: added #4
941   \gdef\themtc{\arabic{mtc}}}% %%H0: space removed
942   \expandafter\ifx\csname #1\endcsname\xchapter
943     \stepcounter{mtc}%
```

`\if@longextensions@` The name of the minilof file is build from `\jobname` and a long or short extension:

```
\themtc
\mlfname 944   \if@longextensions@
\closeout 945   \PackageInfo{minitoc}%
\tf@mtc   946   {Writing\space\jobname.mlf\themtc\@gobble}
\openout   947   \def\mlfname{\jobname.mlf\themtc}%
          948   \else
          949   \PackageInfo{minitoc}%
          950   {Writing\space\jobname.F\themtc\@gobble}
          951   \def\mlfname{\jobname.F\themtc}%
          952   \fi
          953   \immediate\closeout\tf@mtc
          954   \immediate\openout\tf@mtc=\mlfname
          955   \fi
```

`\figure` If we found a `\figure` entry, we copy it into the minilof file:

```
\mtc@toks
\MTC@WriteContentsline 956 \expandafter\ifx\csname #1\endcsname\figure
                      957   \mtc@toks{\noexpand\leavevmode#2}%
                      958   \MTC@WriteContentsline{#1}{mlf}{#3}{#4}%
                      959   \fi
                      960 }
```

`\MLF@explist` The loop to read the LOF file; it expands the list of entries and calls `\MLF@next` to process the first one:

```
\MLF@next
\MLF@list
961 \def\MLF@explist{\expandafter\MLF@next\MLF@list\}
```

`\MLF@loop` And now, we scan the `.lof` file:

```
\openin
\@inputcheck 962 \def\MLF@loop#1{\openin\@inputcheck#1\relax
\MLF@lof 963   \ifeof\@inputcheck
\MLF@read 964   \PackageWarning{minitoc}%
          965   {No file #1.\MessageBreak MINILOFS NOT PREPARED}%
          966   \expandafter\MLF@lof
          967   \else
          968   \PackageInfo{minitoc}%
          969   {PREPARING MINILOFS FROM #1}%
          970   \expandafter\MLF@read\fi}
```

```

\MLF@read  Read the next entry in the .lof file:
\read
\@inputcheck 971 \def\MLF@read{%
972   \read\@inputcheck to\MLF@line

\MLF@line  The ..... make sure that \MLF@test has sufficient arguments:
\MLF@test
\MLF@ 973   \expandafter\MLF@test\MLF@line.....\MLF@% %%H0: . added
974   }%

\MLF@test  The \MLF@test macro finds the “interesting” commands in the LOF file, mainly to
delimit chapters.

\contentsline  Look at the first token of the line. If it is an interesting entry, process it. If it is \@input,
\mtc@string    add the file to the list. Otherwise ignore. Go around the loop if not at end of file. Finally
\MLF@contentsline process the next file in the list.
\@input
\MLF@list 975 \long\def\MLF@test#1#2#3#4#5#6\MLF@{% %%H0: #6 added
\chapterend 976   \ifx#1\contentsline
\closeout 977   \let\mtc@string\string
\tf@mtc 978   \MLF@contentsline{#2}{#3}{#4}{#5}% %%H0: #4 added
\openout 979   \let\mtc@string\relax
\chapterbegin 980   \else\ifx#1\@input
\addtocounter 981   \edef\MLF@list{\MLF@list#2\relax}%
\MLF@lof 982   \else\ifx#1\chapterend
\MLF@read 983   \immediate\closeout\tf@mtc
984   \immediate\openout\tf@mtc=\jobname.mtc
985   \else\ifx#1\chapterbegin
986   \addtocounter{mtc}{-1}%
987   \fi\fi\fi\fi
988   \ifeof\@inputcheck\expandafter\MLF@lof
989   \else\expandafter\MLF@read\fi}%

```

## 5.42 Mini-lists of tables

The code is similar to the code for mini-tables of contents, but with less commands to recognize.

### 5.42.1 Analysis and splitting of the list of tables file

```

\MLT@next  This is done via a loop managed by the following macros:
\MLT@list
\MLT@loop  Processes the next entry in the list and removes it from the head of the list:

```

```

990 \def\MLT@next#1\relax#2\{\%
991   \edef\MLT@list{#2}%
992   \MLT@loop{#1}}

```

```

\MLT@lot    Checks if the list is empty:
\MLT@list
\MLT@explist 993 \def\MLT@lot{%
994   \ifx\MLT@list\@empty\else\expandafter\MLT@explist\fi}

```

\MLT@contentsline The macro \MLT@contentsline analyses the lines read from the LOT file and detects interesting keywords. If \xchapter is found, mtc is incremented and a new minilot file is created.

```

995 \def\MLT@contentsline#1#2#3#4{% %%HO: added #4
996   \gdef\themtc{\arabic{mtc}} %%HO: space removed
997   \expandafter\ifx\csname #1\endcsname\xchapter
998     \stepcounter{mtc}%

```

\if@longextensions@ The name of the minilot file is build from \jobname and a long or short extension:

```

\themtc
\mltname 999   \if@longextensions@%
\closeout 1000   \PackageInfo{minitoc}%
\tf@mtc 1001     {Writing\space\jobname.mlt\themtc\@gobble}%
\openout 1002     \def\mltname{\jobname.mlt\themtc}%
1003   \else
1004     \PackageInfo{minitoc}%
1005     {Writing\space\jobname.T\themtc\@gobble}%
1006     \def\mltname{\jobname.T\themtc}%
1007   \fi
1008   \immediate\closeout\tf@mtc
1009   \immediate\openout\tf@mtc=\mltname
1010   \fi

```

\table If we found a \table entry, we copy it into the minilot file:

```

\mtc@toks
\MTC@WriteContentsline 1011 \expandafter\ifx\csname #1\endcsname\table
1012   \mtc@toks{\noexpand\leavevmode#2}%
1013   \MTC@WriteContentsline{#1}{mlt}{#3}{#4}%
1014   \fi
1015 }

```

\MLT@explist The loop to read the LOT file; it expands the list of entries and calls \MLT@next to process the first one:

```

\MLT@list
1016 \def\MLT@explist{\expandafter\MLT@next\MLT@list\}

```

```

\MLT@loop And now, we scan the .lot file:
\openin
\@inputcheck 1017 \def\MLT@loop#1{\openin\@inputcheck#1\relax
\MLT@lot 1018 \ifeof\@inputcheck
\MLT@read 1019 \PackageWarning{minitoc}%
1020 {No file #1.\MessageBreak MINILOTS NOT PREPARED}%
1021 \expandafter\MLT@lot
1022 \else
1023 \PackageInfo{minitoc}%
1024 {PREPARING MINILOTS FROM #1}%
1025 \expandafter\MLT@read\fi}

\MLT@read Read the next entry in the .lot file:
\read
\@inputcheck 1026 \def\MLT@read{%
1027 \read\@inputcheck to\MLT@line

\MLT@line The ..... make sure that \MLT@test has sufficient arguments:
\MLT@test
\MLT@ 1028 \expandafter\MLT@test\MLT@line.....\MLT@% %%H0: . added
1029 }%

\MLT@test The \MLT@test macro finds the “interesting” commands in the LOT file, mainly to
delimit chapters.

\contentsline Look at the first token of the line. If it is an interesting entry, process it. If it is \@input,
\mtc@string add the file to the list. Otherwise ignore. Go around the loop if not at end of file. Finally
\MLT@contentsline process the next file in the list.
\@input
\MLT@list 1030 \long\def\MLT@test#1#2#3#4#5#6\MLT@{% %%H0: #6 added
\chapterend 1031 \ifx#1\contentsline
\closeout 1032 \let\mtc@string\string
\tf@mtc 1033 \MLT@contentsline{#2}{#3}{#4}{#5}% %%H0: #4 added
\openout 1034 \let\mtc@string\relax
\chapterbegin 1035 \else\ifx#1\@input
\addtocounter 1036 \edef\MLT@list{\MLT@list#2\relax}%
\MLT@lot 1037 \else\ifx#1\chapterend
1038 \immediate\closeout\tf@mtc
\MLT@read 1039 \immediate\openout\tf@mtc=\jobname.mtc
1040 \else\ifx#1\chapterbegin
1041 \addtocounter{mtc}{-1}%
1042 \fi\fi\fi\fi
1043 \ifeof\@inputcheck\expandafter\MLT@lot
1044 \else\expandafter\MLT@read\fi}%

```

Note that we terminate with a closing brace to end the chapter-level macros (end of the *else* branch of a `\@ifundefined{chapter}` alternative).

1045 }%

### 5.43 Macro to write a contents line

`\mtc@dot` The `\MTC@WriteContentsline` macro makes the definition of `\MTC@contentsline` shorter. An extra `\edef` level is removed (Heiko OBERDIEK):

`\mtc@param`  
`\write` The arguments of `\MTC@WriteContentsline` are:  
`\tf@mtc`  
`\@resetfont`  
`\mtc@string` #1: the #1 argument of `\MTC@contentsline`;  
`\contentsline` #2: font shorthand  $\Rightarrow$  `\csname #2font\endcsname`;  
`\mtc@toks` #3: the #3 argument of `\MTC@contentsline`;  
#4: the #4 argument of `\MTC@contentsline` (hyperlink).

```

1046 \def\mtc@dot{.}
1047 \def\MTC@WriteContentsline#1#2#3#4{%
1048   \def\mtc@param{#4}%
1049   \immediate\write\tf@mtc{%
1050     {\string\reset@font
1051       \expandafter\string\csname #2font\endcsname
1052       \string\mtc@string
1053       \string\contentsline{#1}%
1054       {\the\mtc@toks}%
1055       {\string\reset@font
1056         \expandafter\string\csname #2font\endcsname
1057         \space #3%
1058       }%
1059       \ifx\mtc@dot\mtc@param
1060       \else
1061         {#4}% %%HO/BJ: #4 is hyperlink
1062       \fi
1063     }%
1064   }%
1065 }
```

`\MTC@WriteCoffeeline` And the same for a “coffee” line. The arguments of `\MTC@WriteCoffeeline` are:

`\write`  
`\tf@mtc` #1: the #1 argument of `\MTC@contentsline`;  
`\@resetfont` #2: the #3 argument of `\MTC@contentsline`.  
`\coffeefont`  
`\mtc@string`  
`\mtc@toks`

```

1066 \def\MTC@WriteCoffeeline#1#2#3{%
1067   \immediate\write\tf@mtc{%
1068     {\string\reset@font
```

```

1069   \string\coffefont
1070   \string\mtc@string
1071 %% COFFEE \string\textbf{#1}%
1072   {\the\mtc@toks}%
1073   {\string\reset@font
1074   \string\coffefont
1075   \space #3%
1076   }%
1077   }%
1078   }%
1079 }

```

## 5.44 Depth counters for partlofs and partlots

```

\AtBeginDocument If the counters lofdepth and lotdepth are defined, we create the corresponding new
\newcounter counters: partlofdepth and partlotdepth. These counters are initialised to 2. This
\setcounter is done after the loading of the packages, in an \AtBeginDocument block:
\c@lofdepth
\c@lotdepth
1080 \AtBeginDocument{%
1081   \@ifundefined{c@lofdepth}{}%
1082   {\newcounter{partlofdepth}\setcounter{partlofdepth}{2}}%
1083   \@ifundefined{c@lotdepth}{}%
1084   {\newcounter{partlotdepth}\setcounter{partlotdepth}{2}}%
1085 }

```

## 5.45 Part level commands

```

\xpart If \part is defined, we define some utility commands, a counter (ptc) for the
\theptc parttoc and related commands (\theptc, \Thepart, \adjustptc, \decrementptc,
\Thepart \incrementptc), the obsolete command \firstpartis, and the depth counter
\adjustptc parttocdepth.
\decrementptc
\incrementptc
1086 \@ifundefined{part}{}%
\firstpartis 1087 {%
\firstpartis 1088 \def\xpart{xpart}
\if@firstpartis@used@ 1089 \def\Thepart{\arabic{ptc}}
\newcounter 1090 \def\firstpartis#1%
\setcounter 1091 {\PackageWarning{minitoc}%
1092   {*** \string\firstpartis \space is an obsolete command ***}%
1093   \@firstpartis@used@true}
1094 \newcounter{ptc}
1095 \setcounter{ptc}{0}
1096 \newcommand{\adjustptc}[1][1]{\addtocounter{ptc}{#1}}
1097 \def\decrementptc{\addtocounter{ptc}{-1}}
1098 \def\incrementptc{\addtocounter{ptc}{+1}}

```

```

1099 \def\theptc{\arabic{ptc}}
1100 \newcounter{parttocdepth}
1101 \setcounter{parttocdepth}{2}

```

`\ptc@rule` But, sometimes, we need to make a difference between book/report and article classes (is `\chapter` defined?), to have a different layout: the definition of `\ptc@rule` is empty except if `\chapter` is undefined. By default, there is no rule before/after parttocs, partlofs, and partlots for books. You should redeclare `\ptc@rule` if you want these rules.

```

1102 \@ifundefined{chapter}%
1103   {\def\ptc@rule{\rule[3\p@]{\columnwidth}{.4\p@}\vspace*{2.6\p@}}}%
1104   {\let\ptc@rule\relax}

```

`\ptcindent` And we declare the default indentation (both sides) of the parttocs:

```

1105 \newlength\ptcindent
1106 \@ifundefined{chapter}{\ptcindent=24\p@}{\ptcindent=\z@}

```

## 5.46 Fonts for the parttocs

`\ptcfont` We define the fonts for the parttocs. Note that they are larger if `\chapter` is defined (book/report-like document classes) than when it is not (article-like document classes):

```

\ptcSSfont
\ptcSSSfont
\ptcPfont 1107 \@ifundefined{chapter}{%
\ptcSPfont 1108   \def\ptcfont{\small\rmfamily\upshape\mdseries} % the parttoc
\plffont 1109   \def\ptcSfont{\small\rmfamily\upshape\bfseries}% (sections)
\pltfont 1110   \let\ptcSSfont\ptcfont      % (subsections)
\ptifont 1111   \let\ptcSSSfont\ptcfont      % (subsubsections)
1112   \let\ptcPfont\ptcfont      % (paragraphs)
1113   \let\ptcSPfont\ptcfont      % (subparagraphs)
1114   \let\plffont\ptcfont      % (figures)
1115   \let\pltfont\ptcfont      % (tables)
1116   \def\ptifont{\Large\rmfamily\upshape\bfseries}% titles
1117 }%

```

`\ptcfont` If `\chapter` is defined, the fonts are larger and `\ptcCfont` must be defined:

```

\ptcCfont
\ptcSfont 1118 {%
\ptcSSfont 1119   \def\ptcfont{\normalsize\rmfamily\upshape\mdseries} % the parttoc
\ptcSSSfont 1120   \def\ptcCfont{\normalsize\rmfamily\upshape\bfseries}% (chapters)
\ptcPfont 1121   \def\ptcSfont{\normalsize\rmfamily\upshape\mdseries}% (sections)
\ptcSPfont 1122   \let\ptcSSfont\ptcfont      % (subsections)
\plffont 1123   \let\ptcSSSfont\ptcfont      % (subsubsections)
\pltfont 1124   \let\ptcPfont\ptcfont      % (paragraphs)
\ptifont

```

```

1125 \let\ptcSPfont\ptcfont % (subparagraphs)
1126 \let\plffont\ptcfont % (figures)
1127 \let\pltfont\ptcfont % (tables)
1128 \def\ptifont{\LARGE\rmfamily\upshape\bfseries}% titles
1129 }

```

## 5.47 Default titles for part-level mini-tables

`\parttoc` We define the default position, the fonts and the layout for titles of the part-level mini-tables (`\parttoc`, `\partlof` and `\partlot`). This formatting is different if `\chapter` is undefined or defined.

```

\c@pti If \chapter is undefined, the definitions are very simple, for centered, flushleft,
\l@pti flushright or empty titles. Empty titles need a vertical correction (Frank MITTELBACH).
\r@pti
\@pti 1130 \ifundefined{chapter}{%
\n@pti 1131 \def\c@pti#1{\null\hfill #1\hfill\null}
1132 \def\l@pti#1{\null #1\hfill\null}
1133 \def\r@pti#1{\null\hfill #1\null}
1134 \def\@pti#1{\vspace{-\baselineskip}}
1135 \def\n@pti#1{\vspace{-\baselineskip}}
1136 }%

```

`\@pti` But, if `\chapter` is defined, we must simulate the formatting of a chapter head, which is more complex. Empty titles need a vertical correction (Frank MITTELBACH).

```

\if@twocolumn
\@topnewpage 1137 {%
\@makehead@l 1138 \def\@pti#1{\vspace{-\baselineskip}}
\@makehead@r 1139 \def\n@pti#1{\vspace{-\baselineskip}}
\@makehead@c
\@afterheading
\if@twocolumn
\ptifont For a title on the left, we must test if the main text is on two columns:
\@topnewpage
\@afterheading 1140 \def\l@pti#1{\if@twocolumn
\ptifont 1141 \@topnewpage[\@makehead@l{#1}]%
\l@pti 1142 \else
\@makehead@l 1143 \@makehead@l{#1}%
1144 \@afterheading
1145 \fi}
1146 \def\@makehead@l#1{%
1147 \vspace*{50\p@}%
1148 {\parindent \z@ \raggedright
1149 \ptifont
1150 #1\par
1151 \nobreak
1152 \vskip 40\p@
1153 }}

```



`\r@pti` For a title on the right, we must also test if the main text is on two columns:  
`\@makehead@l`

```

1154 \def\r@pti#1{\if@twocolumn
1155     \@topnewpage[\@makehead@r{#1}]%
1156     \else
1157     \@makehead@r{#1}%
1158     \@afterheading
1159     \fi}
1160 \def\@makehead@r#1{%
1161     \vspace*{50\p@}%
1162     {\parindent \z@ \raggedleft
1163     \ptifont
1164     #1\par
1165     \nobreak
1166     \vskip 40\p@
1167     }}

```

`\c@pti` For a centered title, we must also test if the main text is on two columns:  
`\@makehead@c`

```

1168 \def\c@pti#1{\if@twocolumn
1169     \@topnewpage[\@makehead@c{#1}]%
1170     \else
1171     \@makehead@c{#1}%
1172     \@afterheading
1173     \fi}
1174 \def\@makehead@c#1{%
1175     \vspace*{50\p@}%
1176     {\parindent \z@ \centering
1177     \ptifont
1178     #1\par
1179     \nobreak
1180     \vskip 40\p@
1181     }}%
1182 }

```

`\l@pti` By default, titles are on left:

```

\do@ptitc
\df@ptitc 1183 \let\do@ptitc\l@pti
\do@ptilf 1184 \let\df@ptitc\l@pti
\df@ptilf 1185 \let\do@ptilf\l@pti
\do@ptilt 1186 \let\df@ptilf\l@pti
\df@ptilt 1187 \let\do@ptilt\l@pti
1188 \let\df@ptilt\l@pti

```

## 5.48 The ptc@verse environment

ptc@verse Each parttoc is placed inside a ptc@verse environment. This environment is analog to the standard verse environment and hence defined via two commands: \ptc@verse and \endptc@verse. As it is a list environment, we first define (in a local way) \list, then call \list{} and set some dimensions like \itemsep, \itemindent, \listparindent, \@centercr, \itemindent, \partopsep, \topsep. \parsep is set to zero if the tight option is active (this reduces the spacing between the lines). \parskip is set to zero if the k-tight option is active (this reduces the spacing between the lines). Both margins are set to \ptcindent. \endptc@verse terminates the list and discourages a page break.

```

\listparindent
\topsep 1189 \def\ptc@verse{\let\=\@centercr
\parsep 1190 \list{}\itemsep\z@
\parskip 1191 \itemindent \z@
\partopsep 1192 \listparindent \itemindent
\ptcindent 1193 \partopsep\z@
\iftightmtc 1194 \iftightmtc \parsep\z@ \fi
\ifktightmtc 1195 \ifktightmtc \parskip\z@ \fi
1196 \topsep=lex
1197 \leftmargin\ptcindent
1198 \rightmargin\leftmargin\item[]
1199 \def\endptc@verse{\nopagebreak[4]\endlist}

```

## 5.49 The part level mini-tables: \parttoc, \partlof, and \partlot

\parttoc These commands are essentially similar to the \minitoc command, except that they should be placed after a \part command to produce a parttoc, a partlof or a partlot, and the formatting is different and depends of the availability of the \chapter command (for the fonts and the horizontal rules). The code is very similar. The \partlof and \partlot commands are siblings of the \parttoc command. Note that \parttoc, \partlof and \partlot use page styles, because \beforepart... and \afterpart... commands imply usually a \clear[double]page command, and hence \markboth{...}{...} must be called.

### 5.49.1 The \parttoc command

\parttoc This command must be used after \part if you need a parttoc (no automatic parttoc). \parttoc@ First, \parttoc detects the presence of its optional argument, and uses its default value, d, if it is missing. Then, \parttoc@ is called with the effective position as argument:

```
1200 \def\parttoc{\@ifnextchar[\parttoc@]{\parttoc@d}}
```

`\parttoc@` The `\parttoc@` macro does the real work. It first sets the flag `\if@parttoc@used@` (for  
`\if@parttoc@used@` a coherence hint) and checks if long extensions are used or not (to create the name of the  
`\if@longextensions@` parttoc file):

```
\@tocfile
\Thepart 1201 \def\parttoc@[#1]{%
1202 \global\@parttoc@used@true
1203 \if@longextensions@%
1204 \def\@tocfile{ptc\Thepart}%
1205 \else
1206 \def\@tocfile{P\Thepart}%
1207 \fi
```

`\mtc@CkFile` Then, we check the presence of the parttoc file and give a warning if it is not here:

```
\if@mtc@FE
\@tocfile 1208 \mtc@CkFile{\jobname.\@tocfile}
1209 \if@mtc@FE
1210 \PackageInfo{minitoc}%
1211 {\jobname.\@tocfile\space is empty}
1212 \@mtc@empty@parttoc@true
1213 \else
```

`\beforeparttoc` If the parttoc file is present, we can insert it, but we must add some presentation code:  
first, `\beforeparttoc`, of course:

```
1214 \beforeparttoc
```

`\mtc@markboth` If `\chapter` is defined, we just set the page marks with the parttoc title and set the page  
`\@mkboth` style:

```
\thispageparttocstyle
\ptctitle 1215 \@ifundefined{chapter}{}{%
1216 \global\let\mtc@markboth\markboth
1217 \global\let\@mkboth\markboth
1218 \thispageparttocstyle
1219 \mtc@markboth{\uppercase{\ptctitle}}{\uppercase{\ptctitle}}%
1220 }%
```

`\do@ptitc` A `samepage` environment is begun, then the argument is treated to set the position of the  
`\e@pti` parttoc title. If the title string is empty, this forces the positionning.

```
\n@pti
\c@pti 1221 \relax\begin{samepage}%
\l@pti 1222 \if #1e\let\do@ptitc\e@pti
\r@pti 1223 \else\if #1n\let\do@ptitc\n@pti
\df@pti 1224 \else\if #1c\let\do@ptitc\c@pti
\mtc@CkStr 1225 \else\if #1l\let\do@ptitc\l@pti
\ptctitle 1226 \else\if #1r\let\do@ptitc\r@pti
1227 \else\if #1d\let\do@ptitc\df@ptitc
\if@mtc@FE 1228 \fi\fi\fi\fi\fi
samepage 1229 \mtc@CkStr{\ptctitle}\if@mtc@FE \let\do@ptitc\e@pti\relax\fi
```

`\raggedright` We adjust some formatting parameters and avoid a page break between the title and the  
`\parskip` parttoc, then we set the font:  
`\ptcfont`

```

1230      \raggedright
1231      \parskip=\z@%
1232      \reset@font\ptcfont%
1233      \parindent=\z@%
1234      \nopagebreak[4]%
1235      \kern-0.8\baselineskip\nopagebreak[4]%
1236      \par\noindent
1237      \nopagebreak[4]%

```

`\ptc@rule` The parttoc title is set in a tabular environment (to inhibit a page break between the title  
`tabular` and the top rule), with a rule at its bottom if necessary. This rule is an `\hline`. It is the  
`\ptifont` top rule of the parttoc.

```

\do@ptitc
  \mtc@v 1238      \ifx\ptc@rule\relax
\ptctitle 1239      \begin{tabular}{@{}p{\columnwidth}@{}}
  \hline 1240      \reset@font\ptifont\do@ptitc{\mtc@v\ptctitle}\\
1241      \end{tabular}%
1242      \else
1243      \begin{tabular}{@{}p{\columnwidth}@{}}
1244      \reset@font\ptifont\do@ptitc{\mtc@v\ptctitle}\\ \hline
1245      \end{tabular}%
1246      \fi

```

`\mtc@zrule` Then, we adjust the position under the top rule and set the indentation and some  
`\mtc@BBR` formatting parameters:

```

\ptcindent
1247      \nopagebreak[4]\null\leavevmode\mtc@zrule\\*[-\baselineskip]\mtc@BBR
1248      \leftmargin\ptcindent
1249      \rightmargin\ptcindent
1250      \itemindent=\z@\labelwidth=\z@%
1251      \labelsep=\z@\listparindent=\z@%

```

`ptc@verse` We enter in a `ptc@verse` environment to format the parttoc. The toc depth is forced  
`\c@tocdepth` (locally) to `parttocdepth`. A little trick is necessary to adjust the position. A blank line  
`\c@parttocdepth` is necessary to avoid a negative indentation.

```

\mtc@BBR
1252      \begin{ptc@verse}\c@tocdepth=\c@parttocdepth%
1253      \leavevmode\\ \mtc@BBR
1254      \vskip -.5\baselineskip

```

```

\ptc@pgno  If the contents lines must have no numbers, we replace the macro \@dottedtocline
\@dottedtocline with its undotted version. For chapter-level entries, we must invoke \l@chapter ig-
\@undottedtocline noring the page number argument. A hook (redefinissable coommand) is added, and
\mtc@hook@beforeinputfile the formatting settings coming from \mtcsetformat are activated via \ptc@setform.
\ptc@setform Then the parttoc file is inserted, followed by a strut, and the ptc@verse environment is
\@tocfile terminated.
\mtc@strut
ptc@verse 1255 \beginngroup
1256 \makeatletter
1257 \@ifundefined{ptc@pgno}%
1258 {\let\@dottedtocline\@undottedtocline}{}
1259 \@ifundefined{ptc@pgno}%
1260 {\let\l@chapter@SVPN\l@chapter%
1261 \def\l@chapter##1##2{\l@chapter@SVPN{##1}{\hbox{}}}}{}
1262 \@filesfalse\mtc@hook@beforeinputfile
1263 \ptc@setform
1264 \@input{\jobname.\@tocfile}
1265 \vspace{-1ex} \vspace{-1\baselineskip}
1266 \leavevmode\mtc@strut
1267 \global\@nobreakfalse\endgroup
1268 \end{ptc@verse}%

\ptc@rule The final part is just to add the bottom rule, if necessary, a possible page break (if
\mtc@zrule \chapter is not defined), and \afterparttoc.
samepage
\afterparttoc 1269 \kern-1.\baselineskip%
1270 \nopagebreak[4]\ptc@rule\null\leavevmode\\%
1271 \vskip-1.0\baselineskip\mtc@zrule\end{samepage}%
1272 \par\@ifundefined{chapter}{\pagebreak[1]\vspace*{-1ex}}%
1273 \afterparttoc\fi%

```

### 5.49.2 The \partlof command

\partlof This command must be used after \part if you need a partlof (no automatic partlof). First, \partlof detects the presence of its optional argument, and uses its default value, d, if it is missing. Then, \partlof@ is called with the effective position as argument:

```
1274 \def\partlof{\@ifnextchar[{\partlof@}{\partlof@d]}}
```

\partlof@ The \partlof@ macro does the real work. It first sets the flag \if@partlof@used@ (for a coherence hint) and checks if long extensions are used or not (to create the name of the partlof file):

```

\@tocfile
\Thepart 1275 \def\partlof@[#1]{%
1276 \global\@partlof@used@true

```

```

1277 \if@longextensions@%
1278 \def\tocfile{plf\Thepart}%
1279 \else
1280 \def\tocfile{G\Thepart}%
1281 \fi

```

\mtc@CkFile Then, we check the presence of the partlof file and give a warning if it is not here:

```

\if@mtc@FE
\@tocfile 1282 \mtc@CkFile{\jobname.\@tocfile}
1283 \if@mtc@FE
1284 \PackageInfo{minitoc}%
1285 {\jobname.\@tocfile\space is empty}
1286 \@mtc@empty@partlof@true
1287 \else

```

\beforepartlof If the partlof file is present, we can insert it, but we must add some presentation code: first, \beforepartlof, of course:

```

1288 \beforepartlof

```

\mtc@markboth If \chapter is defined, we just set the page marks with the partlof title and set the page style:

```

\@mkboth
\thispagepartlofstyle
\plftitle 1289 \@ifundefined{chapter}{}{}%
1290 \global\let\mtc@markboth\markboth
1291 \global\let\@mkboth\markboth
1292 \thispagepartlofstyle
1293 \mtc@markboth{\uppercase{\plftitle}}{\uppercase{\plftitle}}%
1294 }%

```

\do@ptilf A samepage environment is begun, then the argument is treated to set the position of the partlof title. If the title string is empty, this forces the positionning.

```

\@pti
\n@pti
\c@pti 1295 \relax\begin{samepage}%
\l@pti 1296 \if #1e\let\do@ptilf\@pti
\l@pti 1297 \else\if #1n\let\do@ptilf\n@pti
\l@pti 1298 \else\if #1c\let\do@ptilf\c@pti
\l@pti 1299 \else\if #1l\let\do@ptilf\l@pti
\l@pti 1300 \else\if #1r\let\do@ptilf\r@pti
\l@pti 1301 \else\if #1d\let\do@ptilf\df@ptilf
\l@pti 1302 \fi\fi\fi\fi\fi\fi
samepage 1303 \mtc@CkStr{\plftitle}\if@mtc@FE \let\do@ptilf\@pti\relax\fi

```

`\raggedright` We adjust some formatting parameters and avoid a page break between the title and the  
`\parskip` parttoc, then we set the font:  
`\plffont`

```
1304      \raggedright
1305      \parskip=\z@%
1306      \reset@font\plffont%
```

`\plf@rule` The parttoc title is set in a tabular environment (to inhibit a page break between the title  
`tabular` and the top rule), with a rule at its bottom if necessary. This rule is an `\hline`. It is the  
`\ptifont` top rule of the partlof.

```
\do@ptilf
  \mtc@v 1307      \parindent=\z@%
\plftitle 1308      \nopagebreak[4]%
  \hline 1309      \kern-0.8\baselineskip\nopagebreak[4]%
          1310      \par\noindent
          1311      \ifx\plf@rule\relax
          1312      \begin{tabular}{@{}p{\columnwidth}@{}}
          1313      \reset@font\ptifont\do@ptilf{\mtc@v\plftitle}\\
          1314      \end{tabular}%
          1315      \else
          1316      \begin{tabular}{@{}p{\columnwidth}@{}}
          1317      \reset@font\ptifont\do@ptilf{\mtc@v\plftitle}\\
          1318      \mtc@hstrut\\
          1319      \end{tabular}%
          1320      \fi
```

`\mtc@zrule` Then, we adjust the position under the top rule and set the indentation and some  
`\mtc@BBR` formatting parameters:

```
\ptcindent
          1321      \nopagebreak[4]\null\leavevmode\mtc@zrule\\*[-\baselineskip]\mtc@BBR
          1322      \leftmargin\ptcindent
          1323      \rightmargin\ptcindent
          1324      \itemindent=\z@\labelwidth=\z@%
          1325      \labelsep=\z@\listparindent=\z@%
```

`ptc@verse` We enter in a `ptc@verse` environment to format the partlof. If necessary, the toc depth  
`\mtc@BBR` is forced (locally) to `partlofdepth`. A little trick is necessary to adjust the position. A  
blank line is necessary to avoid a negative indentation.

```
1326      \begin{ptc@verse}%
1327      \@ifundefined{c@lofdepth}{}{\c@tocdepth=\c@partlofdepth}
1328      \ifnum\c@tocdepth<1\relax\c@tocdepth=1\fi
1329      \leavevmode\\
1330      \vskip -.5\baselineskip
```

`\plf@pgno` If the contents lines must have no numbers, we replace the macro `\@dottedtocline`  
`\@dottedtocline` with its undotted version. A hook is added, and the formatting settings coming from  
`\@undottedtocline` `\mtcsetformat` are activated via `\plf@setform`. Then the partlof file is inserted,  
`\mtc@hook@beforeinputfile` followed by a strut, and the `ptc@verse` environment is terminated.  
`\plf@setform`  
`\@tocfile` 1331 `\beginngroup`  
`\mtc@strut` 1332 `\makeatletter`  
`ptc@verse` 1333 `\@ifundefined{plf@pgno}%`  
1334 `{\let\@dottedtocline\@undottedtocline}\}`  
1335 `\@fileswfalse\mtc@hook@beforeinputfile`  
1336 `\plf@setform`  
1337 `\@input{\jobname.\@tocfile}`  
1338 `\vspace{-1ex} \vspace{-0\baselineskip}`  
1339 `\leavevmode\mtc@strut`  
1340 `\global\@nobreakfalse\endgroup`  
1341 `\end{ptc@verse}%`

`\plf@rule` The final part is just to add the bottom rule, if necessary, a possible page break (if  
`\mtc@zrule` `\chapter` is not defined), and `\afterpartlof`. The blank line (`\`) is essential.  
`samepage`  
`\afterpartlof` 1342 `\kern-1.\baselineskip%`  
1343 `\nopagebreak[4]\plf@rule\null\leavevmode\%`  
1344 `\vskip-1.0\baselineskip\mtc@zrule\end{samepage}%`  
1345 `\par\@ifundefined{chapter}{\pagebreak[1]\vspace*{-1ex}}%`  
1346 `\afterpartlof\fi}`

### 5.49.3 The `\partlot` command

`\partlot` This command must be used after `\part` if you need a partlot (no automatic partlot). First,  
`\partlot@` `\partlot` detects the presence of its optional argument, and uses its default value, `d`, if it  
is missing. Then, `\partlot@` is called with the effective position as argument:

```
1347 \def\partlot{\@ifnextchar[{\partlot@}{\partlot@d}}
```

`\partlot@` The `\partlot@` macro does the real work. It first sets the flag `\if@partlot@used@` (for  
`\if@partlot@used@` a coherence hint) and checks if long extensions are used or not (to create the name of the  
`\if@longextensions@` partlot file):  
`\@tocfile`  
`\Thepart` 1348 `%`  
1349 `\def\partlot@[#1]{%`  
1350 `\global\@partlot@used@true`  
1351 `\if@longextensions%`  
1352 `\def\@tocfile{plt\Thepart}%`  
1353 `\else`  
1354 `\def\@tocfile{U\Thepart}%`  
1355 `\fi`



`\mtc@CkFile` Then, we check the presence of the partlot file and give a warning if it is not here:

```

\if@mtc@FE
  \@tocfile 1356      \mtc@CkFile{\jobname.\@tocfile}
              1357      \if@mtc@FE
              1358      \PackageInfo{minitoc}%
              1359          {\jobname.\@tocfile\space is empty}
              1360      \@mtc@empty@partlof@true
              1361      \else

```

`\beforepartlot` If the partlot file is present, we can insert it, but we must add some presentation code: first, `\beforepartlot`, of course:

```

1362      \beforepartlot

```

`\mtc@markboth` If `\chapter` is defined, we just set the page marks with the partlot title and set the page style:

```

\@mkboth
\thispagepartlotstyle
  \plttitle 1363      \@ifundefined{chapter}{}{%
              1364      \global\let\mtc@markboth\markboth
              1365      \global\let\@mkboth\markboth
              1366      \thispagepartlotstyle
              1367      \mtc@markboth{\uppercase{\plttitle}}{\uppercase{\plttitle}}%
              1368      }%

```

`\do@ptilt` A `samepage` environment is begun, then the argument is treated to set the position of the partlof title. If the title string is empty, this forces the positionning.

```

\@e@pti
\@n@pti
\@c@pti 1369      \relax\begin{samepage}%
\@l@pti 1370      \if #1e\let\do@ptilt\@e@pti
\@r@pti 1371      \else\if #1n\let\do@ptilt\@n@pti
\@df@pti 1372      \else\if #1c\let\do@ptilt\@c@pti
\mtc@CkStr 1373      \else\if #1l\let\do@ptilt\@l@pti
\plttitle 1374      \else\if #1r\let\do@ptilt\@r@pti
\if@mtc@FE 1375      \else\if #1d\let\do@ptilt\@df@ptilt
samepage 1376      \fi\fi\fi\fi\fi\fi
          1377      \mtc@CkStr{\plttitle}\if@mtc@FE \let\do@ptilt\@e@pti\relax\fi

```

`\raggedright` We adjust some formatting parameters and avoid a page break between the title and the partlot, then we set the font:

```

\pltfont
          1378      \raggedright
          1379      \parskip=\z@%
          1380      \reset@font\pltfont%
          1381      \parindent=\z@%
          1382      \nolinebreak[4]%
          1383      \kern-0.8\baselineskip\nolinebreak[4]%
          1384      \par\noindent

```

`\ptc@rule` The partlot title is set in a tabular environment (to inhibit a page break between the title and the top rule), with a rule at its bottom if necessary. This rule is an `\hline`. It is the top rule of the partlot.

```

\do@ptilt
  \mtc@v 1385      \ifx\plt@rule\relax
\plttitle 1386      \begin{tabular}{@{}p{\columnwidth}@{}}
  \hline 1387      \reset@font\ptifont\do@ptilt{\mtc@v\plttitle}\\
          1388      \end{tabular}%
          1389      \else
          1390      \begin{tabular}{@{}p{\columnwidth}@{}}
          1391      \reset@font\ptifont\do@ptilt{\mtc@v\plttitle}\\ \hline
          1392      \mtc@hstrut\\
          1393      \end{tabular}%
          1394      \fi

```

`\mtc@zrule` Then, we adjust the position under the top rule and set the indentation and some formatting parameters:

```

\mtc@BBR
\ptcindent
          1395      \nopagebreak[4]\null\leavevmode\mtc@zrule\\*[-\baselineskip]\mtc@BBR
          1396      \leftmargin\ptcindent
          1397      \rightmargin\ptcindent
          1398      \itemindent=\z@\labelwidth=\z@%
          1399      \labelsep=\z@\listparindent=\z@%

```

`ptc@verse` We enter in a `ptc@verse` environment to format the parttoc. If necessary, the toc depth is forced (locally) to `partlotdepth`. A little trick is necessary to adjust the position.

```

          1400      \begin{ptc@verse}%
          1401      \@ifundefined{c@lotdepth}{}{\c@tocdepth=\c@partlotdepth}
          1402      \ifnum\c@tocdepth<1\relax\c@tocdepth=1\fi
          1403      \leavevmode\\ \mtc@BBR
          1404      \vskip -.5\baselineskip

```

`\plt@pgno` If the contents lines must have no numbers, we replace the macro `\@dottedtocline` with its undotted version. A hook is added, and the formatting settings coming from `\@undottedtocline` `\mtcsetformat` are activated via `\plt@setform`. Then the partlot file is inserted, followed by a strut, and the `ptc@verse` environment is terminated.

```

\plt@setform
  \@tocfile 1405 \begingroup
\mtc@strut 1406   \makeatletter
ptc@verse 1407   \ifundefined{plt@pgno}%
          1408   {\let\@dottedtocline\@undottedtocline}%
          1409   \@filesfalse\mtc@hook@beforeinputfile
          1410   \plt@setform
          1411   \@input{\jobname.\@tocfile}
          1412   \vspace{-1ex} \vspace{-0\baselineskip}
          1413   \leavevmode\mtc@strut

```

```

1414 \global\@nobeakfalse\endgroup
1415 \end{ptc@verse}%

\ptc@rule The final part is just to add the bottom rule, if necessary, a possible page break (if
\mtc@zrule \chapter is not defined), and \afterpartlot. The blank line (\\) is essential.
samepage
\afterpartlot 1416 \kern-1.\baselineskip%
1417 \nopagebreak[4]\plt@rule\null\leavevmode\\%
1418 \vskip-1.0\baselineskip\mtc@zrule\end{samepage}%
1419 \par\@ifundefined{chapter}{\pagebreak[1]\vspace*{-1ex}}%
1420 \afterpartlot\fi}

```

## 5.50 Auxiliary commands for printing parttocs

\@dottedtocline The following auxiliary commands are used in the printing of parttocs. Note that \l@xpart uses a huge depth to inhibit the printing of its contents line (except if you cheat). These command are similar to \l@section, only the arguments have been altered:

```

\pchapter
\psection 1421 \def\l@xpart{\@dottedtocline{\@M}{1.0em}{2.3em}}
1422 \def\l@pchapter{\@dottedtocline{1}{1.0em}{2.3em}}
1423 \def\l@psection{\@dottedtocline{2}{1.0em}{2.3em}}
1424 \def\pchapter{pchapter}
1425 \def\psection{psection}

```

## 5.51 Patching the \part command, continued

\sv@part We patch both branches of the \part command: \@part (unstarred \part) and \@spart (\part\*). We add the incrementation of the ptc counter to both branches. In the unstarred branch, we add xpart entries in the TOC, the LOF and the LOT. In the starred branch, we add a \partbegin line in the TOC. This command is just a marker and does nothing real (\relax).

```

\mtc@svpart
\@part
\addtocontents 1426 \let\sv@part\mtc@svpart
\sv@spart
\ptc@spart
\@spart 1427 \def\@part[#1]#2{\sv@part[#1]{#2}\relax
\stepcounter 1428 \addcontentsline{lof}{xpart}{#1}%
\partbegin 1429 \addcontentsline{lot}{xpart}{#1}%
\partend 1430 \addcontentsline{toc}{xpart}{#1}%
1431 \stepcounter{ptc}}
1432 \let\sv@spart\@spart
1433 \def\@spart{\stepcounter{ptc}\sv@spart}
1434 \let\ptc@spart\@spart
1435 \def\@spart{\addtocontents{toc}{\protect\partend}\ptc@spart}

```

```

1436 \def\@spart{\addtocontents{toc}{\protect\partbegin}\ptc@spart}
1437 \let\partend\relax
1438 \let\partbegin\relax

```

## 5.52 The \doparttoc command and its siblings

`\doparttoc` The `\doparttoc` command works like the `\dominitoc` command, `\dopartlof` like `\dominiloof` and `\dopartlot` like `\dominilot`.

`\@doparttoc` The `\doparttoc` command extracts information from the `.toc` file and creates the `\PTC@next` `.ptc(N)` files (`.ptc` becomes `.P` on MS-DOS).

```

1439 \def\@doparttoc#1{%
1440   \makeatletter
1441   \setcounter{ptc}{0}%
1442   \PTC@next#1.toc\relax\}\setcounter{ptc}{0}}%

```

`\@dopartlof` The `\dopartlof` command extracts information from the `.lof` file and creates the `\PLF@next` `.plf(N)` files (`.plf` becomes `.G` on MS-DOS).

```

1443 \def\@dopartlof#1{%
1444   \makeatletter
1445   \setcounter{ptc}{0}%
1446   \PLF@next#1.lof\relax\}\setcounter{ptc}{0}}%

```

`\@dopartlot` The `\dopartlot` command extracts information from the `.lot` file and creates the `\PLT@next` `.plt(N)` files (`.plt` becomes `.U` on MS-DOS).

```

1447 \def\@dopartlot#1{%
1448   \makeatletter
1449   \setcounter{ptc}{0}%
1450   \PLT@next#1.lot\relax\}\setcounter{ptc}{0}}%

```

`\doparttoc` We define the user macros, who detect the optional argument:

`\dopartlof`

```

\dopartlot 1451 \def\doparttoc{\@ifnextchar[{\doparttoc@}{\doparttoc@[1]}}
1452 \def\dopartlof{\@ifnextchar[{\dopartlof@}{\dopartlof@[1]}}
1453 \def\dopartlot{\@ifnextchar[{\dopartlot@}{\dopartlot@[1]}}

```

`\doparttoc@` We treat the optional argument of `\doparttoc` (it becomes the default position for titles of parttocs) and flag this macro as used:

```

\dftitc
\etiti 1454 \def\doparttoc@[#1]{%
\npti
\cpti
\lpti
\rpti

```

```

1455 \global\@doparttoc@used@true
1456 \if #1e\let\df@ptitc\e@pti%
1457 \else\if #1n\let\df@ptitc\n@pti%
1458 \else\if #1c\let\df@ptitc\c@pti%
1459 \else\if #1l\let\df@ptitc\l@pti%
1460 \else\if #1r\let\df@ptitc\r@pti%
1461 \fi\fi\fi\fi\fi%
1462 \@@doparttoc}

```

\dopartlof@ We treat the optional argument of \dopartlof (it becomes the default position for titles of partlofs) and flag this macro as used:

```

\dof@ptilf
  \e@pti 1463 \def\dopartlof@[#1]{%
  \n@pti 1464 \global\@dopartlof@used@true
  \c@pti 1465 \if #1e\let\df@ptilf\e@pti%
  \l@pti 1466 \else\if #1n\let\df@ptilf\n@pti%
  \r@pti 1467 \else\if #1c\let\df@ptilf\c@pti%
          1468 \else\if #1l\let\df@ptilf\l@pti%
          1469 \else\if #1r\let\df@ptilf\r@pti%
          1470 \fi\fi\fi\fi\fi%
          1471 \@@dopartlof}

```

\dopartlot@ We treat the optional argument of \dopartlot (it becomes the default position for titles of partlofs) and flag this macro as used:

```

\dof@ptilt
  \e@pti 1472 \def\dopartlot@[#1]{%
  \n@pti 1473 \global\@dopartlot@used@true
  \c@pti 1474 \if #1e\let\df@ptilt\e@pti%
  \l@pti 1475 \else\if #1n\let\df@ptilt\n@pti%
  \r@pti 1476 \else\if #1c\let\df@ptilt\c@pti%
          1477 \else\if #1l\let\df@ptilt\l@pti%
          1478 \else\if #1r\let\df@ptilt\r@pti%
          1479 \fi\fi\fi\fi\fi%
          1480 \@@dopartlot}

```

\@@doparttoc These macros invoke the \@dopart... commands to create the mini-table file, then close the file descriptor.

```

\dof@ptilt
  \tf@mtc 1481 \def\@@doparttoc{\@doparttoc{\jobname}\immediate\closeout\tf@mtc}
          1482 \def\@@dopartlof{\@dopartlof{\jobname}\immediate\closeout\tf@mtc}
          1483 \def\@@dopartlot{\@dopartlot{\jobname}\immediate\closeout\tf@mtc}

```

### 5.52.1 Processing macros for the parttocs

```

\PTC@next Processing the next entry in the list and remove it from the head of the list:
\PTC@list
\PTC@loop

```

```

1484 \def\PTC@next#1\relax#2\{\%
1485   \edef\PTC@list{#2}%
1486   \PTC@loop{#1}}

```

```

\PTC@toc    Check if the list is empty:
\PTC@list
\PTC@explist 1487 \def\PTC@toc{%
1488   \ifx\PTC@list\@empty\else\expandafter\PTC@explist\fi}

```

```

\PTC@contentsline The macro \PTC@contentsline analyses the lines read from the TOC file and detects
\part             interesting keywords. If \part is found, ptc is incremented and a new partlof file is
\theptc          created.
\tf@mtc
\ptcname 1489 \def\PTC@contentsline#1#2#3#4{% %%HO/DV: 4 instead of 3 parameters
\MTC@WriteContentsLine 1490   \expandafter\ifx\csname #1\endcsname\part
1491     \stepcounter{ptc}%
1492     \if@longextensions@%
1493       \PackageInfo{minitoc}%
1494       {Writing\space\jobname.ptc\theptc\@gobble}%
1495       \def\ptcname{\jobname.ptc\theptc}%
1496     \else
1497       \PackageInfo{minitoc}%
1498       {Writing\space\jobname.P\theptc\@gobble}%
1499       \def\ptcname{\jobname.P\theptc}%
1500     \fi
1501     \immediate\closeout\tf@mtc
1502     \immediate\openout\tf@mtc=\ptcname
1503   \fi
1504   \expandafter\ifx\csname #1\endcsname\starpart\relax
1505     \stepcounter{ptc}%

```

```

\if@longextensions@ We test if long or short extensions are used, to build the name of the mini-table file, then
\ptcname            open it:

```

```

1506   \if@longextensions@%
1507     \PackageInfo{minitoc}%
1508     {Writing\space\jobname.ptc\theptc}%
1509     \def\ptcname{\jobname.ptc\theptc}%
1510   \else
1511     \PackageInfo{minitoc}%
1512     {Writing\space\jobname.P\theptc}%
1513     \def\ptcname{\jobname.P\theptc}%
1514   \fi
1515   \immediate\closeout\tf@mtc
1516   \immediate\openout\tf@mtc=\ptcname
1517 \fi

```

`\mtc@toks` The token register `\mtc@toks` is used to pass the entry to `\MTC@WriteContentsline`:

```
1518 \mtc@toks{\noexpand\leavevmode #2}%
```

`\MTC@WriteContentsline` Now, we filter the relevant contents lines; this code extracts and writes info for chapters, sections, etc.:

```
\chapter
\pchapter
\section 1519 \expandafter\ifx\csname #1\endcsname\chapter
\coffee 1520 \MTC@WriteContentsline{#1}{ptcC}{#3}{#4}%
\subsection 1521 \fi
\subsubsection 1522 \expandafter\ifx\csname #1\endcsname\pchapter
\paragraph 1523 \MTC@WriteContentsline{#1}{ptcC}{#3}{#4}%
\subparagraph 1524 \fi
1525 \expandafter\ifx\csname #1\endcsname\section
1526 \MTC@WriteContentsline{#1}{ptcS}{#3}{#4}%
1527 \fi
1528 \expandafter\ifx\csname #1\endcsname\coffee
1529 \MTC@WriteCoffeeline{#1}{#3}%
1530 \fi
1531 \expandafter\ifx\csname #1\endcsname\subsection
1532 \MTC@WriteContentsline{#1}{ptcSS}{#3}{#4}%
1533 \fi
1534 \expandafter\ifx\csname #1\endcsname\subsubsection
1535 \MTC@WriteContentsline{#1}{ptcSSS}{#3}{#4}%
1536 \fi
1537 \expandafter\ifx\csname #1\endcsname\paragraph
1538 \MTC@WriteContentsline{#1}{ptcP}{#3}{#4}%
1539 \fi
1540 \expandafter\ifx\csname #1\endcsname\subparagraph
1541 \MTC@WriteContentsline{#1}{ptcSP}{#3}{#4}%
1542 \fi
```

`\MTC@WriteContentsline` And for the starred sectionning commands:

```
\starchapter
\starsection 1543 \expandafter\ifx\csname #1\endcsname\starchapter
\starsubsection 1544 %%HO: the following line should be disabled: \stepcounter{ptc}%
\starsubsubsection 1545 \MTC@WriteContentsline{#1}{ptcC}{#3}{#4}%
\starparagraph 1546 \fi
\starsubparagraph 1547 \expandafter\ifx\csname #1\endcsname\starsection
1548 \MTC@WriteContentsline{#1}{ptcS}{#3}{#4}%
1549 \fi
1550 \expandafter\ifx\csname #1\endcsname\starsubsection
1551 \MTC@WriteContentsline{#1}{ptcSS}{#3}{#4}%
1552 \fi
1553 \expandafter\ifx\csname #1\endcsname\starsubsubsection
1554 \MTC@WriteContentsline{#1}{ptcSSS}{#3}{#4}%
1555 \fi
1556 \expandafter\ifx\csname #1\endcsname\starparagraph
1557 \MTC@WriteContentsline{#1}{ptcP}{#3}{#4}%
1558 \fi
1559 \expandafter\ifx\csname #1\endcsname\starsubparagraph
```

```

1560     \MTC@WriteContentsline{#1}{ptcSP}{#3}{#4}%
1561     \fi
1562 }

```

\PTC@explist The loop to read the lines of the TOC file; expands the list of entries and call \PTC@next to process the first one:  
 \PTC@next  
 \PTC@list

```

1563 \def\PTC@explist{\expandafter\PTC@next\PTC@list\}

```

\PTC@loop If an entry is found, loop through line by line, looking for interesting entries. Otherwise, process the next entry in the list.  
 \PTC@toc  
 \PTC@read

```

1564 \def\PTC@loop#1{\openin\@inputcheck#1\relax
1565   \ifeof\@inputcheck
1566     \PackageWarning{minitoc}%
1567       {No file #1\MessageBreak PARTTOCS NOT PREPARED}%
1568     \expandafter\PTC@toc
1569   \else
1570     \PackageInfo{minitoc}%
1571       {PREPARING PARTTOCS FROM #1}%
1572     \expandafter\PTC@read\fi}

```

\PTC@read Read the next entry of the .toc file.  
 \PTC@line

```

1573 \def\PTC@read{%
1574   \read\@inputcheck to\PTC@line

```

\PTC@test The ..... make sure that \PTC@test has sufficient arguments:  
 \PTC@line

```

1575   \expandafter\PTC@test\PTC@line.....\PTC@% %%H0: . added
1576  }%

```

\PTC@test The \PTC@test macro finds the “interesting” commands in the TOC file, mainly to delimit parts:

```

1577 %%H0/BJ: now patch \PTC@test,
1578 %%H0/BJ: call \PTC@contentsline with 4 instead of 3 parameters

```

\PTC@contentsline Look at the first token of the line. If it is an interesting entry, process it. If it is \@input, add the file to the list. Otherwise ignore. Go around the loop if not at end of file. Finally process the next file in the list.  
 \mtc@string  
 \PTC@list  
 \PTC@toc

```

1579 \long\def\PTC@test#1#2#3#4#5#6\PTC@{% %%H0: #6 added
1580   \ifx#1\contentsline
1581     \let\mtc@string\string
1582     \PTC@contentsline{#2}{#3}{#4}{#5}%

```



```

1583    %%HO/DV: 4 instead of 3 parameters
1584    \let\mtc@string\relax
1585    \else\ifx#1\@input
1586        \edef\PTC@list{\PTC@list#2\relax}%
1587    \else\ifx#1\partend
1588        \immediate\closeout\tf@mtc
1589        \immediate\openout\tf@mtc=\jobname.mtc
1590    \else\ifx#1\partbegin
1591        \addtocounter{ptc}{-1}%
1592    \fi\fi\fi\fi
1593    \ifeof\@inputcheck\expandafter\PTC@toc
1594    \else\expandafter\PTC@read\fi}%

```

### 5.52.2 Processing macros for the partlofs

`\PLF@next` Processing the next entry in the list and remove it from the head of the list:

`\PLF@list`

```

\PLF@loop 1595 \def\PLF@next#1\relax#2\{\%
1596    \edef\PLF@list{#2}%
1597    \PLF@loop{#1}}

```

`\PLF@lof` Check if the list is empty:

`\PLF@list`

```

\PLF@explist 1598 \def\PLF@lof{%
1599    \ifx\PLF@list\@empty\else\expandafter\PLF@explist\fi}

```

`\PLF@contentsline` The macro `\PLF@contentsline` analyses the lines read from the LOF file and detects interesting keywords. If `\part` is found, `ptc` is incremented and a new partlof file is created.

`\tf@mtc`

```

\plfname 1600 \def\PLF@contentsline#1#2#3#4{% %%HO: #4 added

```

```

\PLF@WriteContentsLine 1601    \expandafter\ifx\csname #1\endcsname\part
1602        \stepcounter{ptc}%

```

`\if@longextensions@` We test if long or short extensions are used, to build the name of the mini-table file, then open it:

`\plfname`

```

1603    \if@longextensions@
1604        \PackageInfo{minitoc}%
1605            {Writing\space\jobname.plf\theptc}%
1606        \def\plfname{\jobname.plf\theptc}%
1607    \else
1608        \PackageInfo{minitoc}%
1609            {Writing\space\jobname.G\theptc}%
1610        \def\plfname{\jobname.G\theptc}%
1611    \fi

```

```

1612 \immediate\closeout\tf@mtc
1613 \immediate\openout\tf@mtc=\plfname
1614 \fi

```

\mtc@toks The token register \mtc@toks is used to pass the entry to \MTC@WriteContentsline:

```

\figure Now, we filter the relevant contents lines:
\mtc@toks
\MTC@WriteContentsline 1615 \expandafter\ifx\csname #1\endcsname\figure
1616 \mtc@toks{\noexpand\leavevmode#2}%
1617 \MTC@WriteContentsline{#1}{\plf}{#3}{#4}%
1618 \fi
1619 }

```

\PLF@explist The loop to read the lines of the LOF file; expands the list of entries and call \PLF@next to process the first one:

```

\PLF@list
1620 \def\PLF@explist{\expandafter\PLF@next\PLF@list\}

```

\PLF@loop If an entry is found, loop through line by line, looking for interesting entries. Otherwise, process the next entry in the list.

```

\PLF@read
1621 \def\PLF@loop#1{\openin\@inputcheck#1\relax
1622 \ifeof\@inputcheck
1623 \PackageWarning{minitoc}%
1624 {No file #1\MessageBreak PARTLOFS NOT PREPARED}%
1625 \expandafter\PLF@lof
1626 \else
1627 \PackageInfo{minitoc}%
1628 {PREPARING PARTLOFS FROM #1}%
1629 \expandafter\PLF@read\fi}

```

\PLF@read Read the next entry of the .lof file.

```

\PLF@line
1630 \def\PLF@read{%
1631 \read\@inputcheck to\PLF@line

```

\PLF@test The ..... make sure that \PLF@test has sufficient arguments:

```

\PLF@line
1632 \expandafter\PLF@test\PLF@line.....\PLF@% %%H0: . added
1633 }%

```

\PLF@test The \PLF@test macro finds the “interesting” commands in the LOF file, mainly to delimit parts:

```

1634 %%H0/BJ: now patch \PLF@test,
1635 %%H0/BJ: call \PLF@contentsline with 4 instead of 3 parameters

```

```

\PLF@contentsline Look at the first token of the line. If it is an interesting entry, process it. If it is \@input,
\mtc@string      add the file to the list. Otherwise ignore. Go around the loop if not at end of file. Finally
\PLF@list        process the next file in the list.
\PLF@lof
\PLF@read 1636 \long\def\PLF@test#1#2#3#4#5#6\PLF@{% %%H0: #6 added
\partend 1637 \ifx#1\contentsline
          1638 \let\mtc@string\string
          1639 \PLF@contentsline{#2}{#3}{#4}{#5}% %%H0: #4 added
          1640 \let\mtc@string\relax
          1641 \else\ifx#1\@input
          1642 \edef\PLF@list{\PLF@list#2\relax}%
          1643 \else\ifx#1\partend
          1644 \immediate\closeout\tf@mtc
          1645 \immediate\openout\tf@mtc=\jobname.mtc
          1646 \else\ifx#1\partbegin
          1647 \addtocounter{ptc}{-1}%
          1648 \fi\fi\fi\fi
          1649 \ifeof\@inputcheck\expandafter\PLF@lof
          1650 \else\expandafter\PLF@read\fi}%

```

### 5.52.3 Processing macros for the partlots

\PLT@next Processing the next entry in the list and remove it from the head of the list:

```

\PLT@list
\PLT@loop 1651 \def\PLT@next#1\relax#2\{\%
          1652 \edef\PLT@list{#2}%
          1653 \PLT@loop{#1}}

```

\PLT@lot Check if the list is empty:

```

\PLT@list
\PLT@explist 1654 \def\PLT@lot{%
          1655 \ifx\PLT@list\@empty\else\expandafter\PLT@explist\fi}

```

\PLT@contentsline The macro \PLT@contentsline analyses the lines read from the LOT file and detects interesting keywords. If \part is found, ptc is incremented and a new partlot file is created.

```

\theptc
\tf@mtc
\plname 1656 \def\PLT@contentsline#1#2#3#4{% %%H0: #4 added
\PLT@WriteContentsLine 1657 \expandafter\ifx\csname #1\endcsname\part
          1658 \stepcounter{ptc}%

```

`\if@longextensions@` We test if long or short extensions are used, to build the name of the mini-table file, then  
`\pltname` open it:

```

1659 \if@longextensions@%
1660 \PackageInfo{minitoc}%
1661 {Writing\space\jobname.plt\theptc}%
1662 \def\pltname{\jobname.plt\theptc}%
1663 \else
1664 \PackageInfo{minitoc}%
1665 {Writing\space\jobname.U\theptc}%
1666 \def\pltname{\jobname.U\theptc}%
1667 \fi
1668 \immediate\closeout\tf@mtc
1669 \immediate\openout\tf@mtc=\pltname
1670 \fi

```

`\mtc@toks` The token register `\mtc@toks` is used to pass the entry to `\MTC@WriteContentsline`:

```

\table Now, we filter the relevant contents lines:
\mtc@toks
\MTC@WriteContentsline 1671 \expandafter\ifx\csname #1\endcsname\table
1672 \mtc@toks{\noexpand\leavevmode#2}%
1673 \MTC@WriteContentsline{#1}\plt{#3}{#4}%
1674 \fi
1675 }

```

`\PLT@explist` The loop to read the lines of the LOT file; expands the list of entries and call `\PLT@next`  
`\PLT@next` to process the first one:  
`\PLT@list`

```

1676 \def\PLT@explist{\expandafter\PLT@next\PLT@list\}

```

`\PLT@loop` If an entry is found, loop through line by line, looking for interesting entries. Otherwise,  
`\PLT@lot` process the next entry in the list.  
`\PLT@read`

```

1677 \def\PLT@loop#1{\openin\@inputcheck#1\relax
1678 \ifeof\@inputcheck
1679 \PackageWarning{minitoc}%
1680 {No file #1\MessageBreak PARTLOTS NOT PREPARED}%
1681 \expandafter\PLT@lot
1682 \else
1683 \PackageInfo{minitoc}%
1684 {PREPARING PARTLOTS FROM #1}%
1685 \expandafter\PLT@read\fi}

```

`\PLT@read` Read the next entry of the .lot file.

```

\PLT@line
1686 \def\PLT@read{%
1687 \read\@inputcheck to\PLT@line

```

\PLT@test The ..... make sure that \PLT@test has sufficient arguments:  
 \PLT@line

```
1688 \expandafter\PLT@test\PLT@line.....\PLT@% %%H0: . added
1689 }%
```

\PLT@test The \PLT@test macro finds the “interesting” commands in the LOT file, mainly to delimit parts:

```
1690 %%H0/BJ: now patch \PLT@test,
1691 %%H0/BJ: call \PLT@contentsline with 4 instead of 3 parameters
```

\PLT@contentsline Look at the first token of the line. If it is an interesting entry, process it. If it is \@input, add the file to the list. Otherwise ignore. Go around the loop if not at end of file. Finally  
 \mtc@string process the next file in the list.  
 \PLT@list  
 \PLT@lot

```
\PLT@read 1692 \long\def\PLT@test#1#2#3#4#5#6\PLT@{% %%H0: #6 added
\partend 1693 \ifx#1\contentsline
1694 \let\mtc@string\string
1695 \PLT@contentsline{#2}{#3}{#4}{#5}% %%H0: #4 added
1696 \let\mtc@string\relax
1697 \else\ifx#1\@input
1698 \edef\PLT@list{\PLT@list#2\relax}%
1699 \else\ifx#1\partend
1700 \immediate\closeout\tf@mtc
1701 \immediate\openout\tf@mtc=\jobname.mtc
1702 \else\ifx#1\partbegin
1703 \addtocounter{ptc}{-1}%
1704 \fi\fi\fi\fi
1705 \ifeof\@inputcheck\expandafter\PLT@lot
1706 \else\expandafter\PLT@read\fi}%
```

End of the part level stuff (begun in section 5.45 on page 118):

```
1707 }%
```

## 5.53 Depth counters for sectlofs and sectlots

\AtBeginDocument If the counters lofdepth and lotdepth are defined, we create new counters for the depths of the corresponding mini-tables: sectlofdepth and sectlotdepth. These counters are initialised to 2. This is done after the loading of the packages, in an \AtBeginDocument block:  
 \c@lofdepth  
 \c@lotdepth  
 \newcounter  
 \setcounter

```
1708 \AtBeginDocument{%
1709 \ifundefined{c@lofdepth}{}%
1710 {\newcounter{sectlofdepth}\setcounter{sectlofdepth}{2}}%
```

```

1711 \@ifundefined{c@lotdepth}{}%
1712   {\newcounter{sectlotdepth}\setcounter{sectlotdepth}{2}}%
1713 }

```

## 5.54 Section-level commands

`\if@mtc@chapter@undef@` The section-level commands are defined only if `\chapter` is *not* defined, hence in article-like document classes, and only if `\section` is defined:

```

1714 \if@mtc@chapter@undef@
1715 \if@mtc@section@def@

```

`\firstsectionis` We define the obsolete command `\firstsectionis` (with its harmless warning), the `\adjuststc` counter `stc` of `secttocs`, the `\adjuststc`, `\decrementstc` and `\incrementstc` commands, the depth counter `sectocdepth` and its default value 2 (to include at least the subsections), the horizontal rule `\stc@rule` (rule before/after `secttoc`/`sectlof`/`sectlot`), the indentation (both sides) `\stcindent` for the `secttocs` (with its default values).

```

1716 \def\firstsectionis#1%
1717   {\PackageWarning{minitoc}%
1718    {*** \string\firstsectionis \space is an obsolete command ***}%
1719    \@firstsectionis@used@true}
1720 \newcounter{stc}\setcounter{stc}{0}%
1721 \newcommand{\adjuststc}[1][1]{\addtocounter{stc}{#1}}%
1722 \def\decrementstc{\addtocounter{stc}{-1}}%
1723 \def\incrementstc{\addtocounter{stc}{+1}}%
1724 \newcounter{sectocdepth}\setcounter{sectocdepth}{2}%
1725 \def\stc@rule{\rule[3\p@]{\columnwidth}{.4\p@}\vspace*{2.6\p@}}%
1726 \newlength\stcindent \stcindent=24\p@

```

## 5.55 Fonts commands for secttocs and co.

`\stcfont` We define the fonts commands for the `secttocs`, `sectlofs` and `sectlots` and their titles:

```

\stcSSfont
\stcSSSfont 1727 \def\stcfont{\small\rmfamily\upshape\mdseries} % secttoc
\stcPfont 1728 \def\stcSSfont{\small\rmfamily\upshape\bfseries} % (subsections)
\stcSPfont 1729 \let\stcSSSfont\stcfont % (subsubsections)
\slffont 1730 \let\stcPfont\stcfont % (paragraphs)
\sltfont 1731 \let\stcSPfont\stcfont % (subparagraphs)
\stifont 1732 \let\slffont\stcfont % sectlof (figures)
1733 \let\sltfont\stcfont % sectlot (tables)
1734 \def\stifont{\large\rmfamily\upshape\bfseries} % titles

```

## 5.56 Internal macros for title positionning

Some internal macros for title positionning, from the optional arguments of `\dosecttoc` and `\secttoc` commands (and siblings):

```

\l@sti  Centering, flushleft, flushright or empty titles (with a vertical correction for empty titles,
\c@sti  from Frank MITTELBACH):
\r@sti
\l@sti  1735 \def\c@sti#1{\null\hfill #1\hfill\null}
\c@sti  1736 \def\l@sti#1{\null #1\hfill\null}
\r@sti  1737 \def\r@sti#1{\null\hfill #1\null}
\l@sti  1738 \def\l@sti#1{\vspace{-\baselineskip}}
\c@sti  1739 \def\c@sti#1{\vspace{-\baselineskip}}

\do@sttc  By default, titles are flushleft.
\df@sttc
\do@stilf 1740 \let\do@sttc\l@sti
\df@stilf 1741 \let\df@sttc\l@sti
\do@stilt 1742 \let\do@stilf\l@sti
\df@stilt 1743 \let\df@stilf\l@sti
\l@sti    1744 \let\do@stilt\l@sti
          1745 \let\df@stilt\l@sti

```

## 5.57 The stc@verse environment

`stc@verse` The `stc@verse` environment is a very simple list environment, analog to the standard `verse` environment. Some formatting parameters are adjusted. The `tight/loose` and `k-tight/k-loose` package options are honored.

```

1746 \def\stc@verse{\let\=\@centercr
1747   \list{}\itemsep\z@
1748   \itemindent \z@
1749   \listparindent \itemindent
1750   \partopsep \z@
1751   \iftightmtc \parsep \z@ \fi
1752   \ifktightmtc \parskip \z@ \fi
1753   \topsep \z@
1754   \leftmargin\stcindent
1755   \rightmargin\leftmargin}\item[]}
1756 \def\endstc@verse{\nopagebreak[4]\endlist}

```

## 5.58 The `\secttoc`, `\sectlof`, and `\sectlot` commands

These three commands are very similar.

### 5.58.1 The `\secttoc` command

`\secttoc` The `\secttoc` command must be used after `\section` if you need a secttoc (no automatic secttoc). Its code is similar to the code of `\minitoc` (but simpler). First, `\secttoc` detects the presence of its optional argument, and uses its default value, `d`, if it is missing. Then, `\secttoc@` is called with the effective position as argument:

```
1757 \def\secttoc{\@ifnextchar[{\secttoc@}{\secttoc@d}}
```

`\secttoc@` The `\secttoc@` macro does the real work. It first sets the flag `\if@secttoc@used@` (for a coherence hint) and checks if long extensions are used or not (to create the name of the secttoc file):

```
\@tocfile
\thestc 1758 \def\secttoc@[#1]{%
1759 \global\@secttoc@used@true
1760 \if@longextensions@%
1761 \def\@tocfile{stc\thestc}%
1762 \else
1763 \def\@tocfile{S\thestc}%
1764 \fi
```

`\mtc@CkFile` Then, we check the presence and the emptiness of the secttoc file and give a warning if it is not here or is empty:

```
\@tocfile
1765 \mtc@CkFile{\jobname.\@tocfile}
1766 \if@mtc@FE
1767 \PackageInfo{minitoc}%
1768 {\jobname.\@tocfile\space is empty}
1769 \@mtc@empty@secttoc@true
1770 \else
```

`\beforesecttoc` If the secttoc file is present and not empty, we can insert it, but we must add some presentation code: first, `\beforesecttoc`, of course, and the page style feature:

```
1771 \beforesecttoc
1772 \thispagesecttocstyle
```



```

\do@sttc We begin a samepage environment, then treat the positionning argument. If the title is
  \e@sti empty, we simulate the “e” positionning.
  \n@sti
  \c@sti 1773 %%      \mtc@markboth{\uppercase{\stctitle}}{\uppercase{\stctitle}}%
  \l@sti 1774      \relax\begin{samepage}%
  \r@sti 1775      \if #1e\let\do@sttc\e@sti
  \df@sti 1776      \else\if #1n\let\do@sttc\n@sti
\mtc@CkStr 1777      \else\if #1c\let\do@sttc\c@sti
\stctitle 1778      \else\if #1l\let\do@sttc\l@sti
\if@mtc@FE 1779      \else\if #1r\let\do@sttc\r@sti
  samepage 1780      \else\if #1d\let\do@sttc\df@sttc
  1781      \fi\fi\fi\fi\fi\fi
  1782      \mtc@CkStr{\stctitle}\if@mtc@FE \let\do@sttc\e@sti\relax\fi

\raggedright We adjust some formatting parameters and avoid a page break between the title and the
  \parskip secttoc, then we set the font:
  \stcfont
  1783      \raggedright
  1784      \parskip=\z@%
  1785      \reset@font\stcfont%
  1786      \parindent=\z@%
  1787      \nopagebreak[4]%

\stc@rule The secttoc title is set in a tabular environment (to inhibit a page break between the title
  tabular and the top rule), with a rule at its bottom if necessary. This rule is an \hline. It is the
  \stifont top rule of the secttoc.
\do@sttc
  \mtc@v 1788      \kern-0.8\baselineskip\nopagebreak[4]%
\stctitle 1789      \par\noindent
  \hline 1790      \nopagebreak[4]%
  1791      \ifx\stc@rule\relax
  1792      \begin{tabular}{@{}p{\columnwidth}@{}}
  1793      \reset@font\stifont\do@sttc{\mtc@v\stctitle}\\
  1794      \end{tabular}%
  1795      \else
  1796      \begin{tabular}{@{}p{\columnwidth}@{}}
  1797      \reset@font\stifont\do@sttc{\mtc@v\stctitle}\\\hline
  1798      \end{tabular}%
  1799      \fi

\mtc@zrule Then, we adjust the position under the top rule and set the indentation and some
  \mtc@BBR formatting parameters:
\stcindent
  1800      \nopagebreak[4]\null\leavevmode\mtc@zrule\\\mtc@BBR
  1801      \leftmargin\stcindent
  1802      \rightmargin\stcindent
  1803      \itemindent=\z@\labelwidth=\z@%
  1804      \labelsep=\z@\listparindent=\z@%
```

```

    stc@verse We enter in a stc@verse environment to format the secttoc. The toc depth is forced
    \c@tocdepth (locally) to secttocdepth. A little trick is necessary to adjust the position.
\c@secttocdepth
    \mtc@BBR 1805      \begin{stc@verse}\c@tocdepth=\c@secttocdepth%
    1806      \leavevmode\\mtc@BBR
    1807      \vskip -.5\baselineskip

    \stc@pgno If the contents lines must have no numbers, we replace the macro \@dottedtocline
    \@dottedtocline with its undotted version. A hook is added, and the formatting settings coming from
    \@undottedtocline \mtcsetformat are activated via \stc@setform. Then the secttoc file is inserted,
    \mtc@hook@beforeinputfile followed by a strut, and the stc@verse environment is terminated.
    \stc@setform
    \@tocfile 1808 \begingroup
    \mtc@strut 1809 \makeatletter
    stc@verse 1810 \stc@setform%
    1811 \ifundefined{stc@pgno}%
    1812 {\let\@dottedtocline\@undottedtocline}{}
    1813 \@filesfalse\mtc@hook@beforeinputfile
    1814 \stc@setform%
    1815 \@input{\jobname.\@tocfile}
    1816 \vspace{-1ex} \vspace{-\baselineskip}
    1817 \leavevmode\mtc@strut
    1818 \global\@nbreakfalse\endgroup
    1819 \end{stc@verse}%

    \stc@rule The final part is just to add the bottom rule, if necessary, a possible page break and
    \mtc@zrule \aftersectoc.
    samepage
\aftersecttoc 1820 \kern-1.\baselineskip%
    1821 \nopagebreak[4]\stc@rule\null\leavevmode\\%
    1822 \vskip-1.0\baselineskip\mtc@zrule\end{samepage}%
    1823 \par\pagebreak[1]\vspace*{-1ex}\aftersecttoc\fi%
```

### 5.58.2 The \sectlof command

\sectlof The \sectlof command must be used after \section if you need a sectlof (no automatic sectlof). Its code is similar to the code of \minilof (but simpler). First, \sectlof detects the presence of its optional argument, and uses its default value, d, if it is missing. Then, \sectlof@ is called with the effective position as argument:

```
1824 \def\sectlof{\ifnextchar[{\sectlof@}{\sectlof@d}}}
```

```

\sectlof@ The \sectlof@ macro does the real work. It first sets the flag \if@sectlof@used@ (for
\if@sectlof@used@ a coherence hint) and checks if long extensions are used or not (to create the name of the
\if@longextensions@ sectlof file):
\@tocfile
\thetc 1825 \def\sectlof@[#1]{%
1826 \global\@sectlof@used@true
1827 \if@longextensions@%
1828 \def\@tocfile{slf\thetc}%
1829 \else
1830 \def\@tocfile{H\thetc}%
1831 \fi

\mtc@CkFile Then, we check the presence and the emptiness of the sectlof file and give a warning if it
\if@mtc@FE is not here or is empty:
\@tocfile
1832 \mtc@CkFile{\jobname.\@tocfile}
1833 \if@mtc@FE
1834 \PackageInfo{minitoc}%
1835 {\jobname.\@tocfile\space is empty}
1836 \@mtc@empty@sectlof@true
1837 \else

\beforesectlof If the sectlof file is present and not empty, we can insert it, but we must add some
\thispagesectlofsty presentation code: first, \beforesectlof, of course, and the page style feature:
1838 \thispagesectlofsty
1839 %% \mtc@markboth{\uppercase{\slftitle}}{\uppercase{\slftitle}}%
1840 \beforesectlof

\do@stilf We begin a samepage environment, then treat the positionning argument. If the title is
\e@sti empty, we simulate the “e” positionning.
\n@sti
\c@sti 1841 \relax\begin{samepage}%
\l@sti 1842 \if #1e\let\do@stilf\e@sti
\r@sti 1843 \else\if #1n\let\do@stilf\n@sti
\df@sti 1844 \else\if #1c\let\do@stilf\c@sti
\mtc@CkStr 1845 \else\if #1l\let\do@stilf\l@sti
\slftitle 1846 \else\if #1r\let\do@stilf\r@sti
\if@mtc@FE 1847 \else\if #1d\let\do@stilf\df@stilf
1848 \fi\fi\fi\fi\fi\fi
samepage 1849 \mtc@CkStr{\slftitle}\if@mtc@FE \let\do@stilf\e@sti\relax\fi

\raggedright We adjust some formatting parameters and avoid a page break between the title and the
\parskip sectlof, then we set the font:
\slffont
1850 \raggedright
1851 \parskip=\z@%
1852 \reset@font\slffont%

```

```

1853      \parindent=\z@%
1854      \nopagebreak[4]%

```

`\slf@rule` The sectlof title is set in a tabular environment (to inhibit a page break between the title and the top rule), with a rule at its bottom if necessary. This rule is an `\hline`. It is the top rule of the sectlof.

```

\do@stilf
  \mtc@v 1855      \kern-0.8\baselineskip\nopagebreak[4]%
\slftitle 1856      \par\noindent
  \hline 1857      \ifx\slf@rule\relax
           1858      \begin{tabular}{@{}p{\columnwidth}@{}}
           1859      \reset@font\stifont\do@stilf{\mtc@v\slftitle}\\
           1860      \end{tabular}%
           1861      \else
           1862      \begin{tabular}{@{}p{\columnwidth}@{}}
           1863      \mtc@hstrut
           1864      \reset@font\stifont\do@stilf{\mtc@v\slftitle}\\ \hline
           1865      \end{tabular}%
           1866      \fi

```

`\mtc@zrule` Then, we adjust the position under the top rule and set the indentation and some formatting parameters:

```

\mtc@BBR
\stcindent
           1867      \nopagebreak[4]\null\leavevmode\mtc@zrule\\ \mtc@BBR
           1868      \leftmargin\stcindent
           1869      \rightmargin\stcindent
           1870      \itemindent=\z@\labelwidth=\z@%
           1871      \labelsep=\z@\listparindent=\z@%

```

`stc@verse` We enter in a `stc@verse` environment to format the sectlof. The toc depth is forced (locally) to `sectlofdepth`. A little trick is necessary to adjust the position.

```

\c@tocdepth
\c@sectlofdepth
  \mtc@BBR 1872      \begin{stc@verse}%
           1873      \@ifundefined{c@lofdepth}{\c@tocdepth=\c@sectlofdepth}
           1874      \ifnum\c@tocdepth<1\relax\c@tocdepth=1\fi
           1875      \leavevmode\\ \mtc@BBR
           1876      \vskip -.5\baselineskip

```

`\slf@pgno` If the contents lines must have no numbers, we replace the macro `\@dottedtocline` with its undotted version. A hook is added, and the formatting settings coming from `\@undottedtocline` `\mtcsetformat` are activated via `\slf@setform`. Then the sectlof file is inserted, followed by a strut, and the `stc@verse` environment is terminated.

```

\slf@setform
  \@tocfile 1877 \begingroup
  \mtc@strut 1878 \makeatletter
  stc@verse 1879 \@ifundefined{slf@pgno}%
           1880 {\let\@dottedtocline\@undottedtocline}\fi

```

```

1881 \@fileswfalse\mtc@hook@beforeinputfile
1882 \slf@setform%
1883 \@input{\jobname.\@tocfile}
1884 \global\@nobreakfalse\endgroup
1885 \end{stc@verse}%

```

\stc@rule The final part is just to add the bottom rule, if necessary, a possible page break and

\mtc@zrule \afterseclof. The blank line (\\) is essential.

samepage

```

\aftersecttoc 1886 \kern-0.\baselineskip% ABCD
1887 \nopagebreak[4]\slf@rule\null\leavevmode\\%
1888 \vskip-1.0\baselineskip\mtc@zrule\end{samepage}%
1889 \par\pagebreak[1]\vspace*{-1ex}\aftersectlof\fi}%

```

### 5.58.3 The \sectlot command

\sectlot The \sectlot command must be used after \section if you need a sectlot (no automatic sectlot). Its code is similar to the code of \minilot (but simpler). First, \sectlot detects the presence of its optional argument, and uses its default value, d, if it is missing. Then, \sectlot@ is called with the effective position as argument:

```

1890 \def\sectlot{\@ifnextchar[{\sectlot@}{\sectlot@d}}

```

\sectlot@ The \sectlot@ macro does the real work. It first sets the flag \if@sectlot@used@ (for a coherence hint) and checks if long extensions are used or not (to create the name of the sectlot file):

```

\@tocfile
\thetc 1891 \def\sectlot@[#1]{%
1892 \global\@sectlot@used@true
1893 \if@longextensions@%
1894 \def\@tocfile{slt\thetc}%
1895 \else
1896 \def\@tocfile{I\thetc}%
1897 \fi

```

\mtc@CkFile Then, we check the presence and the emptiness of the sectlot file and give a warning if it is not here or is empty:

```

\@tocfile
1898 \mtc@CkFile{\jobname.\@tocfile}
1899 \if@mtc@FE
1900 \PackageInfo{minitoc}%
1901 {\jobname.\@tocfile\space is empty}
1902 \@mtc@empty@sectlot@true
1903 \else

```

`\beforesectlot` If the sectlot file is present and not empty, we can insert it, but we must add some presentation code: first, `\beforesectlot`, of course, and the page style feature:

```
1904      \thispagesectlotstyle
1905 %%      \mtc@markboth{\uppercase{\slttitle}}{\uppercase{\slttitle}}%
1906      \beforesectlot
```

`\do@stilt` We begin a `samepage` environment, then treat the positioning argument. If the title is empty, we simulate the “e” positioning.

```
\n@sti
\c@sti 1907      \relax\begin{samepage}%
\l@sti 1908      \if #1e\let\do@stilt\e@sti
\r@sti 1909      \else\if #1n\let\do@stilt\n@sti
\df@sti 1910      \else\if #1c\let\do@stilt\c@sti
\mtc@CkStr 1911      \else\if #1l\let\do@stilt\l@sti
\slttitle 1912      \else\if #1r\let\do@stilt\r@sti
\if@mtc@FE 1913      \else\if #1d\let\do@stilt\df@stilt
1914      \fi\fi\fi\fi\fi
samepage 1915      \mtc@CkStr{\slttitle}\if@mtc@FE \let\do@stilt\e@sti\relax\fi
```

`\raggedright` We adjust some formatting parameters and avoid a page break between the title and the sectlot, then we set the font:

```
\parskip
\sltfont
1916      \raggedright
1917      \parskip=\z@%
1918      \reset@font\sltfont%
1919      \parindent=\z@%
1920      \nolinebreak[4]%
```

`\stc@rule` The sectlot title is set in a tabular environment (to inhibit a page break between the title and the top rule), with a rule at its bottom if necessary. This rule is an `\hline`. It is the top rule of the sectlot.

```
\do@stilt
\mtc@v 1921      \kern-0.8\baselineskip\nolinebreak[4]%
\slttitle 1922      \par\noindent
\hline 1923      \ifx\slt@rule\relax
1924      \begin{tabular}{@{}p{\columnwidth}@{}}
1925      \reset@font\stifont\do@stilt{\mtc@v\slttitle}\\
1926      \end{tabular}%
1927      \else
1928      \begin{tabular}{@{}p{\columnwidth}@{}}
1929      \mtc@hstrut
1930      \reset@font\stifont\do@stilt{\mtc@v\slttitle}\\
1931      \end{tabular}%
1932      \fi
```

`\mtc@zrule` Then, we adjust the position under the top rule and set the indentation and some formatting parameters:

`\mtc@BBR`

`\stcindent`

```

1933      \nopagebreak[4]\null\leavevmode\mtc@zrule\\\mtc@BBR
1934      \leftmargin\stcindent
1935      \rightmargin\stcindent
1936      \itemindent=\z@\labelwidth=\z@%
1937      \labelsep=\z@\listparindent=\z@%

```

`stc@verse` We enter in a `stc@verse` environment to format the sectlot. The toc depth is forced  
`\c@tocdepth` (locally) to `sectlotdepth`. A little trick is necessary to adjust the position.  
`\c@sectlotdepth`

```

\mtc@BBR 1938      \begin{stc@verse}%
1939      \@ifundefined{c@lotdepth}{}{\c@tocdepth=\c@sectlotdepth}
1940      \ifnum\c@tocdepth<1\relax\c@tocdepth=1\fi
1941      \leavevmode\\\mtc@BBR
1942      \vskip -.5\baselineskip

```

`\slt@pgno` If the contents lines must have no numbers, we replace the macro `\@dottedtocline`  
`\@dottedtocline` with its undotted version. A hook is added, and the formatting settings coming from  
`\@undottedtocline` `\mtcsetformat` are activated via `\slt@setform`. Then the sectlot file is inserted,  
`\mtc@hook@beforeinputfile` followed by a strut, and the `stc@verse` environment is terminated.

```

\slt@setform
\@tocfile 1943 \begingroup
\mtc@strut 1944 \makeatletter
stc@verse 1945 \@ifundefined{slt@pgno}%
1946 {\let\@dottedtocline\@undottedtocline}{}
1947 \gdef\thestc{\arabic{stc}}
1948 \@filesfalse\mtc@hook@beforeinputfile
1949 \slt@setform%
1950 \input{\jobname.\@tocfile}
1951 \global\@nobeckfalse\endgroup
1952 \end{stc@verse}%

```

`\stc@rule` The final part is just to add the bottom rule, if necessary, a possible page break and  
`\mtc@zrule` `\aftersectlot`.  
`samepage`

```

\aftersectlot 1953 \kern-0.\baselineskip% ABCD
1954 \nopagebreak[4]\slt@rule\null\leavevmode\\%
1955 \vskip-1.0\baselineskip\mtc@zrule\end{samepage}%
1956 \par\pagebreak[1]\vspace*{-1ex}\aftersectlot\fi}%

```

## 5.59 Auxiliary internal commands, section level

`\l@xsect` We define auxiliary commands, used for the mini-tables and as delimiters in the TOC file (and LOF and LOT files). The depth of `xsect` is huge to inhibit the printing of its contents line (except if you cheat).

`\@dottedtocline`

`\l@schapter`

`\xsect`

`\schapter`

```

1957 \def\l@xsect{\@dottedtocline{\@M}{1.0em}{2.3em}}
1958 \def\l@schapter{\@dottedtocline{1}{1.0em}{2.3em}}
1959 \def\xsect{xsect}
1960 \def\schapter{schapter}

```

## 5.60 Patching the `\section` command (continued)

`\@sect` We patch the both branches of the `\section` command: `\@sect` for the unstarred version and `\@ssect` for the starred version. First, for the unstarred version (`\@sect`), we add a `xsect` contents line in the LOF and in the LOT. The test `\ifnum #2=1` restricts the action to the section level macros (because `\@sect` is also used by `\subsection` and below, which have no mini-tables).

`\addcontentsline`

```

1961 \let\sv@sect\@sect
1962 \gdef\@sect#1#2#3#4#5#6[#7]#8{%
1963 \ifnum #2=1
1964   \addcontentsline{lof}{xsect}{#7}%
1965   \addcontentsline{lot}{xsect}{#7}%
1966 \fi
1967 \sv@sect{#1}{#2}{#3}{#4}{#5}{#6}[#7]{#8}}

```

`\section` If it is a section (unstarred or starred via `\starsection`), we add a `xsect` entry in the LOF and in the LOT.

`\starsection`

`\addcontentsline`

```

1968 \def\@sect#1#2#3#4#5#6[#7]#8{
1969 \expandafter
1970 \ifx\csname #1\endcsname\section\relax
1971   \addcontentsline{lof}{xsect}{#7}%
1972   \addcontentsline{lot}{xsect}{#7}%
1973 \fi
1974 \ifx\csname #1\endcsname\starsection\relax
1975   \addcontentsline{lof}{xsect}{#7}%
1976   \addcontentsline{lot}{xsect}{#7}%
1977 \fi

```

`\@svsec` And the remainder of the section header formatting:

`\refstepcounter`

`\@tempskipa`

`\@hangfrom`

`\addcontentsline`

`\numberline`

`\@svsechd`

`\@xsect`

```

1978 \ifnum #2>\c@secnumdepth
1979   \let\@svsec\@empty

```



```

1980 \else
1981   \refstepcounter{#1}%
1982   \edef\@svsec{\csname the#1\endcsname\hskip 1em}%
1983 \fi
1984 \@tempskipa #5\relax
1985 \ifdim \@tempskipa>\z@
1986   \begingroup #6\relax
1987     \@hangfrom{\hskip #3\relax\@svsec}%
1988     {\interlinepenalty \@M #8\par}%
1989   \endgroup
1990   \csname #1mark\endcsname{#7}\addcontentsline
1991   {toc}{#1}{\ifnum #2>\c@secnumdepth
1992             \else
1993             \protect\numberline{\csname the#1\endcsname}%
1994             \fi
1995             #7}%
1996 \else
1997   \def\@svsechd{#6\hskip #3\relax
1998   \@svsec #8\csname #1mark\endcsname
1999   {#7}\addcontentsline
2000   {toc}{#1}{\ifnum #2>\c@secnumdepth
2001             \else
2002             \protect\numberline{\csname the#1\endcsname}
2003             \fi
2004             #7}}%
2005 \fi
2006 \@xsect{#5}}

```

```

\@ssect Then we patch the starred branch (\@ssect). We define also the delimiting commands
\sectbegin \sectbegin and \sectend commands.
\sectend
\stc@ssect 2007 \let\stc@ssect\@ssect
\addtocontents 2008 \def\@ssect{\addtocontents{toc}{\protect\sectend}\stc@ssect}
2009 \def\@ssect{\addtocontents{toc}{\protect\sectbegin}\stc@ssect}
2010 \let\sectend\relax
2011 \let\sectbegin\relax

```

## 5.61 The \dosecttoc command and siblings

The \dosectoc command is very similar to \dominitoc.

```

\dosecttoc The \dosecttoc command extracts information from the .toc file and creates the
\@dosecttoc .stc(N) files (.stc becomes .S on MS-DOS).
\STC@next
2012 \def\@dosecttoc#1{
2013   \makeatletter
2014   \setcounter{stc}{0}
2015   \STC@next#1.toc\relax\}\setcounter{stc}{0}}

```

`\dosectlof` The `\dosectlof` command extracts information from the `.lof` file and creates the `\@dosectlof` `.slf(N)` files (`.slf` becomes `.H` on MS-DOS).

`\SLF@next`

```
2016 \def\@dosectlof#1{%
2017   \makeatletter
2018   \setcounter{stc}{0}
2019   \SLF@next#1.lof\relax\}\setcounter{stc}{0}}
```

`\dosectlot` The `\dosectlot` command extracts information from the `.lot` file and creates the `\@dosectlot` `.slt(N)` files (`.slt` becomes `.V` on MS-DOS).

`\PLT@next`

```
2020 \def\@dosectlot#1{%
2021   \makeatletter
2022   \setcounter{stc}{0}
2023   \SLT@next#1.lot\relax\}\setcounter{stc}{0}}
```

`\dosecttoc` We define the user-level macros, who detect the optional argument:

`\dosectlof`

```
\dosectlot 2024 \def\dosecttoc{\@ifnextchar[{\dosecttoc@}{\dosecttoc@[1]}}
2025 \def\dosectlof{\@ifnextchar[{\dosectlof@}{\dosectlof@[1]}}
2026 \def\dosectlot{\@ifnextchar[{\dosectlot@}{\dosectlot@[1]}}
```

`\dosecttoc@` We treat the optional argument of `\dosecttoc` (it becomes the default position for titles of secttocs) and flag this macro as used:

`\df@sttc`

```
\e@sti 2027 \def\dosecttoc@[#1]{%
\n@sti 2028 \global\@dosecttoc@used@true
\c@sti 2029 \if #1e\let\df@sttc\e@sti%
\l@sti 2030 \else\if #1n\let\df@sttc\n@sti%
\r@sti 2031 \else\if #1c\let\df@sttc\c@sti%
2032 \else\if #1l\let\df@sttc\l@sti%
2033 \else\if #1r\let\df@sttc\r@sti%
2034 \fi\fi\fi\fi\fi%
2035 \@@dosecttoc}
```

`\dosectlof@` We treat the optional argument of `\dosectlof` (it becomes the default position for titles of sectlofs) and flag this macro as used:

`\df@stilf`

```
\e@sti 2036 \def\dosectlof@[#1]{%
\n@sti 2037 \global\@dosectlof@used@true
\c@sti 2038 \if #1e\let\df@stilf\e@sti%
\l@sti 2039 \else\if #1n\let\df@stilf\n@sti%
\r@sti 2040 \else\if #1c\let\df@stilf\c@sti%
2041 \else\if #1l\let\df@stilf\l@sti%
2042 \else\if #1r\let\df@stilf\r@sti%
2043 \fi\fi\fi\fi\fi%
2044 \@@dosectlof}
```

```

\dosectlot@ We treat the optional argument of \dosectlot (it becomes the default position for titles
\if@dosectlot@used@ of sectlofs) and flag this macro as used:
\df@stilt
\@sti 2045 \def\dosectlot@[#1]{%
\n@sti 2046 \global\@dosectlot@used@true
\c@sti 2047 \if #1e\let\df@stilt\@sti%
\l@sti 2048 \else\if #1n\let\df@stilt\n@sti%
\r@sti 2049 \else\if #1c\let\df@stilt\c@sti%
2050 \else\if #1l\let\df@stilt\l@sti%
2051 \else\if #1r\let\df@stilt\r@sti%
2052 \fi\fi\fi\fi\fi%
2053 \@@dosectlot}

\@@dosecttoc These macros invoke the \@dosect... commands to create the mini-table file, then close
\@@dosectlof the file descriptor.
\@@dosectlot
\tf@mtc 2054 \def\@@dosecttoc{\@dosecttoc{\jobname}\immediate\closeout\tf@mtc}
2055 \def\@@dosectlof{\@dosectlof{\jobname}\immediate\closeout\tf@mtc}
2056 \def\@@dosectlot{\@dosectlot{\jobname}\immediate\closeout\tf@mtc}

\STC@next Processing the next entry in the list and remove it from the head of the list:
\STC@list
\STC@loop 2057 \def\STC@next#1\relax#2\{\%
2058 \edef\STC@list{#2}%
2059 \STC@loop{#1}}

\STC@toc Check if the list is empty:
\STC@list
\STC@explist 2060 \def\STC@toc{%
2061 \ifx\STC@list\@empty\else\expandafter\STC@explist\fi}

\STC@contentsline The macro \STC@contentsline analyses the lines read from the TOC file. If \section
\section is found, stc is incremented and a new secttoc file is created.
\thestc
\tf@mtc 2062 \def\STC@contentsline#1#2#3#4{\% %%HO: #4 added
\stcname 2063 \gdef\thestc{\arabic{stc}}\% %%HO: space removed
\MTC@WriteContentsLine 2064 \expandafter\ifx\csname #1\endcsname\section
2065 \stepcounter{stc}%
2066 \% \def\thestc{\arabic{stc}}\% HO: removed

\if@longextensions@ We test if long or short extensions are used, to build the name of the mini-table file, then
\stcname open it:

2067 \if@longextensions@%
2068 \PackageInfo{minitoc}%
2069 {Writing\space\jobname.stc\thestc}%
2070 \def\stcname{\jobname.stc\thestc}%

```

```

2071 \else
2072   \PackageInfo{minitoc}%
2073   {Writing\space\jobname.S\thestc}%
2074   \def\stcname{\jobname.S\thestc}%
2075 \fi
2076 \immediate\closeout\tf@mtc
2077 \immediate\openout\tf@mtc=\stcname
2078 \fi

```

\mtc@toks Now, we filter the relevant contents lines:

```

\MTC@WriteCoffeeline
\MTC@WriteContentsline 2079 \mtc@toks{\noexpand\leavevmode #2}%
\coffee 2080 \expandafter\ifx\csname #1\endcsname\coffee
\subsection 2081 \MTC@WriteCoffeeline{#1}{#3}%
\subsubsection 2082 \fi
\paragraph 2083 \expandafter\ifx\csname #1\endcsname\subsection
\subparagraph 2084 \MTC@WriteContentsline{#1}{stcSS}{#3}{#4}%
2085 \fi
2086 \expandafter\ifx\csname #1\endcsname\subsubsection
2087 \MTC@WriteContentsline{#1}{stcSSS}{#3}{#4}%
2088 \fi
2089 \expandafter\ifx\csname #1\endcsname\paragraph
2090 \MTC@WriteContentsline{#1}{stcP}{#3}{#4}%
2091 \fi
2092 \expandafter\ifx\csname #1\endcsname\subparagraph
2093 \MTC@WriteContentsline{#1}{stcSP}{#3}{#4}%
2094 \fi

```

\starsection A starred section terminates the current section and creates a new secttoc file:

```

\stepcounter
\thestc 2095 \ifx\csname #1\endcsname\starsection
\arabic 2096 \stepcounter{stc}%
\if@longextensions@ 2097 \gdef\thestc{\arabic{stc}}
\stcname 2098 \if@longextensions@
\closeout 2099 \PackageInfo{minitoc}%
\openout 2100 {Writing\space\jobname.stc\thestc}%
2101 \def\stcname{\jobname.stc\thestc}%
2102 \else
2103 \PackageInfo{minitoc}%
2104 {Writing\space\jobname.S\thestc}%
2105 \def\stcname{\jobname.S\thestc}%
2106 \fi
2107 \immediate\closeout\tf@mtc
2108 \immediate\openout\tf@mtc=\stcname
2109 \fi

```

\MTC@WriteContentsline We process the entries for starred sectionning commands:

```

\starsubsection
\starsubsubsection 2110 \expandafter\ifx\csname #1\endcsname\starsubsubsection
\starparagraph 2111 \MTC@WriteContentsline{#1}{stcSS}{#3}{#4}%
\starsubparagraph

```

```

2112 \fi
2113 \expandafter\ifx\csname #1\endcsname\starsubsubsection
2114 \MTC@WriteContentsline{#1}{stcSSS}{#3}{#4}%
2115 \fi
2116 \expandafter\ifx\csname #1\endcsname\starparagraph
2117 \MTC@WriteContentsline{#1}{stcP}{#3}{#4}%
2118 \fi
2119 \expandafter\ifx\csname #1\endcsname\starsubparagraph
2120 \MTC@WriteContentsline{#1}{stcSP}{#3}{#4}%
2121 \fi
2122 }

```

\STC@explist The loop to read the lines of the TOC file; expands the list of entries and call \STC@next to process the first one.

\STC@list

```

2123 \def\STC@explist{\expandafter\STC@next\STC@list\}

```

\STC@loop If an entry is found, loop through line by line, looking for interesting entries. Otherwise, process the next entry in the list.

\STC@read

```

2124 \def\STC@loop#1{\openin\@inputcheck#1\relax
2125 \ifeof\@inputcheck
2126 \PackageWarning{minitoc}%
2127 {No file #1\MessageBreak SECTTOCS NOT PREPARED}%
2128 \expandafter\STC@toc
2129 \else
2130 \PackageInfo{minitoc}%
2131 {PREPARING SECTTOCS FROM #1}%
2132 \expandafter\STC@read\fi}

```

\STC@read Read the next entry of the .toc file.

\STC@line

```

2133 \def\STC@read{%
2134 \read\@inputcheck to\STC@line

```

\STC@test The . . . . . make sure that \STC@test has sufficient arguments:

\STC@line

```

2135 \expandafter\STC@test\STC@line. . . . \STC@% %%H0: . added
2136 }%

```

\STC@test The \STC@test macro finds the “interesting” commands in the TOC file, mainly to delimit sections;

\mtc@string

```

\STC@list 2137 \long\def\STC@test#1#2#3#4#5#6\STC@{% %%H0: #6 added

```

```

\STC@toc 2138 \ifx#1\contentsline

```

```

\STC@read 2139 \let\mtc@string\string

```

```

\sectend 2140 \STC@contentsline{#2}{#3}{#4}{#5}% %%H0: #4 added

```

\sectbegin

```

2141 \let\mtc@string\relax
2142 \else\ifx#1\@input
2143 \edef\STC@list{\STC@list#2\relax}%
2144 \else\ifx#1\sectend
2145 \immediate\closeout\tf@mtc
2146 \immediate\openout\tf@mtc=\jobname.mtc
2147 \else\ifx#1\sectbegin
2148 \addtocounter{stc}{-1}%
2149 \fi\fi\fi\fi
2150 \ifeof\@inputcheck\expandafter\STC@toc
2151 \else\expandafter\STC@read\fi}%

```

\SLF@next Processing the next entry in the list and remove it from the head of the list:

\SLF@list

```

\SLF@loop 2152 \def\SLF@next#1\relax#2\\{%
2153 \edef\SLF@list{#2}%
2154 \SLF@loop{#1}}

```

\SLF@lof Check if the list is empty:

\SLF@list

```

\SLF@explist 2155 \def\SLF@lof{%
2156 \ifx\SLF@list\@empty\else\expandafter\SLF@explist\fi}

```

\SLF@contentsline The macro \SLF@contentsline analyses the lines read from the LOF file. If \section is found, stc is incremented and a new sectlof file is created.

\thestc

\tf@mtc 2157 \def\SLF@contentsline#1#2#3#4{% %%HO: #4 added

\slfname 2158 \gdef\thestc{\arabic{stc}} %%HO: space removed

\MTC@WriteContentsLine 2159 \expandafter\ifx\csname #1\endcsname\xsect

2160 \stepcounter{stc}%

2161 %% \gdef\thestc{\arabic{stc}} %%HO: removed

\if@longextensions@ We test if long or short extensions are used, to build the name of the mini-table file, then open it:

\slfname

```

2162 \if@longextensions@%
2163 \PackageInfo{minitoc}%
2164 {Writing\space\jobname.slf\thestc}%
2165 \def\slfname{\jobname.slf\thestc}%
2166 \else
2167 \PackageInfo{minitoc}%
2168 {Writing\space\jobname.H\thestc}%
2169 \def\slfname{\jobname.H\thestc}%
2170 \fi
2171 \immediate\closeout\tf@mtc
2172 \immediate\openout\tf@mtc=\slfname
2173 \fi

```

```

\mtc@toks Now, we filter the relevant contents lines:
\MTC@WriteContentsline
  \figure 2174 \mtc@toks{\noexpand\leavevmode #2}%
          2175 \expandafter\ifx\csname #1\endcsname\figure
          2176 \MTC@WriteContentsline{#1}{slf}{#3}{#4}%
          2177 \fi
          2178 }

\SLF@explist The loop to read the lines of the LOF file; expands the list of entries and call \SLF@next
\SLF@next to process the first one.
\SLF@list
2179 \def\SLF@explist{\expandafter\SLF@next\SLF@list\}

\SLF@loop If an entry is found, loop through line by line, looking for interesting entries. Otherwise,
\SLF@lof process the next entry in the list.
\SLF@read
2180 \def\SLF@loop#1{\openin\@inputcheck#1\relax
2181 \ifeof\@inputcheck
2182 \PackageWarning{minitoc}%
2183 {No file #1\MessageBreak SECTLOFS NOT PREPARED}%
2184 \expandafter\SLF@lof
2185 \else
2186 \PackageInfo{minitoc}%
2187 {PREPARING SECTLOFS FROM #1}%
2188 \expandafter\SLF@read\fi}

Read the next entry of the .lof file.

\SLF@read The ..... make sure that \SLF@test has sufficient arguments:
\SLF@test
\SLF@line 2189 \def\SLF@read{%
2190 \read\@inputcheck to\SLF@line
2191 \expandafter\SLF@test\SLF@line.....\SLF@% %%H0: . added
2192 }%

\SLF@test The \SLF@test macro finds the “interesting” commands in the LOF file, mainly to
\SLF@contentsline delimit sections;
\mtc@string
\SLF@list 2193 \long\def\SLF@test#1#2#3#4#5#6\SLF@{% %%H0: #6 added
\SLF@lof 2194 \ifx#1\contentsline
\SLF@read 2195 \let\mtc@string\string
\sectend 2196 \SLF@contentsline{#2}{#3}{#4}{#5}% %%H0: #4 added
\sectbegin 2197 \let\mtc@string\relax
2198 \else\ifx#1\@input
2199 \edef\SLF@list{\SLF@list#2\relax}%
2200 \else\ifx#1\sectend
2201 \immediate\closeout\tf@mtc
2202 \immediate\openout\tf@mtc=\jobname.mtc

```

```

2203 \else\ifx#1\sectbegin
2204     \addtocounter{stc}{-1}%
2205 \fi\fi\fi\fi
2206 \ifeof\@inputcheck\expandafter\SLF@lof
2207 \else\expandafter\SLF@read\fi}%

```

\SLT@next Processing the next entry in the list and remove it from the head of the list:

\SLT@list

```

\SLT@loop 2208 \def\SLT@next#1\relax#2\{\%
2209     \edef\SLT@list{#2}%
2210     \SLT@loop{#1}}

```

\SLT@lot Check if the list is empty:

\SLT@list

```

\SLT@explist 2211 \def\SLT@lot{%
2212     \ifx\SLT@list\@empty\else\expandafter\SLT@explist\fi}

```

\SLT@contentsline The macro \SLT@contentsline analyses the lines read from the LOT file. If \section is found, stc is incremented and a new sectlot file is created.

\thestc

\tf@mtc 2213 \def\SLT@contentsline#1#2#3#4{% %%H0: #4 added

\sltname 2214 \gdef\thestc{\arabic{stc}}% %%H0: space removed

```

\MTC@WriteContentsLine 2215 \expandafter\ifx\csname #1\endcsname\xsect
2216     \stepcounter{stc}%
2217 %%     \gdef\thestc{\arabic{stc}} %%H0: removed

```

\if@longextensions@ We test if long or short extensions are used, to build the name of the mini-table file, then open it:

\sltname

```

2218 \if@longextensions@%
2219     \PackageInfo{minitoc}%
2220     {Writing\space\jobname.slt\thestc}%
2221     \def\sltname{\jobname.slt\thestc}%
2222 \else
2223     \PackageInfo{minitoc}%
2224     {Writing\space\jobname.V\thestc}%
2225     \def\sltname{\jobname.V\thestc}%
2226 \fi
2227 \immediate\closeout\tf@mtc
2228 \immediate\openout\tf@mtc=\sltname
2229 \fi

```

\mtc@toks Now, we filter the relevant contents lines:

\MTC@WriteContentsline

```

\table 2230 \mtc@toks{\noexpand\leavevmode #2}%
2231 \expandafter\ifx\csname #1\endcsname\table
2232     \MTC@WriteContentsline{#1}{\sltname}{#3}{#4}%

```



```

2233 \fi
2234 }

```

\SLT@explist The loop to read the lines of the LOT file; expands the list of entries and call \SLT@next to process the first one.  
 \SLT@next  
 \SLT@list

```

2235 \def\SLT@explist{\expandafter\SLT@next\SLT@list\}

```

\SLT@loop If an entry is found, loop through line by line, looking for interesting entries. Otherwise, process the next entry in the list.  
 \SLT@lot  
 \SLT@read

```

2236 \def\SLT@loop#1{\openin\@inputcheck#1\relax
2237 \ifeof\@inputcheck
2238 \PackageWarning{minitoc}%
2239 {No file #1\MessageBreak SECTLOTS NOT PREPARED}%
2240 \expandafter\SLT@lot
2241 \else
2242 \PackageInfo{minitoc}%
2243 {PREPARING SECTLOTS FROM #1}%
2244 \expandafter\SLT@read\fi}

```

Read the next entry of the .lot file.

\SLT@read The ..... make sure that \SLT@test has sufficient arguments:

\SLT@test

```

\SLT@line 2245 \def\SLT@read{%
2246 \read\@inputcheck to\SLT@line
2247 \expandafter\SLT@test\SLT@line.....\SLT@% %%H0: . added
2248 }%

```

\SLT@test The \SLT@test macro finds the “interesting” commands in the LOT file, mainly to delimit sections;  
 \SLT@contentsline

\mtc@string

```

\SLT@list 2249 \long\def\SLT@test#1#2#3#4#5#6\SLT@{% %%H0: #6 added
\SLT@lot 2250 \ifx#1\contentsline
\SLT@read 2251 \let\mtc@string\string
\sectend 2252 \SLT@contentsline{#2}{#3}{#4}{#5}% %%H0: #4 added
\sectbegin 2253 \let\mtc@string\relax
2254 \else\ifx#1\@input
2255 \edef\SLT@list{\SLT@list#2\relax}%
2256 \else\ifx#1\sectend
2257 \immediate\closeout\tf@mtc
2258 \immediate\openout\tf@mtc=\jobname.mtc
2259 \else\ifx#1\sectbegin
2260 \addtocounter{stc}{-1}%
2261 \fi\fi\fi\fi
2262 \ifeof\@inputcheck\expandafter\SLT@lot
2263 \else\expandafter\SLT@read\fi}%

```

## 5.62 End of section-level commands

We terminate the *else* branch of the test `\@ifundefined{section}`, the *true* branch of the test `\@ifundefined{chapter}` and add an empty *else* branch to the test `\@ifundefined{chapter}`:

```
2264 \fi% end of \@ifmtc@section@def@
2265 \fi% end of \@ifmtc@chapter@undef@
```

## 5.63 Necessary \l@... commands

```
\l@listof  Some \l@... commands (analog to \l@section or \l@paragraph) are required to
\l@startpart format some entries in the mini-tables, for starred sectioning commands essentially:
\l@starchapter
\l@starsection 2266 \@ifundefined{section}{}{\let\l@listof\l@section}
\l@starsubsection 2267 \@ifundefined{chapter}{}{\let\l@listof\l@chapter}
\l@starsubsubsection 2268 \@ifundefined{part}{}{\let\l@startpart\l@part}
\l@starparagraph 2269 \@ifundefined{chapter}{}{\let\l@starchapter\l@chapter}
\l@starsubparagraph 2270 \@ifundefined{section}{}{\let\l@starsection\l@section}
2271 \@ifundefined{subsection}{}{\let\l@starsubsection\l@subsection}
2272 \@ifundefined{subsubsection}{}{\let\l@starsubsubsection\l@subsubsection}
2273 \@ifundefined{paragraph}{}{\let\l@starparagraph\l@paragraph}
2274 \@ifundefined{subparagraph}{}{\let\l@starsubparagraph\l@subparagraph}
```

## 5.64 The horizontal rules and their default values

```
\noptcrule
\nomtcrule
\nostcrule
\ptcrule We define here the various commands to activate ou inhibit the horizontal rules in
\mtcrule the various kinds of mini-tables. Each such command is an indirect definition of the
\stcrule corresponding horizontal rule. The rules are .4 pt high horizontal rules.
\noplfrule
\nomlfrule 2275 \def\noptcrule{\let\ptc@rule\relax}
\noslfrule 2276 \def\nomtcrule{\let\mtc@rule\relax}
\plfrule 2277 \def\nostcrule{\let\stc@rule\relax}
\mlfrule 2278 \def\ptcrule{\def\ptc@rule{\kern-3\p@ \hrule width \columnwidth \kern2.6\p@}}
\slfrule 2279 \def\mtcrule{\def\mtc@rule{\kern-3\p@ \hrule width \columnwidth \kern2.6\p@}}
\nopltrule 2280 \def\stcrule{\def\stc@rule{\kern-3\p@ \hrule width \columnwidth \kern2.6\p@}}
\nomltrule 2281 \def\ptc@rule{\rule[3\p@]{\columnwidth}{.4\p@}\vspace*{2.6\p@}}
\nosltrule 2282 \def\mtc@rule{\rule[3\p@]{\columnwidth}{.4\p@}\vspace*{2.6\p@}}
2283 \def\stc@rule{\rule[3\p@]{\columnwidth}{.4\p@}\vspace*{2.6\p@}}
\pltrule
\mltrule
\sltrule 2284 \def\noplfrule{\let\plf@rule\relax}
2285 \def\nomlfrule{\let\mlf@rule\relax}
2286 \def\noslfrule{\let\slf@rule\relax}
```

```

2287 \def\plfrule{\def\plf@rule{\kern-3\p@ \hrule width \columnwidth \kern2.6\p@}}
2288 \def\mlfrule{\def\mlf@rule{\kern-3\p@ \hrule width \columnwidth \kern2.6\p@}}
2289 \def\slfrule{\def\slf@rule{\kern-3\p@ \hrule width \columnwidth \kern2.6\p@}}
2290 \def\plf@rule{\rule[3\p@]{\columnwidth}{.4\p@}\vspace*{2.6\p@}}
2291 \def\mlf@rule{\rule[3\p@]{\columnwidth}{.4\p@}\vspace*{2.6\p@}}
2292 \def\slf@rule{\rule[3\p@]{\columnwidth}{.4\p@}\vspace*{2.6\p@}}

2293 \def\nopltrule{\let\plt@rule\relax}
2294 \def\nomltrule{\let\mlt@rule\relax}
2295 \def\nosltrule{\let\slt@rule\relax}
2296 \def\pltrule{\def\plt@rule{\kern-3\p@ \hrule width \columnwidth \kern2.6\p@}}
2297 \def\mltrule{\def\mlt@rule{\kern-3\p@ \hrule width \columnwidth \kern2.6\p@}}
2298 \def\sltrule{\def\slt@rule{\kern-3\p@ \hrule width \columnwidth \kern2.6\p@}}
2299 \def\plt@rule{\rule[3\p@]{\columnwidth}{.4\p@}\vspace*{2.6\p@}}
2300 \def\mlt@rule{\rule[3\p@]{\columnwidth}{.4\p@}\vspace*{2.6\p@}}
2301 \def\slt@rule{\rule[3\p@]{\columnwidth}{.4\p@}\vspace*{2.6\p@}}

```

## 5.65 The `\mtcset...` commands

These commands<sup>7</sup> have been introduced to build a nicer user interface, and each of them replaces numerous user commands, offering a rather unified and logical syntax.

### 5.65.1 Keywords for the `\mtcset...` commands

`\@namedef` We define some common keywords for the `\mtcset...` commands. A keyword is created via the `\@namedef – \@nameuse` mechanism the following way:

```
\@namedef{mtc@family@name}{abbreviation}
```

where *family* is the name of a group of keywords relative to one or several `\mtcset...` macros, *name* is the keyword that the user gives as argument to the `\mtcset...` macro, and *abbreviation* is a string used to build the name of the macro effectively used. As some `\mtcset...` macros have several keyword parameters, this method can reduce the number of macros at the user level, at the cost of few keyword families.

`\@namedef` We define a family (typetable) of keywords for the types of mini-tables:

```

2302 \@namedef{mtc@typetable@parttoc}{ptc}\def\mtc@typetable@parttoc{ptc}
2303 \@namedef{mtc@typetable@partlot}{plt}\def\mtc@typetable@partlot{plf}
2304 \@namedef{mtc@typetable@partlof}{plf}\def\mtc@typetable@partlof{plt}
2305 \@namedef{mtc@typetable@minitoc}{mtc}\def\mtc@typetable@minitoc{mtc}
2306 \@namedef{mtc@typetable@minilof}{mlf}\def\mtc@typetable@minilof{mlf}

```

<sup>7</sup>The general concept of the `\mtcset...` commands was proposed by Benjamin BAYART.

```

2307 \@namedef{mtc@typetable@minilot}{mlt}\def\mtc@typetable@minilot{mlt}
2308 \@namedef{mtc@typetable@secttoc}{stc}\def\mtc@typetable@secttoc{stc}
2309 \@namedef{mtc@typetable@sectlof}{slf}\def\mtc@typetable@sectlof{slf}
2310 \@namedef{mtc@typetable@sectlot}{slt}\def\mtc@typetable@sectlot{slt}

```

\@namedef Then another family (typetitle) for the titles of the mini-tables:

```

2311 \@namedef{mtc@typetitle@parttoc}{pti}\def\mtc@typetitle@parttoc{pti}
2312 \@namedef{mtc@typetitle@partlof}{pti}\def\mtc@typetitle@partlof{pti}
2313 \@namedef{mtc@typetitle@partlot}{pti}\def\mtc@typetitle@partlot{pti}
2314 \@namedef{mtc@typetitle@minitoc}{mti}\def\mtc@typetitle@minitoc{mti}
2315 \@namedef{mtc@typetitle@minilof}{mti}\def\mtc@typetitle@minilof{mti}
2316 \@namedef{mtc@typetitle@minilot}{mti}\def\mtc@typetitle@minilot{mti}
2317 \@namedef{mtc@typetitle@secttoc}{sti}\def\mtc@typetitle@secttoc{sti}
2318 \@namedef{mtc@typetitle@sectlof}{sti}\def\mtc@typetitle@sectlof{sti}
2319 \@namedef{mtc@typetitle@sectlot}{sti}\def\mtc@typetitle@sectlot{sti}

```

\@namedef We define a family (YN) of keywords to recognize the keywords “off” and “on”, with their many synonyms<sup>8</sup> and meaning false or true:

```

2320 \@namedef{mtc@YN@off}{no}\def\mtc@YN@off{no}
2321 \@namedef{mtc@YN@OFF}{no}\def\mtc@YN@OFF{no}
2322 \@namedef{mtc@YN@no}{no}\def\mtc@YN@no{no}
2323 \@namedef{mtc@YN@NO}{no}\def\mtc@YN@NO{no}
2324 \@namedef{mtc@YN@n}{no}\def\mtc@YN@n{no}
2325 \@namedef{mtc@YN@N}{no}\def\mtc@YN@N{no}
2326 \@namedef{mtc@YN@false}{no}\def\mtc@YN@false{no}
2327 \@namedef{mtc@YN@FALSE}{no}\def\mtc@YN@FALSE{no}
2328 \@namedef{mtc@YN@faux}{no}\def\mtc@YN@faux{no}
2329 \@namedef{mtc@YN@FAUX}{no}\def\mtc@YN@FAUX{no}
2330 \@namedef{mtc@YN@f}{no}\def\mtc@YN@f{no}
2331 \@namedef{mtc@YN@F}{no}\def\mtc@YN@F{no}
2332 \@namedef{mtc@YN@NON}{no}\def\mtc@YN@NON{no}
2333 \@namedef{mtc@YN@non}{no}\def\mtc@YN@non{no}
2334 \@namedef{mtc@YN@0}{no}\expandafter\def\csname mtc@YN@0\endcsname{no}
2335 \@namedef{mtc@YN@on}{}\def\mtc@YN@on{}
2336 \@namedef{mtc@YN@ON}{}\def\mtc@YN@ON{}
2337 \@namedef{mtc@YN@yes}{}\def\mtc@YN@yes{}
2338 \@namedef{mtc@YN@YES}{}\def\mtc@YN@YES{}
2339 \@namedef{mtc@YN@y}{}\def\mtc@YN@y{}
2340 \@namedef{mtc@YN@Y}{}\def\mtc@YN@Y{}
2341 \@namedef{mtc@YN@true}{}\def\mtc@YN@true{}
2342 \@namedef{mtc@YN@TRUE}{}\def\mtc@YN@TRUE{}
2343 \@namedef{mtc@YN@t}{}\def\mtc@YN@t{}
2344 \@namedef{mtc@YN@T}{}\def\mtc@YN@T{}
2345 \@namedef{mtc@YN@vrai}{}\def\mtc@YN@vrai{}
2346 \@namedef{mtc@YN@VRAI}{}\def\mtc@YN@VRAI{}
2347 \@namedef{mtc@YN@v}{}\def\mtc@YN@v{}

```

<sup>8</sup>This (extreme) case shows the easyness for creating synonyms of frequently used keywords. Note also that when a keyword contains a non-letter character, we must use a hack with \expandafter \csname ... \endcsname.

```

2348 \@namedef{mtc@YN@V}{ }\def\mtc@YN@TRUEV
2349 \@namedef{mtc@YN@OUI}{ }\def\mtc@YN@OUI{ }
2350 \@namedef{mtc@YN@oui}{ }\def\mtc@YN@oui{ }
2351 \@namedef{mtc@YN@O}{ }\def\mtc@YN@O{ }
2352 \@namedef{mtc@YN@o}{ }\def\mtc@YN@o{ }
2353 \@namedef{mtc@YN@1}{ }\expandafter\def\csname mtc@YN@1\endcsname{ }

```

### 5.65.2 The `\mtcsetfont` command

`\@namedef` We define the sectionning level keywords (note that `part` is not a member of this family (`sectlevel`), because no contents line for a `part` can appear in a mini-table, `part` being the highest sectionning level); “\*” represents “any level”, and is used to set the global default font for a given kind of mini-table.

```

2354 \@namedef{mtc@sectlevel@chapter}{C}\def\mtc@sectlevel@chapter{C}
2355 \@namedef{mtc@sectlevel@section}{S}\def\mtc@sectlevel@section{S}
2356 \@namedef{mtc@sectlevel@subsection}{SS}\def\mtc@sectlevel@subsection{SS}
2357 \@namedef{mtc@sectlevel@subsubsection}{SSS}\def\mtc@sectlevel@subsubsection{SSS}
2358 \@namedef{mtc@sectlevel@paragraph}{P}\def\mtc@sectlevel@paragraph{P}
2359 \@namedef{mtc@sectlevel@subparagraph}{SP}\def\mtc@sectlevel@subparagraph{SP}
2360 \@namedef{mtc@sectlevel@*}{ }\expandafter\def\csname mtc@sectlevel@*\endcsname{ }

```

`\mtcsetfont` The `\mtcsetfont` command has the following syntax:

```
\mtcsetfont{mini-table}{level-name}{font commands}
```

The *mini-table* type is a keyword like `minitoc`, the *level-name* is a sectionning level like `subsection` (no backslash). The *font commands* are a font specification, using NFSS [28] basic commands usually.

`\if@mtc@setfont@` First, we declare a flag, set true:

```
2361 \newif\if@mtc@setfont@\@mtc@setfont@true
```

`\mtcsetfont` Then, we begin the command, which has three arguments:

```
2362 \newcommand{\mtcsetfont}[3]{%
```

`\mtc@mta@abbrev` The two first arguments of this command are keywords. They must be translated into the effective strings. We process the first argument which is a keyword from the `typetable` family. The result is stored in `\mtc@mta@abbrev`. Example: if #1 is `minitoc`, we get `mtc`.

```

2363 \def\mtc@mta@abbrev{X}
2364 \@mtc@setfont@true
2365 \expandafter\ifx\csname mtc@typetable@#1\endcsname\relax
2366   \@mtc@setfont@false
2367   \def\mtc@mta@abbrev{X}
2368   \PackageError{minitoc}%
2369     {*** \string\mtcsetfont \space has a wrong first argument (#1).
2370     \MessageBreak
2371     It should be a mini-table type (parttoc...sectlot)}}%
2372   {Correct the source code.\MessageBreak
2373   Type <return> and rerun LaTeX}
2374 \else
2375   \edef\mtc@mta@abbrev{\@nameuse{mtc@typetable@#1}}
2376 \fi

```

`\mtc@level@abbrev` The second argument, a keyword from the family `sectlevel`, is processed the same way  
`\if@mtc@setfont@` and the result is stored into `\mtc@level@abbrev`. Example: if #2 is subsection, we  
`\@nameuse` get SS.

```

2377 \def\mtc@level@abbrev{X}
2378 \expandafter\ifx\csname mtc@sectlevel@#2\endcsname\relax
2379   \@mtc@setfont@false
2380   \def\mtc@level@abbrev{X}
2381   \PackageError{minitoc}%
2382     {*** \string\mtcsetfont \space has a wrong second argument (#2).
2383     \MessageBreak
2384     It should be a sectionning level (part...subparagraph) or * }%
2385   {Correct the source code.\MessageBreak
2386   Type <return> and rerun LaTeX}
2387 \else
2388   \edef\mtc@level@abbrev{\@nameuse{mtc@sectlevel@#2}}
2389 \fi

```

`\mtc@tmp@name` Then, we construct the effective macro to be applied:  
`\mtc@mta@abbrev`  
`\mtc@level@abbrev` 2390 \def\mtc@tmp@name{\mtc@mta@abbrev\mtc@level@abbrev font}

Example: if #1 is minitoc and #2 is subsection, we get `mtcSSfont`, which is the name of the command for the font of a subsection entry in a minitoc (the backslash is missing, but we will use a `\csname ... \endcsname` pair to apply the constructed command).

`\if@mtc@setfont@` But all combinaisons are not legal (the level of the entry must be lower than the level of the mini-table), so we must test:

```

2391 \expandafter\ifx\csname #1\endcsname\minitoc
2392   \expandafter\ifx\csname #2\endcsname\part\@mtc@setfont@false\fi
2393   \expandafter\ifx\csname #2\endcsname\chapter\@mtc@setfont@false\fi
2394 \fi
2395 \expandafter\ifx\csname #1\endcsname\minilof

```

```

2396 \expandafter\ifx\csname #2\endcsname\part\@mtc@setfont@false\fi
2397 \expandafter\ifx\csname #2\endcsname\chapter\@mtc@setfont@false\fi
2398 \fi
2399 \expandafter\ifx\csname #1\endcsname\minilof
2400 \expandafter\ifx\csname #2\endcsname\part\@mtc@setfont@false\fi
2401 \expandafter\ifx\csname #2\endcsname\chapter\@mtc@setfont@false\fi
2402 \fi
2403 \expandafter\ifx\csname #1\endcsname\secttoc
2404 \expandafter\ifx\csname #2\endcsname\part\@mtc@setfont@false\fi
2405 \expandafter\ifx\csname #2\endcsname\chapter\@mtc@setfont@false\fi
2406 \expandafter\ifx\csname #2\endcsname\section\@mtc@setfont@false\fi
2407 \fi
2408 \expandafter\ifx\csname #1\endcsname\sectlof
2409 \expandafter\ifx\csname #2\endcsname\part\@mtc@setfont@false\fi
2410 \expandafter\ifx\csname #2\endcsname\chapter\@mtc@setfont@false\fi
2411 \expandafter\ifx\csname #2\endcsname\section\@mtc@setfont@false\fi
2412 \fi
2413 \expandafter\ifx\csname #1\endcsname\sectlot
2414 \expandafter\ifx\csname #2\endcsname\part\@mtc@setfont@false\fi
2415 \expandafter\ifx\csname #2\endcsname\chapter\@mtc@setfont@false\fi
2416 \expandafter\ifx\csname #2\endcsname\section\@mtc@setfont@false\fi
2417 \fi

```

`\if@mtc@setfont@` If the combinaison is legal, we apply it, *i.e.* we redefine the meaning of the constructed  
`\mtc@tmp@name` macro with the sequence of commands given as third argument of `\mtcsetfont` and we  
`\mtc@mta@abbrev` log that event (we store the third argument in a token register to can print it *verbatim*).  
`\mtc@level@abbrev`  
`\mtc@toks` If the combinaison is not legal, an error message is displayed.

```

2418 \if@mtc@setfont@
2419 \def\mtc@tmp@name{\mtc@mta@abbrev\mtc@level@abbrev font}
2420 \mtc@toks{#3}
2421 \PackageInfo{minitoc}%
2422 {\string\mtcsetfont\space redefines the macro
2423 \mtc@tmp@name \space as "\the\mtc@toks"}
2424 \expandafter\def\csname\mtc@tmp@name\endcsname{#3}
2425 \else
2426 \PackageError{minitoc}%
2427 {\string\mtcsetfont\space has incompatible\MessageBreak
2428 first (#1) and second (#2) arguments}{}
2429 \fi
2430 }

```

### 5.65.3 The `\mtcsettitlefont` command

`\mtcsettitlefont` This command is very similar to the `\mtcsetfont` command. Its syntax is almost identical:

```
\mtcsettitlefont{mini-table}{font commands}
```

`\if@mtc@settitlefont@` The *mini-table* type is a keyword like `minitoc`. The *font commands* are a font specification, using NFSS [28] basic commands usually. The difference is the absence of the second keyword argument, because the *font commands* will be applied to the title of each mini-table of the given kind.

First, we declare a flag, set true:

```
2431 \newif\if@mtc@settitlefont@\@mtc@settitlefont@true
```

`\mtcsettitlefont` And we begin the definition of the `\mtcsettitlefont` command, which has two arguments:

```
2432 \newcommand{\mtcsettitlefont}[2]{%
```

`\mtc@mtatf@abbrev` We process the first argument, a keyword of the `typetitle` family, then the result is stored into `\mtc@mtatf@abbrev`:  
`\if@mtc@settitlefont@`  
`\@nameuse`

```
2433 \def\mtc@mtatf@abbrev{X}
2434 \@mtc@settitlefont@true
2435 \expandafter\ifx\csname mtc@typetitle@#1\endcsname\relax
2436   \@mtc@settitlefont@false
2437   \def\mtc@mtatf@abbrev{X}
2438   \PackageError{minitoc}%
2439     {*** \string\mtcsettitlefont \space has a wrong first argument (#1).
2440     \MessageBreak
2441     It should be a mini-table type (parttoc...sectlot)}%
2442     {Correct the source code.\MessageBreak
2443     Type <return> and rerun LaTeX}
2444 \else
2445   \edef\mtc@mtatf@abbrev{\@nameuse{mtc@typetitle@#1}}
2446 \fi
```

`\if@mtc@settitlefont@` Then we build the name of the effective command and apply this command:

```
\mtc@tmptf@name
\mtc@mtatf@abbrev 2447 \if@mtc@settitlefont@
\mtc@toks 2448   \def\mtc@tmptf@name{\mtc@mtatf@abbrev font}
2449   \mtc@toks{#2}
2450   \PackageInfo{minitoc}%
2451     {\string\mtcsettitlefont\space redefines the macro
2452     \mtc@tmptf@name \space as "\the\mtc@toks"}
2453   \expandafter\def\csname\mtc@tmptf@name\endcsname{#2}
2454 \else
2455   \PackageError{minitoc}%
2456     {Illegal type of table (#1)}%
2457     {Correct the source code.\MessageBreak
2458     Type <return> and rerun LaTeX}{\relax}
```



```
2459 \fi
2460 }
```

### 5.65.4 The `\mtcsettitle` command

`\mtcsettitle` This command is very similar to the `\mtcsettitlefont` command. Its syntax is almost identical:

```
\mtcsettitle{mini-table}{text}
```

The *mini-table* type is a keyword like `minitoc`. The *text* is the text for a mini-table title.

`\if@mtc@settitle@` First, we declare a flag, set true:

```
2461 \newif\if@mtc@settitle@\@mtc@settitle@true
```

`\mtcsettitle` Then we define the `\mtcsettitle` command, which has two arguments:

```
2462 \newcommand{\mtcsettitle}[2]{%
```

`\mtc@mtati@abbrev` We process the first argument, which is a keyword of the `typetable` family. The result  
`\if@mtc@settitle@` is stored in `\mtc@mtati@abbrev`:  
`\@nameuse`

```
2463 \def\mtc@mtati@abbrev{X}
2464 \@mtc@settitle@true
2465 \expandafter\ifx\csname mtc@typetable@#1\endcsname\relax
2466   \@mtc@settitle@false
2467   \def\mtc@mtati@abbrev{X}
2468   \PackageError{minitoc}%
2469     {*** \string\mtcsettitle \space has a wrong first argument (#1).
2470     \MessageBreak
2471     It should be a mini-table type (parttoc...sectlot)}%
2472   {Correct the source code.\MessageBreak
2473   Type <return> and rerun LaTeX}
2474 \else
2475   \edef\mtc@mtati@abbrev{\@nameuse{mtc@typetable@#1}}
2476 \fi
```

`\if@mtc@settitle@` And we construct the name of the effective macro and apply it:

```
\mtc@tmpti@name
\mtc@mtati@abbrev 2477 \if@mtc@settitle@
\mtc@toks 2478   \def\mtc@tmpti@name{\mtc@mtati@abbrev title}
2479   \mtc@toks{#2}
2480   \PackageInfo{minitoc}%
```

```

2481      {\string\mtcsettitle\space redefines the macro
2482       \mtc@tmpti@name \space as "\the\mtc@toks"}
2483      \expandafter\def\csname\mtc@tmpti@name\endcsname{#2}
2484 \else
2485   \PackageError{minitoc}%
2486     {Illegal type of table (#1)}%
2487     {Correct the source code.\MessageBreak
2488      Type <return> and rerun LaTeX}{\relax}
2489 \fi
2490 }

```

### 5.65.5 The `\mtcsetformat` command

`\@namedef` We define first the keywords (family `formatparam`) for the three formatting parameters that this command can alter:

```

2491 \@namedef{mtc@formatparam@dotinterval}{dotsep}%
2492 \def\mtc@arg@dotinterval{dotsep}
2493 \@namedef{mtc@formatparam@tocrightmargin}{tocrmarg}%
2494 \def\mtc@arg@tocrightmargin{tocrightmargin}
2495 \@namedef{mtc@formatparam@pagenumwidth}{pnumwidth}%
2496 \def\mtc@arg@pagenumwidth{\mtc@arg@pagenumwidth}
2497 %% \@namedef{mtc@arg@numwidth}{numwidth} %not yet available
2498 %% \def\mtc@arg@numwidth{\mtc@arg@numwidth} %not yet available

```

`\AtBeginDocument` The `\mtcsetformat` command needs an initialization to be done at the beginning of the document, to set the defaults values of the formatting parameters:

```

2499 \AtBeginDocument{%

```

`\@pnumwidth` We take, if possible, the default value of `\@pnumwidth` for each type of mini-tables:

```

\ptcpnumwidth
\mtcpnumwidth
\stcpnumwidth 2500 \@ifundefined{ptcpnumwidth}{\let\ptcpnumwidth\@pnumwidth}{}%
\plfpnumwidth 2501 \@ifundefined{stcpnumwidth}{\let\stcpnumwidth\@pnumwidth}{}%
\mlfpnumwidth 2502 \@ifundefined{mtcpnumwidth}{\let\mtcpnumwidth\@pnumwidth}{}%
\slfpnumwidth 2503 \@ifundefined{plfpnumwidth}{\let\plfpnumwidth\@pnumwidth}{}%
\pltpnumwidth 2504 \@ifundefined{mlfpnumwidth}{\let\mlfpnumwidth\@pnumwidth}{}%
\mltpnumwidth 2505 \@ifundefined{slfpnumwidth}{\let\slfpnumwidth\@pnumwidth}{}%
\sltpnumwidth 2506 \@ifundefined{pltpnumwidth}{\let\pltpnumwidth\@pnumwidth}{}%
                2507 \@ifundefined{mltpnumwidth}{\let\mltpnumwidth\@pnumwidth}{}%
                2508 \@ifundefined{sltpnumwidth}{\let\sltpnumwidth\@pnumwidth}{}%

```

```

\@tocrmarg We take, if possible, the default value of \@tocrmarg for each type of mini-tables:
\ptctocrmarg
\mtclofrmarg 2509 \@ifundefined{ptctocrmarg}{\let\ptctocrmarg\@tocrmarg}{}%
\stclotrarg 2510 \@ifundefined{mtctocrmarg}{\let\mtctocrmarg\@tocrmarg}{}%
\plftocrmarg 2511 \@ifundefined{stctocrmarg}{\let\stctocrmarg\@tocrmarg}{}%
\mlflofrmarg 2512 \@ifundefined{plftocrmarg}{\let\plftocrmarg\@tocrmarg}{}%
\slflotrarg 2513 \@ifundefined{mlftocrmarg}{\let\mlftocrmarg\@tocrmarg}{}%
\plttocrmarg 2514 \@ifundefined{slftocrmarg}{\let\slftocrmarg\@tocrmarg}{}%
\mltlofrmarg 2515 \@ifundefined{plttocrmarg}{\let\plttocrmarg\@tocrmarg}{}%
\sltlotrarg 2516 \@ifundefined{mlttocrmarg}{\let\mlttocrmarg\@tocrmarg}{}%
2517 \@ifundefined{slttocrmarg}{\let\slttocrmarg\@tocrmarg}{}%

```

```

\@dotsep We take, if possible, the default value of \@dotsep for each type of mini-tables:
\ptcdotsep
\mtcdotsep 2518 \@ifundefined{ptcdotsep}{\let\ptcdotsep\@dotsep}{}%
\stcdotsep 2519 \@ifundefined{mtcdotsep}{\let\mtcdotsep\@dotsep}{}%
\plfdotsep 2520 \@ifundefined{stcdotsep}{\let\stcdotsep\@dotsep}{}%
\mlfdotsep 2521 \@ifundefined{plfdotsep}{\let\plfdotsep\@dotsep}{}%
\slfdotsep 2522 \@ifundefined{mlfdotsep}{\let\mlfdotsep\@dotsep}{}%
\pltdotsep 2523 \@ifundefined{slfdotsep}{\let\slfdotsep\@dotsep}{}%
\mltdotsep 2524 \@ifundefined{pltdotsep}{\let\pltdotsep\@dotsep}{}%
2525 \@ifundefined{mltdotsep}{\let\mltdotsep\@dotsep}{}%
\sltdotsep 2526 \@ifundefined{sltdotsep}{\let\sltdotsep\@dotsep}{}%

```

And we terminate the \AtBeginDocument block:

```
2527 }
```

\mtcsetformat The executive part is done via the following macros, which are invoked in the \mtc@verse-like environments for each kind of mini-table. These commands activate the values recorded by \mtcsetformat.

\ptc@setform The \ptc@setform macro is invoked in ptc@verse to set format parameters:

```

ptc@verse
2528 \def\ptc@setform{%
2529 \let\@pnumwidth\ptcpnumwidth\relax
2530 \let\@tocrmarg\ptctocrmarg\relax
2531 \let\@dotsep\ptcdotsep\relax
2532 }

```

\mtc@setform The \mtc@setform macro is invoked in mtc@verse to set format parameters:

```

mtc@verse
2533 \def\mtc@setform{%
2534 \let\@pnumwidth\mtcpnumwidth\relax
2535 \let\@tocrmarg\mtctocrmarg\relax
2536 \let\@dotsep\mtcdotsep\relax
2537 }

```

`\stc@setform` The `\stc@setform` macro is invoked in `stc@verse` to set format parameters:  
`stc@verse`

```
2538 \def\stc@setform{%
2539 \let\@pnumwidth\stcpnumwidth\relax
2540 \let\@tocrmarg\stctocrmarg\relax
2541 \let\@dotsep\stcdotsep\relax
2542 }
```

`\plf@setform` The `\plf@setform` macro is invoked in `ptc@verse` to set format parameters:  
`ptc@verse`

```
2543 \def\plf@setform{%
2544 \let\@pnumwidth\plfpnumwidth\relax
2545 \let\@tocrmarg\plftocrmarg\relax
2546 \let\@dotsep\plfdotsep\relax
2547 }
```

`\mlf@setform` The `\mlf@setform` macro is invoked in `mtc@verse` to set format parameters:  
`mtc@verse`

```
2548 \def\mlf@setform{%
2549 \let\@pnumwidth\mlfpnumwidth\relax
2550 \let\@tocrmarg\mlftocrmarg\relax
2551 \let\@dotsep\mlfdotsep\relax
2552 }
```

`\slf@setform` The `\slf@setform` macro is invoked in `stc@verse` to set format parameters:  
`stc@verse`

```
2553 \def\slf@setform{%
2554 \let\@pnumwidth\slfpnumwidth\relax
2555 \let\@tocrmarg\slftocrmarg\relax
2556 \let\@dotsep\slfdotsep\relax
2557 }
```

`\plt@setform` The `\plt@setform` macro is invoked in `ptc@verse` to set format parameters:  
`ptc@verse`

```
2558 \def\plt@setform{%
2559 \let\@pnumwidth\pltpnumwidth\relax
2560 \let\@tocrmarg\plttocrmarg\relax
2561 \let\@dotsep\pltdotsep\relax
2562 }
```

`\mlt@setform` The `\mlt@setform` macro is invoked in `mtc@verse` to set format parameters:  
`mtc@verse`

```
2563 \def\mlt@setform{%
2564 \let\@pnumwidth\plfpnumwidth\relax
2565 \let\@tocrmarg\plftocrmarg\relax
2566 \let\@dotsep\plfdotsep\relax
2567 }
```

`\slt@setform` The `\slt@setform` macro is invoked in `stc@verse` to set format parameters:  
`stc@verse`

```
2568 \def\slt@setform{%
2569 \let\@pnumwidth\plfpnumwidth\relax
2570 \let\@tocrmarg\plftocrmarg\relax
2571 \let\@dotsep\plfdotsep\relax
2572 }
```

`\if@mtc@setformat@` We now define a flag and the `\mtcsetformat` command, which has the following syntax:  
`\mtcsetformat`

`\mtcsetformat{mini-table}{parameter-name}{value}`

where *mini-table* is a keyword of the `typetable` family, *parameter-name* is a keyword of the `formatparam` family and *value*, the value of this parameter for the given kind of mini-table.

```
2573 \newif\if@mtc@setformat@\@mtc@setformat@true
2574 \newcommand{\mtcsetformat}[3]{%
```

`\mtc@mtf@abbrev` We now process the first argument and store the result in `\mtc@mtf@abbrev`:  
`\mtc@fparam@abbrev`

```
2575 \def\mtc@mtf@abbrev{X}
2576 \def\mtc@fparam@abbrev{X}
2577 \@mtc@setformat@true
2578 \expandafter\ifx\csname mtc@typetable@#1\endcsname\relax
2579   \@mtc@setformat@false
2580   \def\mtc@mtf@abbrev{X}
2581   \PackageError{minitoc}%
2582     {*** \string\mtcsetformat \space has a wrong first argument (#1).
2583     \MessageBreak
2584     It should be a mini-table type (parttoc...sectlot)}%
2585     {Correct the source code.\MessageBreak
2586     Type <return> and rerun LaTeX}
2587 \else
2588   \edef\mtc@mtf@abbrev{\@nameuse{mtc@typetable@#1}}
2589 \fi
```

`\mtc@fparam@abbrev` Then the second argument is processed and the result is stored in `\mtc@fparam@abbrev`:  
`\@nameuse`

```
2590 \expandafter\ifx\csname mtc@formatparam@#2\endcsname\relax
2591   \@mtc@setformat@false
2592   \def\mtc@fparam@abbrev{X}
2593   \PackageError{minitoc}%
2594     {*** \string\mtcsetformat \space has a wrong second argument (#2).
2595     \MessageBreak
2596     It should be a formatting param
2597     (pagenumwidth, tocrighthmargin, dotinterval)}%
2598     {Correct the source code.\MessageBreak
```

```

2599      Type <return> and rerun LaTeX}
2600 \else
2601   \edef\mtc@fparam@abbrev{\@nameuse{mtc@formatparam@#2}}
2602 \fi

```

```

\if@mtc@setformat@ The name of the storage macro is build and it receives the third parameter as value:
  \mtc@tmpfm@name
  \mtc@mtf@abbrev 2603 \if@mtc@setformat@
\mtc@fparam@abbrev 2604   \def\mtc@tmpfm@name{\mtc@mtf@abbrev\mtc@fparam@abbrev}
2605   \PackageInfo{minitoc}%
2606     {\string\mtcsetformat\space redefines
2607      the macro \mtc@tmpfm@name \space as \string#3}
2608   \expandafter\def\csname\mtc@tmpfm@name\endcsname{#3}
2609 \else
2610   \PackageError{minitoc}%
2611     {\string\mtcsetformat\space has incompatible
2612      \MessageBreak
2613      first (#1) and second (#2) arguments}{}
2614 \fi
2615 }

```

### 5.65.6 The `\mtcsetpagenumbers` command

This command activates or inhibits page numbers in the mini-tables of a given kind. Its syntax is the following:

```
\mtcsetpagenumbers{mini-table}{on|off}
```

where *mini-table* is a keyword for a kind of mini-table (`parttoc`, ... `sectlot`), or `on` and `off` a keyword to activate (`on`) or inhibit (`off`) the page numbers. `on` and `off` have many synonyms.

```

\if@mtc@setpagenumbers@ We define some flags:
  \if@mtc@spn@ok@
2616 \newif\if@mtc@setpagenumbers@ \@mtc@setpagenumbers@false
2617 \newif\if@mtc@spn@ok@

```

`\mtcsetpagenumber` Then the `\mtcsetpagenumbers` macro, with two arguments:

```
2618 \newcommand{\mtcsetpagenumbers}[2]{%
```

```

\mtc@mttpn@abbrev We process the first argument, a keyword of the typetable family, and store the result
\mtc@pnsw@abbrev  in \mtc@mttpn@abbrev:

```

```

2619 \def\mtc@mttprn@abbrev{X}
2620 \@mtc@setpagenumbers@true
2621 \def\mtc@pnsr@abbrev{}
2622 \expandafter\ifx\csname mtc@typetable@#1\endcsname\relax
2623   \@mtc@setpagenumbers@false
2624   \def\mtc@pnsr@abbrev{X}
2625   \def\mtc@mttprn@abbrev{X}
2626   \PackageError{minitoc}%
2627     {*** \string\mtcsetpagenumbers \space has a wrong first argument (#1).}%
2628     {It should be a mini-table type (parttoc...sectlot)\MessageBreak
2629       Correct the source code, type <return> and rerun LaTeX}
2630 \else
2631   \edef\mtc@mttprn@abbrev{\@nameuse{mtc@typetable@#1}}
2632 \fi

```

```

\if@mtc@spn@ok@ Then the second argument, a keyword of the YN family, and store the result in
\mtc@pnsr@abbrev \mtc@pnsr@abbrev. The name of the effective macro is build and the macro executed.
\mtc@mttprn@abbrev
\if@mtc@setpagenumbers@ 2633 \@mtc@spn@ok@true
\mtc@tmppn@name 2634 \expandafter\ifx\csname mtc@YN@#2\endcsname\relax
2635   \@mtc@spn@ok@false
2636   \def\mtc@pnsr@abbrev{X}
2637   \def\mtc@mttprn@abbrev{X}
2638   \@mtc@setpagenumbers@false
2639   \def\mtc@mttprn@abbrev{X}
2640   \PackageError{minitoc}%
2641     {*** \string\mtcsetpagenumbers \space has a wrong second argument (#2).}%
2642     {It should be a boolean value (0/1, yes/no, on/off)\MessageBreak
2643       Correct the source code, type <return> and rerun LaTeX}
2644 \else
2645   \edef\mtc@pnsr@abbrev{\@nameuse{mtc@YN@#2}}
2646   \def\mtc@pnsr@abbrevX{X}
2647   \def\mtc@noX{mtc@noX}
2648   \def\mtc@tmppn@name{\mtc@pnsr@abbrev\mtc@mttprn@abbrev pagenumbers}
2649   \expandafter\ifx\csname mtc@\mtc@pnsr@abbrev X\endcsname\mtc@noX
2650     \PackageInfo{minitoc}%
2651       {Page numbers are inhibited for the #1s}
2652   \else
2653     \PackageInfo{minitoc}%
2654       {Page numbers are activated for the #1s}
2655   \fi
2656   \csname\mtc@tmppn@name\endcsname{}
2657 \fi
2658 }

```

### 5.65.7 The \mtcsetrules command

This macro is very similar to \mtcsetpagenumbers and its syntax is the same:

```
\mtcsetrules{mini-table}{on|off}
```

where *mini-table* is a keyword for a kind of mini-table (`parttoc`, ... `sectlot`), or `on` and `off` a keyword to activate (on) or inhibit (off) the horizontal rules. `on` and `off` have many synonyms.

Hence the code is similar.

```
\if@mtc@setrules@ We define some flags:
\if@mtc@sru@ok@
2659 \newif\if@mtc@setrules@ \@mtc@setrules@false
2660 \newif\if@mtc@sru@ok@

\mtcsetrules Then the \mtcsetrules macro, which has two arguments:

2661 \newcommand{\mtcsetrules}[2]{%

\mtc@mttru@abbrev We process the first argument, a keyword of the typetable family and store the result in
\if@mtc@setrules@ \mtc@mttru@abbrev:
\mtc@rusw@abbrev
\@nameuse 2662 \def\mtc@mttru@abbrev{X}
2663 \@mtc@setrules@true
2664 \def\mtc@rusw@abbrev{}
2665 \expandafter\ifx\csname mtc@typetable@#1\endcsname\relax
2666   \@mtc@setrules@false
2667   \def\mtc@rusw@abbrev{X}
2668   \def\mtc@mttru@abbrev{X}
2669   \PackageError{minitoc}%
2670     {*** \string\mtcsetrules \space has a wrong first argument (#1).}%
2671     {It should be a mini-table type (parttoc...sectlot)}
2672     \MessageBreak
2673     Correct the source code, type <return> and rerun LaTeX}
2674 \else
2675   \edef\mtc@mttru@abbrev{\@nameuse{mtc@typetable@#1}}
2676 \fi

\if@mtc@sru@ok@ Then the second argument, a keyword of the YN family, and store the result in
\mtc@rusw@abbrev \mtc@rusw@abbrev. The name of the effective macro is build and the macro executed.
\mtc@mttru@abbrev
if@setrules@false 2677 \@mtc@sru@ok@true
\mtc@noX 2678 \expandafter\ifx\csname mtc@YN@#2\endcsname\relax
\mtc@tmppn@name 2679   \@mtc@sru@ok@false
2680   \def\mtc@rusw@abbrev{X}
2681   \def\mtc@mttru@abbrev{X}
2682   \@mtc@setrules@false
2683   \PackageError{minitoc}%
2684     {*** \string\mtcsetrules \space has a wrong second argument (#2).}%
2685     {It should be a boolean value (0/1, yes/no, on/off}
```



```

2686      \MessageBreak
2687      Correct the source code, type <return> and rerun LaTeX}
2688 \else
2689   \edef\mtc@rusw@abbrev{\@nameuse{mtc@YN@#2}}
2690   \def\mtc@rusw@abbrevX{X}
2691   \def\mtc@noX{mtc@noX}
2692   \def\mtc@tmppn@name{\mtc@rusw@abbrev\mtc@mttru@abbrev rule}
2693   \expandafter\ifx\csname mtc@\mtc@rusw@abbrev X\endcsname\mtc@noX
2694     \PackageInfo{minitoc}%
2695     {Horizontal rules are inhibited for the #1s}
2696   \else
2697     \PackageInfo{minitoc}%
2698     {Horizontal rules are activated for the #1s}
2699   \fi
2700   \csname\mtc@tmppn@name\endcsname{}
2701 \fi
2702 }

```

### 5.65.8 The `\mtcsetfeature` command

For this command, we must define three families of keywords, but the third is just used to add the word “style” for the “pagestyle” when “thispage” is used.

A family for the long names of the types of mini-tables:

```

2703 \@namedef{mtc@ltypetable@parttoc}{parttoc}\def\mtc@ltypetable@parttoc{parttoc}
2704 \@namedef{mtc@ltypetable@partlot}{partlot}\def\mtc@ltypetable@partlot{partlot}
2705 \@namedef{mtc@ltypetable@partlof}{partlof}\def\mtc@ltypetable@partlof{partlof}
2706 \@namedef{mtc@ltypetable@minitoc}{minitoc}\def\mtc@ltypetable@minitoc{minitoc}
2707 \@namedef{mtc@ltypetable@minilot}{minilot}\def\mtc@ltypetable@minilot{minilot}
2708 \@namedef{mtc@ltypetable@minilof}{minilof}\def\mtc@ltypetable@minilof{minilof}
2709 \@namedef{mtc@ltypetable@secttoc}{secttoc}\def\mtc@ltypetable@secttoc{secttoc}
2710 \@namedef{mtc@ltypetable@sectlof}{sectlof}\def\mtc@ltypetable@sectlof{sectlof}
2711 \@namedef{mtc@ltypetable@sectlot}{sectlot}\def\mtc@ltypetable@sectlot{sectlot}

```

A family for the type of feature:

```

2712 \@namedef{mtc@featureparam@before}{before}%
2713   \def\mtc@featureparam@before{before}
2714 \@namedef{mtc@featureparam@after}{after}%
2715   \def\mtc@featureparam@after{after}
2716 \@namedef{mtc@featureparam@pagestyle}{thispage}%
2717   \def\mtc@featureparam@pagestyle{thispage}

```

And a family to add “style” if it is a “pagestyle” feature:

```

2718 \@namedef{mtc@ft3@before}{}\expandafter\def\csname mtc@ft3@before\endcsname{}
2719 \@namedef{mtc@ft3@after}{}\expandafter\def\csname mtc@ft3@after\endcsname{}
2720 \@namedef{mtc@ft3@pagestyle}{style}%

```

```
2721 \expandafter\def\csname mtc@ft3@pagestyle\endcsname{style}
```

The `\mtcsetfeature` command has the following syntax:

```
\mtcsetfeature{mini-table}{feature-name}{commands}
```

where *mini-table* is a keyword of the `ltypetable` family, *feature-name* is a keyword of the `featureparam` family (but also of the `ft3` family), and *commands* are the commands which constitute the selected feature.

```
\if@mtc@setfeature@ We define a flag and the \mtcsetfeature command, with three arguments:
\mtcsetfeature
```

```
2722 \newif\if@mtc@setfeature@\@mtc@setfeature@true
2723 \newcommand{\mtcsetfeature}[3]{%
```

```
\mtc@mtfeat@abbrev We process the first argument, a keyword of the ltypetable family and store the result
\mtc@featparam@abbrev in \mtc@mtfeat@abbrev:
```

```
2724 \def\mtc@mtfeat@abbrev{X}
2725 \def\mtc@featparam@abbrev{X}
2726 \@mtc@setfeature@true
2727 \expandafter\ifx\csname mtc@ltypetable@#1\endcsname\relax
2728   \@mtc@setfeature@false
2729   \def\mtc@mtfeat@abbrev{X}
2730   \PackageError{minitoc}%
2731     {*** \string\mtcsetfeature \space has a wrong first argument (#1).
2732     \MessageBreak
2733     It should be a mini-table type (parttoc...sectlot)}%
2734   {Correct the source code.\MessageBreak
2735     Type <return> and rerun LaTeX}
2736 \else
2737   \edef\mtc@mtfeat@abbrev{\@nameuse{mtc@ltypetable@#1}}
2738 \fi
```

```
\if@mtc@setfeature@ The second argument is a keyword of the featureparam family, the result is stored in
\mtc@featparam@abbrev \mtc@featparam@; and the complement is computed from the first argument, interpreted
\mtc@featparam@third as a keyword of the ft3 family and whose result is stored in \mtc@featparam@third.
\@nameuse
```

```
2739 \expandafter\ifx\csname mtc@featureparam@#2\endcsname\relax
2740   \@mtc@setfeature@false
2741   \def\mtc@featparam@abbrev{X}
2742   \def\mtc@featparam@third{X}
2743   \PackageError{minitoc}%
2744     {*** \string\mtcsetfeature \space has a wrong second argument (#2).
2745     \MessageBreak
2746     It should be a feature param (before, after, pagestyle)}%
2747   {Correct the source code.\MessageBreak
2748     Type <return> and rerun LaTeX}
```

```

2749 \else
2750   \edef\mtc@featparam@abbrev{\@nameuse{mtc@featureparam@#2}}
2751   \edef\mtc@featparam@third{\@nameuse{mtc@ft3@#2}}
2752 \fi

\if@mtc@setfeature@ The name of the effective macro is built by concatenating these three pieces (named
\mtc@tmpfeat@name \mtc@featparam@abbrev, \mtc@mtfeat@abbrev, and \mtc@featparam@third re-
\mtc@featparam@abbrev spectively), then this macro is executed:
\mtc@mtfeat@abbrev
\mtc@featparam@third
2753 \if@mtc@setfeature@
2754   \def\mtc@tmpfeat@name%
2755     {\mtc@featparam@abbrev\mtc@mtfeat@abbrev\mtc@featparam@third}
2756   \PackageInfo{minitoc}%
2757   {\string\mtcsetfeature\space redefines the macro
2758    "\csname mtc@tmpfeat@name\endcsname" as "\string#3"}
2759   \expandafter\def\csname\mtc@tmpfeat@name\endcsname{#3}
2760 \else
2761   \PackageError{minitoc}%
2762     {\string\mtcsetfeature\space has incompatible\MessageBreak
2763      first (#1) and second (#2) arguments}{}
2764 \fi}

```

### 5.65.9 The `\mtcsetdepth` command

This command is very similar to the `\mtcsettitle` command. Its syntax is almost identical:

```
\mtcsetdepth{mini-table}{depth}
```



The *mini-table* type is a keyword like `minitoc`. The *depth* is the depth for a mini-table. If it is a mini-table for a list of figures or tables, the corresponding depth counter *must be available*, i.e. must have been created (often by an adequate package, like the `subfig` package [13]).

`\if@mtc@setdepth@` First, we declare a flag, set true:

```
2765 \newif\if@mtc@setdepth@\@mtc@setdepth@true
```

`\mtcsetdepth` Then we define the `\mtcsetdepth` command, with two arguments:

```
2766 \newcommand{\mtcsetdepth}[2]{%
```

`\mtc@mtade@abbrev` We process the first argument, which is a keyword of the `ltypetable` family. The result  
`\if@mtc@setdepth@` is stored in `\mtc@mtade@abbrev`:  
`\@nameuse`

```

2767 \def\mtc@mtade@abbrev{X}
2768 \@mtc@setdepth@true
2769 \expandafter\ifx\csname mtc@ltypetable@#1\endcsname\relax
2770   \@mtc@setdepth@false
2771   \def\mtc@mtade@abbrev{X}
2772   \PackageError{minitoc}%
2773     {*** \string\mtcsetdepth \space has a wrong first argument (#1).
2774     \MessageBreak
2775     It should be a mini-table type (parttoc...sectlot)}%
2776     {Correct the source code.\MessageBreak
2777     Type <return> and rerun LaTeX}
2778 \else
2779   \edef\mtc@mtade@abbrev{\@nameuse{mtc@ltypetable@#1}}
2780 \fi

```

```

\if@mtc@setdepth@ And we construct the name of the effective counter and gave it the value:
\mtc@tmpde@name
\mtc@mtade@abbrev 2781 \if@mtc@setdepth@
\mtc@toks 2782   \def\mtc@tmpde@name{\mtc@mtade@abbrev depth}
\setcounter 2783   \expandafter\noexpand\@ifundefined{c@\mtc@mtade@abbrev depth}%
2784     {\PackageError{minitoc}%
2785     {*** \string\mtcsetdepth \space attempt to use\MessageBreak
2786     an undefined counter (#1depth).}%
2787     {Correct the source code.\MessageBreak
2788     Type <return> and rerun LaTeX}}{%
2789     \mtc@toks{#2} % trick for explicit message using \the.
2790     \PackageInfo{minitoc}%
2791     {\string\mtcsetdepth\space redefines the counter
2792     \mtc@tmpde@name \space as "\the\mtc@toks"}}%
2793     \expandafter\setcounter{\mtc@tmpde@name}{#2}}%
2794 \else
2795   \PackageError{minitoc}%
2796     {\string\mtcsetdepth:\space Illegal type of table (#1)}%
2797     {Correct the source code.\MessageBreak
2798     Type <return> and rerun LaTeX}{\relax}
2799 \fi}% end of \mtcsettitle

```

## 5.66 The mtchideinmaintoc environment

\if@mtc@Himtoc@ The flag \if@mtc@Himtoc@ is used to detect an incorrect imbrication of this environment:

```

2800 \newif\if@mtc@Himtoc@ \@mtc@Himtoc@false

```

```

mtchideinmaintoc Then we define this environment, which inserts into the TOC file commands of the
\if@mtc@Himtoc@ form \setcounter{tocdepth}{...}. It also save and restore the value of the counter
\mtc@sv@tocdepth tocdepth, as the optional argument is the hiding depth of the entries in the main TOC.
\arabic
\addtocontents 2801 \newenvironment{mtchideinmaintoc}[1][-1]%
\setcounter 2802 {\if@mtc@Himtoc@\PackageError{minitoc}%
2803 {Imbrication of mtchideinmaintoc environments}%
2804 {The hiding in main ToC could be incorrect}\fi
2805 \global\@mtc@Himtoc@true
2806 \def\mtc@sv@tocdepth{\arabic{tocdepth}}%
2807 \addtocontents{toc}{\protect\setcounter{tocdepth}{#1}}%
2808 {\if@mtc@Himtoc@\else\PackageError{minitoc}%
2809 {Unbalanced mtchideinmaintoc environment}%
2810 {The hiding in main ToC could be incorrect}\fi
2811 \global\@mtc@Himtoc@false
2812 \addtocontents{toc}{\protect\setcounter{tocdepth}{\mtc@sv@tocdepth}}}

\AtBeginDocument The mtchideinmainlof and mtchideinmainlot environments are similar, but we
\if@mtc@Himlof@ must verify the presence of the associated depth counter, so we have two versions of
mtchideinmainlof each of these environments. This must be done after the loading of the packages.
\mtc@sv@lofdepth
\mtc@sv@tocdepth First, for the list of figures:
\mtc@svf@tocdepth
\arabic 2813 \newif\if@mtc@Himlof@ \@mtc@Himlof@false
\addtocontents 2814 \AtBeginDocument{%
\setcounter 2815 \@ifundefined{c@lofdepth}{%
2816 \newenvironment{mtchideinmainlof}[1][-1]%
2817 {\if@mtc@Himlof@\PackageError{minitoc}%
2818 {Imbrication of mtchideinmainlof environments}%
2819 {The hiding in main LoF could be incorrect}\fi
2820 \global\@mtc@Himlof@true
2821 \def\mtc@sv@tocdepth{\arabic{tocdepth}}%
2822 \def\mtc@sv@lofdepth{\arabic{tocdepth}}%
2823 \addtocontents{lof}{\protect\setcounter{tocdepth}{#1}}%
2824 {\if@mtc@Himlof@\else\PackageError{minitoc}%
2825 {Unbalanced mtchideinmainlof environment}%
2826 {The hiding in main LoF could be incorrect}\fi
2827 \global\@mtc@Himtoc@false
2828 \addtocontents{lof}{\protect\setcounter{tocdepth}{\mtc@sv@lofdepth}}}
2829}%
2830}%
2831 \newenvironment{mtchideinmainlof}[1][-1]%
2832 {\if@mtc@Himlof@\PackageError{minitoc}%
2833 {Imbrication of mtchideinmainlof environments}%
2834 {The hiding in main LoF could be incorrect}\fi
2835 \global\@mtc@Himlof@true
2836 \def\mtc@svf@tocdepth{\arabic{tocdepth}}%
2837 \addtocontents{lof}{\protect\setcounter{tocdepth}{#1}}%
2838 {\if@mtc@Himlof@\else\PackageError{minitoc}%
2839 {Unbalanced mtchideinmainlof environment}%
2840 {The hiding in main LoF could be incorrect}\fi
2841 \global\@mtc@Himlof@false

```

```

2842 \addtocontents{lof}{\protect\setcounter{tocdepth}{\mtc@svf@tocdepth}}}}

\AtBeginDocument Then for the list of tables:
\if@mtc@Himlot@
mtchideinmainlot 2843 \newif\if@mtc@Himlot@ \@mtc@Himlot@false
\mtc@sv@lotdepth 2844 \AtBeginDocument{%
\mtc@sv@tocdepth 2845 \@ifundefined{c@lotdepth}{%
\mtc@svt@tocdepth 2846 \newenvironment{mtchideinmainlot}[1][-1]%
\arabic 2847 {\if@mtc@Himlot@\PackageError{minitoc}%
\addtocontents 2848 {Imbrication of mtchideinmainlot environments}%
\setcounter 2849 {The hiding in main LoT could be incorrect}\fi
2850 \global\@mtc@Himlot@true
2851 \def\mtc@sv@tocdepth{\arabic{tocdepth}}%
2852 \def\mtc@sv@lotdepth{\arabic{tocdepth}}%
2853 \addtocontents{lot}{\protect\setcounter{tocdepth}{#1}}}%
2854 {\if@mtc@Himlot@\else\PackageError{minitoc}%
2855 {Unbalanced mtchideinmainlot environment}%
2856 {The hiding in main LoT could be incorrect}\fi
2857 \global\@mtc@Himlot@false
2858 \addtocontents{lot}{\protect\setcounter{tocdepth}{\mtc@sv@lotdepth}}}
2859 }%
2860 {%
2861 \newenvironment{mtchideinmainlot}[1][-1]%
2862 {\if@mtc@Himlot@\PackageError{minitoc}%
2863 {Imbrication of mtchideinmainlot environments}%
2864 {The hiding in main LoT could be incorrect}\fi
2865 \global\@mtc@Himlot@true
2866 \def\mtc@svt@tocdepth{\arabic{tocdepth}}%
2867 \addtocontents{lot}{\protect\setcounter{tocdepth}{#1}}}%
2868 {\if@mtc@Himlot@\else\PackageError{minitoc}%
2869 {Unbalanced mtchideinmainlot environment}%
2870 {The hiding in main LoT could be incorrect}\fi
2871 \global\@mtc@Himlot@false
2872 \addtocontents{lot}{\protect\setcounter{tocdepth}{\mtc@svt@tocdepth}}}}

```

## 5.67 Fixing the “Index” entry in the TOC

This macro is complex. Its syntax is:

```
\mtcfixindex[part|chapter|section]
```

\ifundefined Depending of the document class, the “Index” entry in the TOC is treated as a starred chapter or a starred section. Hence we must first determine the default value of the optional argument. The default value is then stored in the macro `\mtc@ixfix@level`. This is done by the following code, which eventually gives a warning message:

```

2873 \ifundefined{chapter}{%
2874   \ifundefined{section}%
2875     {\PackageWarningNoLine{minitoc}%
2876      {\string\chapter\space and \string\section\space are undefined.%
2877       \MessageBreak Cannot use \string\mtcfixindex\space %
2878        without optional argument [part]}}%
2879     \ifundefined{part}%
2880       {\PackageError{minitoc}%
2881        {But \string\part\space is undefined}%
2882         {\string\mtcfixindex\space not usable}}%
2883       {\PackageWarningNoLine{minitoc}%
2884        {\string\mtcfixindex\space can only be used with the [part]
2885         optional argument, which becomes the default}%
2886        \def\mtc@ixfix@level{part}}%
2887       }%
2888       {\def\mtc@ixfix@level{section}}}%
2889 {\def\mtc@ixfix@level{chapter}}

```

\if@mtcfixindex@ Then we define a flag and the command \mtcfixindex, which adds the necessary lines  
 \mtcfixindex in the TOC, the LOF and the LOT.  
 \addcontentsline

```

2890 \newif\if@mtcfixindex@ \@mtcfixindex@false
2891 \newcommand{\mtcfixindex}[1][\mtc@ixfix@level]{%
2892   \if@mtcfixindex@false
2893     \expandafter%
2894     \ifx\csname #1\endcsname\part\relax\@mtcfixindex@true\fi
2895     \expandafter%
2896     \ifx\csname #1\endcsname\chapter\relax\@mtcfixindex@true\fi
2897     \expandafter%
2898     \ifx\csname #1\endcsname\section\relax\@mtcfixindex@true\fi
2899     \if@mtcfixindex@
2900       \addcontentsline{lof}{x\mtc@ixfix@level}{}%
2901       \addcontentsline{lot}{x\mtc@ixfix@level}{}%
2902       \csname mtcadd\mtc@ixfix@level\endcsname\relax
2903     \else
2904       \PackageError{minitoc}%
2905       {The optional argument of \string\mtcfixindex\space is wrong}%
2906       {It must be omitted (\mtc@ixfix@level),
2907        or be part, chapter or section}%
2908     \fi
2909 }%

```

## 5.68 Fixing the “Glossary” entry in the TOC

This macro is complex. Its syntax is:

```
\mtcfixglossary[part|chapter|section]
```

`\@ifundefined` Depending of the document class, the “Glossary” entry in the TOC is treated as a starred chapter or a starred section. Hence we must first determine the default value of the optional argument. The default value is then stored in the macro `\mtc@glofix@level`. This is done by the following code, which eventually gives a warning message:

```

2910 \@ifundefined{chapter}{%
2911   \@ifundefined{section}%
2912     {\PackageWarningNoLine{minitoc}%
2913      {\string\chapter\space and \string\section\space are undefined.%
2914       \MessageBreak Cannot use \string\mtcfixglossary\space %
2915        without optional argument [part]}}%
2916   \@ifundefined{part}%
2917     {\PackageError{minitoc}%
2918      {But \string\part\space is undefined}%
2919      {\string\mtcfixglossary\space not usable}}%
2920   {\PackageWarningNoLine{minitoc}%
2921    {\string\mtcfixglossary\space can only be used with the [part]
2922     optional argument, which becomes the default}%
2923    \def\mtc@glofix@level{part}}%
2924   }%
2925   {\def\mtc@glofix@level{section}}}%
2926 {\def\mtc@glofix@level{chapter}}

```

`\if@mtcfixglossary@` Then we define a flag (`\if@mtcfixglossary@`) and the command `\mtcfixglossary`,  
`\mtcfixglossary` which adds the necessary lines in the TOC, the LOF and the LOT.  
`\addcontentsline`

```

2927 \newif\if@mtcfixglossary@ \@mtcfixglossary@false
2928 \newcommand{\mtcfixglossary}[1][\mtc@glofix@level]{%
2929   \@mtcfixglossary@false
2930   \expandafter%
2931   \ifx\csname #1\endcsname\part\relax\@mtcfixglossary@true\fi
2932   \expandafter%
2933   \ifx\csname #1\endcsname\chapter\relax\@mtcfixglossary@true\fi
2934   \expandafter%
2935   \ifx\csname #1\endcsname\section\relax\@mtcfixglossary@true\fi
2936   \if@mtcfixglossary@
2937     \addcontentsline{lof}{x\mtc@glofix@level}{}%
2938     \addcontentsline{lot}{x\mtc@glofix@level}{}%
2939     \csname mtcadd\mtc@glofix@level\endcsname\relax
2940   \else
2941     \PackageError{minitoc}%
2942       {The optional argument of \string\mtcfixglossary\space is wrong}%
2943       {It must be omitted (\mtc@glofix@level),
2944        or be part, chapter or section}%
2945   \fi
2946 }%

```



## 5.69 The `\mtcselectlanguage` command

`\mtcselectlanguage` This command loads a minitoc language definition file *language.mld* to set the language-dependent titles for the mini-tables. But first, we verify that this file exists. The flag `\if@mtc@insellang@` `\IfFileExists` `\if@mtc@insellang@` is true while we are in this macro.

```

\@input
2947 \newif\if@mtc@insellang@ \@mtc@insellang@false
2948 \def\mtcselectlanguage#1{%
2949     \InputIfFileExists{#1.mld}%
2950     {\PackageInfo{minitoc}{#1 language selected.%
2951       \MessageBreak}}%
2952     {\PackageError{minitoc}%
2953       {#1 is not a known language, \MessageBreak
2954         #1.mld not found. \MessageBreak
2955         Command ignored}%
2956       {See the minitoc documentation. \MessageBreak
2957         Correct the source using a valid language name. \MessageBreak
2958         Press RETURN}}}%
2959 }
```

## 5.70 The `\mtcloadmlo` internal command

`\mtcloadmlo` This command loads a minitoc language object file *language.mlo* to set the language-dependent titles for the mini-tables when exotic characters are needed. This command is used only in some .mld files when the title strings can not be generated by the normal processing of minitoc.dtx. The .mlo files are generated by filecontents environments in the minitoc.ins file. But first, we verify that this .mlo file exists.



This command should not be directly invoked by the user. This is verified via the flag `\if@mtc@insellang@`.

```

2960 \def\mtcloadmlo#1{%
2961     \if@mtc@insellang@
2962     \InputIfFileExists{#1.mlo}%
2963     {\PackageInfo{minitoc}{#1 minitoc language object selected.%
2964       \MessageBreak}}%
2965     {\PackageError{minitoc}%
2966       {#1 is not a known minitoc language object file (.mlo), \MessageBreak
2967         #1.mlo not found. \MessageBreak
2968         Command ignored}%
2969       {See the minitoc documentation. \MessageBreak
2970         Correct the source using a valid language name. \MessageBreak
2971         Press RETURN}}}%
2972     \else
2973     \PackageError{minitoc}%
2974       {You are using the \string\mtcloadmlo\space command\MessageBreak
2975         outside of a .mld file}%
2976       {It will be ignored}
```

```

2977      \@mtc@insellang@false
2978      \fi
2979 }

```

## 5.71 The “coffee breaks”

`\addcoffeeline` For the minutes package [30] (by Knut LICKERT), we need some commands to insert special entries, undotted, in the TOC to mark “coffee breaks” in a conference. Hence we define `\addcoffeeline`, `\coffeeline` and `\l@coffee`, and internal commands analog to the standard internal commands to format the TOC.

```

\@Undottedtocline
2980 \def\addcoffeeline#1#2#3{%
2981   \addtocontents{#1}{\protect\coffeeline{#2}{#3}{\null}}}
2982 \def\coffeeline#1{\csname l@#1\endcsname}
2983 \newcommand*\l@coffee{\@Undottedtocline{1}{1.5em}{2.3em}}

```

## 5.72 Initialization of counters

`\AtBeginDocument` At the beginning of the document, we initialize the absolute counters for parts, chapters and sections, if they are defined.

```

\@ifundefined
\setcounter
2984 \AtBeginDocument{%
2985   \@ifundefined{c@ptc}{\setcounter{ptc}{0}}
2986   \@ifundefined{c@mtc}{\setcounter{mtc}{0}}
2987   \@ifundefined{c@stc}{\setcounter{stc}{0}}

```

## 5.73 Declarations for simple options

These options are just setting a flag.

### 5.73.1 Options `tight` and `loose`, `k-tight` and `k-loose`

`\DeclareOption` These options influence the interline separation in the mini-tables.

```

\iftightmtc
\kiftightmtc
2988 \DeclareOption{tight}{\tightmtctrue}
2989 \DeclareOption{loose}{\tightmtcfalse} % default
2990 \DeclareOption{k-tight}{\ktightmtctrue}
2991 \DeclareOption{k-loose}{\ktightmtcfalse} % default

```

### 5.73.2 Options `checkfiles` and `nocheckfiles`

```
\DeclareOption These options activate or inhibit the checking for empty mini-table files.
\if@mtc@checkfiles
2992 \DeclareOption{checkfiles}{\@mtc@checkfilestrue} % default
2993 \DeclareOption{nocheckfiles}{\@mtc@checkfilesfalse}
```

### 5.73.3 Options `dotted` and `undotted`

```
\DeclareOption These options activate or inhibit the leaders (lines of dots) in the mini-tables.
\ifundottedmtc
2994 \DeclareOption{undotted}{\undottedmtctrue}
2995 \DeclareOption{dotted}{\undottedmtcfalse} % default
```

### 5.73.4 Option `notoccite`

This option will later load the `notoccite` package [\[2\]](#).

```
\DeclareOption
\if@mtc@notoccite@
2996 \DeclareOption{notoccite}{\@mtc@notoccite@true}
```

### 5.73.5 Option `shorttext`

```
\DeclareOption This option forces the use of short extensions.
\if@longextensions@
2997 \DeclareOption{shorttext}{\@longextensions@false}
2998 \PackageWarningNoLine{minitoc}%
2999 {*** You have forced the use of short extensions ***}}
```

## 5.74 The `insection` option

```
\if@mtc@ss@insection@ This option is available only if \chapter is not defined and \section defined. It is to
\ifundefined be revised when chapter/section level commands will ever be allowed together, sometime
\DeclareOption in the far away future.

3000 \newif\if@mtc@ss@insection@ \@mtc@ss@insection@false
3001 \ifundefined{chapter}{%
3002 \ifundefined{section}{%}
```

```

3003             {%
3004             \DeclareOption{insection}%
3005             {\@mtc@ss@insection@true}%
3006             }%
3007 {}{}

```

## 5.75 Language options

`\DeclareOption` Each language option reads the corresponding *language.mld* file via the specialized macro `\mtcselectlanguage`. If the file does not exist, a standard error message is displayed. The language options are (should be) in alphabetical order. Several options could load the same file, but, by convention, there should be a *language.mld* file for each language option, but this file can load another one (as *francais.mld* loads *french.mld*).

```

3008 \DeclareOption{acadian}{\mtcselectlanguage{acadian}}%
3009 \DeclareOption{acadien}{\mtcselectlanguage{acadien}}%
3010 \DeclareOption{afrikaan}{\mtcselectlanguage{afrikaan}}%
3011 \DeclareOption{afrikaans}{\mtcselectlanguage{afrikaans}}%
3012 \DeclareOption{american}{\mtcselectlanguage{american}}%
3013 \DeclareOption{arab}{\mtcselectlanguage{arab}}%
3014 \DeclareOption{arabic}{\mtcselectlanguage{arabic}}%
3015 \DeclareOption{armenian}{\mtcselectlanguage{armenian}}%
3016 \DeclareOption{austrian}{\mtcselectlanguage{austrian}}%
3017 \DeclareOption{bahasa}{\mtcselectlanguage{bahasa}}%
3018 \DeclareOption{bangla}{\mtcselectlanguage{bangla}}%
3019 \DeclareOption{basque}{\mtcselectlanguage{basque}}%
3020 \DeclareOption{bicig}{\mtcselectlanguage{bicig}}%
3021 \DeclareOption{brazil}{\mtcselectlanguage{brazil}}%
3022 \DeclareOption{brazilian}{\mtcselectlanguage{brazilian}}%
3023 \DeclareOption{breton}{\mtcselectlanguage{breton}}%
3024 \DeclareOption{british}{\mtcselectlanguage{british}}%
3025 \DeclareOption{bulgarian}{\mtcselectlanguage{bulgarian}}%
3026 \DeclareOption{bulgarianb}{\mtcselectlanguage{bulgarianb}}%
3027 \DeclareOption{buryat}{\mtcselectlanguage{buryat}}%
3028 \DeclareOption{canadian}{\mtcselectlanguage{canadian}}%
3029 \DeclareOption{canadien}{\mtcselectlanguage{canadien}}%
3030 \DeclareOption{castillan}{\mtcselectlanguage{castillan}}%
3031 \DeclareOption{castillian}{\mtcselectlanguage{castillian}}%
3032 \DeclareOption{catalan}{\mtcselectlanguage{catalan}}%
3033 \DeclareOption{chinese1}{\mtcselectlanguage{chinese1}}%
3034 \DeclareOption{chinese2}{\mtcselectlanguage{chinese2}}%
3035 \DeclareOption{croatian}{\mtcselectlanguage{croatian}}%
3036 \DeclareOption{czech}{\mtcselectlanguage{czech}}%
3037 \DeclareOption{danish}{\mtcselectlanguage{danish}}%
3038 \DeclareOption{dutch}{\mtcselectlanguage{dutch}}%
3039 \DeclareOption{english}{\mtcselectlanguage{english}}%
3040 \DeclareOption{esperant}{\mtcselectlanguage{esperant}}%
3041 \DeclareOption{esperanto}{\mtcselectlanguage{esperanto}}%

```

```

3042 \DeclareOption{estonian}{\mtcselectlanguage{estonian}}%
3043 \DeclareOption{ethiopia}{\mtcselectlanguage{ethiopia}}%
3044 \DeclareOption{ethiopian}{\mtcselectlanguage{ethiopian}}%
3045 \DeclareOption{farsi1}{\mtcselectlanguage{farsi1}}%
3046 \DeclareOption{farsi2}{\mtcselectlanguage{farsi2}}%
3047 \DeclareOption{finnish}{\mtcselectlanguage{finnish}}%
3048 \DeclareOption{finnish2}{\mtcselectlanguage{finnish2}}%
3049 \DeclareOption{français}{\mtcselectlanguage{français}}%
3050 \DeclareOption{french}{\mtcselectlanguage{french}}%
3051 \DeclareOption{frenchb}{\mtcselectlanguage{frenchb}}%
3052 \DeclareOption{frenchle}{\mtcselectlanguage{frenchle}}%
3053 \DeclareOption{frenchpro}{\mtcselectlanguage{frenchpro}}%
3054 \DeclareOption{galician}{\mtcselectlanguage{galician}}%
3055 \DeclareOption{german}{\mtcselectlanguage{german}}%
3056 \DeclareOption{germanb}{\mtcselectlanguage{germanb}}%
3057 \DeclareOption{greek}{\mtcselectlanguage{greek}}%
3058 \DeclareOption{greek-mono}{\mtcselectlanguage{greek-mono}}%
3059 \DeclareOption{greek-polydemo}{\mtcselectlanguage{greek-polydemo}}%
3060 \DeclareOption{greek-polykatha}{\mtcselectlanguage{greek-polykatha}}%
3061 \DeclareOption{guarani}{\mtcselectlanguage{guarani}}%
3062 \DeclareOption{hangul1}{\mtcselectlanguage{hangul1}}%
3063 \DeclareOption{hangul2}{\mtcselectlanguage{hangul2}}%
3064 \DeclareOption{hangul3}{\mtcselectlanguage{hangul3}}%
3065 \DeclareOption{hangul4}{\mtcselectlanguage{hangul4}}%
3066 \DeclareOption{hanja1}{\mtcselectlanguage{hanja1}}%
3067 \DeclareOption{hanja2}{\mtcselectlanguage{hanja2}}%
3068 \DeclareOption{hebrew}{\mtcselectlanguage{hebrew}}%
3069 \DeclareOption{hungarian}{\mtcselectlanguage{hungarian}}%
3070 \DeclareOption{icelandic}{\mtcselectlanguage{icelandic}}%
3071 \DeclareOption{interlingua}{\mtcselectlanguage{interlingua}}%
3072 \DeclareOption{irish}{\mtcselectlanguage{irish}}%
3073 \DeclareOption{italian}{\mtcselectlanguage{italian}}%
3074 \DeclareOption{japanese}{\mtcselectlanguage{japanese}}%
3075 \DeclareOption{japanese2}{\mtcselectlanguage{japanese2}}%
3076 \DeclareOption{japanese3}{\mtcselectlanguage{japanese3}}%
3077 \DeclareOption{japanese4}{\mtcselectlanguage{japanese4}}%
3078 \DeclareOption{japanese5}{\mtcselectlanguage{japanese5}}%
3079 \DeclareOption{latin}{\mtcselectlanguage{latin}}%
3080 \DeclareOption{latin2}{\mtcselectlanguage{latin2}}%
3081 \DeclareOption{latvian}{\mtcselectlanguage{latvian}}%
3082 \DeclareOption{letton}{\mtcselectlanguage{letton}}%
3083 \DeclareOption{lithuanian}{\mtcselectlanguage{lithuanian}}%
3084 \DeclareOption{lsorbian}{\mtcselectlanguage{lsorbian}}%
3085 \DeclareOption{magyar}{\mtcselectlanguage{magyar}}%
3086 \DeclareOption{magyar2}{\mtcselectlanguage{magyar2}}%
3087 \DeclareOption{malayalam-keli}{\mtcselectlanguage{malayalam-keli}}%
3088 \DeclareOption{malayalam-rachana}{\mtcselectlanguage{malayalam-rachana}}%
3089 \DeclareOption{malayalam-rachana2}{\mtcselectlanguage{malayalam-rachana2}}%
3090 \DeclareOption{mongol}{\mtcselectlanguage{mongol}}%
3091 \DeclareOption{naustrian}{\mtcselectlanguage{naustrian}}%
3092 \DeclareOption{ngerman}{\mtcselectlanguage{ngerman}}%
3093 \DeclareOption{ngermanb}{\mtcselectlanguage{ngermanb}}%
3094 \DeclareOption{norsk}{\mtcselectlanguage{norsk}}%
3095 \DeclareOption{nynorsk}{\mtcselectlanguage{nynorsk}}%

```

```

3096 \DeclareOption{polish}{\mtcselectlanguage{polish}}%
3097 \DeclareOption{polish2}{\mtcselectlanguage{polish2}}%
3098 \DeclareOption{portuges}{\mtcselectlanguage{portuges}}%
3099 \DeclareOption{portuguese}{\mtcselectlanguage{portuguese}}%
3100 \DeclareOption{romanian}{\mtcselectlanguage{romanian}}%
3101 \DeclareOption{russian}{\mtcselectlanguage{russian}}%
3102 \DeclareOption{russianb}{\mtcselectlanguage{russianb}}%
3103 \DeclareOption{russianc}{\mtcselectlanguage{russianc}}%
3104 \DeclareOption{russian2m}{\mtcselectlanguage{russian2m}}%
3105 \DeclareOption{russian2o}{\mtcselectlanguage{russian2o}}%
3106 \DeclareOption{samin}{\mtcselectlanguage{samin}}%
3107 \DeclareOption{scottish}{\mtcselectlanguage{scottish}}%
3108 \DeclareOption{serbian}{\mtcselectlanguage{serbian}}%
3109 \DeclareOption{serbianc}{\mtcselectlanguage{serbianc}}%
3110 \DeclareOption{slovak}{\mtcselectlanguage{slovak}}%
3111 \DeclareOption{slovene}{\mtcselectlanguage{slovene}}%
3112 \DeclareOption{spanish}{\mtcselectlanguage{spanish}}%
3113 \DeclareOption{spanish2}{\mtcselectlanguage{spanish2}}%
3114 \DeclareOption{spanish3}{\mtcselectlanguage{spanish3}}%
3115 \DeclareOption{swedish}{\mtcselectlanguage{swedish}}%
3116 \DeclareOption{thai}{\mtcselectlanguage{thai}}%
3117 \DeclareOption{turkish}{\mtcselectlanguage{turkish}}%
3118 \DeclareOption{UKenglish}{\mtcselectlanguage{UKenglish}}%
3119 \DeclareOption{ukraine}{\mtcselectlanguage{ukraine}}%
3120 \DeclareOption{USenglish}{\mtcselectlanguage{USenglish}}%
3121 \DeclareOption{usorbian}{\mtcselectlanguage{usorbian}}%
3122 \DeclareOption{vietnam}{\mtcselectlanguage{vietnam}}%
3123 \DeclareOption{vietnamese}{\mtcselectlanguage{vietnamese}}%
3124 \DeclareOption{welsh}{\mtcselectlanguage{welsh}}%

```

## 5.76 The hints option

```

\if@mtc@hints@ First, we define some flags:
\if@mtc@hints@w@
\if@mtc@hints@given@

```

- The flag `\if@mtc@hints@` is true if the hints option is required (default).
- The flag `\if@mtc@hints@w@` is set true if we detect that some sectioning commands have been altered since the loading of the document class.
- The flag `\if@mtc@hints@given@` is set true if the hints option detects something curious and writes messages in the `.log` file.

```

3125 \newif\if@mtc@hints@ \@mtc@hints@true
3126 \newif\if@mtc@hints@w@ \@mtc@hints@w@false
3127 \newif\if@mtc@hints@given@ \@mtc@hints@given@false

```

```

\DeclareOption We declare the hints (default) and nohints options:
\if@mtc@hints@

```

```

3128 \DeclareOption{hints}{\@mtc@hints@true}
3129 \DeclareOption{nohints}{\@mtc@hints@false}

```

`\mtc@hints@begindoc`    The hints option is made of three parts: the first, `\mtc@hints@begindoc`, is executed via `\AtBeginDocument` and looks if some packages or classes are loaded and gives warnings about their compatibility with minitoc.

The second part is made of tiny pieces of code inserted in the minitoc code, to verify that some macros are called in the right order.

`\mtc@hints@enddoc`    The third and last part, `\mtc@hints@enddoc`, is executed via `\AtEndDocument` and examines some flags set by the first and the second parts. Then, if necessary, it writes some infos in the `.log` file and/or warnings on the screen and in the `.log` file. Note that the hints option does not signal errors, only infos and warnings, so it does not stop the  $\LaTeX$  run.

### 5.76.1 First part: `\mtc@hints@begindoc`

`\mtc@hints@begindoc`    We declare the first part of the hints option (for `\AtBeginDocument`):

```

3130 \def\mtc@hints@begindoc{%
3131 \PackageInfo{minitoc}{hints}}%
3132 {***** You requested the hints option *****\MessageBreak
3133 ***** Some hints are eventually given below *****\@gobble}%

```

#### 5.76.1.1 Hint about the appendix package

`\@ifpackageloaded`    We test the presence of the appendix package [46]:  
`\if@mtc@hints@given@`

```

3134 \@ifpackageloaded{appendix}{%
3135 \@mtc@hints@given@true
3136 \PackageInfo{minitoc}{hints}}%
3137 {--- The appendix package is loaded. \MessageBreak
3138 See the minitoc package documentation\MessageBreak
3139 for specific precautions\@gobble}}}%

```

#### 5.76.1.2 Hint about the tocibind package

`\@ifpackageloaded`    We test the presence of the tocibind package [47]:  
`\if@mtc@hints@given@`

```

3140 \@ifpackageloaded{tocbibind}%
3141   {\@mtc@hints@given@true
3142    \PackageInfo{minitoc(hints)}%
3143     {--- The tocbibind package is loaded. \MessageBreak
3144      See the minitoc package documentation
3145      for specific precautions\@gobble}}}%

```

### 5.76.1.3 Hint about the tocloft package

\@ifpackageloaded We test the presence of the tocloft package [44]:  
\if@mtc@hints@given@

```

3146 \@ifpackageloaded{tocloft}%
3147   {\@mtc@hints@given@true
3148    \PackageInfo{minitoc(hints)}%
3149     {--- The tocloft package is loaded. \MessageBreak
3150      See the minitoc package documentation
3151      for specific precautions\@gobble}}}%

```

### 5.76.1.4 Hint about the titletoc package

\@ifpackageloaded We test the presence of the titletoc package [8], and emit a warning, because this  
\if@mtc@hints@given@ package is incompatible with minitoc:

```

3152 \@ifpackageloaded{titletoc}%
3153   {\@mtc@hints@given@true
3154    \PackageWarningNoLine{minitoc(hints)}%
3155     {--- The titletoc package is loaded. \MessageBreak
3156      It is incompatible with the minitoc package}}}%

```

### 5.76.1.5 Hint about the placeins package

\@ifpackageloaded We test if the placeins package [3] is loaded and, if yes, we check that the right options  
\if@mtc@ss@insection@ are selected:

\@ifpackagewith  
\if@mtc@hints@given@

```

3157 \@ifpackageloaded{placeins}%
3158   {\if@mtc@ss@insection@
3159    \@ifpackagewith{placeins}{section,below}}}%
3160   {\@mtc@hints@given@true
3161    \PackageWarningNoLine{minitoc(hints)}%
3162     {--- The placeins package is loaded without
3163      \MessageBreak
3164      section and below options, but minitoc used
3165      \MessageBreak
3166      the insection option which implies them.

```



```

3167         \MessageBreak
3168         Try to inverse the loading order and use coherent options.
3169         \MessageBreak
3170         You may have got a message
3171         ! LaTeX Error: Option clash for package placeins}%
3172     }%
3173     \fi
3174     \@ifpackagelater{placeins}{2005/04/18}{}{}%
3175     \@mtc@hints@given@true
3176     \PackageWarningNoLine{minitoc(hints)}%
3177 {--- The placeins package loaded is
3178     \MessageBreak
3179     too old. It should have a version date of
3180     \MessageBreak
3181     2005/04/18 at least}%
3182 }%
3183 }{}%

```

#### 5.76.1.6 Hint about the memoir class

```

\@ifclassloaded We test if the memoir class [48] is loaded:
\if@mtc@hints@given@
3184 \@ifclassloaded{memoir}%
3185     {\@mtc@hints@given@true
3186     \PackageInfo{minitoc(hints)}%
3187     {--- The memoir class is loaded. \MessageBreak
3188     See the minitoc package documentation for specific precautions\@gobble}{}%
3189     }{}%

```

#### 5.76.1.7 Hint about the amsart and amsproc classes

\@ifclassloaded We test is the amsart or amsproc class is loaded and emit a warning, because thess  
\if@mtc@hints@given@ classes are incompatible with minitoc:

```

3190 \@ifclassloaded{amsart}%
3191     {\@mtc@hints@given@true
3192     \PackageWarningNoLine{minitoc(hints)}%
3193     {--- The amsart class is loaded. \MessageBreak
3194     It is incompatible with the minitoc package}{}{}%
3195 \@ifclassloaded{amsproc}%
3196     {\@mtc@hints@given@true
3197     \PackageWarningNoLine{minitoc(hints)}%
3198     {--- The amsproc class is loaded. \MessageBreak
3199     It is incompatible with the minitoc package}{}{}%

```

**5.76.1.8 Hint about the amsbook class**

```

\@ifclassloaded We test if the amsbook class is loaded:
\if@mtc@hints@given@
3200 \@ifclassloaded{amsbook}%
3201     {\@mtc@hints@given@true
3202     \PackageInfo{minitoc(hints)}%
3203     {--- The amsbook class is loaded. \MessageBreak
3204     See the minitoc package documentation for specific precautions\@gobble}{}}%
3205     }{}%

```

**5.76.1.9 Hint about the abstract package**

```

\@ifpackageloaded We test the presence of the abstract package [45], then its options:
\if@mtc@abstract@loaded@
\@ifpackagewith 3206 \@ifpackageloaded{abstract}%
\if@mtc@hints@given@ 3207     {\@mtc@abstract@loaded@true%
3208     \@ifpackagewith{abstract}{addtotoc}%
3209     {\@mtc@hints@given@true
3210     \PackageInfo{minitoc(hints)}%
3211     {The ‘‘abstract’’ package has been loaded with\MessageBreak
3212     the ‘‘addtotoc’’ option.\MessageBreak
3213     You need to look at the documentation to adjust.}
3214     }{}%

```

**5.76.1.10 Hint about the alteration of the sectionning commands**

```

\part To check if the sectionning commands \part, \chapter or \section have
\chapter been altered by some package or in the preamble, we compare them (when
\section executing a \AtBeginDocument block) with their saved versions (saved by the
\AtBeginDocument minitoc package when it is loaded) \mtc@hints@part, \mtc@hints@chapter
\mtc@hints@part and \mtc@hints@section. For each sectionning command, we must perform the
\mtc@hints@chapter comparaisn for the command itself, its unstarred branch and its starred branch. But the
\mtc@hints@section hyperref package [39] may interfere, hence the formal precautions in the messages.

```

**5.76.1.10.1 Alteration of \part :**

```

\if@mtc@hints@w@ We check the alteration of \part, \@part and \@spart:
\@ifundefined
\part 3215 \@mtc@hints@w@false
\mtc@hints@part 3216 \@ifundefined{part}{}{\ifx\part\mtc@hints@part\relax
\@part 3217 \else\@mtc@hints@w@true\fi}
\mtc@hints@@part 3218 \@ifundefined{part}{}{\ifx\@part\mtc@hints@@part\relax
\@spart
\mtc@hints@@spart
\if@mtc@hints@given@
\if@mtc@hyper@used@

```

```

3219 \else\@mtc@hints@w@true\fi}
3220 \@ifundefined{part}{}{\ifx\@spart\mtc@hints@spart\relax
3221 \else\@mtc@hints@w@true\fi}
3222 \if@mtc@hints@w@\@mtc@hints@given@true%
3223 \PackageWarningNoLine{minitoc(hints)}%
3224 {--- The \string\part\space command is altered after minitoc}
3225 \if@mtc@hyper@used@
3226 \PackageWarningNoLine{minitoc(hints)}%
3227 {--- It may be the consequence of loading the ‘‘hyperref’’ package}
3228 \fi
3229 \fi

```

### 5.76.1.10.2 Alteration of \chapter :

```

\if@mtc@hints@w@ We check the alteration of \chapter, \@chapter and \@schapter:
\@ifundefined
\chapter 3230 \@mtc@hints@w@false
\mtc@hints@chapter 3231 \@ifundefined{chapter}{}{\ifx\chapter\mtc@hints@chapter\relax
\@chapter 3232 \else\@mtc@hints@w@true\fi}%
\mtc@hints@chapter 3233 \@ifundefined{chapter}{}{\ifx\@chapter\mtc@hints@chapter\relax
\@schapter 3234 \else\@mtc@hints@w@true\fi}%
\mtc@hints@schapter 3235 \@ifundefined{chapter}{}{\ifx\@schapter\mtc@hints@schapter\relax
\if@mtc@hints@given@ 3236 \else\@mtc@hints@w@true\fi}%
\if@mtc@hyper@used@ 3237 \if@mtc@hints@w@\@mtc@hints@given@true%
3238 \PackageWarningNoLine{minitoc(hints)}%
3239 {--- The \string\chapter\space command is altered after minitoc}
3240 \if@mtc@hyper@used@
3241 \PackageWarningNoLine{minitoc(hints)}%
3242 {--- It may be the consequence of loading the ‘‘hyperref’’ package}
3243 \fi
3244 \fi

```

### 5.76.1.10.3 Alteration of \section :

```

\if@mtc@hints@w@ We check the alteration of \section, \@sect and \@ssect:
\@ifundefined
\section 3245 \@mtc@hints@w@false
\mtc@hints@section 3246 \@ifundefined{chapter}%
\@sect 3247 {\@ifundefined{section}{}{\ifx\section\mtc@hints@section\relax\else
\mtc@hints@sect 3248 \@mtc@hints@w@true\fi}
\@ssect 3249 \@ifundefined{section}{}{\ifx\@sect\mtc@hints@ssect\relax\else
\mtc@hints@ssect 3250 \@mtc@hints@w@true\fi}
\if@mtc@hints@given@ 3251 \@ifundefined{section}{}{\ifx\@ssect\mtc@hints@ssect\relax\else
\if@mtc@hyper@used@ 3252 \@mtc@hints@w@true\fi}
3253 \if@mtc@hints@w@\@mtc@hints@given@true%
3254 \PackageWarningNoLine{minitoc(hints)}%
3255 {--- The \string\section\space command is altered after minitoc}
3256 \if@mtc@hyper@used@

```

```

3257     \PackageWarningNoLine{minitoc(hints)}%
3258         {--- It may be the consequence of loading the ‘‘hyperref’’ package}
3259     \fi
3260     \relax\else\fi}%

```

### 5.76.1.11 Hint about the coherence of the calling sequences of some commands

And finally, we prepare the coherence tests about the calling sequences of triplets of associated commands like `\doparttoc`, `\parttoc` and `\[fake]listofcontents`, and similar: to be able to use `\parttoc`, a table of contents file must have been created via `\[fake]listofcontents` and splitted into `parttoc` files via `\doparttoc`.

`\if@mtc@hints@` Hence we declare some flags:

```

\if@mtc@toc@used@
\if@mtc@lof@used@ 3261 \if@mtc@hints@
\if@mtc@lot@used@ 3262 \newif\if@mtc@toc@used@ \global\@mtc@toc@used@false
3263 \newif\if@mtc@lof@used@ \global\@mtc@lof@used@false
3264 \newif\if@mtc@lot@used@ \global\@mtc@lot@used@false

```

`\mtc@sv@tableofcontents` Then we patch the involved commands to set the corresponding flag when they are used.

```

\tableofcontents First, the commands for the main summaries:
\if@mtc@toc@used@
\mtc@sv@listoffigures 3265 \let\mtc@sv@tableofcontents\tableofcontents
\listoffigures 3266 \def\tableofcontents%
\if@mtc@lof@used@ 3267 {\global\@mtc@toc@used@true\mtc@sv@tableofcontents}
\mtc@sv@listoftables 3268 \let\mtc@sv@listoffigures\listoffigures
\listoftables 3269 \def\listoffigures%
\if@mtc@lot@used@ 3270 {\global\@mtc@lof@used@true\mtc@sv@listoffigures}
3271 \let\mtc@sv@listoftables\listoftables
3272 \def\listoftables%
3273 {\global\@mtc@lot@used@true\mtc@sv@listoftables}

```

`\mtc@sv@faketableofcontents` Then, their ‘‘fake’’ siblings:

```

\fake tableofcontents
\if@mtc@toc@used@ 3274 \let\mtc@sv@faketableofcontents\fake tableofcontents
\mtc@sv@fakelistoffigures 3275 \def\fake tableofcontents%
\fakelistoffigures 3276 {\global\@mtc@toc@used@true\mtc@sv@faketableofcontents}
\if@mtc@lof@used@ 3277 \let\mtc@sv@fakelistoffigures\fakelistoffigures
\mtc@sv@fakelistoftables 3278 \def\fakelistoffigures%
\fakelistoftables 3279 {\global\@mtc@lof@used@true\mtc@sv@fakelistoffigures}
\if@mtc@lot@used@ 3280 \let\mtc@sv@fakelistoftables\fakelistoftables
3281 \def\fakelistoftables%
3282 {\global\@mtc@lot@used@true\mtc@sv@fakelistoftables}
3283 \fi

```

`\mtc@hints@begindoc` And the `\mtc@hints@begindoc` definition is finished (it begins in section 5.76.1 on page 191):

```
3284 }
```

## 5.76.2 Final part: `\mtc@hints@enddoc`

`\mtc@hints@enddoc` The final part of the hints option is executed via `\AtEndDocument`. Its code is in the `\AtEndDocument` `\mtc@hints@enddoc` macro. It is a sequence of tests on the packages or classes loaded and the flags set during the first and the second parts of this option. First, we declare the `\mtc@hints@enddoc` macro:

```
3285 \def\mtc@hints@enddoc{%
```

### 5.76.2.1 Hint about `\sect-lof|lot` and the `insection` option

`\if@mtc@sect@floats@` We look if some section-level lists of figures or tables have been requested.

`\if@dosectlof@used@`

`\if@dosectlot@used@` 3286 `\if@dosectlof@used@\@mtc@sect@floats@true\fi`

`\if@sectlof@used@` 3287 `\if@dosectlot@used@\@mtc@sect@floats@true\fi`

`\if@sectlot@used@` 3288 `\if@sectlof@used@\@mtc@sect@floats@true\fi`

`\if@mtc@section@def@` 3289 `\if@sectlot@used@\@mtc@sect@floats@true\fi`

```
3290 \if@mtc@section@def@
```

`\if@mtc@placeinsLoaded@` If yes, we verify that the `placeins` package [3] has been loaded with the correct options or that the `insection` option of the `minitoc` package has been invoked. If not, a warning is given.

`\if@mtc@hints@w@`

`\if@mtc@hints@given@`

```
3291 \if@mtc@placeinsLoaded@
```

```
3292 \else
```

```
3293 \if@mtc@sect@floats@%
```

```
3294 \PackageWarningNoLine{minitoc(hints)}%
```

```
3295 {***** You are using \string\dosectlof\space and/or \MessageBreak
```

```
3296 \string\dosectlot, \string\sectlof\space and/or \string\sectlot,
```

```
3297 \MessageBreak
```

```
3298 hence the ‘‘insection’’ package option is recommended. *****}%
```

```
3299 \@mtc@hints@w@true
```

```
3300 \@mtc@hints@given@true
```

```
3301 \fi
```

```
3302 \fi
```

**5.76.2.2 Final part of the coherence tests**

We test if `\parttoc` has been used without `\doparttoc`, etc., for each pair of preparation/insertion commands.

```

\if@mtc@part@def@ For the part level commands:
\if@parttoc@used@
\if@doparttoc@used@ 3303 \if@mtc@part@def@
\if@mtc@hints@given@ 3304 \if@parttoc@used@
\if@partlof@used@ 3305 \if@doparttoc@used@\else
\if@dopartlof@used@ 3306 \PackageWarningNoLine{minitoc(hints)}%
\if@partlot@used@ 3307 {**** You have used \string\parttoc\space
\if@dopartlot@used@ 3308 but not \string\doparttoc\space****}
3309 \@mtc@hints@given@true
3310 \fi
3311 \fi
3312 \if@partlof@used@
3313 \if@dopartlof@used@\else
3314 \PackageWarningNoLine{minitoc(hints)}%
3315 {**** You have used \string\partlof\space
3316 but not \string\dopartlof\space****}
3317 \@mtc@hints@given@true
3318 \fi
3319 \fi
3320 \if@partlot@used@
3321 \if@dopartlot@used@\else
3322 \PackageWarningNoLine{minitoc(hints)}%
3323 {**** You have used \string\partlot\space
3324 but not \string\dopartlot\space****}
3325 \@mtc@hints@given@true
3326 \fi
3327 \fi
3328 \fi

\if@mtc@chapter@def@ For the chapter level commands:
\if@minitoc@used@
\if@dominitoc@used@ 3329 \if@mtc@chapter@def@
\if@mtc@hints@given@ 3330 \if@minitoc@used@
\if@minilof@used@ 3331 \if@dominitoc@used@\else
\if@dominilof@used@ 3332 \PackageWarningNoLine{minitoc(hints)}%
\if@minilot@used@ 3333 {**** You have used \string\minitoc\space
\if@dominilot@used@ 3334 but not \string\dominitoc\space****}
3335 \@mtc@hints@given@true
3336 \fi
3337 \fi
3338 \if@minilof@used@
3339 \if@dominilof@used@\else
3340 \PackageWarningNoLine{minitoc(hints)}%
3341 {**** You have used \string\minilof\space
3342 but not \string\dominilof\space****}
3343 \@mtc@hints@given@true

```

```

3344 \fi
3345 \fi
3346 \if@minilot@used@
3347 \if@dominilot@used@\else
3348 \PackageWarningNoLine{minitoc(hints)}%
3349 {**** You have used \string\minilot\space
3350 but not \string\dominilot\space****}
3351 \@mtc@hints@given@true
3352 \fi
3353 \fi
3354 \fi

```

\if@mtc@section@def@ For the section level commands:

```

\if@secttoc@used@
\if@dosecttoc@used@ 3355 \if@mtc@section@def@
\if@mtc@hints@given@ 3356 \if@secttoc@used@
\if@sectlof@used@ 3357 \if@dosecttoc@used@\else
\if@dosectlof@used@ 3358 \PackageWarningNoLine{minitoc(hints)}%
\if@sectlot@used@ 3359 {**** You have used \string\secttoc\space
\if@dosectlot@used@ 3360 but not \string\dosecttoc\space****}
3361 \@mtc@hints@given@true
3362 \fi
3363 \fi
3364 \if@sectlof@used@
3365 \if@dosectlof@used@\else
3366 \PackageWarningNoLine{minitoc(hints)}%
3367 {**** You have used \string\sectlof\space
3368 but not \string\dosectlof\space****}
3369 \@mtc@hints@given@true
3370 \fi
3371 \fi
3372 \if@sectlot@used@
3373 \if@dosectlot@used@\else
3374 \PackageWarningNoLine{minitoc(hints)}%
3375 {**** You have used \string\sectlot\space
3376 but not \string\dosectlot\space****}
3377 \fi
3378 \fi
3379 \fi

```

### 5.76.2.3 Check if the main tables have been prepared (first part)

Now, we test if a \doparttoc macro has been called but without any matching \parttoc, hence its a vain call. We do the same for each analog command.

```

\if@mtc@part@def@ Part level commands:
\if@doparttoc@used@
\if@parttoc@used@ 3380 \if@mtc@part@def@
\if@mtc@hints@given@
\if@dopartlof@used@
\if@partlof@used@
\if@dopartlot@used@
\if@partlot@used@

```

```

3381 \if@doparttoc@used@
3382   \if@parttoc@used@\else
3383     \PackageWarningNoLine{minitoc(hints)}%
3384     {**** You have used \string\doparttoc\space
3385       but not \string\parttoc\space****}
3386     \@mtc@hints@given@true
3387   \fi
3388 \fi
3389 \if@dopartlof@used@
3390   \if@partlof@used@\else
3391     \PackageWarningNoLine{minitoc(hints)}%
3392     {**** You have used \string\dopartlof\space
3393       but not \string\partlof\space****}
3394     \@mtc@hints@given@true
3395   \fi
3396 \fi
3397 \if@dopartlot@used@
3398   \if@partlot@used@\else
3399     \PackageWarningNoLine{minitoc(hints)}%
3400     {**** You have used \string\dopartlot\space
3401       but not \string\partlot\space****}
3402     \@mtc@hints@given@true
3403   \fi
3404 \fi
3405 \fi

```

```

\if@mtc@chapter@def@ Chapter level commands:
\if@dominitoc@used@
  \if@minitoc@used@ 3406 \if@mtc@chapter@def@
\if@mtc@hints@given@ 3407 \if@dominitoc@used@
\if@dominilof@used@ 3408   \if@minitoc@used@\else
\if@minilof@used@ 3409     \PackageWarningNoLine{minitoc(hints)}%
\if@dominilot@used@ 3410     {**** You have used \string\dominitoc\space
\if@minilot@used@ 3411       but not \string\minitoc\space****}
3412     \@mtc@hints@given@true
3413   \fi
3414 \fi
3415 \if@dominilof@used@
3416   \if@minilof@used@\else
3417     \PackageWarningNoLine{minitoc(hints)}%
3418     {**** You have used \string\dominilof\space
3419       but not \string\minilof\space****}
3420     \@mtc@hints@given@true
3421   \fi
3422 \fi
3423 \if@dominilot@used@
3424   \if@minilot@used@\else
3425     \PackageWarningNoLine{minitoc(hints)}%
3426     {**** You have used \string\dominilot\space
3427       but not \string\minilot\space****}
3428     \@mtc@hints@given@true
3429   \fi

```



```
3430 \fi
3431 \fi
```

```
\if@mtc@section@def@ Section level commands:
\if@dosecttoc@used@
\if@secttoc@used@ 3432 \if@mtc@section@def@
\if@mtc@hints@given@ 3433 \if@dosecttoc@used@
\if@dosectlof@used@ 3434 \if@secttoc@used@\else
\if@sectlof@used@ 3435 \PackageWarningNoLine{minitoc(hints)}%
\if@dosectlot@used@ 3436 {**** You have used \string\dosecttoc\space
\if@sectlot@used@ 3437 but not \string\secttoc\space****}
3438 \@mtc@hints@given@true
3439 \fi
3440 \fi
3441 \if@dosectlof@used@
3442 \if@sectlof@used@\else
3443 \PackageWarningNoLine{minitoc(hints)}%
3444 {**** You have used \string\dosectlof\space
3445 but not \string\sectlof\space****}
3446 \@mtc@hints@given@true
3447 \fi
3448 \fi
3449 \if@dosectlot@used@
3450 \if@sectlot@used@\else
3451 \PackageWarningNoLine{minitoc(hints)}%
3452 {**** You have used \string\dosectlot\space
3453 but not \string\sectlot\space****}
3454 \fi
3455 \fi
3456 \fi
3457 \fi
```

#### 5.76.2.4 Check if the main tables have been prepared (second part)

Another coherence test verifies that if `\parttoc` has been called, then the macro `\tableofcontents` or `\faketableofcontents` has also been called (to create the necessary contents file); and similar tests are made for the other mini-table commands.

```
\if@mtc@part@def@ Part level commands:
\if@mtc@hints@given@
\if@parttoc@used@ 3458 \if@mtc@part@def@
\ifmtc@toc@used@ 3459 \if@parttoc@used@
\if@partlof@used@ 3460 \if@mtc@toc@used@\else
\ifmtc@lof@used@ 3461 \PackageWarningNoLine{minitoc(hints)}%
\if@partlot@used@ 3462 {**** You have used \string\parttoc\space but not\MessageBreak
\ifmtc@lot@used@ 3463 \string\tableofcontents\space nor
3464 \string\faketableofcontents\space ****}
3465 \@mtc@hints@given@true
```

```

3466 \fi
3467 \fi
3468 \if@partlof@used@
3469 \if@mtc@lof@used@\else
3470 \PackageWarningNoLine{minitoc(hints)}%
3471 {**** You have used \string\partlof\space but not\MessageBreak
3472 \string\listoffigures\space nor
3473 \string\fake\listoffigures\space ****}
3474 \@mtc@hints@given@true
3475 \fi
3476 \fi
3477 \if@partlot@used@
3478 \if@mtc@lot@used@\else
3479 \PackageWarningNoLine{minitoc(hints)}%
3480 {**** You have used \string\partlot\space but not\MessageBreak
3481 \string\listoftables\space nor
3482 \string\fake\listoftables\space ****}
3483 \@mtc@hints@given@true
3484 \fi
3485 \fi
3486 \fi

```

\if@mtc@chapter@def@ Chapter level commands:

\if@mtc@hints@given@

```

\if@minitoc@used@ 3487 \if@mtc@chapter@def@
\ifmtc@toc@used@ 3488 \if@minitoc@used@
\if@minilof@used@ 3489 \if@mtc@toc@used@\else
\ifmtc@lof@used@ 3490 \PackageWarningNoLine{minitoc(hints)}%
\if@minilot@used@ 3491 {**** You have used \string\minitoc\space but not\MessageBreak
\ifmtc@lot@used@ 3492 \string\tableofcontents\space nor
3493 \string\fake\tableofcontents\space ****}
3494 \@mtc@hints@given@true
3495 \fi
3496 \fi
3497 \if@minilof@used@
3498 \if@mtc@lof@used@\else
3499 \PackageWarningNoLine{minitoc(hints)}%
3500 {**** You have used \string\minilof\space but not\MessageBreak
3501 \string\listoffigures\space nor
3502 \string\fake\listoffigures\space ****}
3503 \@mtc@hints@given@true
3504 \fi
3505 \fi
3506 \if@minilot@used@
3507 \if@mtc@lot@used@\else
3508 \PackageWarningNoLine{minitoc(hints)}%
3509 {**** You have used \string\minilot\space but not\MessageBreak
3510 \string\listoftables\space nor
3511 \string\fake\listoftables\space ****}
3512 \@mtc@hints@given@true
3513 \fi
3514 \fi

```

3515 \fi

\if@mtc@section@def@ Section level commands:

\if@mtc@hints@given@

\if@secttoc@used@ 3516 \if@mtc@section@def@

\ifmtc@toc@used@ 3517 \if@secttoc@used@

\if@sectlof@used@ 3518 \if@mtc@toc@used@\else

\ifmtc@lof@used@ 3519 \PackageWarningNoLine{minitoc(hints)}%

\if@sectlot@used@ 3520 {\*\*\*\* You have used \string\secttoc\space but not\MessageBreak

\ifmtc@lot@used@ 3521 \string\tableofcontents\space nor

3522 \string\fake\tableofcontents\space \*\*\*\*}

3523 \@mtc@hints@given@true

3524 \fi

3525 \fi

3526 \if@sectlof@used@

3527 \if@mtc@lof@used@\else

3528 \PackageWarningNoLine{minitoc(hints)}%

3529 {\*\*\*\* You have used \string\sectlof\space but not\MessageBreak

3530 \string\listoffigures\space nor

3531 \string\fake\listoffigures\space \*\*\*\*}

3532 \@mtc@hints@given@true

3533 \fi

3534 \fi

3535 \if@sectlot@used@

3536 \if@mtc@lot@used@\else

3537 \PackageWarningNoLine{minitoc(hints)}%

3538 {\*\*\*\* You have used \string\sectlot\space but not\MessageBreak

3539 \string\listoftables\space nor

3540 \string\fake\listoftables\space \*\*\*\*}

3541 \@mtc@hints@given@true

3542 \fi

3543 \fi

3544 \fi

### 5.76.2.5 Check the number of mini-tables, in case of short extensions

\mtc@hints@checklongext If short extensions are used, you can use only 99 mini-tables of each kind. If more are created, the auxiliary files can be overwritten: the hundredth minitoc file \jobname.U100 has its name truncated to \jobname.U10, which is already the tenth minitoc file. Thus, we need a hint to signal this situation. The code is rather simple, but the remedy is bitter and costly: either use a better operating system <sup>9</sup>, either redesign the document.

\if@mtc@hints@given@true

\if@mtc@chapter@def@

\if@mtc@section@def@ 3545 \def\mtc@hints@checklongext{%

3546 \if@longextensions@

3547 \else

3548 \if@mtc@part@def@

3549 \ifnum 99 < \value{ptc}

---

<sup>9</sup>On the long term, a good investment.

```

3550     \@mtc@hints@given@true
3551     \PackageWarningNoLine{minitoc(hints)}%
3552     {**** You have used short extensions
3553     and more than 99 parts ****}
3554 \fi
3555 \fi
3556 \if@mtc@chapter@def@
3557     \ifnum 99 < \value{mtc}
3558     \@mtc@hints@given@true
3559     \PackageWarningNoLine{minitoc(hints)}%
3560     {**** You have used short extensions
3561     and more than 99 chapters ****}
3562 \fi
3563 \else
3564     \if@mtc@section@def@
3565     \ifnum 99 < \value{stc}
3566     \@mtc@hints@given@true
3567     \PackageWarningNoLine{minitoc(hints)}%
3568     {**** You have used short extensions
3569     and more than 99 sections ****}
3570 \fi
3571 \fi
3572 \fi
3573 \fi}
3574 \mtc@hints@checklongext

```

#### 5.76.2.6 Final part of the hint about the sectsty package

\if@mtc@sectstyLoaded@ We test if sectsty has been loaded before (correct) or after (incorrect) minitoc. See  
 \if@mtc@sectstyLoaded@a@ section 5.9 on page 76.  
 \if@mtc@hints@given@

```

3575 \if@mtc@sectstyLoaded@\else
3576     \if@mtc@sectstyLoaded@a@
3577     \PackageWarningNoLine{minitoc(hints)}%
3578     {**** The sectsty package should be loaded \MessageBreak
3579     **before** the minitoc package ****}
3580     \@mtc@hints@given@true
3581 \fi
3582 \fi

```

#### 5.76.2.7 Check if empty mini-tables have been detected

We test for each kind of mini-tables.

\if@mtc@empty@parttoc@ For parttocs:

```
3583 \if@mtc@empty@parttoc@
3584   \PackageWarningNoLine{minitoc(hints)}%
3585   {**** You have attempted to insert empty parttocs ****}
3586   \@mtc@hints@given@true
3587 \fi
```

\if@mtc@empty@partlof@ For partlofs:

```
3588 \if@mtc@empty@partlof@
3589   \PackageWarningNoLine{minitoc(hints)}%
3590   {**** You have attempted to insert empty partlofs ****}
3591   \@mtc@hints@given@true
3592 \fi
```

\if@mtc@empty@partlot@ For partlots:

```
3593 \if@mtc@empty@partlot@
3594   \PackageWarningNoLine{minitoc(hints)}%
3595   {**** You have attempted to insert empty partlots ****}
3596   \@mtc@hints@given@true
3597 \fi
```

\if@mtc@empty@minitoc@ For minitocs:

```
3598 \if@mtc@empty@minitoc@
3599   \PackageWarningNoLine{minitoc(hints)}%
3600   {**** You have attempted to insert empty minitocs ****}
3601   \@mtc@hints@given@true
3602 \fi
```

\if@mtc@empty@minilof@ For minilofs:

```
3603 \if@mtc@empty@minilof@
3604   \PackageWarningNoLine{minitoc(hints)}%
3605   {**** You have attempted to insert empty minilofs ****}
3606   \@mtc@hints@given@true
3607 \fi
```

`\if@mtc@empty@minilot@` For minilots:

```
3608 \if@mtc@empty@minilot@
3609   \PackageWarningNoLine{minitoc(hints)}%
3610   {**** You have attempted to insert empty minilots ****}
3611   \@mtc@hints@given@true
3612 \fi
```

`\if@mtc@empty@secttoc@` For secttocs:

```
3613 \if@mtc@empty@secttoc@
3614   \PackageWarningNoLine{minitoc(hints)}%
3615   {**** You have attempted to insert empty secttocs ****}
3616   \@mtc@hints@given@true
3617 \fi
```

`\if@mtc@empty@sectlof@` For sectlofs:

```
3618 \if@mtc@empty@sectlof@
3619   \PackageWarningNoLine{minitoc(hints)}%
3620   {**** You have attempted to insert empty sectlofs ****}
3621   \@mtc@hints@given@true
3622 \fi
```

`\if@mtc@empty@sectlot@` For sectlots:

```
3623 \if@mtc@empty@sectlot@
3624   \PackageWarningNoLine{minitoc(hints)}%
3625   {**** You have attempted to insert empty sectlots ****}
3626   \@mtc@hints@given@true
3627 \fi
```

### 5.76.2.8 Check if obsolete commands have been used

This hint is just a reminder if you have used obsolete commands, which are also signalled in the .log file.

`\if@firstpartis@used@` Obsolete macro `\firstpartis`:

```
3628 \if@firstpartis@used@
3629   \PackageWarningNoLine{minitoc(hints)}%
3630   {**** You have invoked an obsolete command: \string\firstpartis\space ****}
3631   \@mtc@hints@given@true
3632 \fi
```

`\if@firstchapteris@used@` Obsolete macro `\firstchapteris`:

```
3633 \if@firstchapteris@used@
3634   \PackageWarningNoLine{minitoc(hints)}%
3635   {**** You have invoked an obsolete command: \string\firstchapteris\space ****}
3636   \@mtc@hints@given@true
3637 \fi
```

`\if@firstsectionis@used@` Obsolete macro `\firstsectionis`:

```
3638 \if@firstsectionis@used@
3639   \PackageWarningNoLine{minitoc(hints)}%
3640   {**** You have invoked an obsolete command: \string\firstsectionis\space ****}
3641   \@mtc@hints@given@true
3642 \fi
```

### 5.76.2.9 Check if some hints have been written

`\if@mtc@hints@given@` We come at the end of the third part of the `hints` option: if problems have been detected, a warning is displayed; the warning is not displayed but only written in the `.log` file if no problems have been detected. And we terminate the `\mtc@hints@enddoc` macro by a closing brace.

```
3643 \if@mtc@hints@given@
3644   \PackageWarningNoLine{minitoc(hints)}%
3645   {***** Some hints have been written\MessageBreak
3646       in the \jobname.log file. *****}
3647 \else
3648   \PackageInfo{minitoc(hints)}%
3649   {***** No hints have been written\MessageBreak
3650       in the \jobname.log file. *****\@gobble}
3651 \fi
3652 }
```

## 5.77 Processing of options

`\ExecuteOptionsenglish` First, we apply the default language option, `english`:

```
3653 \ExecuteOptions{english}
```

`\ProcessOptions*` Then, we execute all requested options: for most options, it is just setting a flag, or loading a file for the language options.

```
3654 \ProcessOptions*
```

We now examine the flags for some options and execute the necessary actions.

### 5.77.1 Processing the `insection` option

```
\if@mtc@ss@insection@ For the insection option, we load the placeins package [3] with its options verbose,
\RequirePackage section and below; the correct loading is verified:
\@ifpackageloaded
\if@mtc@placeinsLoaded@ 3655 \if@mtc@ss@insection@
3656 \RequirePackage[section,below,verbose]{placeins}[2005/04/18]%
3657 \@ifpackageloaded{placeins}%
3658 {\@mtc@placeinsLoaded@true}%
3659 {\@mtc@placeinsLoaded@false}%
3660 \fi
```

### 5.77.2 Processing the `notoccite` option

```
\if@mtc@notoccite@ For the notoccite option, we just load the notoccite package [2]:
\RequirePackage
3661 \if@mtc@notoccite@
3662 \RequirePackage{notoccite}%
3663 \fi
```

### 5.77.3 Processing the `hints` option

```
\if@mtc@hints@ For the hints option, we set its first part in a \AtBeginDocument block and its third
\AtBeginDocument (last) part in a \AtEndDocument block:
\mtc@hints@begindoc
\AtEndDocument 3664 \if@mtc@hints@
\mtc@hints@enddoc 3665 \AtBeginDocument{\mtc@hints@begindoc}%
3666 \AtEndDocument{\mtc@hints@enddoc}%
3667 \fi
```

### 5.77.4 Saving the sectionning commands

And, at least, we save the definitions of sectionning commands (and of their unstarred and starred branches), for comparaisons (this is a part of the `hints` option executed in the preamble):

```
\ifundefined For the \part command:
\mtc@hints@part
\part
\mtc@hints@@part
\@part
\mtc@hints@spart
\@spart
```



```

3668 \ifundefined{part}{}{\let\mtc@hints@part\part
3669                               \let\mtc@hints@@part\@part
3670                               \let\mtc@hints@@spart\@spart}

```

```

\ifundefined For the \chapter command:
\mtc@hints@chapter
\chapter 3671 \ifundefined{chapter}{}{\let\mtc@hints@chapter\chapter
\mtc@hints@@chapter 3672                               \let\mtc@hints@@chapter\@chapter
\chapter 3673                               \let\mtc@hints@@schapter\@schapter}
\mtc@hints@@schapter
\@schapter

```

```

\ifundefined For the \section command:
\mtc@hints@section
\section 3674 \ifundefined{section}{}{\let\mtc@hints@section\section
\mtc@hints@@sect 3675                               \let\mtc@hints@@sect\@sect
\@sect 3676                               \let\mtc@hints@@ssect\@ssect}
\mtc@hints@@ssect
\@ssect

```

And the package is terminated.

```
3677 </minitoc>
```

## 5.78 The `fminitoc.dtx` file

`\jobname` This short file is necessary to create the french documentation. Its rôle is to set `\jobname`  
`\input` to `fminitoc` in place of `fminitoc`. As `minitoc.ins` generates the files `minitoc.lan` and `fminitoc.lan` which sets a language number `\LANG`, and `minitoc.dtx` reads `\jobname.lan`, the documentation can be in several languages (english and french) in `minitoc.dtx`, the language being selected by `\ifcase\LANG\relax ... \or\relax ... \fi` constructs. The `\relax` primitives are necessary to avoid bad surprises.

```

3678 <fminitoc>
3679 \ProvidesFile{fminitoc.dtx}[2005/09/16 minitoc french documentation start file]
3680 \input minitoc.dtx
3681 </fminitoc>

```

## Chapter 6

# Commented code of `mtcoff.sty`

### Contents

---

6.1	Why <code>mtcoff.sty</code> ?	210
6.2	Identification of the package	211
6.3	Faking counters and dimensions	211
6.4	Faking simple user commands	212
6.5	Faking user commands with one optional argument	212
6.6	Disabling the internal commands	213
6.7	Disabling the font commands	214
6.8	Disabling the <code>\mtcset...</code> commands	215
6.9	Disabling the new <code>\l@...</code> commands	215
6.10	Ignore the obsolete commands	215
6.11	Disabling the <code>\mtcselectlanguage</code> command	216
6.12	Disabling the <code>\mtcloadmlo</code> command	216
6.13	Disabling the commands for the horizontal rules	216
6.14	Disabling the commands for the page numbers	217
6.15	Disabling the mini-table features commands	217
6.16	Disabling miscellaneous flags and commands	218
6.17	Caution for some commands	218
6.18	Disabling commands for “coffee”	219
6.19	Disabling the <code>mtchideinmain...</code> environments	220
6.20	Disabling the <code>\mtcfixindex</code> command	220
6.21	Disabling the <code>\mtcfixglossary</code> command	220
6.22	Disabling the <code>\addstarred...</code> commands	220

---

### 6.1 Why `mtcoff.sty`?

The `minitoc` package [20, 19] requires that the user inserts many commands in the source code of her/his document, and not only into the preamble of the document. Hence the concept of a replacement package, `mtcoff` (means “minitoc off”), which substitutes to

all commands and environments of the `minitoc` package some alternative commands and environments with the same names and syntaxes, but doing nothing (except emitting some harmless warnings, for special cases). This way, to turn off easily the `minitoc` package, you just have to write, in the preamble of your document, something like:

```
\usepackage[...options...]{minitoc}
%\usepackage{mtcoff}
```

then the `minitoc` package is activated with the specified options. If you modify this two lines this way:

```
%\usepackage[...options...]{minitoc}
\usepackage{mtcoff}
```

then the `minitoc` package is deactivated and all its commands and environments are ignored. This is much easier, faster and safer than commenting out all the commands and environments of `minitoc`. Moreover, this operation is reversible.

## 6.2 Identification of the package

```
\NeedsTeXFormat{First, we identify the package and check the version of LATEX1:
\ProvidesPackage{
3682 \*mtcoff}
3683 \NeedsTeXFormat{LaTeX2e}%
3684 \ProvidesPackage{mtcoff}[2005/09/16 v43 The mtcoff package]}
```

## 6.3 Faking counters and dimensions

```
\count@ As minitoc declares some counters and dimensions registers, we fake them us-
\dimen@ ing \count@ or \dimen@. For \mtcskipamount, we use its default definition,
\c@minitocdepth \bigskipamount.
\mtcindent
\mtcskipamount 3685 \let\c@minitocdepth\count@
\bigskipamount 3686 \let\mtcindent\dimen@
\c@parttocdepth 3687 \let\mtcskipamount\bigskipamount
\mtcindent 3688 \let\c@parttocdepth\count@
\c@secttocdepth 3689 \let\ptcindent\dimen@
\stcindent 3690 \let\c@secttocdepth\count@
\c@mtc 3691 \let\stcindent\dimen@
\c@ptc
\c@stc
```

---

<sup>1</sup>This checking is not really useful for the `mtcoff` package itself, but it is good to check that your version of L<sup>A</sup>T<sub>E</sub>X is not too old to support `minitoc`.

```

3692 \let\c@mtc\count@
3693 \let\c@ptc\count@
3694 \let\c@stc\count@

```

## 6.4 Faking simple user commands

```

\mtcskip Some user commands are easy to fake:
\fakeableofcontents
\fakeableofcontents 3695 \let\mtcskip\relax
\fakeableoftables 3696 \let\fakeableofcontents\relax
\adjustptc 3697 \let\fakeableoftables\relax
\adjustmtc 3698 \let\fakeableoftables\relax
\adjuststc 3699 \newcommand{\adjustptc}[1][1]{\relax}
\decrementptc 3700 \newcommand{\adjustmtc}[1][1]{\relax}
\decrementmtc 3701 \newcommand{\adjuststc}[1][1]{\relax}
\decrementstc 3702 \let\decrementptc\relax
\incrementptc 3703 \let\decrementmtc\relax
\incrementmtc 3704 \let\decrementstc\relax
\incrementstc 3705 \let\incrementptc\relax
\incrementstc 3706 \let\incrementmtc\relax
\incrementstc 3707 \let\incrementstc\relax

```

```

\partend The following commands are not directly called by the user, in normal circumstances, but
\partbegin must be faked:
\chapterend
\chapterbegin 3708 \let\partend\relax
\sectend 3709 \let\partbegin\relax
\sectbegin 3710 \let\chapterend\relax
3711 \let\chapterbegin\relax
3712 \let\sectend\relax
3713 \let\sectbegin\relax

```

## 6.5 Faking user commands with one optional argument

\gobbleopt@ The user commands with an optional argument are faked using the internal  $\LaTeX$  macro \@ifnextchar (to get the optional argument) and the new utility command \gobbleopt@.

```

3714 \def\gobbleopt@[#1]{\relax}

```

```

\@ifnextchar  Commands for part level mini-tables:
\doarttoc
\doartlof 3715 \def\doarttoc{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[1]}}
\doartlot 3716 \def\doartlof{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[1]}}
\parttoc 3717 \def\doartlot{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[1]}}
\partlof 3718 \def\parttoc{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[1]}}
\partlot 3719 \def\partlof{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[1]}}
3720 \def\partlot{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[1]}}

```

```

\@ifnextchar  Commands for chapter level mini-tables:
\dominitoc
\dominilof 3721 \def\dominitoc{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[1]}}
\dominilot 3722 \def\dominilof{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[1]}}
\minitoc 3723 \def\dominilot{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[1]}}
\minilof 3724 \def\minitoc{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[1]}}
\minilot 3725 \def\minilof{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[1]}}
3726 \def\minilot{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[1]}}

```

```

\@ifnextchar  Commands for section level mini-tables:
\dosecttoc
\dosectlof 3727 \def\dosecttoc{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[1]}}
\dosectlot 3728 \def\dosectlof{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[1]}}
\secttoc 3729 \def\dosectlot{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[1]}}
\sectlof 3730 \def\secttoc{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[1]}}
\sectlot 3731 \def\sectlof{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[1]}}
3732 \def\sectlot{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[1]}}

```

## 6.6 Disabling the internal commands

```

\@gobbletwo  We need also to disable some minitoc commands, with \relax (macros with no
\tf@mtc      argument) or \@gobbletwo (macros with two arguments):
\mtc@string
\appendixmtc 3733 \let\tf@mtc\count@
\l@xchapter 3734 \let\mtc@string\relax
\l@xchapter 3735 \let\appendixmtc\relax
\pchapter 3736 \let\l@xchapter\@gobbletwo
\psection 3737 \let\l@xchapter\relax
\l@xpart 3738 \let\pchapter\relax
\l@xpart 3739 \let\psection\relax
\l@xsect 3740 \let\l@xpart\@gobbletwo
\l@xsect 3741 \let\l@xpart\relax
\l@xsect 3742 \let\l@xsect\@gobbletwo
3743 \let\l@xsect\relax

```

## 6.7 Disabling the font commands

`\empty` We disable the minitoc font commands (like `\mtcSSfont`) with `\empty`, because some users might have used:

```
\renewcommand{\mtcSSfont}{...}
```

which will not work if we use `\relax` here.

```
\ptcfont  Fonts for part level mini-tables:
\ptcCfont
\ptcSfont 3744 \let\ptcfont\empty
\ptcSSfont 3745 \let\ptcCfont\empty
\ptcSSSfont 3746 \let\ptcSfont\empty
\ptcPfont 3747 \let\ptcSSfont\empty
\ptcSPfont 3748 \let\ptcSSSfont\empty
\plffont 3749 \let\ptcPfont\empty
\pltfont 3750 \let\ptcSPfont\empty
\ptifont 3751 \let\plffont\empty
          3752 \let\pltfont\empty
          3753 \let\ptifont\empty

\mtcfont  Fonts for chapter level mini-tables:
\mtcSfont
\mtcSSfont 3754 \let\mtcfont\empty
\mtcSSSfont 3755 \let\mtcSfont\empty
\mtcPfont 3756 \let\mtcSSfont\empty
\mtcSPfont 3757 \let\mtcSSSfont\empty
\mlffont 3758 \let\mtcPfont\empty
\mltfont 3759 \let\mtcSPfont\empty
\mtifont 3760 \let\mlffont\empty
          3761 \let\mltfont\empty
          3762 \let\mtifont\empty

\stcfont  Fonts for section level mini-tables:
\stcSSfont
\stcSSSfont 3763 \let\stcfont\empty
\stcPfont 3764 \let\stcSSfont\empty
\stcSPfont 3765 \let\stcSSSfont\empty
\slffont 3766 \let\stcPfont\empty
\sltfont 3767 \let\stcSPfont\empty
\stifont 3768 \let\slffont\empty
          3769 \let\sltfont\empty
          3770 \let\stifont\empty
```

`\coffeeont` Font for “coffee” lines:

```
3771 \let\coffeeont\empty
```

## 6.8 Disabling the `\mtcset...` commands

`\@gobbletwo` These commands use two or three mandatory arguments:

```
\mtcsetdepth
\mtcsetfont 3772 \let\mtcsetdepth\@gobbletwo
\mtcsettitlefont 3773 \def\mtcsetfont#1#2#3{\empty}
\mtcsettitle 3774 \let\mtcsettitlefont\@gobbletwo
\mtcsetformat 3775 \let\mtcsettitle\@gobbletwo
\mtcsetfeature 3776 \def\mtcsetformat#1#2#3{\empty}
\mtcsetpagenumbers 3777 \def\mtcsetfeature#1#2#3{\empty}
\mtcsetrules 3778 \let\mtcsetpagenumbers\@gobbletwo
3779 \let\mtcsetrules\@gobbletwo
```

## 6.9 Disabling the new `\l@...` commands

`\l@starpart` The minitoc package defines the `\l@starXXX` commands to format TOC entries for starred sectioning commands. We reset to the unstarred version, when necessary:

```
\l@starchapter
\l@starsection
\l@starsubsection 3780 \@ifundefined{part}{}{\let\l@starpart\l@part}
\l@starsubsubsection 3781 \@ifundefined{chapter}{}{\let\l@starchapter\l@chapter}
\l@starparagraph 3782 \@ifundefined{section}{}{\let\l@starsection\l@section}
\l@starsubparagraph 3783 \@ifundefined{subsection}{}{\let\l@starsubsection\l@subsection}
3784 \@ifundefined{subsubsection}{}{\let\l@starsubsubsection\l@subsubsection}
3785 \@ifundefined{paragraph}{}{\let\l@starparagraph\l@paragraph}
3786 \@ifundefined{subparagraph}{}{\let\l@starsubparagraph\l@subparagraph}
```

## 6.10 Ignore the obsolete commands

`\@gobble` We just ignore the obsolete commands (with one mandatory argument):

```
\firstpartis
\firstchapteris 3787 \let\firstpartis\@gobble
\firstsectionis 3788 \let\firstchapteris\@gobble
3789 \let\firstsectionis\@gobble
```

## 6.11 Disabling the `\mtcselectlanguage` command

```

\@gobble This command has one mandatory argument:
\mtcselectlanguage
3790 \let\mtcselectlanguage\@gobble

```

## 6.12 Disabling the `\mtcloadmlo` command

```

\@gobble This command has one mandatory argument:
\mtcloadmlo
3791 \let\mtcloadmlo\@gobble

```

## 6.13 Disabling the commands for the horizontal rules

```

\ptcrule These commands have no argument:
\noptcrule
\mtcrule 3792 \let\ptcrule\relax
\nomtcrule 3793 \let\noptcrule\relax
\stcrule 3794 \let\mtcrule\relax
\nostcrule 3795 \let\nomtcrule\relax
\plfrule 3796 \let\stcrule\relax
\noplfrule 3797 \let\nostcrule\relax
\mlfrule 3798 \let\plfrule\relax
\nomlfrule 3799 \let\noplfrule\relax
\slfrule 3800 \let\mlfrule\relax
\noslfrule 3801 \let\nomlfrule\relax
\pltrule 3802 \let\noslfrule\relax
\nopltrule 3803 \let\pltrule\relax
\mltrule 3804 \let\nopltrule\relax
\nomltrule 3805 \let\mltrule\relax
\sltrule 3806 \let\nomltrule\relax
\nosltrule 3807 \let\sltrule\relax
3808 \let\nosltrule\relax
3809 \let\nosltrule\relax

```



## 6.14 Disabling the commands for the page numbers

```

\ptcpagenumbers  These commands have no argument:
\noptcpagenumbers
\mtcpagenumbers 3810 \let\mtcpagenumbers\relax
\nomtcpagenumbers 3811 \let\nomtcpagenumbers\relax
\stcpagenumbers 3812 \let\stcpagenumbers\relax
\nostcpagenumbers 3813 \let\nostcpagenumbers\relax
\plfpagenumbers 3814 \let\plfpagenumbers\relax
\noplfpagenumbers 3815 \let\noplfpagenumbers\relax
\mlfpagenumbers 3816 \let\mlfpagenumbers\relax
\nomlfpagenumbers 3817 \let\nomlfpagenumbers\relax
\slfpagenumbers 3818 \let\slfpagenumbers\relax
\noslfpagenumbers 3819 \let\noslfpagenumbers\relax
\pltpagenumbers 3820 \let\pltpagenumbers\relax
\nopltpagenumbers 3821 \let\nopltpagenumbers\relax
\mltpagenumbers 3822 \let\mltpagenumbers\relax
\nomltpagenumbers 3823 \let\nomltpagenumbers\relax
\sltpagenumbers 3824 \let\sltpagenumbers\relax
\nosltpagenumbers 3825 \let\nosltpagenumbers\relax
\pltpagenumbers 3826 \let\pltpagenumbers\relax
\nopltpagenumbers 3827 \let\nopltpagenumbers\relax

```

## 6.15 Disabling the mini-table features commands

We disable the commands for features (like `\beforeparttoc`) with `\empty`, because some users may have used:

```
\renewcommand{\beforeparttoc}{...}
```

which will not work if we use `\relax` here. These commands have no argument,

```

\beforeparttoc  Commands for part level mini-tables:
\beforepartlof
\beforepartlot 3828 \let\beforeparttoc\empty
\afterparttoc 3829 \let\beforepartlof\empty
\afterpartlof 3830 \let\beforepartlot\empty
\afterpartlot 3831 \let\afterparttoc\empty
\thispageparttocstyle 3832 \let\afterpartlof\empty
\thispagepartlofstyle 3833 \let\afterpartlot\empty
\thispagepartlotstyle 3834 \let\thispageparttocstyle\empty
3835 \let\thispagepartlofstyle\empty
3836 \let\thispagepartlotstyle\empty

```

```

\beforeminitoc  Commands for chapter level mini-tables:
\beforeminilof
\beforeminilot 3837 \let\beforeminitoc\empty
\afterminitoc  3838 \let\beforeminilof\empty
\afterminilof  3839 \let\beforeminilot\empty
\afterminilot  3840 \let\afterminitoc\empty
\thispageminitlet 3841 \let\afterminilof\empty
\thispageminilofstyle 3842 \let\afterminilot\empty
\thispageminilotstyle 3843 \let\thispageminitlet\empty
\thispageminilofstyle 3844 \let\thispageminilofstyle\empty
\thispageminilotstyle 3845 \let\thispageminilotstyle\empty

```

```

\beforesecttoc  Commands for section level mini-tables:
\beforesectlof
\beforesectlot 3846 \let\beforesecttoc\empty
\aftersecttoc  3847 \let\beforesectlof\empty
\aftersectlof  3848 \let\beforesectlot\empty
\aftersectlot  3849 \let\aftersecttoc\empty
\thispagesecttocstyle 3850 \let\aftersectlof\empty
\thispagesectlofstyle 3851 \let\aftersectlot\empty
\thispagesectlotstyle 3852 \let\thispagesecttocstyle\empty
\thispagesectlofstyle 3853 \let\thispagesectlofstyle\empty
\thispagesectlotstyle 3854 \let\thispagesectlotstyle\empty

```

## 6.16 Disabling miscellaneous flags and commands

```

\if@longextensions@  There are some flags and commands that it is wise to declare:
\iftightmtc
\ifktightmtc 3855 \newif\if@longextensions@ \@longextensions@true
\ifundottedmtc 3856 \newif\iftightmtc \tightmtcfalse
\l@listof 3857 \newif\ifktightmtc \ktightmtcfalse
\chapter 3858 \newif\ifundottedmtc \undottedmtcfalse
\chapter 3859 \let\l@listof\chapter

```



## 6.17 Caution for some commands

`\AtBeginDocument` Some minitoc commands should eventually be replaced if you decide to *definitely* stop using the minitoc package with your document. So we declare a flag and a `\AtEndDocument` block to signal that you have used these commands:

```

3860 \newif\ifmtcoffwarn@ \mtcoffwarn@false
3861 \AtEndDocument{\ifmtcoffwarn@
3862   \PackageWarningNoLine{mtcoff}%

```

```

3863      {You should scan (backwards) your .log file to find
3864      \MessageBreak
3865      some commands needing to be replaced if you decide to
3866      \MessageBreak
3867      DEFINITELY stop using minitoc for this document.
3868      \MessageBreak
3869      It is more wise to keep the \string\usepackage\space lines for
3870      \MessageBreak
3871      minitoc and mtcoff and to comment out only one of them}
3872 \fi}

```

```

\mtcaddchapter Then these commands are disabled and they set the flag and give a warning (useful to get
\mtcaddsection the line number):
\mtcaddpart
\ifmtcoffwarn@ 3873 \newcommand{\mtcaddchapter}[1][\mtcoffwarn@true
\mtc@ck 3874 \PackageWarning{mtcoff}%
\addcontentsline 3875 {\protect\mtcaddchapter{...} should be replaced\MessageBreak
3876 by \protect\addcontentsline{toc}{chapter}{...}\MessageBreak}
3877 \def\mtc@ck{#1}
3878 \ifx\mtc@ck\empty
3879 \else
3880 \addcontentsline{toc}{chapter}{#1}%
3881 \fi}
3882 \newcommand{\mtcaddsection}[1][\mtcoffwarn@true
3883 \PackageWarning{mtcoff}%
3884 {\protect\mtcaddsection{...} should be replaced\MessageBreak
3885 by \protect\addcontentsline{toc}{section}{...}\MessageBreak}
3886 \def\mtc@ck{#1}
3887 \ifx\mtc@ck\empty
3888 \else
3889 \addcontentsline{toc}{part}{#1}%
3890 \fi}
3891 \newcommand{\mtcaddpart}[1][\mtcoffwarn@true
3892 \PackageWarning{mtcoff}%
3893 {\protect\mtcaddpart{...} should be replaced\MessageBreak
3894 by \protect\addcontentsline{toc}{part}{...}\MessageBreak}
3895 \def\mtc@ck{#1}
3896 \ifx\mtc@ck\empty
3897 \else
3898 \addcontentsline{toc}{part}{#1}%
3899 \fi}

```

## 6.18 Disabling commands for “coffee”

```

\addcoffeeline We disable the commands relative to “coffee” lines, and the specific version of contents
\coffeeline lines without leaders of dots:
\@gobble
\@Undottedtocline 3900 \def\addcoffeeline#1#2#3{\relax}
\@Undottedtoclinep

```

```

3901 \let\coffeeline@gobble
3902 \let\l@coffee\relax
3903 \def\@Undottedtocline#1#2#3#4#5{\relax}
3904 \def\@Undottedtoclinep#1#2#3#4#5{\relax}

```

## 6.19 Disabling the `mtchideinmain...` environments

`mtchideinmaintoc` These environments accept one optional argument:  
`mtchideinmainlof`  
`mtchideinmainlot`

```

3905 \newenvironment{mtchideinmaintoc}[1][-1]%
3906     {\empty}{\empty}
3907 \newenvironment{mtchideinmainlof}[1][-1]%
3908     {\empty}{\empty}
3909 \newenvironment{mtchideinmainlot}[1][-1]%
3910     {\empty}{\empty}

```

## 6.20 Disabling the `\mtcfindex` command

`\mtcfindex` This command accepts one optional argument:

```

3911 \newcommand{\mtcfindex}[1][\relax]

```

## 6.21 Disabling the `\mtcfindexglossary` command

`\mtcfindex` This command accepts one optional argument:

```

3912 \newcommand{\mtcfindexglossary}[1][\relax]

```

## 6.22 Disabling the `\addstarred...` commands

`\ifmtcoffwarn@` These commands should be replaced by standard commands, but `mtcoff` simulates and  
`\addstarredpart` gives a warning, which will be reminded at the end of document:  
`\addstarredchapter`  
`\addstarredsection`

```

3913 \def\addstarredpart#1{\mtcoffwarn@true
\addcontentsline 3914     \PackageWarning{mtcoff}%
3915     {\protect\addstarredpart{...} should be replaced by\MessageBreak

```

```

3916   \protect\addcontentsline{toc}{part}{...}\MessageBreak}
3917   \addcontentsline{toc}{part}{#1}}
3918 \def\addstarredchapter#1{\mtcoffwarn@true
3919   \PackageWarning{mtcoff}%
3920   {\protect\addstarredchapter{...} should be replaced by\MessageBreak
3921   \protect\addcontentsline{toc}{chapter}{...}\MessageBreak}
3922   \addcontentsline{toc}{chapter}{#1}}
3923 \def\addstarredsection#1{\mtcoffwarn@true
3924   \PackageWarning{mtcoff}%
3925   {\protect\addstarredsection{...} should be replaced by\MessageBreak
3926   \protect\addcontentsline{toc}{section}{...}\MessageBreak}
3927   \addcontentsline{toc}{section}{#1}}

```

And the `mtcoff` package is terminated.

```

3928 \end{mtcoff}

```

## Chapter 7

# Patch for the memoir class

This code must be loaded to fix an incompatibility of the minitoc package with recent versions of the memoir.cls class.

```
3929 (*mtcpatchmem)
3930 \NeedsTeXFormat{LaTeX2e}[1996/06/01]%
3931 \ProvidesPackage{mtcpatchmem}%
3932   [2005/09/16 v43 Package mtcpatchmem]
3933 \PackageInfo{mtcpatchmem}%
3934   {*** mtcpatchmem package to patch the memoir class ***\@gobble}
3935 \renewcommand{\@m@chapter}[1][]{%
3936   \def\ch@pt@c{#1}% capture first optional arg
3937   \ifnextchar[{\@chapter}{\@chapter[]}%
3938 }
3939 \def\@chapter[#1]#2{%
3940 % if |\ch@pt@c| is empty, no [ was found at all. Use #2| as
3941 % entry for all fields.
3942   \ifx\ch@pt@c\@empty
3943     \def\f@rtoc{#2}%
3944     \def\f@rhdr{#2}%
3945   \else
3946 % otherwise at least one [ was found. If #1| is empty then only
3947 % one was found.
3948     \let\f@rtoc\ch@pt@c
3949     \ifx\@empty#1\@empty
3950       \let\f@rhdr\ch@pt@c
3951     \else
3952       \def\f@rhdr{#1}%
3953     \fi
3954   \fi
3955   \ifnum \c@secnumdepth >\m@ne
3956     \if@mainmatter
3957       \refstepcounter{chapter}%
3958     \fi
3959   \fi
```

```

3960 \chaptermark{\f@rhdr}%
3961 \ifartopt
3962   \@makechapterhead{#2}%
3963   \@afterheading
3964 \else
3965   \insertchapterspace
3966   \if@twocolumn
3967     \@topnewpage[\@makechapterhead{#2}]%
3968   \else
3969     \@makechapterhead{#2}%
3970   \fi
3971   \@afterheading
3972 \fi
3973 \ifnum \c@secnumdepth >\m@ne
3974   \if@mainmatter
3975     \ifanappendix
3976       \addcontentsline{toc}{appendix}{%
3977         \protect\chapternumberline{\thechapter}\f@rtoc}%
3978     \else
3979       \addcontentsline{toc}{chapter}{%
3980         \protect\chapternumberline{\thechapter}\f@rtoc}%
3981     \fi
3982   \else
3983     \addcontentsline{toc}{chapter}{\f@rtoc}%
3984   \fi
3985 \else
3986   \addcontentsline{toc}{chapter}{\f@rtoc}%
3987 \fi
3988 \ifheadnameref\M@gettitle{\f@rhdr}\else\M@gettitle{\f@rtoc}\fi
3989 }
3990 </mtcpatchmem>

```

## Chapter 8

# The minitoc language definition (.mld) and object (.mlo) files

### Contents

---

8.1	“Acadian” language: <code>acadian.mld</code> . . . . .	227
8.2	“Acadien” language: <code>acadien.mld</code> . . . . .	228
8.3	“Afrikaan” language: <code>afrikaan.mld</code> . . . . .	228
8.4	“Afrikaans” language: <code>afrikaans.mld</code> . . . . .	228
8.5	“American” language: <code>american.mld</code> . . . . .	229
8.6	“Arab” language: <code>arab.mld</code> . . . . .	229
8.7	“Arabic” language: <code>arabic.mld</code> . . . . .	229
8.8	“Armenian” language: <code>armenian.mld</code> . . . . .	230
8.9	“Austrian” language: <code>austrian.mld</code> . . . . .	230
8.10	“Bahasa” language: <code>bahasa.mld</code> . . . . .	230
8.11	“Bangla” language: <code>bangla.mld</code> . . . . .	231
8.12	“Basque” language: <code>basque.mld</code> . . . . .	231
8.13	“Bicig” language: <code>bicig.mld</code> . . . . .	232
8.14	“Brazil” language: <code>brazil.mld</code> . . . . .	232
8.15	“brazilian” language: <code>brazilian.mld</code> . . . . .	233
8.16	“Breton” language: <code>breton.mld</code> . . . . .	233
8.17	“british” language: <code>british.mld</code> . . . . .	234
8.18	“Bulgarian” language: <code>bulgarian.mld</code> . . . . .	234
8.19	“Bulgarianb” language: <code>bulgarianb.mld</code> . . . . .	235
8.20	“Buryat” language: <code>buryat.mld</code> . . . . .	235
8.21	“Canadian” language: <code>canadian.mld</code> . . . . .	236
8.22	“Canadien” language: <code>canadien.mld</code> . . . . .	236
8.23	“Castillan” language: <code>castillan.mld</code> . . . . .	236
8.24	“Castillian” language: <code>castillian.mld</code> . . . . .	237
8.25	“Catalan” language: <code>catalan.mld</code> . . . . .	237
8.26	“Chinese1” language: <code>chinese1.mld</code> and <code>chinese1.mlo</code> . . . .	237
8.27	“Chinese2” language: <code>chinese2.mld</code> and <code>chinese2.mlo</code> . . . .	238
8.28	“Croatian” language: <code>croatian.mld</code> . . . . .	238
8.29	“Czech” language: <code>czech.mld</code> . . . . .	239
8.30	“Danish” language: <code>danish.mld</code> . . . . .	239
8.31	“Dutch” language: <code>dutch.mld</code> . . . . .	240



8.32	“English” language: english.mld	240
8.33	“Esperant” language: esperant.mld	241
8.34	“Esperanto” language: esperanto.mld	241
8.35	“Estonian” language: estonian.mld	241
8.36	“Ethiopia” language: ethiopia.mld	242
8.37	“Ethiopian” language: ethiopian.mld	243
8.38	“Farsi1” language: farsi1.mld and farsi1.mlo	243
8.39	“Farsi2” language: farsi2.mld and farsi2.mlo	243
8.40	“Finnish” language: finnish.mld	244
8.41	“Finnish2” language: finnish2.mld	244
8.42	“Francais” language: francais.mld	245
8.43	“French” language: french.mld	245
8.44	“Frenchb” language: frenchb.mld	245
8.45	“Frenchle” language: frenchle.mld	246
8.46	“Frenchpro” language: frenchpro.mld	246
8.47	“Galician” language: galician.mld	246
8.48	“German” language: german.mld	247
8.49	“Germanb” language: germanb.mld	247
8.50	“Greek” language: greek.mld	248
8.51	“Greek-mono” language: greek-mono.mld	248
8.52	“Greek-polydemo” language: greek-polydemo.mld	249
8.53	“Greek-polykatha” language: greek-polykatha.mld	250
8.54	“Guarani” language: guarani.mld	251
8.55	“Hangul1” language: hangul1.mld and hangul1.mlo	252
8.56	“Hangul2” language: hangul2.mld and hangul2.mlo	252
8.57	“Hangul3” language: hangul3.mld and hangul3.mlo	253
8.58	“Hangul4” language: hangul4.mld and hangul4.mlo	253
8.59	“Hanja1” language: hanja1.mld and hanja1.mlo	254
8.60	“Hanja2” language: hanja2.mld and hanja2.mlo	255
8.61	“Hebrew” language: hebrew.mld	255
8.62	“Hungarian” language: hungarian.mld	256
8.63	“Icelandic” language: icelandic.mld	256
8.64	“Interlingua” language: interlingua.mld	257
8.65	“Irish” language: irish.mld	257
8.66	“Italian” language: italian.mld	257
8.67	“Japanese” language: japanese.mld and japanese.mlo	258
8.68	“Japanese2” language: japanese2.mld and japanese2.mlo	258
8.69	“Japanese3” language: japanese3.mld and japanese3.mlo	259
8.70	“Japanese4” language: japanese4.mld and japanese4.mlo	260
8.71	“Japanese5” language: japanese5.mld and japanese.mlo	260
8.72	“Latin” language: latin.mld	261
8.73	“Latin2” language: latin2.mld	261
8.74	“Latvian” language: latvian.mld	262
8.75	“Letton” language: letton.mld	262
8.76	“Lithuanian” language: lithuanian.mld	262
8.77	“Lsorbian” language: lsorbian.mld	263
8.78	“Magyar” language: magyar.mld	263
8.79	“Magyar2” language: magyar2.mld	264
8.80	“Malayalam-keli” language: malayalam-keli.mld	264
8.81	“Malayalam-rachana” language: malayalam-rachana.mld	265
8.82	“Malayalam-rachana2” language: malayalam-rachana2.mld	265
8.83	“Mongol” language: mongol.mld	266
8.84	“Naustrian” language: naustrian.mld	266
8.85	“Ngerman” language: ngerman.mld	267

8.86	“Ngermanb” language: ngermanb.mld	267
8.87	“Norsk” language: norsk.mld	268
8.88	“Nynorsk” language: nynorsk.mld	268
8.89	“Polish” language: polish.mld	269
8.90	“Polish2” language: polish2.mld	269
8.91	“Portuges” language: portuges.mld	270
8.92	“Portuguese” language: portuguese.mld	270
8.93	“Romanian” language: romanian.mld	270
8.94	“Russian” language: russian.mld	271
8.95	“Russian2m” language: russian2m.mld	271
8.96	“Russian2o” language: russian2o.mld	272
8.97	“Russianb” language: russianb.mld	273
8.98	“Russianc” language: russianc.mld	274
8.99	“Samin” language: samin.mld	275
8.100	“Scottish” language: scottish.mld	275
8.101	“Serbian” language: serbian.mld	276
8.102	“Serbianc” language: serbianc.mld	276
8.103	“Slovak” language: slovak.mld	277
8.104	“Slovene” language: slovene.mld	277
8.105	“Spanish” language: spanish.mld	278
8.106	“Spanish2” language: spanish2.mld	278
8.107	“Spanish3” language: spanish3.mld	279
8.108	“Swedish” language: swedish.mld	279
8.109	“Thai” language: thai.mld and thai.mlo	280
8.110	“Turkish” language: turkish.mld	280
8.111	“UKenglish” language: UKenglish.mld	280
8.112	“Ukraineb” language: ukraineb.mld	281
8.113	“USenglish” language: USenglish.mld	281
8.114	“Usorbian” language: lsorbian.mld	282
8.115	“Vietnam” language: vietnam.mld	282
8.116	“Vietnamese” language: vietnamese.mld	283
8.117	“Welsh” language: welsh.mld	283

This chapter shows the code of each .mld file. A .mld file is a *minitoc language definition* file, which defines the titles of the mini-tables for a given language. It contains often some comments about its origin, if you need further details. It is loaded either via a package option <sup>1</sup> in the \usepackage command for the minitoc package, either via the command:

```
\mtcselectlanguage      \mtcselectlanguage{<language>}
```

<sup>1</sup>It can also be a global option for the document.

`\ptctitle` Each .mld file must define the nine following commands (for the mini-tables of contents, mini-lists of figures and mini-lists of tables, at the part, chapter and section levels):

<code>\plftitle</code>			
<code>\plttitle</code>			
<code>\mtctitle</code>	• <code>\ptctitle</code>	• <code>\mtctitle</code>	• <code>\stctitle</code>
<code>\mlftitle</code>	• <code>\plftitle</code>	• <code>\mlftitle</code>	• <code>\slftitle</code>
<code>\mltttitle</code>			
<code>\stctitle</code>	• <code>\plttitle</code>	• <code>\mltttitle</code>	• <code>\sltttitle</code>
<code>\slftitle</code>			
<code>\sltttitle</code>			

Many .mld files require special fonts adequate for the corresponding language; as this is a language-dependent issue, the user must set up the correct language and font context for each language, like using the `babel` package [9, 10, 11], the CJK system [29], the ANTOMEGA system [24], the ArabTeX [25, 26], BangTeX [36], ethiop [6], FarsiTeX (<http://www.farsitex.org>), guarani [7], Malayalam [1], MonTeX [14, 15], or ArmTeX [16] packages. Note that it is often the *english* name of the language which is used to name the corresponding .mld file.



But for some oriental languages<sup>2</sup>, the source of the titles use some *exotic encodings*, difficult to manipulate in a .dtx file, the .mld file is then just a wrapper loading a .mlo file<sup>3</sup>, not generated by the .dtx files in the current version of minitoc package. The adequate input encoding must be set up by the user *before* loading the .mld file via `\mtcselectlanguage`.



`filecontents` To go around this limitation, the `minitoc.ins` file uses `filecontents` environments to generate the .mlo files.

## 8.1 “Acadian” language: `acadian.mld`

The `acadian` language<sup>4</sup> is just french, so we load the `french.mld` file (see section 8.43 on page 245):

```
3991 <*acadian>
3992 \ProvidesFile{acadian.mld}[2004/12/14]
3993 \mtcselectlanguage{french}%
3994 </acadian>
```

<sup>2</sup>Mainly for chinese, farsi (iranian) hangul (korean), hanja (korean), japanese and thai variants.

<sup>3</sup>The extension .mlo means *minitoc language object*.

<sup>4</sup>Spoken in some parts of the south of the USA, like Louisiane.

## 8.2 “Acadien” language: `acadien.mld`

The “acadien” language<sup>5</sup> is just french (“acadien” is the french term for “acadian”), so we load the `french.mld` file (see section 8.43 on page 245):

```
3995 <*acadien>
3996 \ProvidesFile{acadien.mld}[2004/12/14]
3997 \mtcselectlanguage{french}%
3998 </acadien>
```

## 8.3 “Afrikaan” language: `afrikaan.mld`

The titles for the “afrikaan” language<sup>6</sup> are taken from the `babel` package [10] [10]:

```
3999 <*afrikaan>
4000 \ProvidesFile{afrikaan.mld}[1999/03/16]
4001 %% Afrikaan(s) titles for minitoc.sty
4002 \def\ptctitle{Inhoudsopgawe}%
4003 \def\plftitle{Lys van figure}%
4004 \def\plttitle{Lys van tabelle}%
4005 %%
4006 \def\mtctitle{Inhoudsopgawe}%
4007 \def\mlftitle{Lys van figure}%
4008 \def\mlttitle{Lys van tabelle}%
4009 %%
4010 \def\stctitle{Inhoudsopgawe}%
4011 \def\slftitle{Lys van figure}%
4012 \def\slttitle{Lys van tabelle}%
4013 </afrikaan>
```

## 8.4 “Afrikaans” language: `afrikaans.mld`

“Afrikaans” is a synonym of “afrikaan”, so we just load `afrikaan.mld` (see section 8.3):

```
4014 <*afrikaans>
4015 \ProvidesFile{afrikaans.mld}[2004/12/14]
4016 \mtcselectlanguage{afrikaan}%
4017 </afrikaans>
```

---

<sup>5</sup>Spoken in some parts of the south of the USA, like Louisiana.

<sup>6</sup>Spoken in South Africa, it has dutch origins; compare with section 8.31 on page 240.

## 8.5 “American” language: `american.mld`

The “american” language is just like “english”<sup>7</sup>, so we just load `english.mld` (see section 8.32 on page 240):

```
4018 <*american>
4019 \ProvidesFile{american.mld}[2004/12/14]
4020 \mtcselectlanguage{english}%
4021 </american>
```

## 8.6 “Arab” language: `arab.mld`

The titles for the “arab” language are taken from the Arab<sub>T</sub><sub>E</sub>X package [25, 26], which should be used, with the associated fonts.

```
4022 <*arab>
4023 \ProvidesFile{arab.mld}[1999/03/16]
4024 %% Arabic titles for minitoc.sty
4025 %% Needs arabic fonts (cf. documentation of arabtex)
4026 %% (strings taken from arabtex; to be used with arabtex)
4027 \def\ptctitle{al-mu.htawayAtu}
4028 \def\plftitle{qA'imaTu a.s-.suwari}
4029 \def\plttitle{qA'imaTu al-^gadAwili}
4030 %%
4031 \def\mtctitle{al-mu.htawayAtu}
4032 \def\mlftitle{qA'imaTu a.s-.suwari}
4033 \def\mlttitle{qA'imaTu al-^gadAwili}
4034 %%
4035 \def\stctitle{al-mu.htawayAtu}
4036 \def\slftitle{qA'imaTu a.s-.suwari}
4037 \def\slttitle{qA'imaTu al-^gadAwili}
4038 </arab>
```

## 8.7 “Arabic” language: `arabic.mld`

“Arabic” is a synonym for “arab”, so we just load `arab.mld` (see section 8.6):

```
4039 <*arabic>
4040 \ProvidesFile{arabic.mld}[2005/02/09]
4041 \mtcselectlanguage{arab}%
4042 </arabic>
```

---

<sup>7</sup>It should be true for the mini-table titles; the languages themselves have some differences.

## 8.8 “Armenian” language: `armenian.mld`

The titles for the “armenian” language are taken from the `ArmTeX` package [16], which should be used, with the associated fonts.

```

4043 <*armenian>
4044 \ProvidesFile{armenian.mld}[1999/06/28]
4045 %% Armenian titles for minitoc.sty
4046 \def\ptctitle{Bovandakuthyun}
4047 \def\plftitle{Patkernerer cank}
4048 \def\pltttitle{Aghyusaknerer cank}
4049 %%
4050 \def\mtctitle{Bovandakuthyun}
4051 \def\mlftitle{Patkernerer cank}
4052 \def\mltttitle{Aghyusaknerer cank}
4053 %%
4054 \def\stctitle{Bovandakuthyun}
4055 \def\slftitle{Patkernerer cank}
4056 \def\sltttitle{Aghyusaknerer cank}
4057 </armenian>

```

## 8.9 “Austrian” language: `austrian.mld`

For the mini-table titles, the “austrian” language is like the “german” language, so we load `german.mld` (see section 8.48 on page 247):

```

4058 <*austrian>
4059 \ProvidesFile{austrian.mld}[2004/12/14]
4060 \mtcselectlanguage{german}%
4061 </austrian>

```

## 8.10 “Bahasa” language: `bahasa.mld`

The titles of the mini-tables for the “bahasa” language<sup>8</sup> are taken from the `babel` package [10]. Specific fonts are needed.

```

4062 <*bahasa>
4063 \ProvidesFile{bahasa.mld}[1999/03/16]
4064 %% Bahasa titles for minitoc.sty
4065 \def\ptctitle{Daftar Isi}%
4066 \def\plftitle{Daftar Gambar}%
4067 \def\pltttitle{Daftar Tabel}%
4068 %%

```

---

<sup>8</sup>Spoken in Indonesia and Malaysia, with different pronunciations but the same writing.

```

4069 \def\mtctitle{Daftar Isi}%
4070 \def\mlftitle{Daftar Gambar}%
4071 \def\mltttitle{Daftar Tabel}%
4072 %%
4073 \def\stctitle{Daftar Isi}%
4074 \def\slftitle{Daftar Gambar}%
4075 \def\slttitle{Daftar Tabel}%
4076 </bahasa>

```

## 8.11 “Bangla” language: bangla.mld

The titles for the “bangla” language<sup>9</sup> are taken from the `bangtex` package [36]; they need specific fonts.

```

4077 < *bangla>
4078 \ProvidesFile{bangla.mld}[2002/03/14]
4079 %% Bangla titles for minitoc.sty
4080 %% Needs specific fonts
4081 \def\ptctitle{suu\*c*i potRo}
4082 \def\plftitle{cho\*b*ir ta\*l*ika}
4083 \def\pltttitle{cho\*k*er ta\*l*ika}
4084 %%
4085 \def\mtctitle{suu\*c*i}
4086 \def\mlftitle{cho\*b*ir ta\*l*ika}
4087 \def\mltttitle{cho\*k*er ta\*l*ika}
4088 %%
4089 \def\stctitle{suu\*c*i}
4090 \def\slftitle{cho\*b*ir ta\*l*ika}
4091 \def\slttitle{cho\*k*er ta\*l*ika}
4092 </bangla>

```

## 8.12 “Basque” language: basque.mld

The titles for the “basque” language<sup>10</sup> are taken from the `babel` package [10]. It seems that special fonts are needed.

```

4093 < *basque>
4094 \ProvidesFile{basque.mld}[1999/12/06]
4095 %% Basque titles for minitoc.sty
4096 %% Needs special fonts
4097 \def\ptctitle{Gaien Aurkibidea}
4098 \def\plftitle{Irudien Zerrenda}
4099 \def\pltttitle{Taulen Zerrenda}
4100 %%

```

---

<sup>9</sup>Spoken in Bangladesh and some parts of India.

<sup>10</sup>Spoken in the Basque country, in the north of Spain and south-west of France.

```

4101 \def\mtctitle{Gaien Aurkibidea}
4102 \def\mlftitle{Irudien Zerrenda}
4103 \def\mltttitle{Taulen Zerrenda}
4104 %%
4105 \def\stctitle{Gaien Aurkibidea}
4106 \def\slftitle{Irudien Zerrenda}
4107 \def\sltttitle{Taulen Zerrenda}
4108 </basque>

```

### 8.13 “Bicig” language: bicig.mld

The titles for the “bicig” language<sup>11</sup> are taken from the MonT<sub>E</sub>X package [14, 15]. This language requires specific fonts. See also section 8.83 on page 266.

```

4109 <{*bicig}
4110 \ProvidesFile{bicig.mld}[1999/03/16]
4111 %% needs mongol fonts
4112 %% Mongol (Bicig) titles for minitoc.sty
4113 \def\ptctitle{\bcg{GarciG}}
4114 \def\plftitle{\bcg{zuraG-un zigsaaIt}}
4115 \def\pltttitle{\bcg{k"usn"agti"iIn jagsaaIt}}
4116 %%
4117 \def\mtctitle{\bcg{GarciG}}
4118 \def\mlftitle{\bcg{zuraG-un zigsaaIt}}
4119 \def\mltttitle{\bcg{k"usn"agti"iIn jagsaaIt}}
4120 %%
4121 \def\stctitle{\bcg{GarciG}}
4122 \def\slftitle{\bcg{zuraG-un zigsaaIt}}
4123 \def\sltttitle{\bcg{k"usn"agti"iIn jagsaaIt}}
4124 </bicig>

```

### 8.14 “Brazil” language: brazil.mld

The titles for the “brazil” language<sup>12</sup> are taken from the babel package [10]:

```

4125 <{*brazil}
4126 \ProvidesFile{brazil.mld}[1999/03/16]
4127 %% Portugues (brazil) titles for minitoc.sty
4128 \def\ptctitle{Sum\`ario}%
4129 \def\plftitle{Lista de Figuras}%
4130 \def\pltttitle{Lista de Tabelas}%
4131 %%
4132 \def\mtctitle{Sum\`ario}%

```

<sup>11</sup>Bicig is a written form of the mongolian language.

<sup>12</sup>It is the portuguese dialect spoken in Brazil. These titles are *different* in Brazil and in Portugal. See section 8.92 on page 270.



```

4133 \def\mlftitle{Lista de Figuras}%
4134 \def\mltttitle{Lista de Tabelas}%
4135 %%
4136 \def\stctitle{Sum\'ario}%
4137 \def\slftitle{Lista de Figuras}%
4138 \def\sltttitle{Lista de Tabelas}%
4139 </brazil>

```

## 8.15 “brazilian” language: `brazilian.mld`

The “brazilian” language is just like “brazil”, so we just load `brazil.mld` (see section 8.14 on the page before):

```

4140 <{*brazilian}
4141 \ProvidesFile{brazilian.mld}[2005/07/11]
4142 \mtcselectlanguage{brazil}%
4143 </brazilian>

```

## 8.16 “Breton” language: `breton.mld`

The titles for the “breton” language<sup>13</sup> are taken from the `babel` package [10]:

```

4144 <{*breton}
4145 \ProvidesFile{breton.mld}[1999/03/16]
4146 %% Breton titles for minitoc.sty
4147 \def\ptctitle{Taolenn}
4148 \def\plftitle{Listenn ar Figurenno\'u}
4149 \def\pltttitle{Listenn an taolenno\'u}
4150 %%
4151 \def\mtctitle{Taolenn}
4152 \def\mlftitle{Listenn ar Figurenno\'u}
4153 \def\mltttitle{Listenn an taolenno\'u}
4154 %%
4155 \def\stctitle{Taolenn}
4156 \def\slftitle{Listenn ar Figurenno\'u}
4157 \def\sltttitle{Listenn an taolenno\'u}
4158 </breton>

```

---

<sup>13</sup>Spoken as a local celtic dialect in french Brittany.

## 8.17 “british” language: british.mld

The “british” language is just like “english”, so we just load english.mld (see section 8.32 on page 240):

```
4159 <{*british>
4160 \ProvidesFile{british.mld}[2005/07/11]
4161 \mtcselectlanguage{english}%
4162 </british>
```

## 8.18 “Bulgarian” language: bulgarian.mld

The titles for the “bulgarian” language are taken from the babel package [10]; they require specific cyrillic fonts. See also section 8.19 on the next page.

```
4163 <{*bulgarian>
4164 \ProvidesFile{bulgarian.mld}[2001/02/28]
4165 %% Bulgarian titles for minitoc.sty
4166 %% Needs special fonts
4167 \def\ptctitle{%
4168   {\cyr\CYRS\cyrhrdsn\cyrd\cyrhrdsn\cyrr\cyrzh\cyra\cyrn\cyri\cyre}}%
4169 \def\plftitle{%
4170   {\cyr\CYRS\cyrp\cyri\cyr\cyrhrdsn\cyrk\ %
4171     \cyrn\cyra\ \cyrf\cyri\cyrg\cyru\cyrr\cyri\cyrt\cyre}}%
4172 \def\pltttitle{%
4173   {\cyr\CYRS\cyrp\cyri\cyr\cyrhrdsn\cyrk\ %
4174     \cyrn\cyra\ \cyrt\cyra\cyrb\cyrl\cyri\cyrc\cyri\cyrt\cyre}}%
4175 %%
4176 \def\mtctitle{%
4177   {\cyr\CYRS\cyrhrdsn\cyrd\cyrhrdsn\cyrr\cyrzh\cyra\cyrn\cyri\cyre}}%
4178 \def\mlftitle{%
4179   {\cyr\CYRS\cyrp\cyri\cyr\cyrhrdsn\cyrk\ %
4180     \cyrn\cyra\ \cyrf\cyri\cyrg\cyru\cyrr\cyri\cyrt\cyre}}%
4181 \def\mltttitle{%
4182   {\cyr\CYRS\cyrp\cyri\cyr\cyrhrdsn\cyrk\ %
4183     \cyrn\cyra\ \cyrt\cyra\cyrb\cyrl\cyri\cyrc\cyri\cyrt\cyre}}%
4184 %%
4185 \def\stctitle{%
4186   {\cyr\CYRS\cyrhrdsn\cyrd\cyrhrdsn\cyrr\cyrzh\cyra\cyrn\cyri\cyre}}%
4187 \def\slftitle{%
4188   {\cyr\CYRS\cyrp\cyri\cyr\cyrhrdsn\cyrk\ %
4189     \cyrn\cyra\ \cyrf\cyri\cyrg\cyru\cyrr\cyri\cyrt\cyre}}%
4190 \def\sltttitle{%
4191   {\cyr\CYRS\cyrp\cyri\cyr\cyrhrdsn\cyrk\ %
4192     \cyrn\cyra\ \cyrt\cyra\cyrb\cyrl\cyri\cyrc\cyri\cyrt\cyre}}%
4193 </bulgarian>
```

## 8.19 “Bulgarianb” language: bulgarianb.mld

The titles for the “bulgarianb” (upper bulgarian) language are taken from the babel package [10]; they require specific cyrillic fonts. See also section 8.18 on the preceding page.

```

4194 <*bulgarianb>
4195 \ProvidesFile{bulgariand.mld}[2005/03/24]
4196 %% Upper bulgarian titles for minitoc.sty
4197 %% Needs cyrillic fonts for upper bulgarian
4198 \def\ptctitle{%
4199   {\cyr\CYRS\cyrrhdsn\cyrd\cyrrhdsn\cyrr\cyrrzh\cyra\cyrn\cyri\cyre}}%
4200 \def\plftitle{% Figuri
4201   {\cyr \CYRF\cyri\cyrg\cyru\cyrr\cyri}}%
4202 \def\plttitle{% Tablici
4203   {\cyr \CYRT\cyra\cyrb\cyrl\cyri\cyrc\cyri}}%
4204 %%
4205 \def\mtctitle{% Sydyrzhanie
4206   {\cyr\CYRS\cyrrhdsn\cyrd\cyrrhdsn\cyrr\cyrrzh\cyra\cyrn\cyri\cyre}}%
4207 \def\mlftitle{% Figurite
4208   {\cyr \CYRF\cyri\cyrg\cyru\cyrr\cyri}}%
4209 \def\mlttitle{% Tablici
4210   {\cyr \CYRT\cyra\cyrb\cyrl\cyri\cyrc\cyri}}%
4211 %%
4212 \def\stctitle{% Sydyrzhanie
4213   {\cyr\CYRS\cyrrhdsn\cyrd\cyrrhdsn\cyrr\cyrrzh\cyra\cyrn\cyri\cyre}}%
4214 \def\slftitle{% Figuri
4215   {\cyr \CYRF\cyri\cyrg\cyru\cyrr\cyri}}%
4216 \def\slttitle{% Tablici
4217   {\cyr \CYRT\cyra\cyrb\cyrl\cyri\cyrc\cyri}}%
4218 </bulgarianb>

```

## 8.20 “Buryat” language: buryat.mld

The titles for the “buryat” language <sup>14</sup> are taken from the MonTeX package [14, 15]. This language requires specific fonts. See also section 8.83 on page 266.

```

4219 <*buryat>
4220 \ProvidesFile{buryat.mld}[1999/03/16]
4221 %% Buryat titles for minitoc.sty
4222 %% Needs special fonts
4223 \def\ptctitle{{\mnr Gar{\sh}ag}}%
4224 \def\plftitle{{\mnr Zuraga"i jagsaalt}}%
4225 \def\plttitle{{\mnr X"usn"ag"at"a"i jagsaalt}}%
4226 %%
4227 \def\mtctitle{{\mnr Gar{\sh}ag}}%
4228 \def\mlftitle{{\mnr Zuraga"i jagsaalt}}%

```

<sup>14</sup>Spoken in some regions of Mongolia, in the Buryat republic, near Lake Baikal.

```

4229 \def\mltttitle{{\mnr X"usn"ag"at"a"i jagsaalt}}%
4230 %%
4231 \def\stctitle{{\mnr Gar{\sh}ag}}%
4232 \def\slfttitle{{\mnr Zuraga"i jagsaalt}}%
4233 \def\sltttitle{{\mnr X"usn"ag"at"a"i jagsaalt}}%
4234 </buryat>

```

## 8.21 “Canadian” language: canadian.mld



The “canadian” language (note the final “ian”) is just the *english* language spoken in Canada. We just load the file `english.mld` (see section 8.32 on page 240):

```

4235 <*canadian>
4236 \ProvidesFile{canadian.mld}[2004/12/14]
4237 \mtcselectlanguage{english}%
4238 </canadian>

```

## 8.22 “Canadien” language: canadien.mld



The “canadien” language (note the final “ien”) is just the *french* language spoken in Canada. We just load the file `french.mld` (see section 8.43 on page 245):

```

4239 <*canadien>
4240 \ProvidesFile{canadien.mld}[2004/12/14]
4241 \mtcselectlanguage{french}%
4242 </canadien>

```

## 8.23 “Castillan” language: castillan.mld

The “castillan” language is more known as “spanish”, but is spoken mainly in Castile, a part of central Spain. We just load the `spanish.mld` file (see section 8.105 on page 278):

```

4243 <*castillan>
4244 %% Castillan (spanish) titles for minitoc.sty
4245 \ProvidesFile{castillan.mld}[2004/12/14]
4246 \mtcselectlanguage{spanish}%
4247 </castillan>

```

## 8.24 “Castillian” language: `castillian.mld`

The “castillian” language is more known as “spanish”, but is spoken mainly in Castile, a part of central Spain. “Castillian” is the english name for “castillan”. We just load the `spanish.mld` file (see section 8.105 on page 278):

```
4248 <*castillian>
4249 \ProvidesFile{castillian.mld}[2005/07/01]
4250 %% Castillian (spanish) titles for minitoc.sty
4251 \mtcselectlanguage{spanish}%
4252 </castillian>
```

## 8.25 “Catalan” language: `catalan.mld`

The titles for the “catalan” language <sup>15</sup> are taken from the `babel` package [10]:

```
4253 <*catalan>
4254 \ProvidesFile{catalan.mld}[1999/03/16]
4255 %% Catalan titles for minitoc.sty
4256 \def\ptctitle{\'Index}%
4257 \def\plftitle{\'Index de figures}%
4258 \def\plttitle{\'Index de taules}%
4259 %%
4260 \def\mtctitle{\'Index}%
4261 \def\mlftitle{Figures}%
4262 \def\mlttitle{Taules}%
4263 %%
4264 \def\stctitle{\'Index}%
4265 \def\slftitle{Figures}%
4266 \def\slttitle{Taules}%
4267 </catalan>
```

## 8.26 “Chinese1” language: `chinese1.mld` and `chinese1.mlo`

There are several variants for the chinese language. The “chinese1” language uses titles taken from the `Bg5.cap` file in the CJK system [29]. Special fonts are needed, of course. See also section 8.27 on the following page.

The titles for the “chinese1” language contain characters that cannot be easily generated, hence we load `chinese1.mlo`.

```
4268 <*chinese1>
4269 \ProvidesFile{chinese1.mld}[2005/01/28]
```

<sup>15</sup>Spoken in Catalunya, the eastern part of Spain, around Barcelona.

```

4270%% From the file file Bg5.cap of the CJK package
4271%%   for using Asian logographs (Chinese/Japanese/Korean) with LaTeX2e
4272%%   created by Werner Lemberg <wl@gnu.org>
4273%% Version 4.5.2 (28-Mar-2003)
4274%%
4275%% Chinese captions
4276%%
4277%% character set: Big 5
4278%% encoding: Big 5
4279%%
4280\mtcloadmlo{chinese1}
4281</chinese1>

```

## 8.27 “Chinese2” language: chinese2.mld and chinese2.mlo

The “chinese2” language uses titles taken from the Bg5.cpx file in the CJK system [29]. Special fonts are needed, of course. See also section 8.26 on the page before.

The titles for the “chinese2” language contain characters that cannot be easily generated, hence we load chinese2.mlo.

```

4282<*chinese2>
4283\ProvidesFile{chinese2.mld}[2005/01/28]
4284%% From the file Bg5.cpx of the CJK package
4285%% for using Asian logographs (Chinese/Japanese/Korean) with LaTeX2e
4286%% created by Werner Lemberg <wl@gnu.org>
4287%%
4288%% Version 4.5.2 (28-Mar-2003)
4289%%
4290%% Chinese captions
4291%%
4292%% character set: Big 5
4293%% encoding: Big 5
4294%%
4295%% preprocessed
4296%%
4297\mtcloadmlo{chinese2}
4298</chinese2>

```

## 8.28 “Croatian” language: croatian.mld

The titles for the “croatian” language are taken from the babel package [10]:

```

4299<*croatian>
4300\ProvidesFile{croatian.mld}[1999/03/16]
4301%% Croatian titles for minitoc.sty

```

```

4302 \def\ptctitle{Sadr\v{z}aj}%
4303 \def\plftitle{Slike}%
4304 \def\plttitle{Tablice}%
4305 %%
4306 \def\mtctitle{Sadr\v{z}aj}%
4307 \def\mlftitle{Slike}%
4308 \def\mlttitle{Tablice}%
4309 %%
4310 \def\stctitle{Sadr\v{z}aj}%
4311 \def\slftitle{Slike}%
4312 \def\slttitle{Tablice}%
4313 </croatian>

```

## 8.29 “Czech” language: czech.mld

The titles for the “czech” language are taken from the babel package [10]:

```

4314 <*czech>
4315 \ProvidesFile{czech.mld}[1999/03/16]
4316 %% Czech titles for minitoc.sty
4317 %% Warning: defines \w as a ring accent
4318 \def\w#1{\accent'27 #1}
4319 \def\ptctitle{Obsah}%
4320 \def\plftitle{Seznam obr\'azk\w{u}}%
4321 \def\plttitle{Seznam tabulek}%
4322 %%
4323 \def\mtctitle{Obsah}%
4324 \def\mlftitle{Seznam obr\'azk\w{u}}%
4325 \def\mlttitle{Seznam tabulek}%
4326 %%
4327 \def\stctitle{Obsah}%
4328 \def\slftitle{Seznam obr\'azk\w{u}}%
4329 \def\slttitle{Seznam tabulek}%
4330 </czech>

```

## 8.30 “Danish” language: danish.mld

The titles for the “danish” language are taken from the babel package [10]:

```

4331 <*danish>
4332 \ProvidesFile{danish.mld}[1999/03/16]
4333 %% Danish titles for minitoc.sty
4334 \def\ptctitle{Indholdsfortegnelse}%
4335 \def\plftitle{Figurer}%
4336 \def\plttitle{Tabeller}%
4337 %%

```

```

4338 \def\mtctitle{Indholdsfortegnelse}%
4339 \def\mlftitle{Figurer}%
4340 \def\mltttitle{Tabeller}%
4341 %%
4342 \def\stctitle{Indholdsfortegnelse}%
4343 \def\slftitle{Figurer}%
4344 \def\slttitle{Tabeller}%
4345 </danish>

```

### 8.31 “Dutch” language: dutch.mld

The titles for the “dutch” language are taken from the babel package [10]:

```

4346 <{*dutch}>
4347 \ProvidesFile{dutch.mld}[1999/03/16]
4348 % Dutch titles for minitoc.sty
4349 \def\ptctitle{Inhoudsopgave}%
4350 \def\plftitle{Lijst van figuren}%
4351 \def\pltttitle{Lijst van tabellen}%
4352 %%
4353 \def\mtctitle{Inhoudsopgave}%
4354 \def\mlftitle{Lijst van figuren}%
4355 \def\mltttitle{Lijst van tabellen}%
4356 %%
4357 \def\stctitle{Inhoudsopgave}%
4358 \def\slftitle{Lijst van figuren}%
4359 \def\slttitle{Lijst van tabellen}%
4360 </dutch>

```

### 8.32 “English” language: english.mld

The titles for the “english” language are taken from the babel package [10]. See also sections 8.5 on page 229, 8.17 on page 234, 8.21 on page 236, 8.111 on page 280, and 8.113 on page 281. *It is the default language.*



```

4361 <{*english}>
4362 \ProvidesFile{english.mld}[1999/03/16]
4363 %% English titles for minitoc.sty
4364 \def\ptctitle{Table of Contents}
4365 \def\plftitle{List of Figures}
4366 \def\pltttitle{List of Tables}
4367 %%
4368 \def\mtctitle{Contents}
4369 \def\mlftitle{Figures}
4370 \def\mltttitle{Tables}
4371 %%

```



```

4372 \def\stctitle{Contents}
4373 \def\slftitle{Figures}
4374 \def\sltttitle{Tables}
4375 \end{english}

```

### 8.33 “Esperant” language: `esperant.mld`

The titles for the “esperant” (esperanto) language are taken from the `babel` package [10].

```

4376 \begin{esperant}
4377 \ProvidesFile{esperant.mld}[1999/03/16]
4378 %% Esperanto titles for minitoc.sty
4379 \def\ptctitle{Enhavo}%
4380 \def\plftitle{Listo de figuroj}%
4381 \def\plttitle{Listo de tabeloj}%
4382 %%
4383 \def\mtctitle{Enhavo}%
4384 \def\mlftitle{Listo de figuroj}%
4385 \def\mlttitle{Listo de tabeloj}%
4386 %%
4387 \def\stctitle{Enhavo}%
4388 \def\slftitle{Listo de figuroj}%
4389 \def\sltttitle{Listo de tabeloj}%
4390 \end{esperant}

```

### 8.34 “Esperanto” language: `esperanto.mld`

The “esperanto” and “esperant” languages are synonyms, so we just load the `esperant.mld` file (see section 8.33):

```

4391 \begin{esperanto}
4392 \ProvidesFile{esperanto.mld}[2004/12/14]
4393 \mtcselectlanguage{esperant}%
4394 \end{esperanto}

```

### 8.35 “Estonian” language: `estonian.mld`

The titles for the “estonian” language are taken from the `babel` package [10]:

```

4395 \begin{estonian}
4396 \ProvidesFile{estonian.mld}[1999/03/16]
4397 %% Estonian titles for minitoc.sty

```

```

4398 \def\ptctitle{Sisukord}%
4399 \def\plftitle{Joonised}%
4400 \def\plttitle{Tabelid}%
4401 %%
4402 \def\mtctitle{Sisukord}%
4403 \def\mlftitle{Joonised}%
4404 \def\mlttitle{Tabelid}%
4405 %%
4406 \def\stctitle{Sisukord}%
4407 \def\slftitle{Joonised}%
4408 \def\slttitle{Tabelid}%
4409 </estonian>

```

### 8.36 “Ethiopia” language: ethiopia.mld

The titles for the “ethiopia” language are taken from the `ethiop` package [6]. Specific fonts are needed.

```

4410 <*ethiopia>
4411 \ProvidesFile{ethiopia.mld}[1999/03/16]
4412 %% Needs special fonts
4413 \def\ptctitle{yezate}%
4414 \def\plftitle{%
4415   ya\eth@doaltchar{85}'elo\eth@doaltchar{109}
4416   mAwe\eth@doaltchar{187}}%
4417 \def\plttitle{%
4418   yasane\eth@doaltchar{176}ra\eth@doaltchar{149}
4419   mAwe\eth@doaltchar{187}}%
4420 %%
4421 \def\mtctitle{yezate}%
4422 \def\mlftitle{%
4423   ya\eth@doaltchar{85}'elo\eth@doaltchar{109}
4424   mAwe\eth@doaltchar{187}}%
4425 \def\mlttitle{%
4426   yasane\eth@doaltchar{176}ra\eth@doaltchar{149}
4427   mAwe\eth@doaltchar{187}}%
4428 %%
4429 \def\stctitle{yezate}%
4430 \def\slftitle{%
4431   ya\eth@doaltchar{85}'elo\eth@doaltchar{109}
4432   mAwe\eth@doaltchar{187}}%
4433 \def\slttitle{%
4434   yasane\eth@doaltchar{176}ra\eth@doaltchar{149}
4435   mAwe\eth@doaltchar{187}}%
4436 </ethiopia>

```

### 8.37 “Ethiopian” language: `ethiopian.mld`

The “ethiopian” language is just a synonym for the “ethiopia” language, so we just load the `ethiopia.mld` file (see section 8.36 on the preceding page):

```
4437 <{*ethiopian}>
4438 \ProvidesFile{ethiopian.mld}[2004/12/14]
4439 %% Needs ethiopian special fonts
4440 \mtcselectlanguage{ethiopia}%
4441 </ethiopian>
```

### 8.38 “Farsi1” language: `farsi1.mld` and `farsi1.mlo`

There are several variants for the farsi language, spoken in Iran. The “farsi1” language uses titles taken from the `farsi.sty` file in the FarsiTeX system (<http://www.farsitex.org>), by Dr Mohammad GHODSI, Roozbeh POURNADER, HASSAN ABOLHASSANI, and others. Special fonts are needed, of course. See also section 8.39.

The titles for the “farsi1” language contain characters that cannot be easily generated, hence we load `farsi1.mlo`.

```
4442 <{*farsi1}>
4443 \ProvidesFile{farsi1.mld}[2005/09/13]
4444 % From farsi.sty of the FarsiTeX project
4445 % by Dr Mohammad Ghodsi,
4446 % Roozbeh Pournader (roozbeh@sharif.edu),
4447 % Hassan Abolhassani, and others.
4448 % http://www.farsitex.org
4449 \mtcloadmlo{farsi1}
4450 </farsi1>
```

### 8.39 “Farsi2” language: `farsi2.mld` and `farsi2.mlo`

There are several variants for the farsi language, spoken in Iran. The “farsi2” language uses titles taken from the `farsi.sty` file in the FarsiTeX system (<http://www.farsitex.org>), by Dr Mohammad GHODSI, Roozbeh POURNADER, HASSAN ABOLHASSANI, and others. Special fonts are needed, of course. See also section 8.38.

The titles for the “farsi2” language contain characters that cannot be easily generated, hence we load `farsi2.mlo`.

```
4451 <{*farsi2}>
4452 \ProvidesFile{farsi2.mld}[2005/09/13]
4453 % From farsi.sty of the FarsiTeX project
```

```

4454 % by Dr Mohammad Ghodsi,
4455 % Roozbeh Pournader (roozbeh@sharif.edu),
4456 % Hassan Abolhassani, and others.
4457 % http://www.farsitex.org
4458 \mtcloadmlo{farsi2}
4459 </farsi2>

```

## 8.40 “Finnish” language: `finnish.mld`

The titles for the “finnish” language are taken from the babel package [10]. See also section 8.41.

```

4460 (*finnish)
4461 \ProvidesFile{finnish.mld}[1999/03/16]
4462 %% Finnish titles for minitoc.sty
4463 \def\ptctitle{Sis\alt{o}}% /* Could be "Sis\allys" as well */
4464 \def\plftitle{Kuvat}%
4465 \def\plttitle{Taulukot}%
4466 %%
4467 \def\mtctitle{Sis\alt{o}}% /* Could be "Sis\allys" as well */
4468 \def\mlftitle{Kuvat}%
4469 \def\mlttitle{Taulukot}%
4470 %%
4471 \def\stctitle{Sis\alt{o}}% /* Could be "Sis\allys" as well */
4472 \def\slftitle{Kuvat}%
4473 \def\slttitle{Taulukot}%
4474 </finnish>

```

## 8.41 “Finnish2” language: `finnish2.mld`

The titles for the “finnish2” language are taken from a variant proposed in the babel package [10]. See also section 8.40.

```

4475 (*finnish2)
4476 \ProvidesFile{finnish2.mld}[2005/04/06]
4477 %% Finnish titles for minitoc.sty
4478 \def\ptctitle{Sis\allys}%
4479 \def\plftitle{Kuvat}%
4480 \def\plttitle{Taulukot}%
4481 %%
4482 \def\mtctitle{Sis\allys}%
4483 \def\mlftitle{Kuvat}%
4484 \def\mlttitle{Taulukot}%
4485 %%
4486 \def\stctitle{Sis\allys}%
4487 \def\slftitle{Kuvat}%

```

```

4488 \def\sltttitle{Taulukot}%
4489 </finnish2>

```

## 8.42 “Francais” language: `francais.mld`

The “`francais`”<sup>16</sup> language is a synonym for the “`french`” language, so we load the file `french.mld` (see section 8.43):

```

4490 <(*francais)
4491 \ProvidesFile{francais.mld}[2004/12/14]
4492 \mtcselectlanguage{french}%
4493 </francais>

```

## 8.43 “French” language: `french.mld`

The titles for the “`french`” language are taken from the `babel` package [10]. See also sections 8.1 on page 227, 8.2 on page 228, 8.22 on page 236, 8.42, 8.44, 8.45 on the next page, and 8.46 on the following page.

```

4494 <(*french)
4495 \ProvidesFile{french.mld}[2005/06/16]
4496 %%
4497 \def\ptctitle{Table des Mati\`eres}
4498 \def\plftitle{Liste des Figures}
4499 \def\plttitle{Liste des Tableaux}
4500 %%
4501 %% French titles for minitoc.sty
4502 \def\mtctitle{Sommaire}
4503 \def\mlftitle{Figures}
4504 \def\mlttitle{Tableaux}
4505 %%
4506 \def\stctitle{Sommaire}
4507 \def\slftitle{Figures}
4508 \def\sltttitle{Tableaux}
4509 </french>

```

## 8.44 “Frenchb” language: `frenchb.mld`

The “`frenchb`” language is a synonym for the “`french`” language, so we load the `french.mld` file. See section 8.43.

---

<sup>16</sup>The right spelling is “français”, but we do not dare using a cedilla in a file name.

```

4510 <{*frenchb>
4511 \ProvidesFile{frenchb.mld}[2003/02/11]
4512 \mtcselectlanguage{french}%
4513 </frenchb>

```

## 8.45 “Frenchle” language: frenchle.mld

The “frenchle” language is a synonym for the “french” language, so we load the french.mld file. See section 8.43 on the preceding page.

```

4514 <{*frenchle>
4515 \ProvidesFile{frenchle.mld}[2003/02/20]
4516 \mtcselectlanguage{french}%
4517 </frenchle>

```

## 8.46 “Frenchpro” language: frenchpro.mld

The “frenchpro” language is a synonym for the “french” language, so we load the french.mld file. See section 8.43 on the page before.

```

4518 <{*frenchpro>
4519 \ProvidesFile{frenchpro.mld}[2003/02/20]
4520 \mtcselectlanguage{french}%
4521 </frenchpro>

```

## 8.47 “Galician” language: galician.mld

The titles for the “galician” language<sup>17</sup> are taken from the babel package [10]:

```

4522 <{*galician>
4523 \ProvidesFile{galician.mld}[1999/03/16]
4524 %% Galician titles for minitoc.sty
4525 %%
4526 \def\ptctitle{\`Indice Xeral}%
4527 \def\plftitle{\`Indice de Figuras}%
4528 \def\plttitle{\`Indice de T\`aboas}%
4529 %%
4530 \def\mtctitle{\`Indice Xeral}%
4531 \def\mlftitle{\`Indice de Figuras}%
4532 \def\mlttitle{\`Indice de T\`aboas}%
4533 %%

```

---

<sup>17</sup>Spoken in Galice, in the north-west part of Spain, around Santiago de Compostela.

```

4534 \def\stctitle{\`Indice Xeral}%
4535 \def\slftitle{\`Indice de Figuras}%
4536 \def\sltttitle{\`Indice de T\`aboas}%
4537 </galician>

```

## 8.48 “German” language: `german.mld`

The titles for the “german” language are taken from the babel package [10]. See also the section 8.9 on page 230.

```

4538 <*german>
4539 \ProvidesFile{german.mld}[1999/03/16]
4540 %% German titles for minitoc.sty
4541 \def\ptctitle{Inhaltsangabe}
4542 \def\plftitle{Figuren}
4543 \def\pltttitle{Tabellen}
4544 %%
4545 \def\mtctitle{Inhaltsangabe}
4546 \def\mlftitle{Figuren}
4547 \def\mltttitle{Tabellen}
4548 %%
4549 \def\stctitle{Inhaltsangabe}
4550 \def\slftitle{Figuren}
4551 \def\sltttitle{Tabellen}
4552 </german>

```

## 8.49 “Germanb” language: `germanb.mld`

The “germanb” language is a variant for the “german” language. The titles are taken from the babel package [10]:

```

4553 <*germanb>
4554 \ProvidesFile{germanb.mld}[1999/03/16]
4555 %% German titles (variant) for minitoc.sty
4556 \def\ptctitle{Inhaltsverzeichnis}%      % oder nur: Inhalt
4557 \def\plftitle{Abbildungsverzeichnis}
4558 \def\pltttitle{Tabellenverzeichnis}
4559 %%
4560 \def\mtctitle{Inhaltsverzeichnis}%      % oder nur: Inhalt
4561 \def\mlftitle{Abbildungsverzeichnis}
4562 \def\mltttitle{Tabellenverzeichnis}
4563 %%
4564 \def\stctitle{Inhalt}%      % oder nur: Inhalt
4565 \def\slftitle{Abbildungen}
4566 \def\sltttitle{Tabellen}
4567 </germanb>

```

## 8.50 “Greek” language: greek.mld

The titles for the “greek” language (modern greek) are taken from the babel package [10]. Greek fonts are required.

```

4568 <*greek>
4569 \ProvidesFile{greek.mld}[1999/03/16]
4570 %% greek.mld
4571 %% Needs greek fonts.
4572 \def\ptctitle{Perieq'omena}
4573 \def\plftitle{Kat'alogoc Sqhm'atwn}
4574 \def\plttitle{Kat'alogoc Pin'akwn}
4575 %%
4576 \def\mtctitle{Perieq'omena}
4577 \def\mlftitle{Kat'alogoc Sqhm'atwn}
4578 \def\mlttitle{Kat'alogoc Pin'akwn}
4579 %%
4580 \def\stctitle{Perieq'omena}
4581 \def\slftitle{Kat'alogoc Sqhm'atwn}
4582 \def\slttitle{Kat'alogoc Pin'akwn}
4583 </greek>

```

## 8.51 “Greek-mono” language: greek-mono.mld

The titles for the “greek-mono” language<sup>18</sup> are taken from the ANTOMEGA project [24]:

```

4584 <*greek – mono>
4585 \ProvidesFile{greek-mono.mld}[2005/02/08]
4586 %% from omega-greek.ldf (Antomega project)
4587 %% Needs Omega
4588 %% Alexej M. Kryokov
4589 %% Dmitry Ivanov
4590 %%
4591 \def\ptctitle{\localgreek%
4592 {^03a0^03b5^03c1^03b9^03b5^03c7^03cc^03bc%
4593 ^03b5^03bd^03b1}}%
4594 \def\plftitle{\localgreek%
4595 {^039a^03b1^03c4^03ac^03bb^03bf^03b3^03bf%
4596 ^03c2 ^03c3^03c7^03b7^03bc^03ac^03c4^03c9%
4597 ^03bd}}%
4598 \def\mlftitle{\localgreek%
4599 {^039a^03b1^03c4^03ac^03bb^03bf^03b3^03bf%
4600 ^03c2 ^03c0^03b9^03bd^03ac^03ba^03c9^03bd}}%
4601 %%
4602 \def\mtctitle{\localgreek%
4603 {^03a0^03b5^03c1^03b9^03b5^03c7^03cc^03bc%

```

<sup>18</sup>Monotonic greek, from a recent (1982) but strongly contested – and contestable – reform of the greek language.



```

4604 ^^^^03b5^^^^03bd^^^^03b1}}%
4605 \def\mlftitle%\localgreek%
4606 {^^^^039a^^^^03b1^^^^03c4^^^^03ac^^^^03bb^^^^03bf^^^^03b3^^^^03bf%
4607 ^^^^03c2 ^^^^03c3^^^^03c7^^^^03b7^^^^03bc^^^^03ac^^^^03c4^^^^03c9%
4608 ^^^^03bd}}%
4609 \def\mlftitle%\localgreek%
4610 {^^^^039a^^^^03b1^^^^03c4^^^^03ac^^^^03bb^^^^03bf^^^^03b3^^^^03bf%
4611 ^^^^03c2 ^^^^03c0^^^^03b9^^^^03bd^^^^03ac^^^^03ba^^^^03c9^^^^03bd}}%
4612 %%
4613 \def\stctitle%\localgreek%
4614 {^^^^03a0^^^^03b5^^^^03c1^^^^03b9^^^^03b5^^^^03c7^^^^03cc^^^^03bc%
4615 ^^^^03b5^^^^03bd^^^^03b1}}%
4616 \def\slftitle%\localgreek%
4617 {^^^^039a^^^^03b1^^^^03c4^^^^03ac^^^^03bb^^^^03bf^^^^03b3^^^^03bf%
4618 ^^^^03c2 ^^^^03c3^^^^03c7^^^^03b7^^^^03bc^^^^03ac^^^^03c4^^^^03c9%
4619 ^^^^03bd}}%
4620 \def\slftitle%\localgreek%
4621 {^^^^039a^^^^03b1^^^^03c4^^^^03ac^^^^03bb^^^^03bf^^^^03b3^^^^03bf%
4622 ^^^^03c2 ^^^^03c0^^^^03b9^^^^03bd^^^^03ac^^^^03ba^^^^03c9^^^^03bd}}%
4623 </greek – mono>

```

## 8.52 “Greek-polydemo” language: greek-polydemo.mld

The titles for the “greek-polydemo” language<sup>19</sup> are taken from the ANTOMEGA project [24]:

```

4624 <*greek – polydemo>
4625 \ProvidesFile{greek-polydemo.mld}[2005/02/08]
4626 %% from omega-greek.ldf (Antomega project)
4627 %% Needs Omega
4628 %% Alexej M. Kryokov
4629 %% Dmitry Ivanov
4630 %%
4631 \def\ptctitle%\localgreek%
4632 {^^^^03a0^^^^03b5^^^^03c1^^^^03b9^^^^03b5^^^^03c7^^^^1f79^^^^03bc%
4633 ^^^^03b5^^^^03bd^^^^03b1}}%
4634 \def\plftitle%\localgreek%
4635 {^^^^039a^^^^03b1^^^^03c4^^^^1f71^^^^03bb^^^^03bf^^^^03b3^^^^03bf%
4636 ^^^^03c2 ^^^^03c3^^^^03c7^^^^03b7^^^^03bc^^^^1f71^^^^03c4^^^^03c9%
4637 ^^^^03bd}}%
4638 \def\pltttitle%\localgreek%
4639 {^^^^039a^^^^03b1^^^^03c4^^^^1f71^^^^03bb^^^^03bf^^^^03b3^^^^03bf%
4640 ^^^^03c2 ^^^^03c0^^^^03b9^^^^03bd^^^^1f71^^^^03ba^^^^03c9^^^^03bd}}%
4641 %%
4642 \def\mtctitle%\localgreek%
4643 {^^^^03a0^^^^03b5^^^^03c1^^^^03b9^^^^03b5^^^^03c7^^^^1f79^^^^03bc%
4644 ^^^^03b5^^^^03bd^^^^03b1}}%
4645 \def\mlftitle%\localgreek%
4646 {^^^^039a^^^^03b1^^^^03c4^^^^1f71^^^^03bb^^^^03bf^^^^03b3^^^^03bf%

```

<sup>19</sup>Polytonic demotic (popular) greek, for classical greek.

```

4647 ^^^^03c2 ^^^^03c3^^^^03c7^^^^03b7^^^^03bc^^^^1f71^^^^03c4^^^^03c9%
4648 ^^^^03bd}}}%
4649 \def\mltttitle{\localgreek%
4650 {^^^^039a^^^^03b1^^^^03c4^^^^1f71^^^^03bb^^^^03bf^^^^03b3^^^^03bf%
4651 ^^^^03c2 ^^^^03c0^^^^03b9^^^^03bd^^^^1f71^^^^03ba^^^^03c9^^^^03bd}}}%
4652 %%
4653 \def\stcttitle{\localgreek%
4654 {^^^^03a0^^^^03b5^^^^03c1^^^^03b9^^^^03b5^^^^03c7^^^^1f79^^^^03bc%
4655 ^^^^03b5^^^^03bd^^^^03b1}}}%
4656 \def\slfttitle{\localgreek%
4657 {^^^^039a^^^^03b1^^^^03c4^^^^1f71^^^^03bb^^^^03bf^^^^03b3^^^^03bf%
4658 ^^^^03c2 ^^^^03c3^^^^03c7^^^^03b7^^^^03bc^^^^1f71^^^^03c4^^^^03c9%
4659 ^^^^03bd}}}%
4660 \def\sltttitle{\localgreek%
4661 {^^^^039a^^^^03b1^^^^03c4^^^^1f71^^^^03bb^^^^03bf^^^^03b3^^^^03bf%
4662 ^^^^03c2 ^^^^03c0^^^^03b9^^^^03bd^^^^1f71^^^^03ba^^^^03c9^^^^03bd}}}%
4663 </greek – polydemo>

```

### 8.53 “Greek-polykatha” language: greek-polykatha.mld

The titles for the “greek-polykatha” language<sup>20</sup> are taken from the ANTOMEGA project [24]:

```

4664 <*greek – polykatha>
4665 \ProvidesFile{greek-polykatha.mld}[2005/02/08]
4666 %% from omega-greek.ldf (Antomega project)
4667 %% Needs Omega
4668 %% Alexej M. Kryokov
4669 %% Dmitry Ivanov
4670 %%
4671 \def\ptcttitle{\localgreek%
4672 {^^^^03a0^^^^03b5^^^^03c1^^^^03b9^^^^03b5^^^^03c7^^^^1f79^^^^03bc%
4673 ^^^^03b5^^^^03bd^^^^03b1}}}%
4674 \def\plfttitle{\localgreek%
4675 {^^^^039a^^^^03b1^^^^03c4^^^^1f71^^^^03bb^^^^03bf^^^^03b3^^^^03bf%
4676 ^^^^03c2 ^^^^03c3^^^^03c7^^^^03b7^^^^03bc^^^^1f71^^^^03c4^^^^03c9%
4677 ^^^^03bd}}}%
4678 \def\pltttitle{\localgreek%
4679 {^^^^039a^^^^03b1^^^^03c4^^^^1f71^^^^03bb^^^^03bf^^^^03b3^^^^03bf%
4680 ^^^^03c2 ^^^^03c0^^^^03b9^^^^03bd^^^^1f71^^^^03ba^^^^03c9^^^^03bd}}}%
4681 %%
4682 \def\mtcttitle{\localgreek%
4683 {^^^^03a0^^^^03b5^^^^03c1^^^^03b9^^^^03b5^^^^03c7^^^^1f79^^^^03bc%
4684 ^^^^03b5^^^^03bd^^^^03b1}}}%
4685 \def\mlfttitle{\localgreek%
4686 {^^^^039a^^^^03b1^^^^03c4^^^^1f71^^^^03bb^^^^03bf^^^^03b3^^^^03bf%

```

<sup>20</sup>Polytonic greek, « kathaverousa » (purified) style, a form of the Greek language created during the early 19th century by Adamantios KORAI, to purify the language from the Byzantine and non-greek vocabulary. It has now been obsoleted by the demotic (popular) greek, but it has left a very noticeable trace in the modern Greek language.

```

4687 ^^^^03c2 ^^^^03c3^^^^03c7^^^^03b7^^^^03bc^^^^1f71^^^^03c4^^^^03c9%
4688 ^^^^03bd}}}%
4689 \def\mltttitle{\localgreek%
4690 {^^^^039a^^^^03b1^^^^03c4^^^^1f71^^^^03bb^^^^03bf^^^^03b3^^^^03bf%
4691 ^^^^03c2 ^^^^03c0^^^^03b9^^^^03bd^^^^1f71^^^^03ba^^^^03c9^^^^03bd}}}%
4692 %%
4693 \def\stcttitle{\localgreek%
4694 {^^^^03a0^^^^03b5^^^^03c1^^^^03b9^^^^03b5^^^^03c7^^^^1f79^^^^03bc%
4695 ^^^^03b5^^^^03bd^^^^03b1}}}%
4696 \def\slfttitle{\localgreek%
4697 {^^^^039a^^^^03b1^^^^03c4^^^^1f71^^^^03bb^^^^03bf^^^^03b3^^^^03bf%
4698 ^^^^03c2 ^^^^03c3^^^^03c7^^^^03b7^^^^03bc^^^^1f71^^^^03c4^^^^03c9%
4699 ^^^^03bd}}}%
4700 \def\sltttitle{\localgreek%
4701 {^^^^039a^^^^03b1^^^^03c4^^^^1f71^^^^03bb^^^^03bf^^^^03b3^^^^03bf%
4702 ^^^^03c2 ^^^^03c0^^^^03b9^^^^03bd^^^^1f71^^^^03ba^^^^03c9^^^^03bd}}}%
4703 </greek – polykatha>

```

## 8.54 “Guarani” language: guarani.mld

The “guarani” language is the main language spoken in Paraguay. Very often, a mixture of Guarani and Spanish, known as Jopara, is spoken. A special input encoding (win-gn.def) is needed. The titles are taken from the guarani.ldf file by Javier BEZOS [7]. A special input encoding (win-gn.def) is needed. These files are available on the CTAN archives.

```

4704 <*guarani>
4705 \ProvidesFile{guarani.mld}[2005/08/26]
4706 %% Guarani titles for minitoc.sty
4707 %% from guarani.ldf by Javier Bezos.
4708 %% Input encoding win-gn.def is needed.
4709 %%
4710 \def\ptcttitle{\'Indice general}%
4711 \def\plfttitle{\'Indice de figuras}%
4712 \def\pltttitle{\'Indice de cuadros}%
4713 %%
4714 \def\mtcttitle{\'Indice general}%
4715 \def\mlfttitle{\'Indice de figuras}%
4716 \def\mltttitle{\'Indice de cuadros}%
4717 %%
4718 \def\stcttitle{\'Indice general}%
4719 \def\slfttitle{\'Indice de figuras}%
4720 \def\sltttitle{\'Indice de cuadros}%
4721 </guarani>

```

## 8.55 “Hangul1” language: hangul1.mld and hangul1.mlo

The titles for the “hangul1” language (korean in hangul script, first variant) are taken from the file hangul.cap of the CJK system [29]. Special fonts are needed, of course. See also sections 8.56, 8.57 on the following page, 8.58 on the next page, 8.59 on page 254, and 8.60 on page 255.

The titles for the “hangul1” language contain characters that cannot be easily generated, hence we load hangul1.mlo.

```

4722 (*hangul1)
4723 \ProvidesFile{hangul1.mld}[2005/01/28]
4724 %% From the file hangul.cap of the CJK package
4725 %% for using Asian logographs (Chinese/Japanese/Korean) with LaTeX2e
4726 %% created by Werner Lemberg <wl@gnu.org>
4727 %%
4728 %% Version 4.5.2 (28-Mar-2003)
4729 %%
4730 %% Hangul captions
4731 %%
4732 %% character set: KS X 1001:1992 (=KS C 5601-1992)
4733 %% encoding: EUC (=Wansung)
4734 %%
4735 \mtcloadmlo{hangul1}
4736 \hangul1

```

## 8.56 “Hangul2” language: hangul2.mld and hangul2.mlo

The titles for the “hangul2” language (korean in hangul script, second variant) are taken from the file hangul.cpx of the CJK system [29]. Special fonts are needed, of course. See also sections 8.55, 8.57 on the next page, 8.58 on the following page, 8.59 on page 254, and 8.60 on page 255.

The titles for the “hangul2” language contain characters that cannot be easily generated, hence we load hangul2.mlo.

```

4737 (*hangul2)
4738 \ProvidesFile{hangul2.mld}[2005/01/28]
4739 %% From the file hangul.cpx of the CJK package
4740 %% for using Asian logographs (Chinese/Japanese/Korean) with LaTeX2e
4741 %% created by Werner Lemberg <wl@gnu.org>
4742 %%
4743 %% Version 4.5.2 (28-Mar-2003)
4744 %%
4745 %% Hangul captions
4746 %%
4747 %% character set: KS X 1001:1992 (=KS C 5601-1992)
4748 %% encoding: EUC (=Wansung)

```

```

4749 %%
4750 %% preprocessed
4751 %%
4752 \mtcloadmlo{hangul2}
4753 \hangul2

```

## 8.57 “Hangul3” language: hangul3.mld and hangul3.mlo

The titles for the “hangul3” language (korean in hangul script, third variant) are taken from the file hangul2.cap of the CJK system [29]. Special fonts are needed, of course. See also sections 8.55 on the page before, 8.56 on the preceding page, 8.58, 8.59 on the following page, and 8.60 on page 255.

The titles for the “hangul3” language contain characters that cannot be easily generated, hence we load hangul3.mlo.

```

4754 (*hangul3)
4755 \ProvidesFile{hangul3.mld}[2005/01/28]
4756 %% From the file hangul2.cap of the CJK package
4757 %% for using Asian logographs (Chinese/Japanese/Korean) with LaTeX2e
4758 %% created by Werner Lemberg <wl@gnu.org>
4759 %%
4760 %% Version 4.5.2 (28-Mar-2003)
4761 %%
4762 %% Hangul captions set 2
4763 %%
4764 %% character set: KS X 1001:1992 (=KS C 5601-1992)
4765 %% encoding: EUC (=Wansung)
4766 %%
4767 \mtcloadmlo{hangul3}
4768 \hangul3

```

## 8.58 “Hangul4” language: hangul4.mld and hangul4.mlo

The titles for the “hangul4” language (korean in hangul script, fourth variant) are taken from the file hangul2.cpx of the CJK system [29]. Special fonts are needed, of course. See also sections 8.55 on the preceding page, 8.56 on the page before, 8.57, 8.59 on the next page, and 8.60 on page 255.

The titles for the “hangul4” language contain characters that cannot be easily generated, hence we load hangul4.mlo.

```

4769 (*hangul4)
4770 \ProvidesFile{hangul4.mld}[2005/01/28]
4771 %% From the file hangul2.cpx of the CJK package

```

```

4772%% for using Asian logographs (Chinese/Japanese/Korean) with LaTeX2e
4773%% created by Werner Lemberg <wl@gnu.org>
4774%%
4775%% Version 4.5.2 (28-Mar-2003)
4776%%
4777%% Hangul captions set 2
4778%%
4779%% character set: KS X 1001:1992 (=KS C 5601-1992)
4780%% encoding: EUC (=Wansung)
4781%%
4782%% preprocessed
4783%%
4784\mtcloadmlo{hangul4}
4785</hangul4>

```

## 8.59 “Hanja1” language: hanja1.mld and hanja1.mlo

The titles for the “hanja1” language (korean in the old script hanja, first variant) are taken from the file `hanja.cpx` of the CJK system [29]. Special fonts are needed, of course. See also sections 8.55 on page 252, 8.56 on page 252, 8.57 on the page before, 8.58 on the preceding page, and 8.60 on the next page.

The titles for the “hanja1” language contain characters that cannot be easily generated, hence we load `hanja1.mlo`.

```

4786<*hanja1>
4787\ProvidesFile{hanja1.mld}[2005/01/28]
4788%% From the file hanja.cpx of the CJK package
4789%% for using Asian logographs (Chinese/Japanese/Korean) with LaTeX2e
4790%% created by Werner Lemberg <wl@gnu.org>
4791%%
4792%% Version 4.5.2 (28-Mar-2003)
4793%%
4794%% Hanja captions
4795%%
4796%% character set: KS X 1001:1992 (=KS C 5601-1992)
4797%% encoding: EUC (=Wansung)
4798%%
4799%% preprocessed
4800%%
4801\mtcloadmlo{hanja1}
4802</hanja1>

```

## 8.60 “Hanja2” language: hanja2.mld and hanja2.mlo

The titles for the “hanja2” language (Korean in the old script hanja, second variant) are taken from the file hanja.cap of the CJK system [29]. Special fonts are needed, of course. See also sections 8.55 on page 252, 8.56 on page 252, 8.57 on page 253, 8.58 on page 253, and 8.59 on the preceding page.

The titles for the “hanja2” language contain characters that cannot be easily generated, hence we load hanja2.mlo.

```

4803 <(*hanja2)
4804 \ProvidesFile{hanja2.mld}[2005/01/28]
4805 %% From the file hanja.cap of the CJK package
4806 %% for using Asian logographs (Chinese/Japanese/Korean) with LaTeX2e
4807 %% created by Werner Lemberg <a7971428@unet.univie.ac.at>
4808 %%
4809 %% Version 4.1.3 (20-Jun-1997)
4810 %%
4811 %% Hanja captions
4812 %%
4813 %% character set: KS X 1001:1992 (=KS C 5601-1992)
4814 %% encoding: EUC (=Wansung)
4815 %%
4816 \mtcloadmlo{hanja2}
4817 </hanja2>

```

## 8.61 “Hebrew” language: hebrew.mld

The titles for the “hebrew” language are taken from the ArabTeX package [25, 26], which should be used, with the associated fonts.

```

4818 <(*hebrew)
4819 \ProvidesFile{hebrew.mld}[2001/02/28]
4820 %% Hebrew titles for minitoc.sty
4821 %% Need hebrew fonts (see arabtex documentation)
4822 \def\ptctitle{\tav\vav\kaf\finalnun\
4823             \ayin\nun\yod\nun\yod\finalmem}%
4824 \def\plftitle{\resh\shin\yod\mem\tav\
4825             \alef\yod\vav\resh\yod\finalmem}%
4826 \def\plttitle{\resh\shin\yod\mem\tav\
4827             \tet\bet\lamed\alef\vav\tav}%
4828 %%
4829 \def\mtctitle{\tav\vav\kaf\finalnun\
4830             \ayin\nun\yod\nun\yod\finalmem}%
4831 \def\mlftitle{\resh\shin\yod\mem\tav\
4832             \alef\yod\vav\resh\yod\finalmem}%
4833 \def\mlttitle{\resh\shin\yod\mem\tav\
4834             \tet\bet\lamed\alef\vav\tav}%

```

```

4835 %%
4836 \def\stctitle{\tav\vav\kaf\finalnun\
4837             \ayin\nun\yod\nun\yod\finalmem}%
4838 \def\slftitle{\resh\shin\yod\mem\tav\
4839             \alef\yod\vav\resh\yod\finalmem}%
4840 \def\sltttitle{\resh\shin\yod\mem\tav\
4841             \tet\bet\lamed\alef\vav\tav}%
4842 </hebrew>

```

## 8.62 “Hungarian” language: hungarian.mld

The “hungarian” language is a synonym of the “magyar” language, so we load magyar.mld. See section 8.78 on page 263.

```

4843 <{*hungarian}
4844 \ProvidesFile{hungarian.mld}[2004/12/14]
4845 \mtcselectlanguage{magyar}%
4846 </hungarian>

```

## 8.63 “Icelandic” language: icelandic.mld

The titles for the “icelandic” language are taken from the babel package [10]:

```

4847 <{*icelandic}
4848 \ProvidesFile{icelandic.mld}[2001/02/28]
4849 %% Icelandic titles for minitoc.sty
4850 %% need inputenc with 8-bits encoding
4851 \def\ptctitle{Efnisyfirlit}%
4852 \def\plftitle{Myndaskrá}%
4853 \def\pltttitle{Töfluskrá}%
4854 %%
4855 \def\mtctitle{Efnisyfirlit}%
4856 \def\mlftitle{Myndaskrá}%
4857 \def\mltttitle{Töfluskrá}%
4858 %%
4859 \def\stctitle{Efnisyfirlit}%
4860 \def\slftitle{Myndaskrá}%
4861 \def\sltttitle{Töfluskrá}%
4862 </icelandic>

```



## 8.64 “Interlingua” language: `interlingua.mld`

The titles for the “interlingua” language<sup>21</sup> are taken from the babel package [10]:

```

4863 <*interlingua>
4864 \ProvidesFile{interlingua.mld}[2001/07/04]
4865 \def\ptctitle{Contento}
4866 \def\plftitle{Lista de Figuras}
4867 \def\plttitle{Lista de Tabellas}
4868 %%
4869 \def\mtctitle{Contento}
4870 \def\mlftitle{Figuras}
4871 \def\mlttitle{Tabellas}
4872 %%
4873 \def\stctitle{Contento}
4874 \def\slftitle{Figuras}
4875 \def\slttitle{Tabellas}
4876 </interlingua>

```

## 8.65 “Irish” language: `irish.mld`

The titles for the “irish” language are taken from the babel package [10]:

```

4877 <*irish>
4878 \ProvidesFile{irish.mld}[1999/03/16]
4879 %% Irish titles for minitoc.sty
4880 \def\ptctitle{Cl\'ar\'Abhair}
4881 \def\plftitle{L\'ear\'aid\'{\i}}
4882 \def\plttitle{T\'abla\'{\i}}
4883 %%
4884 \def\mtctitle{Cl\'ar\'Abhair}
4885 \def\mlftitle{L\'ear\'aid\'{\i}}
4886 \def\mlttitle{T\'abla\'{\i}}
4887 %%
4888 \def\stctitle{Cl\'ar\'Abhair}
4889 \def\slftitle{L\'ear\'aid\'{\i}}
4890 \def\slttitle{T\'abla\'{\i}}
4891 </irish>

```

## 8.66 “Italian” language: `italian.mld`

The titles for the “italian” language are taken from the babel package [10]:

---

<sup>21</sup>A tentative for an universal language.

```

4892 <*italian>
4893 \ProvidesFile{italian.mld}[1999/03/16]
4894 %% Italian titles for minitoc.sty
4895 \def\ptctitle{Contenuto}%
4896 \def\plftitle{Elenco delle figure}%
4897 \def\pltttitle{Elenco delle tabelle}%
4898 %%
4899 \def\mtctitle{Contenuto}%
4900 \def\mlftitle{Elenco delle figure}%
4901 \def\mltttitle{Elenco delle tabelle}%
4902 %%
4903 \def\stctitle{Contenuto}%
4904 \def\slftitle{Elenco delle figure}%
4905 \def\slttitle{Elenco delle tabelle}%
4906 </italian>

```

## 8.67 “Japanese” language: `japanese.mld` and `japanese.mlo`

They are several variants for the japanese titles. The titles for a first variant of the “japanese” language have been found (by a Google search) on the Web site of Professor Toshiki KUMAZAWA<sup>22</sup>. See also sections 8.68, 8.69 on the next page, 8.70 on page 260, and 8.71 on page 260.

The titles for the “japanese” language contain characters that cannot be easily generated, hence we load `japanese.mlo`.

```

4907 <*japanese>
4908 \ProvidesFile{japanese.mld}[1999/03/16]
4909 %% Japanese titles for minitoc.sty
4910 %% Needs japanese fonts (CJK) and special input encoding.
4911 %% From Kumazawa Toshiki
4912 %% kumazawa@biwako.shiga-u.ac.jp
4913 %% http://www.biwako.shiga-u.ac.jp/sensei/kumazawa/tex/minitoc.html
4914 \mtcloadmlo{japanese}
4915 </japanese>

```

## 8.68 “Japanese2” language: `japanese2.mld` and `japanese2.mlo`

The titles for the “japanese2” language (japanese, second variant) are taken from file JIS.cap of the CJK system [29]. Special fonts are needed, of course. See also sections 8.67, 8.69 on the next page, 8.70 on page 260, and 8.71 on page 260.

---

<sup>22</sup><http://www.biwako.shiga-u.ac.jp/sensei/kumazawa/tex/minitoc.html>

The titles for the “japanese2” language contain characters that cannot be easily generated, hence we load `japanese2.mlo`.

```

4916 <*japanese2>
4917 \ProvidesFile{japanese2.mld}[2005/01/28]
4918 %% From the file JIS.cap of the CJK package
4919 %% for using Asian logographs (Chinese/Japanese/Korean) with LaTeX2e
4920 %% created by Werner Lemberg <wl@gnu.org>
4921 %%
4922 %% Version 4.5.2 (28-Mar-2003)
4923 %%
4924 %% Japanese captions
4925 %%
4926 %% character set: JIS X 0208:1997 (or JIS X 0208-1990)
4927 %% encoding: EUC
4928 %%
4929 \mtcloadmlo{japanese2}
4930 </japanese2>

```

## 8.69 “Japanese3” language: `japanese3.mld` and `japanese3.mlo`

The titles for the “japanese3” language (japanese, third variant) are taken from file `JIS.cpx` of the CJK system [29]. Special fonts are needed, of course. See also sections 8.67 on the preceding page, 8.68 on the page before, 8.70 on the following page, and 8.71 on the next page.

The titles for the “japanese3” language contain characters that cannot be easily generated, hence we load `japanese3.mlo`.

```

4931 <*japanese3>
4932 \ProvidesFile{japanese3.mld}[2005/01/28]
4933 %% From the file JIS.cpx of the CJK package
4934 %% for using Asian logographs (Chinese/Japanese/Korean) with LaTeX2e
4935 %% created by Werner Lemberg <wl@gnu.org>
4936 %%
4937 %% Version 4.5.2 (28-Mar-2003)
4938 %%
4939 %% Japanese captions
4940 %%
4941 %% character set: JIS X 0208:1997 (or JIS X 0208-1990)
4942 %% encoding: EUC
4943 %%
4944 %% preprocessed
4945 %%
4946 \mtcloadmlo{japanese3}
4947 </japanese3>

```

## 8.70 “Japanese4” language: `japanese4.mld` and `japanese4.mlo`

The titles for the “japanese4” language (japanese, fourth version) are taken from file `SJIS.cap` of the CJK system [29]. Special fonts are needed, of course. See also sections 8.67 on page 258, 8.68 on page 258, 8.69 on the preceding page, and 8.71.

The titles for the “japanese4” language contain characters that cannot be easily generated, hence we load `japanese4.mlo`.

```

4948 (*japanese4)
4949 \ProvidesFile{japanese4.mld}[2005/01/28]
4950 % From the file SJIS.cap of the CJK package
4951 % for using Asian logographs (Chinese/Japanese/Korean) with LaTeX2e
4952 % created by Werner Lemberg <wl@gnu.org>
4953 %
4954 % Version 4.5.2 (28-Mar-2003)
4955 %
4956 % Japanese captions
4957 %
4958 % character set: JIS X 0208:1997 (or JIS X 0208-1990)
4959 % encoding: SJIS
4960 %
4961 \mtcloadmlo{japanese4}
4962 \japanese4

```

## 8.71 “Japanese5” language: `japanese5.mld` and `japanese.mlo`

The titles for the “japanese5” (japanese, fifth variant) language are taken from file `SJIS.cpx` of the CJK system [29]. Special fonts are needed, of course. See also sections 8.67 on page 258, 8.68 on page 258, 8.69 on the preceding page, and 8.70.

The titles for the “japanese5” language contain characters that cannot be easily generated, hence we load `japanese5.mlo`.

```

4963 (*japanese5)
4964 \ProvidesFile{japanese5.mld}[[2005/01/28]
4965 %% From the file SJIS.cpx of the CJK package
4966 %% for using Asian logographs (Chinese/Japanese/Korean) with LaTeX2e
4967 %% created by Werner Lemberg <wl@gnu.org>
4968 %%
4969 %% Version 4.5.2 (28-Mar-2003)
4970 %%
4971 %% Japanese captions
4972 %%
4973 %% character set: JIS X 0208:1997 (or JIS X 0208-1990)
4974 %% encoding: SJIS
4975 %%
4976 %% preprocessed

```

```

4977 %%
4978 \mtcloadmlo{japanese5}
4979 </japanese5>

```

## 8.72 “Latin” language: latin.mld

The titles for the “latin” language are taken from the babel package [10]. See also section 8.73.

```

4980 <*\latin>
4981 \ProvidesFile{latin.mld}[2001/02/28]
4982 %% Latin titles for minitoc.sty
4983 \def\ptctitle{Index}%
4984 \def\plftitle{Conspectus descriptionum}%
4985 \def\plttitle{Conspectus tabularum}%
4986 %%
4987 \def\mtctitle{Index}%
4988 \def\mlftitle{Conspectus descriptionum}%
4989 \def\mlttitle{Conspectus tabularum}%
4990 %%
4991 \def\stctitle{Index}%
4992 \def\slftitle{Conspectus descriptionum}%
4993 \def\slttitle{Conspectus tabularum}%
4994 </latin>

```

## 8.73 “Latin2” language: latin2.mld

The titles for the “latin2” language (latin, abbreviated variant) are taken from the babel package [10], but abbreviated. See also section 8.72.

```

4995 <*\latin2>
4996 \ProvidesFile{latin2.mld}[2005/04/06]
4997 %% Latin titles (short) for minitoc.sty
4998 \def\ptctitle{Index}%
4999 \def\plftitle{Conspectus descriptionum}%
5000 \def\plttitle{Conspectus tabularum}%
5001 %%
5002 \def\mtctitle{Index}%
5003 \def\mlftitle{Descriptiones}%
5004 \def\mlttitle{Tabulae}%
5005 %%
5006 \def\stctitle{Index}%
5007 \def\slftitle{Descriptiones}%
5008 \def\slttitle{Tabulae}%
5009 </latin2>

```

## 8.74 “Latvian” language: `latvian.mld`

The titles for the “latvian” language<sup>23</sup> are taken from the ANTOMEGA project [24]. See also section 8.75:

```

5010 <*latvian>
5011 \ProvidesFile{latvian.mld}[2005/02/08]
5012 %% from latvian.ldf (Antomega project)
5013 %% Needs Omega
5014 %% Alexej M. Kryokov
5015 %% Dmitry Ivanov
5016 %%
5017 \def\ptctitle{\locallatvian{Saturdays}}%
5018 \def\plftitle{\locallatvian{Attētie 113. saraksts}}%
5019 \def\plttitle{\locallatvian{Tabulu saraksts}}%
5020 %%
5021 \def\mtctitle{\locallatvian{Saturdays}}%
5022 \def\mlftitle{\locallatvian{Attētie 113. saraksts}}%
5023 \def\mlttitle{\locallatvian{Tabulu saraksts}}%
5024 %%
5025 \def\stctitle{\locallatvian{Saturdays}}%
5026 \def\slftitle{\locallatvian{Attētie 113. saraksts}}%
5027 \def\slttitle{\locallatvian{Tabulu saraksts}}%
5028 </latvian>

```

## 8.75 “Letton” language: `letton.mld`

The “letton” language is a synonym for the “latvian” language, so we just load `latvian.mld`. See section 8.74.

```

5029 <*letton>
5030 \ProvidesFile{letton.mld}[2005/02/08]
5031 \mtcselectlanguage{latvian}%
5032 </letton>

```

## 8.76 “Lithuanian” language: `lithuanian.mld`

The titles for the “lithuanian” language are taken from the babel package [10]:

```

5033 <*lithuanian>
5034 \ProvidesFile{lithuanian.mld}[1999/07/29]
5035 %% Lithuanian titles for minitoc.sty
5036 \def\ptctitle{Turinys}

```

---

<sup>23</sup>“Latvian” is the original name for “letton”.

```

5037 \def\plftitle{Paveiksl\protect\c u s\protect\c ara\protect\v sas}
5038 \def\plttitle{Lentel\protect\es}
5039 %%
5040 \def\mtctitle{Turinys}
5041 \def\mlftitle{Paveiksl\protect\c u s\protect\c ara\protect\v sas}
5042 \def\mlttitle{Lentel\protect\es}
5043 %%
5044 \def\stctitle{Turinys}
5045 \def\slftitle{Paveiksl\protect\c u s\protect\c ara\protect\v sas}
5046 \def\slttitle{Lentel\protect\es}
5047 </lithuanian>

```

## 8.77 “Lsorbian” language: lsorbian.mld

The titles for the “Lsorbian” language<sup>24</sup> are taken from the babel package [10]. See also section 8.114 on page 282.

```

5048 <*lsorbian>
5049 \ProvidesFile{lsorbian.mld}[1999/03/16]
5050 %% Lower sorbian titles for minitoc.sty
5051 \def\ptctitle{Wop\'simje\'se}
5052 \def\plftitle{Zapismobrazow}
5053 \def\plttitle{Zapistabulkow}
5054 %%
5055 \def\mtctitle{Wop\'simje\'se}
5056 \def\mlftitle{Zapismobrazow}
5057 \def\mlttitle{Zapistabulkow}
5058 %%
5059 \def\stctitle{Wop\'simje\'se}
5060 \def\slftitle{Zapismobrazow}
5061 \def\slttitle{Zapistabulkow}
5062 </lsorbian>

```

## 8.78 “Magyar” language: magyar.mld

The titles for the “magyar” language are taken from the babel package [10]. A synonym of “magyar” is “hungarian” (see section 8.62 on page 256). See also section 8.79 on the following page.

```

5063 <*magyar>
5064 \ProvidesFile{magyar.mld}[1999/03/16]
5065 % Magyar titles for minitoc.sty

```

---

<sup>24</sup>Lower sorbian. Sorbian, or wendisch, is a member of the west slavic subgroup of indo-european languages spoken in Lower Lusatia in the german *länder* of Saxony and Brandenburg. The Sorbs are descendents of the Wends, the german name for the slavic tribes who occupied the area between the Elbe and Saale rivers in the west and the Odra (Oder) river in the east during the medieval period (VIth century).

```

5066 \def\ptctitle{Tratalom}%
5067 \def\plftitle{\`Abr\`ak}%
5068 \def\plttitle{T\`abl\`azatok}%
5069 %%
5070 \def\mtctitle{Tratalom}%
5071 \def\mlftitle{\`Abr\`ak}%
5072 \def\mlttitle{T\`abl\`azatok}%
5073 %%
5074 \def\stctitle{Tratalom}%
5075 \def\slftitle{\`Abr\`ak}%
5076 \def\slttitle{T\`abl\`azatok}%
5077 </magyar>

```

## 8.79 “Magyar2” language: magyar2.mld

The titles for the “magyar2” language are taken from a variant proposed in the babel package [10]. See also section 8.78 on the page before.

```

5078 <*magyar2>
5079 \ProvidesFile{magyar.mld}[2005/04/06]
5080 %% Magyar titles for minitoc.sty (variant)
5081 \def\ptctitle{Tratalom}%
5082 \def\plftitle{\`Abr\`ak list\`aja}%
5083 \def\plttitle{T\`abl\`azatok list\`aja}%
5084 %%
5085 \def\mtctitle{Tratalom}%
5086 \def\mlftitle{\`Abr\`ak list\`aja}%
5087 \def\mlttitle{T\`abl\`azatok list\`aja}%
5088 %%
5089 \def\stctitle{Tratalom}%
5090 \def\slftitle{\`Abr\`ak list\`aja}%
5091 \def\slttitle{T\`abl\`azatok list\`aja}%
5092 </magyar2>

```

## 8.80 “Malayalam-keli” language: malayalam-keli.mld

The titles for the “malayalam” language<sup>25</sup>, with the “Keli” fonts, are taken from the malayalam package [1]. This language requires specific fonts. See also sections 8.81 on the following page and 8.82 on the next page.

```

5093 <*malayalam – keli>
5094 \ProvidesFile{malayalam-keli.mld}[2005/06/07]
5095 %

```

<sup>25</sup>The Malayalam language is spoken from the western coast of Malabar to the extreme southern India, mainly in the Kerala state. It is one of the dravidian languages strongly bound to the Tamil language. The alphabet and the script are dated from the 8th or 9th centuries.



```

5096 % Malayalam: Keli fonts
5097 %
5098 \def\ptctitle{\mm \X{\<68>}\X{\<197>}\X{\<83>}\X{\<161>}\<119>}%
5099 \def\plftitle{\mm \X{\<78>}\<110>}\X{\<123>}\<88>}\X{\<167>}\X{\<196>}}%
5100 \def\pltttitle{\mm \X{\<116>}\<83>}\X{\<95>}\<110>}\X{\<102>}\<112>}\X{\<73>}\X{\<196>}}%
5101 %
5102 \def\mtctitle{\mm \X{\<68>}\X{\<197>}\X{\<83>}\X{\<161>}\<119>}%
5103 \def\mlftitle{\mm \X{\<78>}\<110>}\X{\<123>}\<88>}\X{\<167>}\X{\<196>}}%
5104 \def\mltttitle{\mm \X{\<116>}\<83>}\X{\<95>}\<110>}\X{\<102>}\<112>}\X{\<73>}\X{\<196>}}%
5105 %
5106 \def\stctitle{\mm \X{\<68>}\X{\<197>}\X{\<83>}\X{\<161>}\<119>}%
5107 \def\slftitle{\mm \X{\<78>}\<110>}\X{\<123>}\<88>}\X{\<167>}\X{\<196>}}%
5108 \def\sltttitle{\mm \X{\<116>}\<83>}\X{\<95>}\<110>}\X{\<102>}\<112>}\X{\<73>}\X{\<196>}}%
5109 </malayalam – keli>

```

## 8.81 “Malayalam-rachana” language: malayalam-rachana.mld

The titles for the “malayalam” language, with the traditionnal “Rachana” fonts (old lipi), are taken from the malayalam package [1]. This language requires specific fonts. See also sections 8.80 on the preceding page and 8.82.

```

5110 (*malayalam – rachana)
5111 \ProvidesFile{malayalam-rachana.mld}[2005/06/07]
5112 %
5113 % Malayalam: Rachana fonts, traditionnal.
5114 %
5115 \def\ptctitle{\mm \X{\<68>}\X{\<201>}\X{\<83>}\X{\<183>}\<119>}%
5116 \def\plftitle{\mm \X{\<78>}\<111>}\X{\<C\<94>}}\X{\<186>}\X{\<179>}}%
5117 \def\pltttitle{\mm \X{\<117>}\<83>}\X{\<95>}\<111>}\X{\<F\<59>}}\X{\<73>}\X{\<179>}}%
5118 %
5119 \def\mtctitle{\mm \X{\<68>}\X{\<201>}\X{\<83>}\X{\<183>}\<119>}%
5120 \def\mlftitle{\mm \X{\<78>}\<111>}\X{\<C\<94>}}\X{\<186>}\X{\<179>}}%
5121 \def\mltttitle{\mm \X{\<117>}\<83>}\X{\<95>}\<111>}\X{\<F\<59>}}\X{\<73>}\X{\<179>}}%
5122 %
5123 \def\stctitle{\mm \X{\<68>}\X{\<201>}\X{\<83>}\X{\<183>}\<119>}%
5124 \def\slftitle{\mm \X{\<78>}\<111>}\X{\<C\<94>}}\X{\<186>}\X{\<179>}}%
5125 \def\sltttitle{\mm \X{\<117>}\<83>}\X{\<95>}\<111>}\X{\<F\<59>}}\X{\<73>}\X{\<179>}}%
5126 </malayalam – rachana>

```

## 8.82 “Malayalam-rachana2” language: malayalam-rachana2.mld

The titles for the “malayalam” language, with the reformed “Rachana” fonts (new lipi), are taken from the malayalam package [1]. This language requires specific fonts. See also sections 8.80 on the preceding page and 8.81.

```

5127 (*malayalam – rachana2)

```

```

5128 \ProvidesFile{malayalam-rachana2.mld}[2005/06/07]
5129 %
5130 % Malayalam: Rachana fonts, reformed.
5131 %
5132 \def\ptctitle{\mm \X{\<68>}\X{\<201>}\X{\<83>}\X{\<183>}\<119>}
5133 \def\plftitle{\mm \X{\<78>\<111>}\X{\<125>\<88>}\X{\<186>}\X{\<179>}}
5134 \def\pltttitle{\mm \X{\<117>\<83>}\X{\<95>\<111>}\X{\<106>\<113>}\X{\<73>}\X{\<179>}}
5135 %
5136 \def\mtctitle{\mm \X{\<68>}\X{\<201>}\X{\<83>}\X{\<183>}\<119>}
5137 \def\mlftitle{\mm \X{\<78>\<111>}\X{\<125>\<88>}\X{\<186>}\X{\<179>}}
5138 \def\mltttitle{\mm \X{\<117>\<83>}\X{\<95>\<111>}\X{\<106>\<113>}\X{\<73>}\X{\<179>}}
5139 %
5140 \def\stctitle{\mm \X{\<68>}\X{\<201>}\X{\<83>}\X{\<183>}\<119>}
5141 \def\slftitle{\mm \X{\<78>\<111>}\X{\<125>\<88>}\X{\<186>}\X{\<179>}}
5142 \def\slttitle{\mm \X{\<117>\<83>}\X{\<95>\<111>}\X{\<106>\<113>}\X{\<73>}\X{\<179>}}
5143 \</malayalam – rachana2>

```

### 8.83 “Mongol” language: mongol.mld

The titles for the “mongol” language are taken from the MonTeX package [14, 15]. This language requires specific fonts. See also sections 8.13 on page 232 and 8.20 on page 235.

```

5144 \<*mongol>
5145 \ProvidesFile{mongol.mld}[1999/03/16]
5146 %% Mongol (xalx) titles for minitoc.sty
5147 %% Needs mongol fonts
5148 \def\ptctitle{{\mnr Garqig}}
5149 \def\plftitle{{\mnr Zurgin in jagsaalt}}
5150 \def\pltttitle{{\mnr X"usn"agti in jagsaalt}}
5151 %%
5152 \def\mtctitle{{\mnr Garqig}}
5153 \def\mlftitle{{\mnr Zurgin in jagsaalt}}
5154 \def\mltttitle{{\mnr X"usn"agti in jagsaalt}}
5155 %%
5156 \def\stctitle{{\mnr Garqig}}
5157 \def\slftitle{{\mnr Zurgin in jagsaalt}}
5158 \def\slttitle{{\mnr X"usn"agti in jagsaalt}}
5159 \</mongol>

```

### 8.84 “Naustrian” language: naustrian.mld

The “naustrian” language is a synonym of the “ngermanb” language (a revised version of the germanb variant of the german language), so we load ngermanb.mld. See also section 8.86 on the following page.

```

5160 \<*naustrian>

```

```

5161 \ProvidesFile{naustrian.mld}[2004/12/14]
5162 \mtcselectlanguage{ngermanb}%
5163 </naustrian>

```

## 8.85 “Ngerman” language: ngerman.mld

The “ngerman” language is a synonym of the “ngermanb” language<sup>26</sup>, so we load `ngermanb.mld`. See also section 8.86.

```

5164 <*ngerman>
5165 \ProvidesFile{ngerman.mld}[2004/12/14]
5166 \mtcselectlanguage{ngermanb}%
5167 </ngerman>

```

## 8.86 “Ngermanb” language: ngermanb.mld

The titles for the “ngermanb” language<sup>27</sup> are taken from the `babel` package [10]. See also sections 8.84 on the preceding page, and 8.85.

```

5168 <*ngermanb>
5169 \ProvidesFile{ngermanb.mld}[1999/12/06]
5170 %% New german (B) titles for minitoc.sty
5171 \def\ptctitle{Inhaltsverzeichnis}%    % oder nur: Inhalt
5172 \def\plftitle{Abbildungsverzeichnis}%
5173 \def\plttitle{Tabellenverzeichnis}%
5174 %%
5175 \def\mtctitle{Inhalt}%
5176 \def\mlftitle{Abbildungsverzeichnis}%
5177 \def\mlttitle{Tabellenverzeichnis}%
5178 %%
5179 \def\stctitle{Inhalt}%
5180 \def\slftitle{Abbildungsverzeichnis}%
5181 \def\slttitle{Tabellenverzeichnis}%
5182 </ngermanb>

```

---

<sup>26</sup>A revised version of the `germanb` variant of the german language.

<sup>27</sup>A variant of the german language, with revised spelling.

## 8.87 “Norsk” language: norsk.mld

The titles for the “norsk” language (or bokmål, language of the kingdom) are taken from the babel package [10], with help from Dag LANGMYHR. See also section 8.88.

```

5183 <*norsk>
5184 \ProvidesFile{norsk.mld}[1999/03/16]
5185 %% Norsk titles for minitoc.sty
5186 %% Thanks to Dag Langmyhr (dag@ifi.uio.no)
5187 \def\ptctitle{Innhold}%
5188 \def\plftitle{Figurer}% or Figurliste
5189 \def\pltttitle{Tabeller}% or Tabelliste
5190 %%
5191 \def\mtctitle{Innhold}%
5192 \def\mlftitle{Figurer}% or Figurliste
5193 \def\mltttitle{Tabeller}% or Tabelliste
5194 %%
5195 \def\stctitle{Innhold}%
5196 \def\slftitle{Figurer}% or Figurliste
5197 \def\sltttitle{Tabeller}% or Tabelliste
5198 </norsk>

```

## 8.88 “Nynorsk” language: nynorsk.mld

The titles for the “nynorsk” language<sup>28</sup> are taken from the babel package [10], with help from Dag LANGMYHR. See also section 8.87.

```

5199 <*nynorsk>
5200 \ProvidesFile{nynorsk.mld}[1999/03/16]
5201 %% Nynorsk titles for minitoc.sty
5202 %% Thanks to Dag Langmyhr (dag@ifi.uio.no)
5203 \def\mtctitle{Innhald}%
5204 \def\mlftitle{Figurar}% or Figurliste
5205 \def\mltttitle{Tabellar}% or Tabelliste
5206 %%
5207 \def\ptctitle{Innhald}%
5208 \def\plftitle{Figurar}% or Figurliste
5209 \def\pltttitle{Tabellar}% or Tabelliste
5210 %%
5211 \def\stctitle{Innhald}%
5212 \def\slftitle{Figurar}% or Figurliste
5213 \def\sltttitle{Tabellar}% or Tabelliste
5214 </nynorsk>

```

---

<sup>28</sup>Created around 1800 by Ivar Åssen to make a real independant and national norwegian language, in reaction to danish, from the various dialect spoken in the country.

## 8.89 “Polish” language: polish.mld

The titles for the “polish” language are taken from the babel package [10]. See also section 8.90.

```

5215 <*polish>
5216 \ProvidesFile{polish.mld}[1999/03/16]
5217 %% Polish titles for minitoc.sty
5218 \def\ptctitle{Spis rzeczy}%
5219 \def\plftitle{Spis rysunk\'ow}%
5220 \def\pltttitle{Spis tablic}%
5221 %%
5222 \def\mtctitle{Spis rzeczy}%
5223 \def\mlftitle{Spis rysunk\'ow}%
5224 \def\mltttitle{Spis tablic}%
5225 %%
5226 \def\stctitle{Spis rzeczy}%
5227 \def\slftitle{Spis rysunk\'ow}%
5228 \def\slttitle{Spis tablic}%
5229 </polish>

```

## 8.90 “Polish2” language: polish2.mld

The titles for the “polish2” language<sup>29</sup> are taken from the ANTOMEGA project [24]. See also section 8.89.

```

5230 <*polish2>
5231 \ProvidesFile{polish2.mld}[2005/02/08]
5232 %% from omega-polish.ldf (Antomega project)
5233 %% Needs Omega
5234 %% Alexej M. Kryokov
5235 %% Dmitry Ivanov
5236 %%
5237 \def\ptctitle{\localpolish{Spis tre^^^^00b1ci}}%
5238 \def\plftitle{\localpolish{Spis rysunk^^^^00adw}}%
5239 \def\pltttitle{\localpolish{Spis tablic}}%
5240 %%
5241 \def\mtctitle{\localpolish{Spis tre^^^^00b1ci}}%
5242 \def\mlftitle{\localpolish{Spis rysunk^^^^00adw}}%
5243 \def\mltttitle{\localpolish{Spis tablic}}%
5244 %%
5245 \def\stctitle{\localpolish{Spis tre^^^^00b1ci}}%
5246 \def\slftitle{\localpolish{Spis rysunk^^^^00adw}}%
5247 \def\slttitle{\localpolish{Spis tablic}}%
5248 </polish2>

```

---

<sup>29</sup>“Polish2” is a variant of “polish”.

### 8.91 “Portuges” language: portuges.mld

This is another spelling for “portuguese” (see section 8.92), so we just load portuges.mld:

```
5249 <*portuges>
5250 \ProvidesFile{portuges.mld}[2005/06/06]
5251 \mtcselectlanguage{portuguese}
5252 </portuges>
```

### 8.92 “Portuguese” language: portuguese.mld

The titles for the “portuguese” language are taken from the babel package [10]. See also section 8.14 on page 232, because the titles are different in Brazil, even if the language is also portuguese.

```
5253 <*portuguese>
5254 \ProvidesFile{portuguese.mld}[2005/07/08]
5255 %% Portuguese titles for minitoc.sty
5256 \def\ptctitle{Conte\'udo}%
5257 \def\plftitle{Lista de Figuras}%
5258 \def\plttitle{Lista de Tabelas}%
5259 %%
5260 \def\mtctitle{Conte\'udo}%
5261 \def\mlftitle{Lista de Figuras}%
5262 \def\mlttitle{Lista de Tabelas}%
5263 %%
5264 \def\stctitle{Conte\'udo}%
5265 \def\slftitle{Lista de Figuras}%
5266 \def\slttitle{Lista de Tabelas}%
5267 </portuguese>
```

### 8.93 “Romanian” language: romanian.mld

The titles for the “romanian” language are taken from the babel package [10].

```
5268 <*romanian>
5269 \ProvidesFile{romanian.mld}[1999/03/16]
5270 %% Romanian titles for minitoc.sty
5271 \def\ptctitle{Cuprins}%
5272 \def\plftitle{List\u{a} de figuri}%
5273 \def\plttitle{List\u{a} de tabele}%
5274 %%
5275 \def\mtctitle{Cuprins}%
5276 \def\mlftitle{List\u{a} de figuri}%
5277 \def\mlttitle{List\u{a} de tabele}%
5278 </romanian>
```

```

5276 \def\mlfttitle{List\u{a} de figuri}%
5277 \def\mltttitle{List\u{a} de tabele}%
5278 %%
5279 \def\stctitle{Cuprins}%
5280 \def\slfttitle{List\u{a} de figuri}%
5281 \def\slttitle{List\u{a} de tabele}%
5282 \</romanian>

```

## 8.94 “Russian” language: russian.mld

The titles for the “russian” language are taken from the babel package [10]. Specific cyrillic fonts are required.

```

5283 \<*russian>
5284 \ProvidesFile{russian.mld}[1999/03/16]
5285 %% Russian titles for minitoc.sty
5286 \def\ptctitle{Oglavlenie}%
5287 \def\plfttitle{Pere{\cz}en{\mz} risunkov}%
5288 \def\pltttitle{Pere{\cz}en{\mz} tablic}%
5289 %%
5290 \def\mtctitle{Oglavlenie}%
5291 \def\mlfttitle{Pere{\cz}en{\mz} risunkov}%
5292 \def\mltttitle{Pere{\cz}en{\mz} tablic}%
5293 %%
5294 \def\stctitle{Oglavlenie}%
5295 \def\slfttitle{Pere{\cz}en{\mz} risunkov}%
5296 \def\slttitle{Pere{\cz}en{\mz} tablic}%
5297 \</russian>

```

## 8.95 “Russian2m” language: russian2m.mld

The titles for the “russian2m” language (“russian2m” is a modern variant of “russian”) are taken from the ANTOMEGA project [24]. Specific cyrillic fonts are required. See also section 8.94.

```

5298 \<*russian2m>
5299 \ProvidesFile{russian2m.mld}[2005/02/08]
5300 %% from russian2m.ldf (Antomega project, russian modern)
5301 %% Needs Omega
5302 %% Alexej M. Kryokov
5303 %% Dmitry Ivanov
5304 %%
5305 %% Needs cyrillic fonts
5306 %%
5307 \def\ptctitle{\localrussian%
5308 {^^^^041e^^^^0433^^^^043b^^^^0430^^^^0432^^^^043b^^^^0435^^^^043d%

```

```

5309 ^^^^0438^^^^0435}}%
5310 \def\plftitle{\localrussian%
5311 {^^^^0421^^^^043f^^^^0438^^^^0441^^^^043e^^^^043a  ^^^^0438^^^^043b%
5312  ^^^^043b^^^^044e^^^^0441^^^^0442^^^^0440^^^^0430^^^^0446^^^^0438%
5313  ^^^^0439}}}%
5314 \def\pltttitle{\localrussian%
5315 {^^^^0421^^^^043f^^^^0438^^^^0441^^^^043e^^^^043a  ^^^^0442^^^^0430%
5316  ^^^^0431^^^^043b^^^^0438^^^^0446}}}%
5317 %%
5318 \def\mtctitle{\localrussian%
5319 {^^^^041e^^^^0433^^^^043b^^^^0430^^^^0432^^^^043b^^^^0435^^^^043d%
5320  ^^^^0438^^^^0435}}}%
5321 \def\mlftitle{\localrussian%
5322 {^^^^0421^^^^043f^^^^0438^^^^0441^^^^043e^^^^043a  ^^^^0438^^^^043b%
5323  ^^^^043b^^^^044e^^^^0441^^^^0442^^^^0440^^^^0430^^^^0446^^^^0438%
5324  ^^^^0439}}}%
5325 \def\mltttitle{\localrussian%
5326 {^^^^0421^^^^043f^^^^0438^^^^0441^^^^043e^^^^043a  ^^^^0442^^^^0430%
5327  ^^^^0431^^^^043b^^^^0438^^^^0446}}}%
5328 %%
5329 \def\stctitle{\localrussian%
5330 {^^^^041e^^^^0433^^^^043b^^^^0430^^^^0432^^^^043b^^^^0435^^^^043d%
5331  ^^^^0438^^^^0435}}}%
5332 \def\slftitle{\localrussian%
5333 {^^^^0421^^^^043f^^^^0438^^^^0441^^^^043e^^^^043a  ^^^^0438^^^^043b%
5334  ^^^^043b^^^^044e^^^^0441^^^^0442^^^^0440^^^^0430^^^^0446^^^^0438%
5335  ^^^^0439}}}%
5336 \def\sltttitle{\localrussian%
5337 {^^^^0421^^^^043f^^^^0438^^^^0441^^^^043e^^^^043a  ^^^^0442^^^^0430%
5338  ^^^^0431^^^^043b^^^^0438^^^^0446}}}%
5339 </russian2m>

```

## 8.96 “Russian2o” language: russian2o.mld

The titles for the “russian2o” language (“russian2o” is an old variant of “russian”) are taken from the ANTOMEGA project [24]. Specific cyrillic fonts are required. See also section 8.94 on the page before.

```

5340 <*russian2o>
5341 \ProvidesFile{russian2o.mld}[2005/02/08]
5342 %% from russian2o.mld (Antomega project - russian old)
5343 %% Needs Omega
5344 %% Alexej M. Kryokov
5345 %% Dmitry Ivanov
5346 %%
5347 \def\ptctitle{\localrussian%
5348 {^^^^041e^^^^0433^^^^043b^^^^0430^^^^0432^^^^043b^^^^0435^^^^043d%
5349  ^^^^0456^^^^0435}}}%
5350 \def\plftitle{\localrussian%
5351 {^^^^0421^^^^043f^^^^0438^^^^0441^^^^043e^^^^043a^^^^044a  ^^^^0438%

```



```

5352 ^^^^043b^^^043b^^^044e^^^0441^^^0442^^^0440^^^0430^^^0446%
5353 ^^^^0456^^^0439}}%
5354 \def\plttitle{\localrussian%
5355 {^^^^0421^^^043f^^^0438^^^0441^^^043e^^^043a^^^044a ^^^0442%
5356 ^^^0430^^^0431^^^043b^^^0438^^^0446^^^044a}}%
5357 %%
5358 \def\mtctitle{\localrussian%
5359 {^^^^041e^^^0433^^^043b^^^0430^^^0432^^^043b^^^0435^^^043d%
5360 ^^^0456^^^0435}}%
5361 \def\mlftitle{\localrussian%
5362 {^^^^0421^^^043f^^^0438^^^0441^^^043e^^^043a^^^044a ^^^0438%
5363 ^^^043b^^^043b^^^044e^^^0441^^^0442^^^0440^^^0430^^^0446%
5364 ^^^0456^^^0439}}%
5365 \def\mltttitle{\localrussian%
5366 {^^^^0421^^^043f^^^0438^^^0441^^^043e^^^043a^^^044a ^^^0442%
5367 ^^^0430^^^0431^^^043b^^^0438^^^0446^^^044a}}%
5368 %
5369 \def\stctitle{\localrussian%
5370 {^^^^041e^^^0433^^^043b^^^0430^^^0432^^^043b^^^0435^^^043d%
5371 ^^^0456^^^0435}}%
5372 \def\slftitle{\localrussian%
5373 {^^^^0421^^^043f^^^0438^^^0441^^^043e^^^043a^^^044a ^^^0438%
5374 ^^^043b^^^043b^^^044e^^^0441^^^0442^^^0440^^^0430^^^0446%
5375 ^^^0456^^^0439}}%
5376 \def\sltttitle{\localrussian%
5377 {^^^^0421^^^043f^^^0438^^^0441^^^043e^^^043a^^^044a ^^^0442%
5378 ^^^0430^^^0431^^^043b^^^0438^^^0446^^^044a}}%
5379 </russian2o>

```

## 8.97 “Russianb” language: russianb.mld

The titles for the “russianb” language (“russianb” is a variant of “russian”) are taken from the `babel` package [10]. Specific cyrillic fonts are required. See also section 8.94 on page 271.

```

5380 (*russianb)
5381 \ProvidesFile{russianb.mld}[1999/03/16]
5382 %% russianb.mld
5383 \def\ptctitle{%
5384 {\cyr \CYRO\CYRg\CYRl\CYRa\CYRv\CYRl\CYRe\CYRn\CYRi\CYRe}}%
5385 \def\plftitle{%
5386 {\cyr \CYRS\CYRp\CYRi\CYRs\CYRo\CYRk\space
5387 \CYRi\CYRl\CYRl\CYRyu\CYRs\CYRt\CYRr\CYRa\CYRc\CYRi\CYRishrt}}%
5388 \def\plttitle{%
5389 \CYRS\CYRp\CYRi\CYRs\CYRo\CYRk\space
5390 \CYRt\CYRa\CYRb\CYRl\CYRi\CYRc}%
5391 }
5392 %%
5393 \def\mtctitle{%
5394 {\cyr \CYRO\CYRg\CYRl\CYRa\CYRv\CYRl\CYRe\CYRn\CYRi\CYRe}}%

```

```

5395 \def\mlftitle{%
5396   {\cyr \CYRS\CYRp\CYRi\CYRs\CYRo\CYRk\space
5397     \CYRi\CYRL\CYRL\CYRyu\CYRs\CYRt\CYRr\CYRa\CYRc\CYRi\CYRishrt}}%
5398 \def\mltttitle{%
5399   \CYRS\CYRp\CYRi\CYRs\CYRo\CYRk\space
5400   \CYRt\CYRa\CYRb\CYRL\CYRi\CYRc}%
5401 }
5402 %%
5403 \def\stctitle{%
5404   {\cyr \CYRO\CYRg\CYRL\CYRa\CYRv\CYRL\CYRe\CYRn\CYRi\CYRe}}%
5405 \def\slftitle{%
5406   {\cyr \CYRS\CYRp\CYRi\CYRs\CYRo\CYRk\space
5407     \CYRi\CYRL\CYRL\CYRyu\CYRs\CYRt\CYRr\CYRa\CYRc\CYRi\CYRishrt}}%
5408 \def\slttitle{%
5409   \CYRS\CYRp\CYRi\CYRs\CYRo\CYRk\space
5410   \CYRt\CYRa\CYRb\CYRL\CYRi\CYRc}%
5411 }
5412 </russianb>

```

## 8.98 “Russianc” language: russianc.mld

The titles for the “russianc” language (“russianc” is a variant of “russian”, used in the part of Mongolia under russian influence) are taken from the file `russian.def` in the MonTeX package [14, 15]. Specific cyrillic fonts are required. See also section 8.94 on page 271.

```

5413 <(*russianc)
5414 \ProvidesFile{russianc.mld}[1999/03/16]
5415 %% Needs cyrillic fonts
5416 %% Russian titles for minitoc.sty
5417 %% Needs cyrillic fonts
5418 \def\ptctitle{\xalx{Oglawlenie}}
5419 \def\plftitle{\xalx{Spisok risunkow}}
5420 \def\pltttitle{\xalx{Spisok tablic}}
5421 %%
5422 \def\mtctitle{\xalx{Soderjanie}}
5423 \def\mlftitle{\xalx{Spisok risunkow}}
5424 \def\mltttitle{\xalx{Spisok tablic}}
5425 %%
5426 \def\stctitle{\xalx{Soderjanie}}
5427 \def\slftitle{\xalx{Spisok risunkow}}
5428 \def\slttitle{\xalx{Spisok tablic}}
5429 </russianc>

```

## 8.99 “Samin” language: `samin.mld`

The titles for the “samin” language<sup>30</sup> are taken from the `samin.dtx` file of the `babel` package [10]. Specific fonts are required.

```

5430 <*samin>
5431 \ProvidesFile{samin.mld}[2001/02/28]
5432 %% Samin titles for minitoc.sty
5433 \def\ptctitle{Sisdoallu}%
5434 \def\plftitle{Govvosat}%
5435 \def\plttitle{Tabeallat}%
5436 %%
5437 \def\mtctitle{Sisdoallu}%
5438 \def\mlftitle{Govvosat}%
5439 \def\mlttitle{Tabeallat}%
5440 %%
5441 \def\stctitle{Sisdoallu}%
5442 \def\slftitle{Govvosat}%
5443 \def\slttitle{Tabeallat}%
5444 </samin>

```

## 8.100 “Scottish” language: `scottish.mld`

The titles for the “scottish” language are taken from the `babel` language [10]:

```

5445 <*scottish>
5446 \ProvidesFile{scottish.mld}[1999/03/16]
5447 %%Scottish titles for minitoc.sty
5448 \def\ptctitle{Cl\‘ar-obrach}
5449 \def\plftitle{LiostaDhealbh}
5450 \def\plttitle{LiostaChl\‘ar}
5451 %%
5452 \def\mtctitle{Cl\‘ar-obrach}
5453 \def\mlftitle{LiostaDhealbh}
5454 \def\mlttitle{LiostaChl\‘ar}
5455 %%
5456 \def\stctitle{Cl\‘ar-obrach}
5457 \def\slftitle{LiostaDhealbh}
5458 \def\slttitle{LiostaChl\‘ar}
5459 </scottish>

```

---

<sup>30</sup>Several Sami dialects/languages are spoken in Finland, Norway, Sweden, and on the Kola Peninsula (Russia). The alphabets differ, so there will eventually be a need for more `.dtx` files for e.g. Lule and South Sami. Hence the name `samin.dtx` (and not `sami.dtx` or the like) in the North Sami case. This note is copied from the `samin.dtx` file.

### 8.101 “Serbian” language: `serbian.mld`

The titles for the “serbian” language are taken from the `babel` package [10]. See also section 8.102.

```

5460 <*serbian>
5461 \ProvidesFile{serbian.mld}[1999/12/06]
5462 %%Serbian titles for minitoc.sty
5463 \def\ptctitle{Sadr\v{z}aj}%
5464 \def\plftitle{Slike}%
5465 \def\plttitle{Tabele}%
5466 %%
5467 \def\mtctitle{Sadr\v{z}aj}%
5468 \def\mlftitle{Slike}%
5469 \def\mlttitle{Tabele}%
5470 %%
5471 \def\stctitle{Sadr\v{z}aj}%
5472 \def\slftitle{Slike}%
5473 \def\slttitle{Tabele}%
5474 </serbian>

```

### 8.102 “Serbianc” language: `serbianc.mld`

The titles for the “serbianc” language<sup>31</sup> have been given by Marko ĆEHAJA and Frank KÜSTER. Cyrillic fonts are required. See also section 8.101.

```

5475 <*serbianc>
5476 \ProvidesFile{serbianc.mld}[2001/11/25]
5477 %% Provides titles for minitoc.sty in Serbian Cyrillic
5478 %%
5479 %%Marko Ćehaja Internut@Thetaworld.Org
5480 %%Frank Küster, Biozentrum der Univ. Basel, frank@kuesterei.ch
5481 %%Abt. Biophysikalische Chemie
5482 \def\ptctitle{%
5483   {\cyr\CYRS\cyra\cyrd\cyrr\cyrrh\cyra\cyrje}}%
5484 \def\plftitle{%
5485   {\cyr\CYRS\cyrl\cyri\cyrk\cyre}}%
5486 \def\plttitle{\CYRT\cyra\cyrb\cyrl\cyri\cyrc\cyre}%
5487 %%
5488 \def\mtctitle{%
5489   {\cyr\CYRS\cyra\cyrd\cyrr\cyrrh\cyra\cyrje}}%
5490 \def\mlftitle{%
5491   {\cyr\CYRS\cyrl\cyri\cyrk\cyre}}%
5492 \def\mlttitle{\cyr\CYRT\cyra\cyrb\cyrl\cyri\cyrc\cyre}%
5493 %%
5494 \def\stctitle{%
5495   {\cyr\CYRS\cyra\cyrd\cyrr\cyrrh\cyra\cyrje}}%

```

---

<sup>31</sup>The “serbianc” language is written with cyrillic characters.

```

5496 \def\slfttitle{%
5497   {\cyr\CYRS\cyr1\cyri\cyrk\cyre}}%
5498 \def\slttitle{\CYRT\cyra\cyrb\cyr1\cyri\cyrc\cyre}%
5499 </serbianc>

```

### 8.103 “Slovak” language: slovak.mld

The titles for the “slovak” language are taken from the babel package [10]:

```

5500 <{*slovak}
5501 \ProvidesFile{slovak.mld}[1999/03/16]
5502 %% Slovak titles for minitoc.sty
5503 \def\ptctitle{Obsah}%
5504 \def\plfttitle{Zoznam obr\'azkov}%
5505 \def\plttitle{Zoznam tabuliek}%
5506 %%
5507 \def\mtctitle{Obsah}%
5508 \def\mlfttitle{Zoznam obr\'azkov}%
5509 \def\mlttitle{Zoznam tabuliek}%
5510 %%
5511 \def\stctitle{Obsah}%
5512 \def\slfttitle{Zoznam obr\'azkov}%
5513 \def\slttitle{Zoznam tabuliek}%
5514 </slovak>

```

### 8.104 “Slovene” language: slovene.mld

The titles for the “slovene” language are taken from the babel package [10]:

```

5515 <{*slovene}
5516 \ProvidesFile{slovene.mld}[1999/03/16]
5517 %% Slovene titles for minitoc.sty
5518 \def\ptctitle{Kazalo}%
5519 \def\plfttitle{Slike}%
5520 \def\plttitle{Tabele}%
5521 %%
5522 \def\mtctitle{Kazalo}%
5523 \def\mlfttitle{Slike}%
5524 \def\mlttitle{Tabele}%
5525 %%
5526 \def\stctitle{Kazalo}%
5527 \def\slfttitle{Slike}%
5528 \def\slttitle{Tabele}%
5529 </slovene>

```

## 8.105 “Spanish” language: `spanish.mld`

The titles for the “spanish” language are taken from the `babel` package [10]. Note that the “spanish” is in fact “castillan” (see section 8.23 on page 236). Other languages are spoken in Spain: “basque” (section 8.12 on page 231), “catalan” (section 8.25 on page 237), and “galician” (section 8.47 on page 246). “Spanish2” is a version of “spanish” with shorter titles (see section 8.106). “Spanish3” (see section 8.107 on the next page) is a version for the ANTOMEGA [24] project; some titles are different.

```

5530 <*spanish>
5531 \ProvidesFile{spanish.mld}[1999/03/16]
5532 %% Spanish titles for minitoc.sty
5533 \def\ptctitle{\'Indice General}%
5534 \def\plftitle{\'Indice de Figuras}%
5535 \def\plttitle{\'Indice de Tablas}%
5536 %%
5537 \def\mtctitle{\'Indice General}%
5538 \def\mlftitle{\'Indice de Figuras}%
5539 \def\mlttitle{\'Indice de Tablas}%
5540 %%
5541 \def\stctitle{\'Indice General}%
5542 \def\slftitle{\'Indice de Figuras}%
5543 \def\slttitle{\'Indice de Tablas}%
5544 </spanish>

```

## 8.106 “Spanish2” language: `spanish2.mld`

The titles for the “spanish2” language are taken from the `babel` package [10], but made shorter for chapter and section levels. See section 8.105.

```

5545 <*spanish2>
5546 \ProvidesFile{spanish2.mld}[2005/03/31]
5547 %% Spanish titles for minitoc.sty
5548 \def\ptctitle{\'Indice General}%
5549 \def\plftitle{\'Indice de Figuras}%
5550 \def\plttitle{\'Indice de Tablas}%
5551 %%
5552 \def\mtctitle{Contenido}%
5553 \def\mlftitle{Figuras}%
5554 \def\mlttitle{Tablas}%
5555 %%
5556 \def\stctitle{Contenido}%
5557 \def\slftitle{Figuras}%
5558 \def\slttitle{Tablas}%
5559 </spanish2>

```

## 8.107 “Spanish3” language: spanish3.mld

The titles for the “spanish3” language are taken from the omega-spanish.ldf file of the ANTOMEGA project [24]. See section 8.105 on the page before.

```

5560 (*spanish3)
5561 \ProvidesFile{spanish3.mld}[2005/09/06]
5562 %% Spanish titles for minitoc.sty
5563 %% from omega-spanish.ldf of the ANTOMEGA project.
5564 \def\ptctitle{\localspanish{^^^00cdndice general}}%
5565 \def\plftitle{\localspanish{^^^00cdndice de figuras}}%
5566 \def\plttitle{\localspanish{^^^00cdndice de cuadros}}%
5567 %%
5568 \def\mtctitle{\localspanish{^^^00cdndice general}}%
5569 \def\mlftitle{\localspanish{^^^00cdndice de figuras}}%
5570 \def\mlttitle{\localspanish{^^^00cdndice de cuadros}}%
5571 %%
5572 \def\stctitle{\localspanish{^^^00cdndice general}}%
5573 \def\slftitle{\localspanish{^^^00cdndice de figuras}}%
5574 \def\slttitle{\localspanish{^^^00cdndice de cuadros}}%
5575 </spanish3>

```

## 8.108 “Swedish” language: swedish.mld

The titles for the “swedish” language are taken from the babel package [10]:

```

5576 (*swedish)
5577 \ProvidesFile{swedish.mld}[1999/03/16]
5578 %% Swedish titles for minitoc.sty
5579 \def\ptctitle{Inneh\csname aa\endcsname ll}%
5580 \def\plftitle{Figurer}%
5581 \def\plttitle{Tabeller}%
5582 %%
5583 \def\mtctitle{Inneh\csname aa\endcsname ll}%
5584 \def\mlftitle{Figurer}%
5585 \def\mlttitle{Tabeller}%
5586 %%
5587 \def\stctitle{Inneh\csname aa\endcsname ll}%
5588 \def\slftitle{Figurer}%
5589 \def\slttitle{Tabeller}%
5590 </swedish>

```

### 8.109 “Thai” language: `thai.mld` and `thai.mlo`

The titles for the “thai” language are taken from the `babel` package [10], using fonts of the CJK system [29].

The titles for the “thai” language contain characters that cannot be easily generated, hence we load `thai.mlo`.

```
5591 <{*thai}
5592 \ProvidesFile{thai.mld}[2005/01/28]
5593 %% from thaicjk.ldf CJK 4.5.2 Thai support for the babel system
5594 %% by Werner Lemberg <wl@gnu.org>
5595 %%
5596 \mtcloadmlo{thai}
5597 </thai>
```

### 8.110 “Turkish” language: `turkish.mld`

The titles for the “turkish” language are taken from the `babel` package [10]:

```
5598 <{*turkish}
5599 \ProvidesFile{turkish.mld}[1999/03/16]
5600 %% Turkish titles for minitoc.sty
5601 \def\ptctitle{\.I\c cindekiler}%
5602 \def\plftitle{\c Sekiller Listesi}%
5603 \def\plttitle{Tablolar\in Listesi}%
5604 %%
5605 \def\mtctitle{\.I\c cindekiler}%
5606 \def\mlftitle{\c Sekiller Listesi}%
5607 \def\mlttitle{Tablolar\in Listesi}%
5608 %%
5609 \def\stctitle{\.I\c cindekiler}%
5610 \def\slftitle{\c Sekiller Listesi}%
5611 \def\slttitle{Tablolar\in Listesi}%
5612 </turkish>
```

### 8.111 “UKenglish” language: `UKenglish.mld`

The “UKenglish” language is just like “english”, so we just load `english.mld` (see section 8.32 on page 240):

```
5613 <{*UKenglish}
5614 \ProvidesFile{UKenglish.mld}[2005/07/11]
5615 \mtcselectlanguage{english}%
5616 </UKenglish>
```



## 8.112 “Ukraineb” language: `ukraineb.mld`

The titles for the “ukraineb” language are taken from the `babel` package [10]. Cyrillic fonts are required.

```

5617 <*ukraineb>
5618 \ProvidesFile{ukraineb.mld}[1999/12/06]
5619 %% Ukraine (B) titles for minitoc.sty
5620 %% Needs cyrillic fonts
5621 \def\mtctitle{{\cyr\CYZ\cyrm\cyrii\cyrs\cyrt}}%
5622 \def\mlftitle{{\cyr\CYRP\cyre\cyrr\cyre\cyrl\cyrii\cyrk
5623      \ \cyrii\cyrl\cyryu\cyrs\cyrt\cyrr\cyra\cyrc\cyrii\cyrishrt}}%
5624 \def\mltttitle{{\cyr\CYRP\cyre\cyrr\cyre\cyrl\cyrii\cyrk
5625      \ \cyrt\cyra\cyrb\cyrl\cyri\cyrc\cyrsftsn}}%
5626 %%
5627 \def\ptctitle{{\cyr\CYZ\cyrm\cyrii\cyrs\cyrt}}%
5628 \def\plftitle{{\cyr\CYRP\cyre\cyrr\cyre\cyrl\cyrii\cyrk
5629      \ \cyrii\cyrl\cyryu\cyrs\cyrt\cyrr\cyra\cyrc\cyrii\cyrishrt}}%
5630 \def\pltttitle{{\cyr\CYRP\cyre\cyrr\cyre\cyrl\cyrii\cyrk
5631      \ \cyrt\cyra\cyrb\cyrl\cyri\cyrc\cyrsftsn}}%
5632 %%
5633 \def\stctitle{{\cyr\CYZ\cyrm\cyrii\cyrs\cyrt}}%
5634 \def\slftitle{{\cyr\CYRP\cyre\cyrr\cyre\cyrl\cyrii\cyrk
5635      \ \cyrii\cyrl\cyryu\cyrs\cyrt\cyrr\cyra\cyrc\cyrii\cyrishrt}}%
5636 \def\sltttitle{{\cyr\CYRP\cyre\cyrr\cyre\cyrl\cyrii\cyrk
5637      \ \cyrt\cyra\cyrb\cyrl\cyri\cyrc\cyrsftsn}}%
5638 </ukraineb>

```

## 8.113 “USenglish” language: `USenglish.mld`

The “USenglish” language is just like “english”<sup>32</sup>, so we just load `english.mld` (see section 8.32 on page 240):

```

5639 <*USenglish>
5640 \ProvidesFile{USenglish.mld}[2005/07/11]
5641 \mtcselectlanguage{english}%
5642 </USenglish>

```

---

<sup>32</sup>It should be true for the mini-table titles; the languages themselves have some differences.

## 8.114 “Usorbian” language: lsorbian.mld

The titles for the “usorbian” language<sup>33</sup> are taken from the babel package [10]. See also section 8.77 on page 263.

```

5643 (*usorbian)
5644 \ProvidesFile{usorbian.mld}[1999/03/16]
5645 %% Upper sorbian titles for minitoc.sty
5646 %% Needs cyrillic fonts
5647 \def\ptctitle{Wobsah}
5648 \def\plftitle{Zapismobrazow}
5649 \def\plttitle{Zapistabulkow}
5650 %%
5651 \def\mtctitle{Wobsah}
5652 \def\mlftitle{Zapismobrazow}
5653 \def\mlttitle{Zapistabulkow}
5654 %%
5655 \def\stctitle{Wobsah}
5656 \def\slftitle{Zapismobrazow}
5657 \def\slttitle{Zapistabulkow}
5658 </usorbian>

```

## 8.115 “Vietnam” language: vietnam.mld

The titles for the “vietnam” language are taken from the vietnam.sty package. Vietnamese fonts are required. See also section 8.116 on the next page.

```

5659 (*vietnam)
5660 \ProvidesFile{vietnam.mld}[1999/03/16]
5661 %% vietnamese titles for minitoc.sty
5662 %%
5663 \def\ptctitle{M\{d\{u\}c l\{d\{u\}c}
5664 \def\plftitle{Dan h s\{ach h\{inh v\{~e}
5665 \def\plttitle{Dan h s\{ach b\{h\{a\}ng}
5666 %%
5667 \def\mtctitle{M\{d\{u\}c l\{d\{u\}c}
5668 \def\mlftitle{Dan h s\{ach h\{inh v\{~e}
5669 \def\mlttitle{Dan h s\{ach b\{h\{a\}ng}
5670 %%
5671 \def\stctitle{M\{d\{u\}c l\{d\{u\}c}
5672 \def\slftitle{Dan h s\{ach h\{inh v\{~e}
5673 \def\slttitle{Dan h s\{ach b\{h\{a\}ng}
5674 </vietnam>

```

---

<sup>33</sup>Upper sorbian. Sorbian, or wendisch, is a member of the west slavic subgroup of indo-european languages spoken in Upper Lusatia in the german *länder* of Saxony and Brandenburg. The Sorbs are descendents of the Wends, the german name for the slavic tribes who occupied the area between the Elbe and Saale rivers in the west and the Odra (Oder) river in the east during the medieval period (VIth century).

## 8.116 “Vietnamese” language: vietnamese.mld

The “vietnamese” language is just a synonym for the “vietnam” language. So we just load vietnam.mld. Vietnamese fonts are required. See also section 8.115 on the preceding page.

```
5675 <*vietnamese>
5676 \ProvidesFile{vietnamese.mld}[2004/12/14]
5677 \mtcselectlanguage{vietnam}%
5678 </vietnamese>
```

## 8.117 “Welsh” language: welsh.mld

The titles for the “welsh” language are taken from the babel package [10]:

```
5679 <*welsh>
5680 \ProvidesFile{welsh.mld}[1999/12/06]
5681 %% Welsh titles for minitoc.sty
5682 \def\ptctitle{Cynnwys}%
5683 \def\plftitle{Rhestr Ddarluniau}%
5684 \def\plttitle{Rhestr Dablau}%
5685 %%
5686 \def\mtctitle{Cynnwys}%
5687 \def\mlftitle{Rhestr Ddarluniau}%
5688 \def\mlttitle{Rhestr Dablau}%
5689 %%
5690 \def\stctitle{Cynnwys}%
5691 \def\slftitle{Rhestr Ddarluniau}%
5692 \def\slttitle{Rhestr Dablau}%
5693 </welsh>
```

# Complements

# Bibliography

- [1] Alex AJ. Typesetting Malayalam Using  $\text{\LaTeX}$  2 $\epsilon$ . <http://www.tug.org/tex-archive/language/malayalam/doc/mm-usr.pdf>, April 2003. alexaj@myrealbox.com. 31, 227, 264, 265
- [2] Donald Arseneau. The notoccite package. <http://www.tug.org/tex-archive/macros/latex/contrib/misc/notoccite.sty>, July 2000. asnd@triumf.ca. 38, 75, 97, 100, 103, 187, 208
- [3] Donald Arseneau. The placeins package. <http://www.tug.org/tex-archive/macros/latex/contrib/misc/placeins.sty> and <http://www.tug.org/tex-archive/macros/latex/contrib/misc/placeins.sty>, June 2002. asnd@triumf.ca. 21, 192, 197, 208
- [4] Benjamin Bayart. *Joli manuel pour  $\text{\LaTeX}$  2 $\epsilon$* . Guide local pour l’ESIEE, December 1995. <http://www.tug.org/tex-archive/info/JMPL.ps.gz>. 36
- [5] Emmanuel Beffara. Rubber Manual 1.0. <http://www.pps.jussieu.fr/~beffara/soft/rubber/doc.html>, July 2005. 46
- [6] Berhanu Beyene, Manfred Kudlek, Olaf Kummer, and Jochen Metzinger. Ethiopian Language Support for the babel Package. <http://www.tug.org/tex-archive/language/ethiopia/ethiop/doc/ethiodoc.ps>, December 1998. Universität Hamburg. 31, 227, 242
- [7] Javier Bezos. Typesetting Guaraní with  $\text{\TeX}$ . <http://www.tug.org/tex-archive/language/guarani/guarani.pdf>, July 2004. 31, 227, 251
- [8] Javier Bezos. The titlesec and titletoc packages. <http://www.tug.org/tex-archive/macros/latex/contrib/titlesec/titlesec.pdf>, January 2005. <http://www.texytipografia.com/contact.php>. 192
- [9] Johannes Braams. Babel, a multilingual style-option system for use with  $\text{\LaTeX}$ ’s standard document styles. *TUGboat*, 12(2):291–301, June 1991. <http://www.tug.org/TUGboat/Articles/tb12-2/tb32braa.pdf>. 227
- [10] Johannes Braams. Babel, a multilingual package for use with  $\text{\LaTeX}$ ’s standard document classes. <http://www.tug.org/tex-archive/macros/latex/required/babel/babel.pdf>

- and <http://www.tug.org/tex-archive/macros/latex/required/babel/user.pdf>, June 2004. [babel@bramms.cistron.nl](mailto:babel@bramms.cistron.nl). 27, 31, 227, 228, 230, 231, 232, 233, 234, 235, 237, 238, 239, 240, 241, 244, 245, 246, 247, 248, 256, 257, 261, 262, 263, 264, 267, 268, 269, 270, 271, 273, 275, 276, 277, 278, 279, 280, 281, 282, 283
- [11] Johannes Braams, Victor Eijkhout, and Nico Poppelier. The development of national L<sup>A</sup>T<sub>E</sub>X styles. *TUGboat*, 10(3):401–406, November 1989. 227
- [12] David Carlisle. The xr package. <http://www.tug.org/tex-archive/macros/required/tools/xr.dtx>, May 1994. [carlisle@cs.man.ac.uk](mailto:carlisle@cs.man.ac.uk). 20, 108
- [13] Steven Douglas Cochran. The subfig Package. <http://www.tug.org/tex-archive/macros/latex/contrib/subfig/subfig.pdf>, January 2004. [sds@cs.cmu.edu](mailto:sds@cs.cmu.edu), [cochran@ieee.org](mailto:cochran@ieee.org). 22, 179
- [14] Oliver Corff. MonT<sub>E</sub>X – A Quick Guide (*draft*). <http://www.tug.org/tex-archive/language/mongolian/montex/doc/mlsquick.pdf>, November 2001. [corff@zedat.fu-berlin.de](mailto:corff@zedat.fu-berlin.de). 31, 227, 232, 235, 266, 274
- [15] Oliver Corff. MonT<sub>E</sub>X – mongolian for L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>, Implementation Level System Documentation. <http://www.tug.org/tex-archive/language/mongolian/montex/doc/montex.pdf>, July 2002. [corff@zedat.fu-berlin.de](mailto:corff@zedat.fu-berlin.de). 31, 227, 232, 235, 266, 274
- [16] Serguei Dachian, Arnak Dalalyan, and Vardan Vakopian. ArmT<sub>E</sub>X: a System for Writing in Armenian with T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X. <http://www.tug.org/tex-archive/language/armtex/v2.0/manual.ps>, written in armenian, June 1999. [Serguei.Dachian@univ-lemans.fr](mailto:Serguei.Dachian@univ-lemans.fr). 31, 227, 230
- [17] Bernard Desgraupes. *L<sup>A</sup>T<sub>E</sub>X, Apprentissage, guide et référence*. Vuibert, Paris, second edition, 2003.
- [18] Jean-Pierre F. Drucbert. The shorttoc package. <http://www.tug.org/tex-archive/macros/latex/contrib/shorttoc/shorttoc.dtx>, August 2002. [jean-pierre.drucbert@onera.fr](mailto:jean-pierre.drucbert@onera.fr).
- [19] Jean-Pierre F. Drucbert. Le paquetage minitoc. <http://www.tug.org/tex-archive/macros/latex/contrib/minitoc/fminitoc.pdf>, July 2005. [jean-pierre.drucbert@onera.fr](mailto:jean-pierre.drucbert@onera.fr). 210
- [20] Jean-Pierre F. Drucbert. The minitoc package. <http://www.tug.org/tex-archive/macros/latex/contrib/minitoc/minitoc.pdf>, July 2005. [jean-pierre.drucbert@onera.fr](mailto:jean-pierre.drucbert@onera.fr). 210
- [21] Victor Eijkhout. The comment package. <http://www.tug.org/tex-archive/macros/latex/contrib/comment/>, October 1999. [victor@eijkhout.net](mailto:victor@eijkhout.net).
- [22] Donald E. Knuth. *The T<sub>E</sub>Xbook*, volume A of *Computers and Typesetting*. Addison-Wesley, Reading, Massachusetts, sixth edition, 1989. Revised to cover T<sub>E</sub>X3, 1991. 287

- [23] Donald E. Knuth. *Le T<sub>E</sub>Xbook, Composition informatique*. Vuibert, Paris, 2003. French translation of [22], by Jean-Côme Charpentier.
- [24] Alexej M. Kryukov. Typesetting multilingual documents with ANTOMEGA. <http://www.tug.org/tex-archive/systems/omega/contrib/antomega/doc/omega/antomega/antomega.pdf>, September 2003. 31, 227, 248, 249, 250, 262, 269, 271, 272, 278, 279
- [25] Klaus Lagally. ArabT<sub>E</sub>X—Typetting Arabic with vowels and ligatures. In *Proceedings of the 7th European T<sub>E</sub>X Conference*, pages 153–172, Prague, 1992. CsTUG. ISBN 80-210-0480-0, [lagallyk@acm.org](mailto:lagallyk@acm.org). 27, 31, 227, 229, 255
- [26] Klaus Lagally. ArabT<sub>E</sub>X Version 3. <http://www.tug.org/tex-archive/language/arabtex/doc/html/arabtex.htm>, October 2003. [lagallyk@acm.org](mailto:lagallyk@acm.org). 27, 31, 227, 229, 255
- [27] Leslie Michel Lamport. *L<sup>A</sup>T<sub>E</sub>X: A Document Preparation System: User's Guide and Reference Manual*. Addison-Wesley, Reading, Massachusetts, second edition, 1994. 43
- [28] L<sup>A</sup>T<sub>E</sub>X3 Project Team. L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> font selection. <http://www.tug.org/tex-archive/macros/latex/doc/fontguide.pdf>, February 2004. 93, 165, 168
- [29] Werner Lemberg. The CJK package for L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> — Multilingual support beyond babel. *TUGboat*, 18(3):214–224, 1997. <http://www.tug.org/TUGboat/Articles/tb18-3/cjkintro600.pdf>. 27, 31, 227, 237, 238, 252, 253, 254, 255, 258, 259, 260, 280
- [30] Knut Lichert. L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> for people in associations: `minutes.sty`. <http://www.tug.org/tex-archive/macros/latex/contrib/minutes/>, January 2001. [knut@lickert.net](mailto:knut@lickert.net). 93, 186
- [31] Nicolas Markey. Tame the BeaST (Tthe B to X of B<sub>B</sub>T<sub>E</sub>X). [http://www.tug.org/tex-archive/info/bibtex/tamethebeast/ttb\\_en.pdf](http://www.tug.org/tex-archive/info/bibtex/tamethebeast/ttb_en.pdf), [markey@lsv.ens-cachan.fr](mailto:markey@lsv.ens-cachan.fr), June 2005.
- [32] Rowland McDonnell. The sectsty package v2.0.2. <http://www.tug.org/tex-archive/macros/latex/contrib/sectsty/>, April 2002. [rowland.mcdonnell@physics.org](mailto:rowland.mcdonnell@physics.org). 39, 52, 76
- [33] Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, and Chris Rowley. *The L<sup>A</sup>T<sub>E</sub>X Companion 2*. Tools and Techniques for Computer Typesetting. Addison-Wesley Professional, Reading, Massachusetts, 2004.
- [34] Tobias Oetiker, Hubert Partl, Helene Hyna, and Elisabeth Schlegl. Une courte (?) introduction à L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>. *Cahiers GUTenberg*, Hors-série numéro 2, February 2003. Translation in french of [35] by Matthieu Herrb, adaptation and complements by Daniel Flipo, see also: <http://www.tug.org/tex-archive/info/lshort/french/flshort-3.20.pdf>.
- [35] Tobias Oetiker, Hubert Partl, Helene Hyna, and Elisabeth Schlegl. The Not So Short Introduction to L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>. <http://www.tug.org/tex-archive/info/lshort/english/lshort.pdf>, April 2004. 287

- [36] Palash Baran Pal. Bangtex: a package for typesetting documents in Bangla using the  $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  systems. <http://tnp.saha.ernet.in/~pbpal/bangtex/bangtex.html>, January 2001. 31, 227, 231
- [37] Oren Patashnik. Bib $\text{T}_{\text{E}}\text{X}$ ing. Documentation for general Bib $\text{T}_{\text{E}}\text{X}$  users, <http://www.tug.org/tex-archive/biblio/bibtex/contrib/doc/btxdoc.pdf>, February 1988.
- [38] Oren Patashnik. Designing Bib $\text{T}_{\text{E}}\text{X}$  styles. The part of Bib $\text{T}_{\text{E}}\text{X}$ 's documentation that's not meant for general users, <http://www.tug.org/tex-archive/biblio/bibtex/contrib/doc/btxhak.pdf>, February 1988.
- [39] Sebastian Rahtz and Heiko Oberdiek. Hypertext marks in  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ : a manual for hyperref. <http://www.tug.org/tex-archive/macros/latex/contrib/hyperref/doc/manual.pdf>, February 2004. 48, 75, 194
- [40] Young Ryu. The TX Fonts. <http://www.tug.org/tex-archive/fonts/txfonts/doc/txfontsdoc.pdf>, December 2000. 25
- [41] Stephan P. von Bechtolsheim.  $\text{T}_{\text{E}}\text{X}$  in practice: Comments on a 4-volume, 1400-page series on  $\text{T}_{\text{E}}\text{X}$ . *TUGboat*, 11(3):409–412, September 1990. <http://www.tug.org/TUGboat/Articles/tb11-3/tb29bechtolsheim.pdf>.
- [42] Stephan P. von Bechtolsheim. *T<sub>E</sub>X in Practice*. Springer Verlag, Berlin, Germany / Heidelberg, Germany / London, UK / etc., 1992.
  - 1. Basics.
  - 2. Paragraphs, Maths and Fonts.
  - 3. Tokens, Macros.
  - 4. Output Routines.

(A progressive approach to the secrets of  $\text{T}_{\text{E}}\text{X}$ ). 77
- [43] Peter R. Wilson. The needspace package. <http://www.tug.org/tex-archive/macros/latex/contrib/misc/needspace.sty>, January 2003. [pandgwilson@earthlink.net](mailto:pandgwilson@earthlink.net). 43
- [44] Peter R. Wilson. The tocloft package. <http://www.tug.org/tex-archive/macros/latex/contrib/tocloft/tocloft.pdf>, September 2003. [pandgwilson@earthlink.net](mailto:pandgwilson@earthlink.net). 31, 49, 53, 192
- [45] Peter R. Wilson. The abstract package. <http://www.tug.org/tex-archive/macros/latex/contrib/abstract/abstract.pdf>, April 2004. [pandgwilson@earthlink.net](mailto:pandgwilson@earthlink.net). 39, 52, 194
- [46] Peter R. Wilson. The appendix package. <http://www.tug.org/tex-archive/macros/latex/contrib/appendix/appendix.pdf>, April 2004. [pandgwilson@earthlink.net](mailto:pandgwilson@earthlink.net). 49, 191
- [47] Peter R. Wilson. The tocbibend package. <http://www.tug.org/tex-archive/macros/latex/contrib/tocbibind/tocbibind.pdf>, May 2004. [pandgwilson@earthlink.net](mailto:pandgwilson@earthlink.net). 37, 191



- [48] Peter R. Wilson. The Memoir Class. <http://www.tug.org/tex-archive/macros/latex/contrib/memoir/memman.pdf>, February 2005. [pandgwilson@earthlink.net](mailto:pandgwilson@earthlink.net). 50, 76, 86, 193
- [49] Peter R. Wilson and Donald Arseneau. The iftmarg package. <http://www.tug.org/tex-archive/macros/latex/contrib/misc/ifmtarg.sty>, March 2000. [pandgwilson@earthlink.net](mailto:pandgwilson@earthlink.net), [asnd@triumf.ca](mailto:asnd@triumf.ca). 77

## Change History

00 1990/10/01

Original version, by Nigel WARD.

00 1991/11/01

Revised to reuse `\chapter`, `\section`, `\subsection` commands transparently, generate toc-file-name automatically, assorted other cleanup (Dan JURAFSKY).

01 1993/06/01

Added `\chapterend` to terminate the scope of a minitoc. (*If you forgot putting `\chapterend` at the end of each chapter, an entry for the next chapter will appear in each minitoc.*) (Thanks to Yufan HU).

All the layout of the minitoc is in the `\minitableofcontents` command, so if somebody wants to redefine that layout, he has just to rewrite it (and only it).

At least three passes (3) of  $\LaTeX$  are necessary to get correct minitocs (the first pass creates the `.mtc(X)` files, the second uses them (but they may contain wrong page numbers) and recreates them, the third should be ok).

New design, to avoid allocating a newwrite, or file descriptor, for each chapter (a deadly bug) (Jean-Pierre F. DRUCBERT).

Problems: you *must* have `\chapterend` to terminate each chapter with a minitoc. How about avoiding this constraint?

Replaced the `minipage` environment by a `verse` environment, to allow a minitoc being split across pages.

Some mods added to work with `xr.sty` (external references). `xr.sty` version 5 is much more tolerant.

The depth of the minitoc is user-adjustable with the counter `minitocdepth` (similar to `tocdepth` for the table of contents).

Works with `\chapter[xxx]{yyy}` and floating bodies. Works with two columns (but the minitoc is composed in one column; how to make it to spread over the two columns?).

You can inhibit the minitoc for the next chapter by preceding it with `\minitocno`. (`\minitocyes` is useless for the user, because it is implicit *after* the `\chapter*` pseudo-chapters).

02 1993/07/05

Added compatibility with `hangcaption.sty` (the package `hangcaption` (if present) must be loaded *before* the `minitoc` package). *Beware* to options modifying `@caption`.

03 1993/07/09

Version 3 not released (buggy).

04 1993/07/09

Added `\if@realch` to avoid contentslines from pseudo-chapters to go into the toc. The package file `mtcoeff.sty` allows you to use a  $\LaTeX$  document with minitoc commands and to make them transparent: just replace the minitoc option by `mtcoeff`.

05 1993/07/13

Added a selection mechanism to not write spurious things in the minitocs.

06 1993/07/15

Fixed problems about chapters in the toc, removed obsolete `\caption` stuff (filters are better) added compatibility with `toch.sty` (`toch.sty` makes a table of chapters; if used, must be loaded *before* `minitoc.sty`).

07 1993/07/22

(*major differences*) Completely rewritten, using tricks from `xr.sty` (the version 5, by David CARLISLE). The info for minitocs is directly stolen from the `.toc` file. `\chapterend` and `\minitocno` are suppressed, `\minitoc`, `\dominitoc` and `\faketableofcontents` added.

08 1993/07/29

Spacing adjustments.

- 09 1993/08/04  
Added mods for MS-DOS (search for MS-DOS, uncomment; search for UNIX, comment out). MS-DOS allows only 3 characters for extensions in file names (what a pity).
- 10 1993/08/05  
Works now with appendices. Detects now the obsolete versions of `latex.tex` (`\@inputcheck` or `\reset@font` not defined).
- 11 1993/08/18  
Added `\mtcSfont`, font for section entries, `\mtcSSfont` for subsection entries, `\mtcSSSfont` for subsubsection entries, `\mtcPfont` for paragraph entries, `\mtcSPfont` for subparagraph entries.
- 12 1993/12/16  
Use `\kern` in place of `\vspace*`, and added penalties (`\nopagebreak`) to avoid a page break just before last `\mtc@rule`. Also added a `samepage` environment. Removed old commented out lines from previous versions.
- 13 1993/12/17  
Added `minilof` and `minilot` stuff. For MS-DOS, uncomment the definition of `\SHORTEXT`.
- 14 1994/01/03  
Corrected some spacing problems (avoiding `~`'s). `\mtifont` is changed from `\normalsize\bf` to `\large\bf`.  
Corrected space under `minitoc/lof/lot` and added a `\raggedright` setting to avoid "underfull" warnings.  
Removed the setting of `\clubpenalty` and `\widowpenalty` to 10000 (done by `\samepage`), and `\noindent`.  
Simplified processing of optional argument in `\minitoc`, `\minilof` and `\minilot`.  
Some modifications suggested by Donald ARSENEAU (thanks): `\@newread` becomes `\newread`, not outer version of `\newread`; `\empty` replaced by `\relax` in the spare definition of `\reset@font`.
- 15 1994/01/27  
Added `\parttoc`, `\partlof` and `\partlot` for books, `\secttoc`, `\sectlof` and `\sectlot` for articles, with some commands and parameters parallel to those for mini-tables.
- 16 1994/02/02  
Bug fixes (typos).
- 17 1994/06/23  
'n' (null) synonym of 'e' (empty) in the optional argument of `\minitoc`, `\dominitoc`, and siblings.  
Compatibility with "L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>". Thanks to Denis ROEGEL (who found the problem) and Frank MITTELBACH (who gave the hints to solve).
- 18 1994/06/26  
Introduce the language files as options. Many thanks to Michel GOOSSENS (via Frank MITTELBACH) who was inspired by the code of the `babel` package (by Johannes BRAAMS).  
Make `minitoc` really compatible with L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>.
- 19 1994/08/16  
`\mtcrule`, `\nomtcrule` etc. commands added.  
Added stuff for numbering of chapters (parts, sections) not starting at 1. `\firstchapteris` etc. commands added.  
Corrected `mtcswedish.sty` (Jan Michel RYNNING.)  
Corrected a bug in `\c@mti`.  
Corrected appendix in articles.
- 20 1994/08/25  
Added the `\mtcpagenumbers` and `\nomtcpagenumbers` commands (and siblings) to make `minitocs` with/without page numbers. Default: with page numbers.  
Corrected (difficult bug) appendix in articles.

- Corrected a problem with chapters numbered with uppercase roman numbers.
- Corrected spacing before and after minitocs and siblings.
- Corrected vertical spacing.
- 21 1994/09/07
  - Corrected typos in `minitoc.sty` and `minitoc.tex`.
- 22 1994/10/10
  - Corrected typos in `minitoc.sty`.
- 23 1994/11/08
  - Added a missing line in `\sectlof`.
  - Removed appendix stuff.
  - Removed stuff for `\firstchapteris` and `co`. These commands are obsolete.
  - Works with document classes resetting chapter (or section) number at each part (thanks to Denis ROEGEL).
- 24 1994/12/21
  - The `\protect` commands have been removed from the `.toc`, `.lot` and `.lot` files, so some internal macros have been corrected to be compatible with the  $\text{\LaTeX}$  2 $\epsilon$  release of December 1994. Thanks to Denis ROEGEL who did the work.
- 25 1996/09/13
  - Updated `mtcnorsk.sty` and added `mtcnynorsk.sty` on a suggestion from Dag LANGMYHR.
- 26 1996/11/14
  - Added `breton`, `estonian`, `germanb`, `greek`, `irish`, `russianb`, `scottish`, `lower` and `upper sorbian`; renamed “esperanto” by “esperant” like in the `babel` package.
  - Language specific files are now named `\language.mld` (replacing `mtc\language.sty`) because they are not packages and it makes shorter names.
- 27 1996/12/20
  - `english.mld` loaded as default language.
  - Added `vietnam.mld` and `arab.mld`.
  - Corrections for starred sectionning commands.
  - Renamed `minitocoff.sty` into `mtcoff.sty` to keep the name short.
- 28 1997/10/29
  - Added `coffee` stuff.
  - Added `\addstarred` stuff (for starred chapter stuff).
  - Added autoconfiguration of extensions.
  - Added the `shorttext` package option.
  - Added the `afrikaan(s)`, `brazil`, and `ethiopia(n)` languages.
  - Fixed bug in `parttocs`.
- 28 1998/06/15
  - A typo corrected by Donald ARSENEAU:
    - `{\let@dottedtocline@undottedtocline}\}`
    - should probably be
    - `{\let@dottedtocline@undottedtocline}\}`
    - (a backslash was missing after `\let`). Thanks to him.
  - Added the `bahasa` language.
- 28 1998/12/03
  - Added the `tight` and `loose` package options.
- 29 1999/03/16
  - Added the `bicig`, `buryat`, `mongol` and `russianc` languages.
- 29 1999/06/28
  - Added the `armenian` language (from `ArmTeX`).
- 29 1999/07/23
  - Added the `dotted/undotted` package options (default: `dotted`).
- 29 1999/07/29
  - Added the `lithuanian` language.

- 30 1999/12/06
  - Added the basque, ngermanb, serbian, ukraineb, and welsh languages.
  - Corrected a bug in `\sltname` definition (mlt should be slt).
- 31 2000/04/04
  - Added compatibility with the `hyperref` package, thanks to Heiko OBERDIEK, who has also simplified some code and fixed the infamous `\chapter*` bug.
- 32 2000/08/08
  - `\nomtcrule` corrected.
  - Added very (too) numerous new commands for the mini-table features: `\beforeparttoc`, `\beforepartlof`, `\beforepartlot`, `\afterparttoc`, `\afterpartlof`, `\afterpartlot`, `\thispageparttocstyle`, `\thispagepartlofstyle`, and `\thispagepartlotstyle`.
  - Documentation improved by Stefan ULRICH.
- 33 2000/12/07
  - Added new adjustment commands: `\mtcaddchapter`, `\mtcaddsection`, and `\mtcaddpart`. These commands add stuff in the `.toc`, `.lof` and `.lot` files for the `\chapter*` (`\section*` and `\part*`) problem. From a suggestion by Karl F. EVERITT.
- 33 2000/12/08
  - Corrected a feature in `\mtcaddchapter` and `co.` with a blank optional argument.
- 34 2000/12/13
  - Added `.mld` files for alternate names of languages: `so`, `american.mld` just loads `english.mld`, which contains the real definitions.
  - Added in the documentation a section for use with the `tocbibend` package.
- 35 2001/01/09
  - Added macros to test if a file is “empty” (*i.e.* empty, blank or inexistent) or “non empty” (*i.e.* useful). I used some code from Stephan P. von BECHTOLSHEIM.
  - Added the `checkfiles/nocheckfiles` package options.
  - Replaced `\The@chapter` by `\The@mtc`.
- 35 2001/02/26
  - Added `bulgarian.mld`, `hebrew.mld`, `icelandic.mld`, `latin.mld`, and `samin.mld`.
- 35 2001/03/09
  - Added `\mtcselectlanguage`.
- 35 2001/06/01
  - Fixed the `estonian` package option (missing).
- 35 2001/07/04
  - Added the `interlingua` language.
- 36 2002/02/11
  - Corrected an interaction with `\tableofcontents` which creates a `\chapter*` or a `\section*`, perturbing `mtc/stc` counters (problem signalled by Frank MITTELBACH).
- 36 2002/02/18
  - Corrected a spacing problem with empty titles (problem signalled by Frank MITTELBACH).
  - Workaround for the `\parttoc-\chapter*` problem.
- 36 2002/02/19
  - Added `\mtcskip` and `\mtcskipamount`.
- 36 2002/02/27
  - Fixed test for empty files.
- 36 2002/03/13
  - Added the `bangla` language.
- 36 2002/03/15
  - Reduced depth of `\mtc@strutbox`.
- 37 2003/01/24
  - Version #37 dropped.
- 38 2003/01/24
  - `\hrule` and `\vrule` replaced by `\rule` ( $\LaTeX$ ).

- `pt` becomes `\@pt` and `\opt` becomes `\z@`.
  - Added `mtc@zrule` for zero-dims rules.
  - Added the `frenchb` language (synonym of `french`).
- 38 2003/01/30
  - Added the `flsection` and `flsectionb` package options.
  - Changed the test for empty titles.
- 38 2003/01/31
  - The `tight` and `loose` package options are applied to `\parttoc` (Thomas LEONHARDT).
- 38 2003/02/07
  - Package options `flsection` and `flsectionb` removed and replaced by the `insection` package option (like `flsectionb`).
- 38 2003/02/11
  - Corrected numbering of SLF, SLT.
- 38 2003/02/20
  - Added `frenchle` and `frenchpro` languages (synonyms of `french`).
  - Corrected `sectocs`, at least.
- 38 2003/03/18
  - Corrected some vertical spacings and struts (I added some mods by Frank MITTELBACH, many thanks to him.). A lot of cleaning remains to do, but the release seems to be needed now.
- 39 2003/04/09
  - `\nomtcbpagenumbers` and `memoir` class.
  - Modern font commands for compatibility with the `memoir` class.
- 39 2003/06/08
  - Added `\@fileswfalse` and `\mtc@hook@beforeinputfile` for the `notoccite` package (requested by Donald ARSENEAU); added the `notoccite` package option (loads the `notoccite` package).
- 39 2004/09/08
  - Added comments in `.mld` files using special fonts.
  - Added language options and `.mld` files for dialects: `canadian (english)`, `acadian`, `acadien`, `canadien (french)`, `naustrian`, `ngerman (ngermanb)`.
  - Documentation: added a paragraph about making a TOC for appendices, eventually not listed in the main TOC.
- 39 2004/09/17
  - Corrections in the documentation; corrections about rules.
- 40 2004/12/09
  - Added a figure in `minitoc.tex` about the need of three compilations.
  - Added a paragraph about a problem with the `appendix` package.
  - Added some infos in `minitoc.bug`.
  - Added the `japanese` and `castillan` languages.
  - Removed the test on the presence of the `multicol` package in `minitoc.tex`, because `multicol` is a required package.
- 40 2004/12/13
  - Updated `fminitoc.bib` and `minitoc.bib`.
- 40 2004/12/14
  - Added the `hints` package option. This option is still experimental; your advice is welcome.
- 40 2004/12/20
  - Added `fminitoc.pdf` (french documentation in PDF format).
- 41 2005/01/05
  - Corrections in documentation.
  - Message added if some sectioning commands are not available.
  - Replaced `\typeout` commands in `minitoc.sty` by the `\PackageInfo` or `\PackageWarning` commands; with the line number when useful (`\@gobble` if no line number). Hence,

- the package is less verbose (`\PackageInfo` writes only in the `.log` file, not on the terminal).
- 41 2005/01/06  
Added the `\mtcsetfont` (Benjamin BAYART) and `\mtcsettitlefont` commands, with a much simpler syntax.
- 41 2005/01/10  
Added bibliography.
- 41 2005/01/11  
 $\TeX$  classes: `amsart` and `amsproc` are incompatible with `minitoc`, `amsbook` needs precautions.
- 41 2005/01/12  
Added `\mtcsetformat`.
- 41 2005/01/18  
Added `\mtcsettitle`.  
Added a hint for recommending the `insection` package option.
- 41 2005/01/19  
Added a hint about coherence `\dominitoc/\minitoc` and `co`.  
Added a hint about the presence of `\dominitoc` and `co`.  
Improved documentation about hints.
- 41 2005/01/20  
Added a hint about using short extensions with more than 99 parts or 99 chapters or 99 sections.
- 41 2005/01/25  
`\ptifont`: `\Huge\bfseries` becomes `\LARGE\bfseries`.
- 41 2005/01/26  
Added `\mtcsetpagenumbers`.
- 41 2005/01/28  
Added many new language files: `serbianc.mld`, `chinese1.mld`, `chinese2.mld`, `hangul1.mld`, `hangul2.mld`, `hangul3.mld`, `hangul4.mld`, `hanja1.mld`, `hanja2.mld`, `japanese2.mld`, `japanese3.mld`, `japanese4.mld`, `japanese5.mld`, `thai.mld`.
- 41 2005/02/02  
Added `\mtcsetrules`.
- 41 2005/02/03  
Added `\plfrule`, `\noplfrule`, `\mlfrule`, `\nomlfrule`, `\slfrule`, `\noslfrule`, `\pltrule`, `\nopltrule`, `\mltrule`, `\nomltrule`, `\sltrule`, `\nosltrule`.
- 41 2005/02/04  
Added the `mtchideinmaintoc` environment.
- 41 2005/02/08  
Added `latvian.mld`, `letton.mld`, `greek-mono.mld`, `greek-polydemo.mld`, `greek-polykatha.mld`, `polish2.mld`, `russian2m.mld`, and `russian2o.mld` as new language files.
- 41 2005/02/09  
Added the `mtchideinmainlof` and `mtchideinmainlot` environments.
- 41 2005/02/10  
Added tests on the `mtchideinmain*` environments.
- 41 2005/02/14  
Added `\mtcfixindex`.
- 42 2005/02/14  
Replaced “language” by “langue” in the french documentation.
- 42 2005/02/15  
Fixed a minor typo.
- 42 2005/02/16  
Upgraded `\mtcfixindex`.
- 42 2005/02/21  
Added `\mtcsettitle`, forgotten to be inserted in v41.

- 43 2005/02/21
  - Version 43: consolidation of v40, v41 and v42.
- 43 2005/02/24
  - Fixed a big bug in `\mtcsetformat`.
  - Fixed a bug in `mtcoff.sty` about `\mtcfixindex`.
- 43 2005/03/02
  - Added the `INSTALL` file and a chapter about installation.
  - Fixed the `\mtcset...` macros.
  - Moved history to the end of package code.
- 43 2005/03/07
  - Completed the hint about coherence `\dominitoc/\minitoc` and `co`.
  - Fixed a typo (Benjamin BAYART).
- 43 2005/03/08
  - Added a hint about coherence `\minitoc` and `\tableofcontents`.
- 43 2005/03/09
  - Added comments about fonts.
- 43 2005/03/10
  - Corrections in documentation.
- 43 2005/03/11
  - Added `\mtcsetfeature`.
- 43 2005/03/14
  - Added `bulgarianb.mld` (upper bulgarian).
- 43 2005/03/15
  - Added `*[-\baselineskip]` after the `\\` after the top rule of each part level mini-table.
- 43 2005/03/16
  - Corrections in the arguments of `\mtcsetfeature`.
- 43 2005/03/18
  - Removed `\markboth` for `minitocs (...)` and `secttocs (...)`.
- 43 2005/03/21
  - Added `spanish2.mld`.
- 43 2005/03/22
  - Added a hint for the abstract package.
- 43 2005/04/06
  - Added `finnish2.mld`, `latin2.mld`, and `magyar2.mld`.
  - Corrected the `stc@verse` environment.
- 43 2005/04/08
  - Renamed `portuges.mld` as `portugues.mld`.
- 43 2005/04/12
  - Correction in `\mtcskip`.
  - First version in `.dtx` format.
- 43 2005/04/14
  - Removed `\ypart`, `\ychapter`, `\ysection`, and stuff; unused.
- 43 2005/05/11
  - Added `\mtcfixglossary`.
  - Added `minitoc.ist` to format the index correctly.
  - Corrected a typo in `\@dosectlot`.
  - Print the documentation with “oneside” to have all marginal notes on left. Added the (extended) code of `morefloats.sty` (Don HOSEK) to allow more marginpars and floats.
- 43 2005/05/26
  - Fixed rules in `parttocs`, `partlofs` and `partlots`.
- 43 2005/05/30
  - Added a hint about the `sectsty` package (must be loaded *before* `minitoc`).
  - Fixed chapter-level entries in `parttocs`, when page numbers must be removed.



- 43 2005/06/01
  - Added a hint about attempts to insert empty mini-tables.
  - Added a hint about the use of obsolete commands.
  - The mini-lists of figures or tables should not be printed empty even if `tocdepth < 1`.
- 43 2005/06/02
  - Added `\mtcsetdepth`.
  - Added the notion of depth for mini-tables of figures/tables.
  - The `hints` option is the default and no more considered as experimental.
- 43 2005/06/03
  - Added an error message in `\mtcsetdepth` if the counter is not available.
- 43 2005/06/06
  - Added `portuges.mld`, which loads `portugues.mld`.
- 43 2005/06/07
  - Added three variants for the malayalam language: `malayalam-keli.mld`, `malayalam-rachana.mld`, and `malayalam-rachana2.mld`.
- 43 2005/06/14
  - Added method for bilingual documentation
- 43 2005/06/15
  - Added `fminitoc.ist` to format correctly the index in french.
- 43 2005/06/16
  - Changed “Liste des Tables” by “Liste des Tableaux” in `french.mld`, and in the french documentation, to stick to the choices of the `babel` package.
- 43 2005/06/17
  - The file `fminitoc.dtx` is now generated by `minitoc.ins`.
- 43 2005/06/21
  - Added “OUI”, “NON”, “oui”, “non”, “O”, and “o” as true/false keywords.
  - Compacted the code about detection of short/long extensions.
- 43 2005/06/22
  - Added “VRAI”, “FAUX”, “vrai”, “faux”, “V”, and “v” as true/false keywords.
- 43 2005/06/23
  - Correctly set the `\iffTR` flag to have the months in the right language in the bibliography.
- 43 2005/06/29
  - Set the flag `\mtcoffwarn@true` in `mtcoff.sty` if a command `\mtcadd...` is found.
- 43 2005/07/01
  - Added `castillian.mld`.
  - Renamed `portugues.mld` as `portuguese.mld`.
- 43 2005/07/11
  - Added `brazilian.mld`, `british.mld`, `UKenglish.mld`, and `USenglish.mld`.
- 43 2005/07/12
  - Suppressed “General:” in the changes history.
- 43 2005/07/13
  - Replaced some `\PackageWarning` commands by `\PackageInfo`.
- 43 2005/07/18
  - Restoring the correspondence of each language option with a `.mld` file.
- 43 2005/07/20
  - Improving the `mtchideinmainlof` and `mtchideinmainlot` environments.
- 43 2005/07/21
  - Added the `\decrementptc`, `\decrementmtc`, and `\decrementstc` commands.
  - Removing unused some flags `\if@mtc@setpagenumbers@act@` and `\if@mtc@setrules@act@`.
- 43 2005/07/22
  - Added a test on the version of the `placeins` package.
  - Corrected a bug in `mtcoff.sty`.
  - Improved some messages in `mtcoff.sty`.

- 43 2005/08/23  
Added a note about `\FloatBarrier`.
- 43 2005/08/24  
Added a note about an alignment problem in the `minitocs`. Updated `minitoc.bug`.  
Made two versions of the `mtchideinmainlof` and `mtchideinmainlot` environment, depending of the presence of the corresponding depth counter.  
The `memoir` class is incompatible if too recent.
- 43 2005/08/25  
Added a comment about the position of the `\do...` preparation commands.  
Corrections in the `mtchideinmainlof` and `mtchideinmainlot` environments.
- 43 2005/08/26  
Added `guarani.mld`.
- 43 2005/08/29  
Added `\incrementptc`, `\incrementmtc`, and `\incrementstc`. Added an optional argument to `\adjustptc`, `\adjustmtc`, and `\adjuststc`.  
Added the `k-tight` and `k-loose` package options.
- 43 2005/09/02  
Added a patch for the recent version of the `memoir` class.
- 43 2005/09/06  
Added `spanish3.mld`.
- 43 2005/09/08  
Use `\mtcselectlanguage` in language options and in “secondary” `.mld` files.
- 43 2005/09/09  
Added `\mtcloadmlo` to be used in some `.mld` files to load a `.mlo` file.
- 43 2005/09/12  
Added a test to forbid direct calls of `\mtcloadmlo` by the user.
- 43 2005/09/13  
Added `farsi1.mld`, `farsi1.mlo`, `farsi2.mld`, and `farsi2.mlo`.  
Added a note about the `rubber` script.
- 43 2005/09/15  
Added `mtcglo.ist` to format the glossary.
- 43 2005/09/16  
Removed the page numbers in the glossary. Done in the `*mk` scripts.

# Index

## Symbols

.lof	81	\@dosectlof	2016, 2055
.log	71	\@dosectlof@used@false	55
.lot	81	\@dosectlof@used@true	2037
.mld	25, 31, 51, 54, 224, 226, 227	\@dosectlot	2020, 2056
.mlo	31, 54, 185, 224, 227	\@dosectlot@used@false	56
.toc	81	\@dosectlot@used@true	2046
\@@dominilof	805, 823	\@dosecttoc	2012, 2054
\@@dominilot	814, 823	\@dosecttoc@used@false	54
\@@dominitoc	780, 796, 823	\@dosecttoc@used@true	2028
\@@dopartlof	1471, 1481	\@dotsep	2518, 2531, 2536, 2541, 2546, 2551, 2556, 2561, 2566, 2571
\@@dopartlot	1480, 1481	\@dottedtocline	548, 620, 687, 702, 748, 761, 764, 767, 772, 775, 778, 1255, 1331, 1405, 1421, 1808, 1877, 1943, 1957
\@@doparttoc	1462, 1481	\@filesfalse	552, 624, 691, 1262, 1335, 1409, 1813, 1881, 1948
\@@dosectlof	2044, 2054	\@firstchapteris@used@false	58
\@@dosectlot	2053, 2054	\@firstchapteris@used@true	448
\@@dosecttoc	2035, 2054	\@firstoftwo	111
\@M	279, 702, 725, 738, 1421, 1957, 1988	\@firstpartis@used@false	57
\@Undottedtocline	720, 2980, 3900	\@firstpartis@used@true	1093
\@Undottedtoclinep	733, 748, 761, 764, 767, 772, 775, 778, 3900	\@firstsectionis@used@false	59
\@afterheading	1137, 1140, 3963, 3971	\@firstsectionis@used@true	1719
\@afterindenttrue	278, 724, 737	\@gobble	3787, 3790, 3791, 3900
\@centercr	486, 1189, 1746	\@gobbletwo	3733, 3772
\@chapter	215, 704, 3230, 3671, 3937, 3939	\@hangfrom	1978
\@dominilof	785, 824	\@ifclassloaded	85, 3184, 3190, 3200
\@dominilof@used@false	49	\@ifnextchar	497, 568, 635, 784, 789, 795, 1200, 1274, 1347, 1451–1453, 1757, 1824, 1890, 2024–2026, 3715, 3721, 3727, 3937
\@dominilof@used@true	806	\@ifpackagelater	3174
\@dominilot	790, 825	\@ifpackageloaded	68, 82, 83, 3134, 3140, 3146, 3152, 3157, 3206, 3655
\@dominilot@used@false	50	\@ifpackagewith	3157, 3206
\@dominilot@used@true	815	\@ifundefined	2873, 2910, 2984, 3000, 3215, 3230, 3245, 3668, 3671, 3674
\@dominitoc	780, 823	\@input	913, 975, 1030, 2947, 2960
\@dominitoc@used@false	48	\@inputcheck	133, 265, 899, 909, 929, 962, 971, 988, 1017, 1026, 1043, 1564, 1565, 1574, 1593,
\@dominitoc@used@true	797		
\@dopartlof	1443, 1482		
\@dopartlof@used@false	43		
\@dopartlof@used@true	1464		
\@dopartlot	1447, 1483		
\@dopartlot@used@false	44		
\@dopartlot@used@true	1473		
\@doparttoc	1439, 1481		
\@doparttoc@used@false	42		
\@doparttoc@used@true	1455		

- 1621, 1622, 1631, 1649, 1677, 1678,  
1687, 1705, 2124, 2125, 2134, 2150,  
2180, 2181, 2190, 2206, 2236, 2237,  
2246, 2262
- \@longextensions@false . 240, 245, 2997  
\@longextensions@true . . . . . 242, 3855  
\@m@mchapter . . . . . 3935  
\@makechapterhead . . . . 3962, 3967, 3969  
\@makephead@c . . . . . 1137, 1168  
\@makephead@l . . . . . 1137, 1140, 1154  
\@makephead@r . . . . . 1137  
\@minilof@used@false . . . . . 46  
\@minilof@used@true . . . . . 570  
\@minilot@used@false . . . . . 47  
\@minilot@used@true . . . . . 637  
\@minitoc@used@false . . . . . 45  
\@minitoc@used@true . . . . . 499  
\@mkboth . . . . . 419, 1215, 1289, 1363  
\@mtc@FEfalse . . . . . 144, 157  
\@mtc@FEtrue . . . . . 134, 148, 155  
\@mtc@Himlof@false . . . . . 2813, 2841  
\@mtc@Himlof@true . . . . . 2820, 2835  
\@mtc@Himlot@false . . . 2843, 2857, 2871  
\@mtc@Himlot@true . . . . . 2850, 2865  
\@mtc@Himtoc@false . . . 2800, 2811, 2827  
\@mtc@Himtoc@true . . . . . 2805  
\@mtc@LIfalse . . . . . 138, 142, 144  
\@mtc@LItrue . . . . . 133, 148  
\@mtc@abstract@loaded@true . . . 3207  
\@mtc@chapter@def@false . . . . . 29, 30  
\@mtc@chapter@def@true . . . . . 30  
\@mtc@chapter@undef@false . . . . . 36  
\@mtc@chapter@undef@true . . . . . 35, 36  
\@mtc@checkfilesfalse . . . . . 2993  
\@mtc@checkfilestrue . . . . . 135, 2992  
\@mtc@empty@minilof@false . . . . . 22  
\@mtc@empty@minilof@true . . . . . 580  
\@mtc@empty@minilot@false . . . . . 23  
\@mtc@empty@minilot@true . . . . . 647  
\@mtc@empty@minitoc@false . . . . . 21  
\@mtc@empty@minitoc@true . . . . . 509  
\@mtc@empty@partlof@false . . . . . 19  
\@mtc@empty@partlof@true . . 1286, 1360  
\@mtc@empty@partlot@false . . . . . 20  
\@mtc@empty@parttoc@false . . . . . 18  
\@mtc@empty@parttoc@true . . . . . 1212  
\@mtc@empty@sectlof@false . . . . . 25  
\@mtc@empty@sectlof@true . . . . . 1836  
\@mtc@empty@sectlot@false . . . . . 26  
\@mtc@empty@sectlot@true . . . . . 1902  
\@mtc@empty@secttoc@false . . . . . 24  
\@mtc@empty@secttoc@true . . . . . 1769  
\@mtc@hints@false . . . . . 3129  
\@mtc@hints@given@false . . . . . 3127  
\@mtc@hints@given@true 3135, 3141, 3147,  
3153, 3160, 3175, 3185, 3191, 3196,  
3201, 3209, 3222, 3237, 3253, 3300,  
3309, 3317, 3325, 3335, 3343, 3351,  
3361, 3369, 3386, 3394, 3402, 3412,  
3420, 3428, 3438, 3446, 3465, 3474,  
3483, 3494, 3503, 3512, 3523, 3532,  
3541, 3550, 3558, 3566, 3580, 3586,  
3591, 3596, 3601, 3606, 3611, 3616,  
3621, 3626, 3631, 3636, 3641  
\@mtc@hints@true . . . . . 3125, 3128  
\@mtc@hints@w@false . . 3126, 3215, 3230,  
3245  
\@mtc@hints@w@true . . 3217, 3219, 3221,  
3232, 3234, 3236, 3248, 3250, 3252,  
3299  
\@mtc@hyper@used@false . . . . . 66  
\@mtc@hyper@used@true . . . . . 69  
\@mtc@insellang@false . . . . . 2947, 2977  
\@mtc@lof@used@false . . . . . 3263  
\@mtc@lof@used@true . . . . . 3270, 3279  
\@mtc@lot@used@false . . . . . 3264  
\@mtc@lot@used@true . . . . . 3273, 3282  
\@mtc@memoirLoaded@false . . . . . 14, 89  
\@mtc@memoirLoaded@true . . . . . 86  
\@mtc@memoirnew@false . . . . . 15, 92  
\@mtc@memoirnew@true . . . . . 94  
\@mtc@notoccite@false . . . . . 60  
\@mtc@notoccite@true . . . . . 2996  
\@mtc@part@def@false . . . . . 27, 28  
\@mtc@part@def@true . . . . . 28  
\@mtc@part@undef@false . . . . . 34  
\@mtc@part@undef@true . . . . . 33, 34  
\@mtc@placeinsLoaded@false . . 13, 3659  
\@mtc@placeinsLoaded@true . . . . . 3658  
\@mtc@sect@floats@false . . . . . 12  
\@mtc@sect@floats@true . . . 3286–3289  
\@mtc@section@def@false . . . . . 31, 32  
\@mtc@section@def@true . . . . . 32  
\@mtc@section@undef@false . . . . . 38  
\@mtc@section@undef@true . . . . . 37, 38  
\@mtc@sectstyLoaded@a@false . . . . . 17  
\@mtc@sectstyLoaded@a@true . . . . . 83  
\@mtc@sectstyLoaded@false . . . . . 16  
\@mtc@sectstyLoaded@true . . . . . 82  
\@mtc@setdepth@false . . . . . 2770  
\@mtc@setdepth@true . . . . . 2765, 2768  
\@mtc@setfeature@false . . . . . 2728, 2740  
\@mtc@setfeature@true . . . . . 2722, 2726  
\@mtc@setfont@false . . . . . 2366, 2379,  
2392, 2393, 2396, 2397, 2400, 2401,  
2404–2406, 2409–2411, 2414–2416  
\@mtc@setfont@true . . . . . 2361, 2364  
\@mtc@setformat@false . . . . . 2579, 2591

- \@mtc@setformat@true . . . . . 2573, 2577
  - \@mtc@setpagenumbers@false 2616, 2623, 2638
  - \@mtc@setpagenumbers@true . . . . . 2620
  - \@mtc@setrules@false . . 2659, 2666, 2682
  - \@mtc@setrules@true . . . . . 2663
  - \@mtc@settitle@false . . . . . 2466
  - \@mtc@settitle@true . . . . . 2461, 2464
  - \@mtc@settitlefont@false . . . . . 2436
  - \@mtc@settitlefont@true . . . 2431, 2434
  - \@mtc@spn@ok@false . . . . . 2635
  - \@mtc@spn@ok@true . . . . . 2633
  - \@mtc@sru@ok@false . . . . . 2679
  - \@mtc@sru@ok@true . . . . . 2677
  - \@mtc@ss@insection@false . . . . . 3000
  - \@mtc@ss@insection@true . . . . . 3005
  - \@mtc@toc@used@false . . . . . 3262
  - \@mtc@toc@used@true . . . . . 3267, 3276
  - \@mtcfixglossary@false . . . . 2927, 2929
  - \@mtcfixglossary@true . 2931, 2933, 2935
  - \@mtcfixindex@false . . . . . 2890, 2892
  - \@mtcfixindex@true . . . 2894, 2896, 2898
  - \@mtcpatchmemoir@false . . . . . 84
  - \@mtcpatchmemoir@true . . . . . 100
  - \@namedef . . 2302, 2311, 2320, 2354, 2491
  - \@nameuse . 2302, 2363, 2377, 2433, 2463, 2588, 2590, 2631, 2645, 2662, 2689, 2737, 2739, 2767
  - \@nobreakfalse . 412, 557, 629, 696, 1267, 1340, 1414, 1818, 1884, 1951
  - \@part . . . . . 170, 1426, 3215, 3668
  - \@partlof@used@false . . . . . 40
  - \@partlof@used@true . . . . . 1276
  - \@partlot@used@false . . . . . 41
  - \@partlot@used@true . . . . . 1350
  - \@parttoc@used@false . . . . . 39
  - \@parttoc@used@true . . . . . 1202
  - \@plus . . . . . 722, 735
  - \@pnumwidth . 745, 2500, 2529, 2534, 2539, 2544, 2549, 2554, 2559, 2564, 2569
  - \@resetfont . . . . . 1046, 1066
  - \@schapter . . . . . 709, 3230, 3671
  - \@secondoftwo . . . . . 111
  - \@sect . . . . . 203, 1961, 1968, 3245, 3674
  - \@sectlof@used@false . . . . . 52
  - \@sectlof@used@true . . . . . 1826
  - \@sectlot@used@false . . . . . 53
  - \@sectlot@used@true . . . . . 1892
  - \@secttoc@used@false . . . . . 51
  - \@secttoc@used@true . . . . . 1759
  - \@spart . . . . . 170, 1426, 3215, 3668
  - \@sssect . . . . . 203, 2007, 3245, 3674
  - \@svsec . . . . . 1978
  - \@svsechd . . . . . 1978
  - \@tempdima . . . . . 281, 727, 728, 740, 741
  - \@tempskipa . . . . . 1978
  - \@tocfile . . . . . 500, 505, 508, 554, 571, 576, 579, 626, 638, 643, 646, 693, 1201, 1208, 1255, 1275, 1282, 1331, 1348, 1356, 1405, 1758, 1765, 1808, 1825, 1832, 1877, 1891, 1898, 1943
  - \@tocrmarg 277, 723, 736, 2509, 2530, 2535, 2540, 2545, 2550, 2555, 2560, 2565, 2570
  - \@topnewpage . . . . . 1137, 1140, 3967
  - \@undottedtocline . . . . . 274, 548, 620, 687, 1255, 1331, 1405, 1808, 1877, 1943
  - \@width . . . . . 457
  - \@xsect . . . . . 1978
  - \ . 486, 545, 558, 616, 630, 683, 697, 1189
  - \~ . . . . . 5664, 5668, 5672
  - \\_ 4170, 4171, 4173, 4174, 4179, 4180, 4182, 4183, 4188, 4189, 4191, 4192, 4822, 4824, 4826, 4829, 4831, 4833, 4836, 4838, 4840, 5623, 5625, 5629, 5631, 5635, 5637
- A**
- abstract . . . . . **39, 52**
  - acadian.mld . . . . . **227**
  - acadien.mld . . . . . **228**
  - \accent . . . . . 4318
  - add.bib . . . . . **55**
  - add.tex . . . . . **38, 55**
  - \addcoffeeline . . . . . 2980, 3900
  - \addcontentsline 180, 207–210, 218–221, 421, 704, 1428–1430, 1961, 1968, 1978, 2890, 2927, 3873, 3913, 3976, 3979, 3983, 3986
  - \addst@rred . . . . . 421, 440, 714
  - \addstarredchapter . . . . . 714, 3913
  - \addstarredpart . . . . . 714, 3913
  - \addstarredsection . . . . 440, 714, 3913
  - \addtocontents . . 709, 1426, 2007, 2801, 2813, 2843, 2980
  - \addtocounter . . . . . 167, 451–453, 913, 975, 1030, 1096–1098, 1591, 1647, 1703, 1721–1723, 2148, 2204, 2260
  - addtotoc . . . . . **39, 52**
  - \addvspace . . . . . 11
  - \adjustmtc . . . . . 217, 444, 3695
  - \adjustptc . . . . . 180, 1086, 3695
  - \adjuststc . . . . . 206, 1716, 3695
  - afrikaan.mld . . . . . **228**

- afrikaans.mld ..... 228  
 \afterminilof ..... 384, 630, 3837  
 \afterminilot ..... 384, 697, 3837  
 \afterminitoc ..... 384, 558, 3837  
 \afterpartlof ..... 377, 1342, 3828  
 \afterpartlot ..... 377, 1416, 3828  
 \afterparttoc ..... 377, 1269, 3828  
 \aftersectlof ..... 387, 1889, 3846  
 \aftersectlot ..... 387, 1953, 3846  
 \aftersecttoc ..... 387, 1820, 1886, 3846  
 \alef .. 4825, 4827, 4832, 4834, 4839, 4841  
 \alloc@ ..... 224  
 american.mld ..... 229  
 amsart ..... 39  
 amsart.cls ..... 50  
 amsbook ..... 39  
 amsbook.cls ..... 51  
 amsproc ..... 39  
 amsproc.cls ..... 50  
 appendix ..... 39, 49  
 \appendixmtc ..... 3733  
 arab.mld ..... 229  
 \arabic . 833, 940, 995, 2095, 2801, 2813, 2843  
 arabic.mld ..... 229  
 armenian.mld ..... 230  
 article ..... 80  
 \AtBeginDocument ..... 65, 82, 413, 1080, 1708, 2499, 2813, 2843, 2984, 3130, 3215, 3664, 3860  
 \AtEndDocument .. 3130, 3285, 3664, 3861  
 austrian.mld ..... 230  
 \ayin ..... 4823, 4830, 4837
- B**
- bahasa.mld ..... 230  
 bangla.mld ..... 231  
 bangtex ..... 231  
 basque.mld ..... 231  
 \bcg ... 4113–4115, 4117–4119, 4121–4123  
 \beforeminilof ..... 371, 584, 3837  
 \beforeminilot ..... 371, 651, 3837  
 \beforeminitoc ..... 371, 513, 3837  
 \beforepartlof ..... 364, 1288, 3828  
 \beforepartlot ..... 364, 1362, 3828  
 \beforeparttoc ..... 364, 1214, 3828  
 \beforesectlof ..... 374, 1838, 3846  
 \beforesectlot ..... 374, 1904, 3846  
 \beforesecttoc ..... 374, 1771, 3846  
 below ..... 39  
 \bet ..... 4827, 4834, 4841  
 \bfseries 466, 473, 1109, 1116, 1120, 1128, 1728, 1734  
 bicig.mld ..... 232
- \bigskipamount ..... 10, 3685  
 book ..... 80, 82  
 brazil.mld ..... 232, 233  
 brazilian.mld ..... 233  
 breton.mld ..... 233  
 british.mld ..... 234  
 bulgarian.mld ..... 234  
 bulgarianb.mld ..... 235  
 buryat.mld ..... 235
- C**
- \c@lofdepth ..... 413, 1080, 1708  
 \c@lotdepth ..... 413, 1080, 1708  
 \c@minilofdepth ..... 616  
 \c@minilotdepth ..... 683  
 \c@minitocdepth ..... 545, 3685  
 \c@mtc ..... 421, 3685  
 \c@mti .. 475, 513, 584, 651, 796, 805, 814  
 \c@partlofdepth ..... 1327  
 \c@partlotdepth ..... 1401  
 \c@parttocdepth ..... 1252, 3685  
 \c@ptc ..... 421, 3685  
 \c@pti 1130, 1168, 1221, 1295, 1369, 1454, 1463, 1472  
 \c@secnumdepth .. 1978, 1991, 2000, 3955, 3973  
 \c@sectlofdepth ..... 1872  
 \c@sectlotdepth ..... 1938  
 \c@secttocdepth ..... 1805, 3685  
 \c@stc ..... 421, 3685  
 \c@sti 1735, 1773, 1841, 1907, 2027, 2036, 2045  
 \c@tocdepth .... 275, 545, 616, 683, 721, 734, 1252, 1327, 1328, 1401, 1402, 1805, 1872, 1938  
 canadian.mld ..... 236  
 canadien.mld ..... 236  
 castillan.mld ..... 236  
 castillian.mld ..... 237  
 catalan.mld ..... 237  
 catalog ..... 55  
 \centering ..... 1176  
 \cftpagenumbersoff .. 294–298, 305–308, 317–322, 324, 326, 328, 330, 332, 334  
 \cftpagenumberon ... 288–292, 300–303, 310–315, 323, 325, 327, 329, 331, 333  
 \ch@pt@c ... 3936, 3940, 3942, 3948, 3950  
 \chapbegin ..... 780  
 chapter ..... 73, 74  
 \chapter ..... 97, 187, 214, 215, 364, 377, 429, 440, 497, 568, 635, 714, 833, 1519, 2393, 2397, 2401, 2405,

- 2410, 2415, 2876, 2896, 2913, 2933,  
3215, 3230, 3671, 3855
- \chapterbegin . . . 709, 913, 975, 1030, 3708
- \chapterend . . . 709, 913, 975, 1030, 3708
- \chaptermark . . . . . 3960
- \chapternumberline . . . . . 3977, 3980
- \chardef . . . . . 224
- checkfiles . . . . . **20, 22, 46**
- chinese1.mld . . . . . **237**
- chinese1.mlo . . . . . **237**
- chinese2.mld . . . . . **238**
- chinese2.mlo . . . . . **238**
- \cleardoublepage . . . . . 368, 381
- \closeout . . . . . 163, 166, 238, 823–825,  
837, 868, 913, 944, 975, 999, 1030,  
1481–1483, 1501, 1515, 1588, 1612,  
1644, 1668, 1700, 2054–2056, 2076,  
2095, 2145, 2171, 2201, 2227, 2257
- \coffee . . . . . 865, 1519, 2079
- \coffee font . . . 474, 720, 733, 1066, 3771
- \coffeeline . . . . . 2980, 3900
- \columnwidth . . . . . 457, 531, 535,  
567, 602, 606, 669, 673, 1103, 1239,  
1243, 1312, 1316, 1386, 1390, 1725,  
1792, 1796, 1858, 1862, 1924, 1928,  
2278–2283, 2287–2292, 2296–2301
- \contentsline . . . 180, 206, 217, 780, 913,  
975, 1030, 1046, 1580, 1637, 1693,  
2138, 2194, 2250
- \copy . . . . . 229, 233
- \count@ . . . . . 3685
- croatian.mld . . . . . **238**
- \cyr . . . 4168, 4170, 4173, 4177, 4179, 4182,  
4186, 4188, 4191, 4199, 4201, 4203,  
4206, 4208, 4210, 4213, 4215, 4217,  
5384, 5386, 5394, 5396, 5404, 5406,  
5483, 5485, 5489, 5491, 5492, 5495,  
5497, 5621, 5622, 5624, 5627, 5628,  
5630, 5633, 5634, 5636
- \CYRa . . . 5384, 5387, 5390, 5394, 5397, 5400,  
5404, 5407, 5410
- \cyra . . . 4168, 4171, 4174, 4177, 4180, 4183,  
4186, 4189, 4192, 4199, 4203, 4206,  
4210, 4213, 4217, 5483, 5486, 5489,  
5492, 5495, 5498, 5623, 5625, 5629,  
5631, 5635, 5637
- \CYRb . . . . . 5390, 5400, 5410
- \cyrb . . . 4174, 4183, 4192, 4203, 4210, 4217,  
5486, 5492, 5498, 5625, 5631, 5637
- \CYRc . . . 5387, 5390, 5397, 5400, 5407, 5410
- \cyrc . . . 4174, 4183, 4192, 4203, 4210, 4217,  
5486, 5492, 5498, 5623, 5625, 5629,  
5631, 5635, 5637
- \cyrd . . . 4168, 4177, 4186, 4199, 4206, 4213,  
5483, 5489, 5495
- \CYRe . . . . . 5384, 5394, 5404
- \cyre . . . 4168, 4171, 4174, 4177, 4180, 4183,  
4186, 4189, 4192, 4199, 4206, 4213,  
5485, 5486, 5491, 5492, 5497, 5498,  
5622, 5624, 5628, 5630, 5634, 5636
- \CYRF . . . . . 4201, 4208, 4215
- \cyrf . . . . . 4171, 4180, 4189
- \CYRg . . . . . 5384, 5394, 5404
- \cyrg . . . 4171, 4180, 4189, 4201, 4208, 4215
- \cyrhdsn . . . 4168, 4170, 4173, 4177, 4179,  
4182, 4186, 4188, 4191, 4199, 4206,  
4213
- \CYRi . . . 5384, 5386, 5387, 5389, 5390, 5394,  
5396, 5397, 5399, 5400, 5404, 5406,  
5407, 5409, 5410
- \cyri . . . 4168, 4170, 4171, 4173, 4174, 4177,  
4179, 4180, 4182, 4183, 4186, 4188,  
4189, 4191, 4192, 4199, 4201, 4203,  
4206, 4208, 4210, 4213, 4215, 4217,  
5485, 5486, 5491, 5492, 5497, 5498,  
5625, 5631, 5637
- \cyrii . . . 5621–5624, 5627–5630, 5633–5636
- \CYRishrt . . . . . 5387, 5397, 5407
- \cyrishrt . . . . . 5623, 5629, 5635
- \cyrje . . . . . 5483, 5489, 5495
- \CYRk . . . 5386, 5389, 5396, 5399, 5406, 5409
- \cyrk . . . 4170, 4173, 4179, 4182, 4188, 4191,  
5485, 5491, 5497, 5622, 5624, 5628,  
5630, 5634, 5636
- \CYRl . . . 5384, 5387, 5390, 5394, 5397, 5400,  
5404, 5407, 5410
- \cyr1 . . . 4174, 4183, 4192, 4203, 4210, 4217,  
5485, 5486, 5491, 5492, 5497, 5498,  
5622–5625, 5628–5631, 5634–5637
- \cyrm . . . . . 5621, 5627, 5633
- \CYRn . . . . . 5384, 5394, 5404
- \cyrn . . . 4168, 4171, 4174, 4177, 4180, 4183,  
4186, 4189, 4192, 4199, 4206, 4213
- \CYRO . . . . . 5384, 5394, 5404
- \CYRo . . . 5386, 5389, 5396, 5399, 5406, 5409
- \CYRp . . . 5622, 5624, 5628, 5630, 5634, 5636
- \CYRp . . . 5386, 5389, 5396, 5399, 5406, 5409
- \cyrp . . . 4170, 4173, 4179, 4182, 4188, 4191
- \CYRr . . . . . 5387, 5397, 5407
- \cyrr . . . . . 4168, 4171, 4177, 4180, 4186,  
4189, 4199, 4201, 4206, 4208, 4213,  
4215, 5483, 5489, 5495, 5622–5624,  
5628–5630, 5634–5636
- \CYRS . . . 4168, 4170, 4173, 4177, 4179, 4182,  
4186, 4188, 4191, 4199, 4206, 4213,  
5386, 5389, 5396, 5399, 5406, 5409,  
5483, 5485, 5489, 5491, 5495, 5497

\CYRs . . 5386, 5387, 5389, 5396, 5397, 5399,  
           5406, 5407, 5409  
 \cyrS . . 4170, 4173, 4179, 4182, 4188, 4191,  
           5621, 5623, 5627, 5629, 5633, 5635  
 \cyrSftsn . . . . . 5625, 5631, 5637  
 \CYRT . . 4203, 4210, 4217, 5486, 5492, 5498  
 \CYRt . . 5387, 5390, 5397, 5400, 5407, 5410  
 \cyrT . . 4171, 4174, 4180, 4183, 4189, 4192,  
           5621, 5623, 5625, 5627, 5629, 5631,  
           5633, 5635, 5637  
 \cyrU . . 4171, 4180, 4189, 4201, 4208, 4215  
 \CYRv . . . . . 5384, 5394, 5404  
 \CYRyu . . . . . 5387, 5397, 5407  
 \cyrYu . . . . . 5623, 5629, 5635  
 \CYRZ . . . . . 5621, 5627, 5633  
 \cyrzh . 4168, 4177, 4186, 4199, 4206, 4213,  
           5483, 5489, 5495  
 \cz . . . 5287, 5288, 5291, 5292, 5295, 5296  
 czech.mld . . . . . **239**

**D**

danish.mld . . . . . **239**  
 \DeclareOption . 2988, 2992, 2994, 2996,  
           2997, 3000, 3008, 3128  
 \decrementmtc . . . . . 444, 3695  
 \decrementptc . . . . . 1086, 3695  
 \decrementstc . . . . . 1716, 3695  
 \df@mtic . . . . . 513, 584, 651  
 \df@mtilf . . . . . 475, 480, 591, 805  
 \df@mtilt . . . . . 475, 480, 658, 814  
 \df@mtitc . . . . . 475, 480, 520, 796  
 \df@pti . . . . . 1221, 1295, 1369  
 \df@ptilf . . . . . 1183, 1301, 1463  
 \df@ptilt . . . . . 1183, 1375, 1472  
 \df@ptitc . . . . . 1183, 1227, 1454  
 \df@sti . . . . . 1773, 1841, 1907  
 \df@stilf . . . . . 1740, 1847, 2036  
 \df@stilt . . . . . 1740, 1913, 2045  
 \df@stitc . . . . . 1740, 1780, 2027  
 \dimen@ . . . . . 3685  
 \do@mtilf 475, 480, 586–591, 593, 603, 607  
 \do@mtilt 475, 480, 653–658, 660, 670, 674  
 \do@mtitc . . . . . 475, 480, 513, 584, 651  
 \do@ptilf . . . . . 1183, 1295, 1307  
 \do@ptilt . . . . . 1183, 1369, 1385  
 \do@ptitc . . . . . 1183, 1221, 1238  
 \do@stilf . . . . . 1740, 1841, 1855  
 \do@stilt . . . . . 1740, 1907, 1921  
 \do@stitc . . . . . 1740, 1773, 1788  
 \dominilof 568, 780, 785, 3342, 3418, 3721  
 \dominilof@ . . . . . 789, 805  
 \dominilot 635, 780, 790, 3350, 3426, 3721  
 \dominilot@ . . . . . 795, 814  
 \dominitoc . . . 497, 780, 3334, 3410, 3721

\dominitoc@ . . . . . 780, 784, 796  
 \dopartlof . 1439, 1451, 3316, 3392, 3715  
 \dopartlof@ . . . . . 1452, 1463  
 \dopartlot . 1439, 1451, 3324, 3400, 3715  
 \dopartlot@ . . . . . 1453, 1472  
 \doparttoc . 1439, 1451, 3308, 3384, 3715  
 \doparttoc@ . . . . . 1451, 1454  
 \dosectlof . . 12, 2016, 2024, 3295, 3368,  
           3444, 3727  
 \dosectlof@ . . . . . 2025, 2036  
 \dosectlot . . 12, 2020, 2024, 3296, 3376,  
           3452, 3727  
 \dosectlot@ . . . . . 2026, 2045  
 \dosecttoc . 2012, 2024, 3360, 3436, 3727  
 \dosecttoc@ . . . . . 2024, 2027  
 dutch.mld . . . . . **240**

**E**

\e@mti . . 475, 513, 584, 651, 796, 805, 814  
 \e@pti 1130, 1137, 1221, 1295, 1369, 1454,  
           1463, 1472  
 \e@sti 1735, 1773, 1841, 1907, 2027, 2036,  
           2045  
 emk . . . . . **55**  
 \empty . . . . . 365–367, 371–376,  
           378–380, 384–389, 393–398, 3744,  
           3773, 3776, 3777, 3828–3854, 3878,  
           3887, 3896, 3906, 3908, 3910  
 \endlist . . . . . 496, 1199, 1756  
 \endmtc@verse . . . . . 496  
 \endptc@verse . . . . . 1189  
 \endstc@verse . . . . . 1756  
 english.mld . . . **229, 234, 236, 240, 280, 281**  
 filecontents . . . . . 3991  
 mtc@verse . . 486, 539, 558, 610, 630, 677,  
           697, 2533, 2548, 2563  
 mtchideinmainlof . . . . . 2813, 3905  
 mtchideinmainlot . . . . . 2843, 3905  
 mtchideinmaintoc . . . . . 2801, 3905  
 ptc@verse . 1189, 1252, 1255, 1326, 1331,  
           1400, 1405, 2528, 2543, 2558  
 samepage . . . . . 513, 558, 584, 630, 651,  
           697, 1221, 1269, 1295, 1342, 1369,  
           1416, 1773, 1820, 1841, 1886, 1907,  
           1953  
 stc@verse . 1746, 1805, 1808, 1872, 1877,  
           1938, 1943, 2538, 2553, 2568  
 tabular . 513, 584, 651, 1238, 1307, 1385,  
           1788, 1855, 1921  
 esperant.mld . . . . . **241**  
 esperanto.mld . . . . . **241**  
 estonian.mld . . . . . **241**



\eth@doaltchar . . . 4415, 4416, 4418, 4419,  
4423, 4424, 4426, 4427, 4431, 4432,  
4434, 4435  
ethiopia.mld . . . . . **242**  
ethopia.mld . . . . . **243**  
ethopian.mld . . . . . **243**  
\ExecuteOptions . . . . . 3653  
\ExecuteOptions{english} . . . . . 3653

**F**

F . . . . . **98**  
\f@rhdr . . . . . 3944, 3950, 3952, 3960, 3988  
\f@rtoc . . . . . 3943, 3948, 3977, 3980, 3983, 3986,  
3988  
\fake@starttoc . . . . . 399  
\fakelistoffigures 399, 3274, 3473, 3502,  
3531, 3695  
\fakelistoftables 399, 3274, 3482, 3511,  
3540, 3695  
\faketableofcontents . . 399, 3274, 3464,  
3493, 3522, 3695  
farsi1.mld . . . . . **243**  
farsi1.mlo . . . . . **243**  
farsi2.mld . . . . . **243**  
farsi2.mlo . . . . . **243**  
\figure . . . . . 956, 1615, 2174  
filecontents (environment) . . . . . 3991  
\finalmem . . . . . 4823, 4825, 4830, 4832, 4837,  
4839  
\finalnun . . . . . 4822, 4829, 4836  
finnish.mld . . . . . **244**  
finnish2.mld . . . . . **244**  
\firstchapteris . . . . . 444, 3635, 3787  
\firstpartis . . . . . 1086, 3630, 3787  
\firstsectionis . . . . . 1716, 3640, 3787  
fminitoc.bib . . . . . **55**  
fminitoc.dtx . . . . . **54, 55**  
fminitoc.ist . . . . . **55**  
fminitoc.lan . . . . . **55**  
fminitoc.pdf . . . . . **55**  
fminitoc.ps . . . . . **55**  
fmk . . . . . **55**  
franc.sty . . . . . **55**  
francais.mld . . . . . **245**  
frbib.sty . . . . . **55**  
french.mld . . . . . **236, 245, 246**  
frenchb.mld . . . . . **245**  
frenchle.mld . . . . . **246**  
frenchpro.mld . . . . . **246**  
frnew.sty . . . . . **55**  
frplain1.bst . . . . . **55**

**G**

galician.mld . . . . . **246**

german.mld . . . . . **230, 247**  
germanb.mld . . . . . **247**  
\gobbleopt@ . . . . . 3714, 3715–3732  
greek-mono.mld . . . . . **248**  
greek-polydemo.mld . . . . . **249**  
greek-polykatha.mld . . . . . **250**  
greek.mld . . . . . **248**  
guarani.mld . . . . . **251**

**H**

hangul1.mld . . . . . **252**  
hangul2.mld . . . . . **252**  
hangul2.mlo . . . . . **252**  
hangul3.mld . . . . . **253**  
hangul3.mlo . . . . . **253**  
hangul4.mld . . . . . **253**  
hanja1.mld . . . . . **254**  
hanja1.mlo . . . . . **254**  
hanja2.mld . . . . . **255**  
hanja2.mlo . . . . . **255**  
\hb@xt@ . . . . . 745  
hebrew.mld . . . . . **255**  
hints . . . . . **38, 43, 74, 208**  
\hline . . . . . 536, 607, 674, 1238, 1307, 1385,  
1788, 1855, 1921  
\hrule . . . . . 457, 2278–2280, 2287–2289,  
2296–2298  
hungarian.mld . . . . . **256**  
hyperref . . . . . **39, 48, 75**

**I**

icelandic.mld . . . . . **256**  
\if@dominilof@used@ 45, 805, 3329, 3406  
\if@dominilot@used@ 45, 814, 3329, 3406  
\if@dominitoc@used@ 45, 780, 796, 3329,  
3406  
\if@dopartlof@used@ 39, 1463, 3303, 3380  
\if@dopartlot@used@ 39, 1472, 3303, 3380  
\if@doparttoc@used@ 39, 1454, 3303, 3380  
\if@dosectlof@used@ . . . 51, 2036, 3286,  
3355, 3432  
\if@dosectlot@used@ . . . 51, 2045, 3286,  
3355, 3432  
\if@dosecttoc@used@ 51, 2027, 3355, 3432  
\if@files@ . . . . . 408  
\if@firstchapteris@used@ 57, 444, 3633  
\if@firstpartis@used@ . . 57, 1086, 3628  
\if@firstsectionis@used@ . . . 57, 3638  
\if@longextensions . . . . . 500  
\if@longextensions@ 238, 500, 571, 638,  
837, 868, 944, 999, 1201, 1275,  
1348, 1492, 1506, 1603, 1659, 1758,  
1825, 1891, 2067, 2095, 2162, 2218,  
2997, 3545, 3855

- \if@mainmatter . . . . . 3956, 3974
- \if@minilof@used@ . 45, 569, 3329, 3406, 3487
- \if@minilot@used@ . 45, 636, 3329, 3406, 3487
- \if@minitoc@used@ . 45, 498, 3329, 3406, 3487
- \if@mtc@abstract@loaded@ . . . . . 3206
- \if@mtc@chapter@def@ . . . 29, 3329, 3406, 3487, 3545
- \if@mtc@chapter@undef@ . 35, 1714, 2265
- \if@mtc@checkfiles . . . . . 133, 2992
- \if@mtc@empty@minilof@ . . 18, 576, 3603
- \if@mtc@empty@minilot@ . . 23, 643, 3608
- \if@mtc@empty@minitoc@ . . 18, 505, 3598
- \if@mtc@empty@partlof@ . . . . 18, 3588
- \if@mtc@empty@partlot@ . . . . 18, 3593
- \if@mtc@empty@parttoc@ . . . . 18, 3583
- \if@mtc@empty@sectlof@ . . . . 18, 3618
- \if@mtc@empty@sectlot@ . . . . 18, 3623
- \if@mtc@empty@secttoc@ . . . . 24, 3613
- \if@mtc@FE . 133, 505, 513, 576, 584, 643, 651, 1208, 1221, 1282, 1295, 1356, 1369, 1765, 1773, 1832, 1841, 1898, 1907
- \if@mtc@Himlof@ . . . . . 2813
- \if@mtc@Himlot@ . . . . . 2843
- \if@mtc@Himtoc@ . . . . . 2800, 2801
- \if@mtc@hints@ . . 3125, 3128, 3261, 3664
- \if@mtc@hints@giben@ . . . . . 3125
- \if@mtc@hints@given@ . 3127, 3134, 3140, 3146, 3152, 3157, 3184, 3190, 3200, 3206, 3215, 3230, 3245, 3291, 3303, 3329, 3355, 3380, 3406, 3432, 3458, 3487, 3516, 3575, 3643
- \if@mtc@hints@given@true . . . . . 3545
- \if@mtc@hints@w@ 3125, 3215, 3230, 3245, 3291
- \if@mtc@hyper@used@ 65, 3215, 3230, 3245
- \if@mtc@insellang@ . . . . . 2947, 2960
- \if@mtc@LI . . . . . 133
- \if@mtc@lof@used@ . . . 3261, 3265, 3274, 3469, 3498, 3527
- \if@mtc@lot@used@ . . . 3261, 3265, 3274, 3478, 3507, 3536
- \if@mtc@memoirLoaded@ . . . . 13, 84, 286
- \if@mtc@memoirnew@ . . . . . 13, 84
- \if@mtc@notoccite@ . . . . 60, 2996, 3661
- \if@mtc@part@def@ 27, 3303, 3380, 3458, 3545
- \if@mtc@part@undef@ . . . . . 33
- \if@mtc@placeinsLoaded@ 13, 3291, 3655
- \if@mtc@sect@floats@ . . . 12, 3286, 3291
- \if@mtc@section@def@ . . . 31, 1714, 2264, 3286, 3355, 3432, 3516, 3545
- \if@mtc@section@undef@ . . . . . 37
- \if@mtc@sectstyLoaded@ . . . 13, 82, 3575
- \if@mtc@sectstyLoaded@a@ . 13, 82, 3575
- \if@mtc@setdepth@ . . . 2765, 2767, 2781
- \if@mtc@setfeature@ . 2722, 2739, 2753
- \if@mtc@setfont@ 2361, 2363, 2377, 2391, 2418
- \if@mtc@setformat@ . . . . . 2573, 2603
- \if@mtc@setpagenumbers@ . . 2616, 2633
- \if@mtc@setrules@ . . . . . 2659, 2662
- \if@mtc@settitle@ . . . 2461, 2463, 2477
- \if@mtc@settitlefont@ 2431, 2433, 2447
- \if@mtc@spn@ok@ . . . . . 2616, 2633
- \if@mtc@sru@ok@ . . . . . 2659, 2677
- \if@mtc@ss@insection@ 3000, 3157, 3655
- \if@mtc@toc@used@ . . . 3261, 3265, 3274, 3460, 3489, 3518
- \if@mtcfixglossary@ . . . . . 2927
- \if@mtcfixindex@ . . . . . 2890
- \if@mtcpatchmemoir@ . . . . . 84, 102
- \if@partlof@used@ 39, 1275, 3303, 3380, 3458
- \if@partlot@used@ 39, 1348, 3303, 3380, 3458
- \if@parttoc@used@ 39, 1201, 3303, 3380, 3458
- \if@sectlof@used@ 51, 1825, 3286, 3355, 3432, 3516
- \if@sectlot@used@ 51, 1891, 3286, 3355, 3432, 3516
- \if@secttoc@used@ 51, 1758, 3355, 3432, 3516
- \if@setrules@false . . . . . 2677
- \if@twocolumn . . . . . 1137, 1140
- \ifanappendix . . . . . 3975
- \ifartopt . . . . . 3961
- \ifdim . . . . . 1985
- \ifFileExists . . . . . 150, 2947, 2960
- \ifheadnameref . . . . . 3988
- \ifkthightmtc . . 63, 486, 1189, 1746, 3855
- ifmtarg . . . . . 77
- \ifmtc@lof@used@ . . . . 3458, 3487, 3516
- \ifmtc@lot@used@ . . . . 3458, 3487, 3516
- \ifmtc@toc@used@ . . . . 3458, 3487, 3516
- \ifmtcoffwarn@ . . . . . 3860, 3873, 3913
- \iftightmtc . . 63, 486, 1189, 1746, 2988, 3855
- \ifundottedmtc 748, 759, 769, 2994, 3855
- \ignorespaces . . . . . 704
- imk . . . . . 55
- \in . . . . . 5603, 5607, 5611
- \incrementmtc . . . . . 444, 3695

- \incrementptc ..... 1086, 3695
- \incrementstc ..... 1716, 3695
- \input ..... 238, 3678
- \InputIfFileExists .... 103, 2949, 2962
- insection ..... **20, 39**
- \insertchapterspace ..... 3965
- INSTALL ..... **54**
- \interlinepenalty .. 279, 725, 738, 1988
- interlingua.mld ..... **257**
- irish.mld ..... **257**
- italian.mld ..... **257**
- \itemindent 486, 542, 613, 680, 1189, 1250,  
1324, 1398, 1748, 1749, 1803, 1870,  
1936
- \itemsep ..... 486, 1189, 1747
- J**
- japanese.mld ..... **258**
- japanese.mlo ..... **258**
- japanese2.mld ..... **258**
- japanese2.mlo ..... **258**
- japanese3.mld ..... **259**
- japanese3.mlo ..... **259**
- japanese4.mld ..... **260**
- japanese4.mlo ..... **260**
- japanese5.mld ..... **260**
- japanese5.mlo ..... **260**
- \jobname ..... 238, 3678
- K**
- k-loose ..... **29**
- k-tight ..... **29, 47, 75, 122**
- \kaf ..... 4822, 4829, 4836
- \kiftightmtc ..... 2988
- \ktightmtcfalse ..... 64, 2991, 3857
- \ktightmtctrue ..... 2990
- L**
- \l@chapter .. 217, 702, 1260, 1261, 2267,  
2269, 3781
- \l@chapter@SVPN ..... 1260, 1261
- \l@coffee ..... 2980, 3902
- \l@listof ..... 2266, 3855
- \l@mti .. 475, 480, 513, 584, 651, 796, 805,  
814
- \l@paragraph ..... 2273, 3785
- \l@part ..... 180, 2268, 3780
- \l@pchapter ..... 1421
- \l@psection ..... 1421
- \l@pti 1130, 1140, 1183, 1221, 1295, 1369,  
1454, 1463, 1472
- \l@schapter ..... 1957
- \l@section ..... 206, 2266, 2270, 3782
- \l@starchapter ..... 2266, 3780
- \l@starparagraph ..... 2266, 3780
- \l@starpast ..... 2266, 3780
- \l@starsection ..... 2266, 3780
- \l@starsubparagraph ..... 2266, 3780
- \l@starsubsection ..... 2266, 3780
- \l@starsubsubsection ..... 2266, 3780
- \l@sti 1735, 1740, 1773, 1841, 1907, 2027,  
2036, 2045
- \l@subparagraph ..... 2274, 3786
- \l@subsection ..... 2271, 3783
- \l@subsubsection ..... 2272, 3784
- \l@xchapter ..... 217, 702, 3733
- \l@xpart ..... 180, 1421, 3733
- \l@xssect ..... 1957, 3733
- \l@xsection ..... 206
- \labelsep 543, 614, 681, 1251, 1325, 1399,  
1804, 1871, 1937
- \labelwidth ... 542, 613, 680, 1250, 1324,  
1398, 1803, 1870, 1936
- \lamed ..... 4827, 4834, 4841
- \LARGE ..... 1128
- \Large ..... 1116
- \large ..... 473, 1734
- latin.mld ..... **261**
- latin2.mld ..... **261**
- latvian.mld ..... **262**
- \leftmargin 494, 495, 540, 611, 678, 1197,  
1198, 1248, 1322, 1396, 1754, 1755,  
1801, 1868, 1934
- letton.mld ..... **262**
- \list ..... 486, 1189, 1747
- \listoffigures ... 197, 3265, 3472, 3501,  
3530
- \listoftables 197, 3265, 3481, 3510, 3539
- \listparindent . 486, 543, 614, 681, 1189,  
1251, 1325, 1399, 1749, 1804, 1871,  
1937
- lithuanian.mld ..... **262**
- \localgreek 4591, 4594, 4598, 4602, 4605,  
4609, 4613, 4616, 4620, 4631, 4634,  
4638, 4642, 4645, 4649, 4653, 4656,  
4660, 4671, 4674, 4678, 4682, 4685,  
4689, 4693, 4696, 4700
- \locallatvian ... 5017–5019, 5021–5023,  
5025–5027
- \localpolish ... 5237–5239, 5241–5243,  
5245–5247
- \localrussian ... 5307, 5310, 5314, 5318,  
5321, 5325, 5329, 5332, 5336, 5347,  
5350, 5354, 5358, 5361, 5365, 5369,  
5372, 5376
- \localspanish ... 5564–5566, 5568–5570,  
5572–5574
- lofdepth ..... **22, 91**

- loose ..... 29  
lotdepth ..... 22, 91  
lsorbian.mld ..... 263
- M**
- M ..... 95  
\M@getttitle ..... 3988  
magyar.mld ..... 256, 263  
magyar2.mld ..... 264  
malayalam-keli.mld ..... 264  
malayalam-rachana.mld ..... 265  
malayalam-rachana2.mld ..... 265  
\mdseries 465, 474, 1108, 1119, 1121, 1727  
\mem ... 4824, 4826, 4831, 4833, 4838, 4840  
memoir ..... 39, 50, 73, 76, 86  
\MessageBreak ..... 96–98, 193, 252,  
259, 267, 271, 902, 965, 1020, 1567,  
1624, 1680, 2127, 2183, 2239, 2370,  
2372, 2383, 2385, 2427, 2440, 2442,  
2457, 2470, 2472, 2487, 2583, 2585,  
2595, 2598, 2612, 2628, 2642, 2672,  
2686, 2732, 2734, 2745, 2747, 2762,  
2774, 2776, 2785, 2787, 2797, 2877,  
2914, 2951, 2953, 2954, 2956, 2957,  
2964, 2966, 2967, 2969, 2970, 2974,  
3132, 3137, 3138, 3143, 3149, 3155,  
3163, 3165, 3167, 3169, 3178, 3180,  
3187, 3193, 3198, 3203, 3211, 3212,  
3295, 3297, 3462, 3471, 3480, 3491,  
3500, 3509, 3520, 3529, 3538, 3578,  
3645, 3649, 3864, 3866, 3868, 3870,  
3875, 3876, 3884, 3885, 3893, 3894,  
3915, 3916, 3920, 3921, 3925, 3926  
mini-art.tex ..... 55  
\minilof 568, 2395, 3341, 3419, 3500, 3721  
\minilof@ ..... 568, 569, 748  
minilofdepth ..... 91, 100  
\minilot 635, 2399, 3349, 3427, 3509, 3721  
\minilot@ ..... 635, 636, 748  
minilotdepth ..... 102  
\minitoc 497, 2391, 3333, 3411, 3491, 3721  
minitoc-ex.tex ..... 55  
minitoc.bib ..... 55  
minitoc.bug ..... 38, 43, 55  
minitoc.dtx ..... 54, 55  
minitoc.ins ..... 31, 54, 55  
minitoc.ist ..... 55  
minitoc.l ..... 54  
minitoc.lan ..... 55  
minitoc.pdf ..... 55  
minitoc.ps ..... 55  
minitoc.sty ..... 54  
minitoc.sum ..... 55  
\minitoc@ ..... 497, 498, 748  
minitocdepth ..... 22, 92, 96  
minutes ..... 93  
mlf ..... 98  
\MLF@ ..... 973, 975  
\MLF@contentsline ..... 940, 975  
\MLF@explist ..... 938, 961  
\MLF@line ..... 972, 973  
\MLF@list ..... 935, 938, 961, 975  
\MLF@lof ..... 938, 962, 975  
\MLF@loop ..... 935, 962  
\MLF@next ..... 785, 935, 961  
\mlf@pgno ..... 342, 343  
\MLF@read ..... 962, 971, 975  
\mlf@rule . 444, 601, 632, 2285, 2288, 2291  
\mlf@setform ..... 620, 2548  
\MLF@test ..... 973, 975  
\mlf@dotsep ..... 2518, 2551  
\mlffont ..... 465, 584, 3754  
\mlflofrmarg ..... 2509  
\mlfname ..... 944  
\mlfpagenumbers 323, 336, 342, 355, 3810  
\mlfpnumwidth ..... 2500, 2549  
\mlfrule ..... 2275, 3792  
\mlftitle . 583, 593, 603, 607, 3991, 4007,  
4032, 4051, 4070, 4086, 4102, 4118,  
4133, 4152, 4178, 4207, 4228, 4261,  
4307, 4324, 4339, 4354, 4369, 4384,  
4403, 4422, 4468, 4483, 4503, 4531,  
4546, 4561, 4577, 4605, 4609, 4645,  
4685, 4715, 4831, 4856, 4870, 4885,  
4900, 4988, 5003, 5022, 5041, 5056,  
5071, 5086, 5103, 5120, 5137, 5153,  
5176, 5192, 5204, 5223, 5242, 5261,  
5276, 5291, 5321, 5361, 5395, 5423,  
5438, 5453, 5468, 5490, 5508, 5523,  
5538, 5553, 5569, 5584, 5606, 5622,  
5652, 5668, 5687  
\mlftocrmarg ..... 2513, 2550  
mlt ..... 101  
\MLT@ ..... 1028, 1030  
\MLT@contentsline ..... 995, 1030  
\MLT@explist ..... 993, 1016  
\MLT@line ..... 1027, 1028  
\MLT@list ..... 990, 993, 1016, 1030  
\MLT@loop ..... 990, 1017  
\MLT@lot ..... 993, 1017, 1030  
\MLT@next ..... 790, 990, 1016  
\mlt@pgno ..... 348, 349  
\MLT@read ..... 1017, 1026, 1030  
\mlt@rule . 444, 668, 699, 2294, 2297, 2300  
\mlt@setform ..... 687, 2563  
\MLT@test ..... 1028, 1030  
\mltdotsep ..... 2518  
\mltfont ..... 465, 651, 3754

- \mltloformarg ..... 2509
- \mltname ..... 999
- \mltpagenumbers 329, 336, 342, 348, 355,  
3810
- \mltpnumwidth ..... 2500
- \mltrule ..... 2275, 3792
- \mltttitle . 650, 660, 670, 674, 3991, 4008,  
4033, 4052, 4071, 4087, 4103, 4119,  
4134, 4153, 4181, 4209, 4229, 4262,  
4308, 4325, 4340, 4355, 4370, 4385,  
4404, 4425, 4469, 4484, 4504, 4532,  
4547, 4562, 4578, 4649, 4689, 4716,  
4833, 4857, 4871, 4886, 4901, 4989,  
5004, 5023, 5042, 5057, 5072, 5087,  
5104, 5121, 5138, 5154, 5177, 5193,  
5205, 5224, 5243, 5262, 5277, 5292,  
5325, 5365, 5398, 5424, 5439, 5454,  
5469, 5492, 5509, 5524, 5539, 5554,  
5570, 5585, 5607, 5624, 5653, 5669,  
5688
- \mlttocrmarg ..... 2516
- \mmm ... 5098–5100, 5102–5104, 5106–5108,  
5115–5117, 5119–5121, 5123–5125,  
5132–5134, 5136–5138, 5140–5142
- \mmr ... 4223–4225, 4227–4229, 4231–4233,  
5148–5150, 5152–5154, 5156–5158
- mongol.mld ..... **266**
- mtc ..... **92, 95**
- \MTC@ ..... 911, 915
- \mtc@While ..... 119, 121, 126
- \mtc@arg@dotinterval ..... 2492
- \mtc@arg@numwidth ..... 2498
- \mtc@arg@pagenumwidth ..... 2496
- \mtc@arg@tocrightmargin ..... 2494
- \mtc@BBR ... 224, 539, 545, 610, 616, 677,  
683, 1247, 1252, 1321, 1326, 1395,  
1400, 1800, 1805, 1867, 1872, 1933,  
1938
- \mtc@Body ..... 133
- \mtc@bottom@rule ... 558, 564, 630, 697
- \mtc@ck ..... 3873
- \mtc@CkFile ... 133, 505, 576, 643, 1208,  
1282, 1356, 1765, 1832, 1898
- \mtc@CkStr 133, 513, 584, 651, 1221, 1295,  
1369, 1773, 1841, 1907
- \MTC@contentsline ..... 833, 913
- \mtc@dot ..... 1046
- \mtc@EndWhile ..... 108, 133
- \MTC@explist ..... 830, 898
- \mtc@featparam@abbrev 2724, 2739, 2753
- \mtc@featparam@third ..... 2739, 2753
- \mtc@featureparam@after ..... 2715
- \mtc@featureparam@before ..... 2713
- \mtc@featureparam@pagestyle ... 2717
- \mtc@fparam@abbrev .. 2575, 2590, 2603
- \mtc@glofix@level 2910, 2928, 2937–2939,  
2943
- \mtc@hints@@chapter ..... 3230, 3671
- \mtc@hints@@part ..... 3215, 3668
- \mtc@hints@@schapter ..... 3230, 3671
- \mtc@hints@@sect ..... 3245, 3674
- \mtc@hints@@spart ..... 3215, 3668
- \mtc@hints@@sssect ..... 3245, 3674
- \mtc@hints@begindoc . 3130, 3284, 3664
- \mtc@hints@chapter .. 3215, 3230, 3671
- \mtc@hints@checklongext ..... 3545
- \mtc@hints@enddoc 3130, 3285, 3643, 3664
- \mtc@hints@part ..... 3215, 3668
- \mtc@hints@section .. 3215, 3245, 3674
- \mtc@hook@beforeinputfile ... 60, 548,  
620, 687, 1255, 1331, 1405, 1808,  
1877, 1943
- \mtc@hstrut . 224, 1318, 1392, 1863, 1929
- \mtc@hstrutbox ..... 224
- \mtc@ifmtarg .... 108, 143, 180, 206, 217
- \mtc@ixfix@level 2873, 2891, 2900–2902,  
2906
- \mtc@level@abbrev ... 2377, 2390, 2418
- \MTC@line ..... 909, 911
- \MTC@list ..... 826, 830, 898, 913
- \MTC@loop ..... 826, 899
- \mtc@ltypetable@minilof ..... 2708
- \mtc@ltypetable@minilot ..... 2707
- \mtc@ltypetable@minitoc ..... 2706
- \mtc@ltypetable@partlof ..... 2705
- \mtc@ltypetable@partlot ..... 2704
- \mtc@ltypetable@parttoc ..... 2703
- \mtc@ltypetable@sectlof ..... 2710
- \mtc@ltypetable@sectlot ..... 2711
- \mtc@ltypetable@secttoc ..... 2709
- \mtc@markboth .. 419, 512, 583, 650, 1215,  
1289, 1363, 1773, 1839, 1905
- \mtc@mta@abbrev ..... 2363, 2390, 2418
- \mtc@mtade@abbrev ..... 2767, 2781
- \mtc@mtatf@abbrev ..... 2433, 2447
- \mtc@mtati@abbrev ..... 2463, 2477
- \mtc@mtf@abbrev ..... 2575, 2603
- \mtc@mtfeat@abbrev ..... 2724, 2753
- \mtc@mttpn@abbrev ..... 2619, 2633
- \mtc@mttru@abbrev ..... 2662, 2677
- \MTC@next ..... 780, 826, 898
- \mtc@noX ..... 2647, 2649, 2677
- \mtc@onebackchap ..... 168
- \mtc@onebackchapter ..... 167
- \mtc@onebackpart ..... 167
- \mtc@onebacksect ..... 169, 197
- \mtc@onebacksection ..... 167
- \mtc@param ..... 1046

- \mtc@pgno ..... 336, 337, 548, 620, 687
- \mtc@pns@abbrev ..... 2619, 2633
- \mtc@pns@abbrevX ..... 2646
- \MTC@read ..... 899, 909, 913
- \mtc@Rline ..... 133
- \mtc@rule . 444, 513, 564, 584, 651, 2276,  
2279, 2282
- \mtc@rusw@abbrev ..... 2662, 2677
- \mtc@rusw@abbrevX ..... 2690
- \mtc@schapter ..... 709
- \mtc@sectlevel@chapter ..... 2354
- \mtc@sectlevel@paragraph ..... 2358
- \mtc@sectlevel@section ..... 2355
- \mtc@sectlevel@subparagraph ... 2359
- \mtc@sectlevel@subsection ..... 2356
- \mtc@sectlevel@subsubsection ... 2357
- \mtc@setform ..... 548, 2533
- \mtc@string ..... 224, 913,  
975, 1030, 1046, 1066, 1579, 1636,  
1692, 2137, 2193, 2249, 3733
- \mtc@strut 224, 548, 620, 687, 1255, 1331,  
1405, 1808, 1877, 1943
- \mtc@strutbox ..... 224
- \mtc@sv@fakelistoffigures ..... 3274
- \mtc@sv@fakelistoftables ..... 3274
- \mtc@sv@faketableofcontents ... 3274
- \mtc@sv@listoffigures ..... 3265
- \mtc@sv@listoftables ..... 3265
- \mtc@sv@lofdepth ..... 2813
- \mtc@sv@lotdepth ..... 2843
- \mtc@sv@tableofcontents ..... 3265
- \mtc@sv@tocdepth ..... 2801, 2813, 2843
- \mtc@svchapter ..... 215
- \mtc@svf@tocdepth ..... 2813
- \mtc@svpart ..... 170, 1426
- \mtc@svsection ..... 203
- \mtc@svs@part ..... 170
- \mtc@svss ..... 203
- \mtc@svt@tocdepth ..... 2843
- \MTC@test ..... 911, 913
- \mtc@tmp@name ..... 2390, 2418
- \mtc@tmpde@name ..... 2781
- \mtc@tmpfeat@name ..... 2753
- \mtc@tmpfm@name ..... 2603
- \mtc@tmppn@name ..... 2633, 2677
- \mtc@tmptf@name ..... 2447
- \mtc@tmpti@name ..... 2477
- \MTC@toc ..... 830, 899, 913
- \mtc@toks ..... 224, 849,  
956, 1011, 1046, 1066, 1518, 1615,  
1671, 2079, 2174, 2230, 2418, 2447,  
2477, 2781
- \mtc@typetable@minilof ..... 2306
- \mtc@typetable@minilot ..... 2307
- \mtc@typetable@minitoc ..... 2305
- \mtc@typetable@partlof ..... 2304
- \mtc@typetable@partlot ..... 2303
- \mtc@typetable@parttoc ..... 2302
- \mtc@typetable@sectlof ..... 2309
- \mtc@typetable@sectlot ..... 2310
- \mtc@typetable@secttoc ..... 2308
- \mtc@typetitle@minilof ..... 2315
- \mtc@typetitle@minilot ..... 2316
- \mtc@typetitle@minitoc ..... 2314
- \mtc@typetitle@partlof ..... 2312
- \mtc@typetitle@partlot ..... 2313
- \mtc@typetitle@parttoc ..... 2311
- \mtc@typetitle@sectlof ..... 2318
- \mtc@typetitle@sectlot ..... 2319
- \mtc@typetitle@secttoc ..... 2317
- \mtc@v ..... 224,  
532, 536, 603, 607, 670, 674, 1238,  
1307, 1385, 1788, 1855, 1921
- \mtc@verse ..... 486
- mtc@verse (environment) 486, 539, 558, 610,  
630, 677, 697, 2533, 2548, 2563
- \mtc@While ..... 108, 133
- \mtc@WhileBody ..... 108
- \mtc@WhileCondition ..... 108
- \mtc@WhileNext ..... 108
- \mtc@WhilePreCondition ..... 108
- \MTC@WriteCoffeeline .. 865, 1066, 1529,  
2079
- \MTC@WriteContentsLine ... 1489, 2062,  
2157, 2213
- \MTC@WriteContentsline ..... 850, 882,  
956, 1011, 1046, 1519, 1543, 1615,  
1671, 2079, 2110, 2174, 2230
- \mtc@xifmtarg ..... 108
- \mtc@YN@F ..... 2331
- \mtc@YN@f ..... 2330
- \mtc@YN@FALSE ..... 2327
- \mtc@YN@false ..... 2326
- \mtc@YN@FAUX ..... 2329
- \mtc@YN@faux ..... 2328
- \mtc@YN@N ..... 2325
- \mtc@YN@n ..... 2324
- \mtc@YN@NO ..... 2323
- \mtc@YN@no ..... 2322
- \mtc@YN@NON ..... 2332
- \mtc@YN@non ..... 2333
- \mtc@YN@O ..... 2351
- \mtc@YN@o ..... 2352
- \mtc@YN@OFF ..... 2321
- \mtc@YN@off ..... 2320
- \mtc@YN@ON ..... 2336
- \mtc@YN@on ..... 2335
- \mtc@YN@OUI ..... 2349

- \mtc@YN@oui ..... 2350
  - \mtc@YN@T ..... 2344
  - \mtc@YN@t ..... 2343
  - \mtc@YN@TRUE ..... 2342
  - \mtc@YN@true ..... 2341
  - \mtc@YN@TRUEV ..... 2348
  - \mtc@YN@v ..... 2347
  - \mtc@YN@VRAI ..... 2346
  - \mtc@YN@vrai ..... 2345
  - \mtc@YN@Y ..... 2340
  - \mtc@YN@y ..... 2339
  - \mtc@YN@YES ..... 2338
  - \mtc@YN@yes ..... 2337
  - \mtc@zrule . 224, 539, 561, 610, 633, 677,  
700, 1247, 1269, 1321, 1342, 1395,  
1416, 1800, 1820, 1867, 1886, 1933,  
1953
  - \mtcaddchapter ..... 217, 3873
  - \mtcaddpart ..... 180, 3873
  - \mtcaddsection ..... 206, 3873
  - \mtcdotsep ..... 2518, 2536
  - \mtcficindex ..... 3911, 3912
  - \mtcfixglossary . 2914, 2919, 2921, 2927,  
3912
  - \mtcfixindex 2877, 2882, 2884, 2890, 3911
  - \mtcfont ..... 465, 513, 3754
  - mtcglo.ist ..... **55**
  - mtchideinmainlof (environment) ... 2813,  
3905
  - mtchideinmainlot (environment) ... 2843,  
3905
  - mtchideinmaintoc (environment) ... 2801,  
3905
  - \mtcindent 8, 444, 486, 539, 610, 677, 3685
  - \mtcloadmlo 2960, 3791, 4280, 4297, 4449,  
4458, 4735, 4752, 4767, 4784, 4801,  
4816, 4914, 4929, 4946, 4961, 4978,  
5596
  - \mtclofrmargin ..... 2509
  - \mtcname ..... 837, 868
  - mtcoff ..... **41, 43**
  - mtcoff.sty ..... **54**
  - \mtcoffwarn@false ..... 3860
  - \mtcoffwarn@true 3873, 3882, 3891, 3913,  
3918, 3923
  - \mtcpagenumbers ... 287, 336, 355, 3810
  - \mtcPfont ..... 465, 3754
  - \mtcpnumwidth ..... 2500, 2534
  - \mtcrule ..... 2275, 3792
  - \mtcselectlanguage . 2947, 3008, 3790,  
3991, 3993, 3997, 4016, 4020, 4041,  
4060, 4142, 4161, 4237, 4241, 4246,  
4251, 4393, 4440, 4492, 4512, 4516,  
4520, 4845, 5031, 5162, 5166, 5251,  
5615, 5641, 5677
  - \mtcsetdepth 2766, 2773, 2785, 2791, 2796,  
3772
  - \mtcsetfeature ... 399, 2722, 2731, 2744,  
2757, 2762, 3772
  - \mtcsetfont 2361, 2362, 2369, 2382, 2422,  
2427, 3772
  - \mtcsetformat . 548, 620, 687, 2528, 2573,  
2582, 2594, 2606, 2611, 3772
  - \mtcsetpagenumber ..... 2618
  - \mtcsetpagenumbers .. 2618, 2627, 2641,  
3772
  - \mtcsetrules .... 2661, 2670, 2684, 3772
  - \mtcsettitle 2461, 2462, 2469, 2481, 2799,  
3772
  - \mtcsettitlefont 2431, 2432, 2439, 2451,  
3772
  - \mtcSfont ..... 465, 3754
  - \mtcskip ..... 8, 3695
  - \mtcskipamount ..... 8, 3685
  - \mtcSPfont ..... 465, 3754
  - \mtcSSfont ..... 465, 3754
  - \mtcSSSfont ..... 465, 3754
  - \mtcsv@listoffigures ..... 197
  - \mtcsv@listoftables ..... 197
  - \mtcsv@tableofcontents ..... 197
  - \mtctitle . 512, 513, 584, 651, 3991, 4006,  
4031, 4050, 4069, 4085, 4101, 4117,  
4132, 4151, 4176, 4205, 4227, 4260,  
4306, 4323, 4338, 4353, 4368, 4383,  
4402, 4421, 4467, 4482, 4502, 4530,  
4545, 4560, 4576, 4602, 4642, 4682,  
4714, 4829, 4855, 4869, 4884, 4899,  
4987, 5002, 5021, 5040, 5055, 5070,  
5085, 5102, 5119, 5136, 5152, 5175,  
5191, 5203, 5222, 5241, 5260, 5275,  
5290, 5318, 5358, 5393, 5422, 5437,  
5452, 5467, 5488, 5507, 5522, 5537,  
5552, 5568, 5583, 5605, 5621, 5651,  
5667, 5686
  - \mtctocmargin ..... 2510, 2535
  - \mtifont ..... 465, 513, 584, 651, 3754
  - \mz ... 5287, 5288, 5291, 5292, 5295, 5296
- N**
- \n@mti .. 475, 513, 584, 651, 796, 805, 814
  - \n@pti 1130, 1137, 1221, 1295, 1369, 1454,  
1463, 1472
  - \n@sti 1735, 1773, 1841, 1907, 2027, 2036,  
2045
  - naustrian.mld ..... **266**
  - needspace ..... **43**
  - \NeedsTeXFormat ..... 2, 3682, 3930

`\newbox` ..... 227, 231  
`\newcommand` ... 180, 206, 217, 451, 1096,  
 1721, 2362, 2432, 2462, 2574, 2618,  
 2661, 2723, 2766, 2891, 2928, 2983,  
 3699–3701, 3873, 3882, 3891, 3911,  
 3912  
`\newcounter` 415, 417, 449, 455, 1080, 1086,  
 1708, 1720, 1724  
`\newif` . 12–27, 29, 31, 33, 35, 37, 39–60, 63,  
 64, 66, 84, 133–135, 240, 748, 2361,  
 2431, 2461, 2573, 2616, 2617, 2659,  
 2660, 2722, 2765, 2800, 2813, 2843,  
 2890, 2927, 2947, 3000, 3125–3127,  
 3262–3264, 3855–3858, 3860  
`\newlength` ..... 8, 1105, 1726  
`\newread` ..... 224, 268  
`\newwrite` ..... 7, 408  
`ngerman.mld` ..... **267**  
`ngermanb.mld` ..... **266, 267**  
`nocheckfiles` ..... **22, 39**  
`nohints` ..... **39**  
`\noindent` 529, 600, 667, 1236, 1310, 1384,  
 1789, 1856, 1922  
`\nomlfpagenumbers` . 323, 336, 342, 3810  
`\nomlfrule` ..... 2275, 3792  
`\nomltpagenumbers` . 329, 336, 342, 348,  
 3810  
`\nomltrule` ..... 2275, 3792  
`\nomtcpagenumbers` ..... 287, 336, 3810  
`\nomtcrule` ..... 2275, 3792  
`\noplfpagenumbers` ..... 323, 347, 3810  
`\noplfrule` ..... 2275, 3792  
`\nopltpagenumbers` ..... 329, 353, 3810  
`\nopltrule` ..... 2275, 3792  
`\noptcpagenumbers` ..... 309, 341, 3810  
`\noptcrule` ..... 2275, 3792  
`\normalcolor` ..... 745  
`\normalfont` ..... 745  
`\normalsize` ..... 1119–1121  
`norsk.mld` ..... **268**  
`\noslfpagenumbers` ..... 323, 345, 3810  
`\noslfrule` ..... 2275, 3792  
`\nosltpagenumbers` ..... 329, 351, 3810  
`\nosltrule` ..... 2275, 3792  
`\nostcpagenumbers` ..... 299, 339, 3810  
`\nostcrule` ..... 2275, 3792  
`notoccite` ..... **38, 75**  
`\numberline` ..... 1978  
`\nun` ..... 4823, 4830, 4837  
`nynorsk.mld` ..... **268**

## O

`\openin` ..... 899, 962, 1017

`\openout` ..... 161, 165,  
 238, 409, 837, 868, 913, 944, 975,  
 999, 1030, 1502, 1516, 1589, 1613,  
 1645, 1669, 1701, 2077, 2095, 2146,  
 2172, 2202, 2228, 2258

## P

`\Package` ..... 104  
`\PackageError` ... 2368, 2381, 2426, 2438,  
 2455, 2468, 2485, 2581, 2593, 2610,  
 2626, 2640, 2669, 2683, 2730, 2743,  
 2761, 2772, 2784, 2795, 2802, 2808,  
 2817, 2824, 2832, 2838, 2847, 2854,  
 2862, 2868, 2880, 2904, 2917, 2941,  
 2952, 2965, 2973  
`\PackageInfo` ..... 2, 65, 87, 92,  
 94, 96, 174, 187, 189, 195, 214, 238,  
 249, 251, 507, 578, 645, 838, 842,  
 871, 875, 905, 945, 949, 968, 1000,  
 1004, 1023, 1210, 1284, 1358, 1493,  
 1497, 1507, 1511, 1570, 1604, 1608,  
 1627, 1660, 1664, 1683, 1767, 1834,  
 1900, 2068, 2072, 2099, 2103, 2130,  
 2163, 2167, 2186, 2219, 2223, 2242,  
 2421, 2450, 2480, 2605, 2650, 2653,  
 2694, 2697, 2756, 2790, 2950, 2963,  
 3131, 3136, 3142, 3148, 3186, 3202,  
 3210, 3648, 3933  
`\PackageWarning` 446, 901, 964, 1019, 1091,  
 1566, 1623, 1679, 1717, 2126, 2182,  
 2238, 3874, 3883, 3892, 3914, 3919,  
 3924  
`\PackageWarningNoLine` ..... 171,  
 191, 255, 257, 266, 270, 2875, 2883,  
 2912, 2920, 2998, 3154, 3161, 3176,  
 3192, 3197, 3223, 3226, 3238, 3241,  
 3254, 3257, 3294, 3306, 3314, 3322,  
 3332, 3340, 3348, 3358, 3366, 3374,  
 3383, 3391, 3399, 3409, 3417, 3425,  
 3435, 3443, 3451, 3461, 3470, 3479,  
 3490, 3499, 3508, 3519, 3528, 3537,  
 3551, 3559, 3567, 3577, 3584, 3589,  
 3594, 3599, 3604, 3609, 3614, 3619,  
 3624, 3629, 3634, 3639, 3644, 3862  
`\pagebreak` 563, 634, 701, 1272, 1345, 1419,  
 1823, 1889, 1956  
`\paragraph` ..... 850, 1519, 2079  
`\parsep` ..... 486, 1189, 1751  
`\parskip` 8, 1189, 1230, 1304, 1378, 1783,  
 1850, 1916  
`part` ..... **73, 74**  
`\part` 170, 424, 714, 1489, 1600, 1656, 2392,  
 2396, 2400, 2404, 2409, 2414, 2881,  
 2894, 2918, 2931, 3215, 3668



- \partbegin . 1426, 1590, 1646, 1702, 3708
- \partend ... 1426, 1579, 1636, 1692, 3708
- \partlof .. 1130, 1200, 1274, 3315, 3393, 3471, 3715
- \partlof@ ..... 759, 1274, 1275
- partlofdepth ..... **118, 127**
- \partlot .. 1130, 1200, 1347, 3323, 3401, 3480, 3715
- \partlot@ ..... 759, 1347, 1348
- partlotdepth ..... **118, 130**
- \partopsep ..... 490, 1189, 1750
- \parttoc ... 1130, 1200, 3307, 3385, 3462, 3715
- \parttoc@ ..... 759, 1200, 1201
- parttocdepth ..... **22, 118, 124**
- \pchapter ..... 1421, 1519, 3733
- placeins ..... **39, 73**
- \PLF@ ..... 1632, 1636
- \PLF@contentsline ... 1600, 1635, 1636
- \PLF@explist ..... 1598, 1620
- \PLF@line ..... 1630, 1632
- \PLF@list ..... 1595, 1598, 1620, 1636
- \PLF@lof ..... 1598, 1621, 1636
- \PLF@loop ..... 1595, 1621
- \PLF@next ..... 1443, 1595, 1620
- \plf@pgno ..... 346, 347, 1331
- \PLF@read ..... 1621, 1630, 1636
- \plf@rule .. 444, 1307, 1342, 2284, 2287, 2290
- \plf@setform ..... 1331, 2543
- \PLF@test ..... 1632, 1634, 1636
- \PLF@WriteContentsLine ..... 1600
- \plfdotsep ..... 2518, 2546, 2566, 2571
- \plffont ..... 1107, 1118, 1304, 3744
- \plfname ..... 1600, 1603
- \plfpagenumbers ... 323, 346, 355, 3810
- \plfpnumwidth ... 2500, 2544, 2564, 2569
- \plfrule ..... 2275, 3792
- \plftitle . 1289, 1295, 1307, 3991, 4003, 4028, 4047, 4066, 4082, 4098, 4114, 4129, 4148, 4169, 4200, 4224, 4257, 4303, 4320, 4335, 4350, 4365, 4380, 4399, 4414, 4464, 4479, 4498, 4527, 4542, 4557, 4573, 4594, 4598, 4634, 4674, 4711, 4824, 4852, 4866, 4881, 4896, 4984, 4999, 5018, 5037, 5052, 5067, 5082, 5099, 5116, 5133, 5149, 5172, 5188, 5208, 5219, 5238, 5257, 5272, 5287, 5310, 5350, 5385, 5419, 5434, 5449, 5464, 5484, 5504, 5519, 5534, 5549, 5565, 5580, 5602, 5628, 5648, 5664, 5683
- \plftocrmarg .... 2509, 2545, 2565, 2570
- \PLT@ ..... 1688, 1692
- \PLT@contentsline ... 1656, 1691, 1692
- \PLT@explist ..... 1654, 1676
- \PLT@line ..... 1686, 1688
- \PLT@list ..... 1651, 1654, 1676, 1692
- \PLT@loop ..... 1651, 1677
- \PLT@lot ..... 1654, 1677, 1692
- \PLT@next ..... 1447, 1651, 1676, 2020
- \plt@pgno ..... 352, 353, 1405
- \PLT@read ..... 1677, 1686, 1692
- \plt@rule 444, 1385, 1417, 2293, 2296, 2299
- \plt@setform ..... 1405, 2558
- \PLT@test ..... 1688, 1690, 1692
- \PLT@WriteContentsLine ..... 1656
- \pltdotsep ..... 2518, 2561
- \pltfont ..... 1107, 1118, 1378, 3744
- \pltname ..... 1656, 1659
- \pltpagenumbers .... 329, 352, 355, 3810
- \pltpnumwidth ..... 2500, 2559
- \pltrule ..... 2275, 3792
- \plttitle . 1363, 1369, 1385, 3991, 4004, 4029, 4048, 4067, 4083, 4099, 4115, 4130, 4149, 4172, 4202, 4225, 4258, 4304, 4321, 4336, 4351, 4366, 4381, 4400, 4417, 4465, 4480, 4499, 4528, 4543, 4558, 4574, 4638, 4678, 4712, 4826, 4853, 4867, 4882, 4897, 4985, 5000, 5019, 5038, 5053, 5068, 5083, 5100, 5117, 5134, 5150, 5173, 5189, 5209, 5220, 5239, 5258, 5273, 5288, 5314, 5354, 5388, 5420, 5435, 5450, 5465, 5486, 5505, 5520, 5535, 5550, 5566, 5581, 5603, 5630, 5649, 5665, 5684
- \plttocrmarg ..... 2509, 2560
- pmk ..... **55**
- polish.mld ..... **269**
- polish2.mld ..... **269**
- portuges.mld ..... **270**
- portuguese.mld ..... **270**
- \ProcessOptions ..... 3654
- \ProcessOptions\* ..... 3654
- \ProvidesFile ... 3679, 3992, 3996, 4000, 4015, 4019, 4023, 4040, 4044, 4059, 4063, 4078, 4094, 4110, 4126, 4141, 4145, 4160, 4164, 4195, 4220, 4236, 4240, 4245, 4249, 4254, 4269, 4283, 4300, 4315, 4332, 4347, 4362, 4377, 4392, 4396, 4411, 4438, 4443, 4452, 4461, 4476, 4491, 4495, 4511, 4515, 4519, 4523, 4539, 4554, 4569, 4585, 4625, 4665, 4705, 4723, 4738, 4755, 4770, 4787, 4804, 4819, 4844, 4848, 4864, 4878, 4893, 4908, 4917, 4932, 4949, 4964, 4981, 4996, 5011, 5030,

- 5034, 5049, 5064, 5079, 5094, 5111,  
5128, 5145, 5161, 5165, 5169, 5184,  
5200, 5216, 5231, 5250, 5254, 5269,  
5284, 5299, 5341, 5381, 5414, 5431,  
5446, 5461, 5476, 5501, 5516, 5531,  
5546, 5561, 5577, 5592, 5599, 5614,  
5618, 5640, 5644, 5660, 5676, 5680
- \ProvidesPackage ..... 2, 3682, 3931
- \psection ..... 1421, 3733
- ptc ..... **79, 118, 131**
- \PTC@ ..... 1575, 1579
- \PTC@contentsline ... 1489, 1578, 1579
- \PTC@explist ..... 1487, 1563
- \PTC@line ..... 1573, 1575
- \PTC@list ..... 1484, 1487, 1563, 1579
- \PTC@loop ..... 1484, 1564
- \PTC@next ..... 1439, 1484, 1563
- \ptc@pgno ..... 340, 341, 1255
- \PTC@read ..... 1564, 1573, 1579
- \ptc@rule . 1102, 1238, 1269, 1385, 1416,  
2275, 2278, 2281
- \ptc@setform ..... 1255, 2528
- \ptc@spart ..... 1426
- \PTC@test ..... 1575, 1577
- \PTC@toc ..... 1487, 1564, 1579
- \ptc@verse ..... 1189
- ptc@verse (environment) 1189, 1252, 1255,  
1326, 1331, 1400, 1405, 2528, 2543,  
2558
- \ptcCfont ..... 1118, 3744
- \ptcdotsep ..... 2518, 2531
- \ptcfont ..... 1107, 1118, 1230, 3744
- \ptcindent 1105, 1189, 1247, 1321, 1395,  
3689
- \ptcname ..... 1489, 1506
- \ptcpagenumbers ... 309, 340, 355, 3810
- \ptcPfont ..... 1107, 1118, 3744
- \ptcpnumwidth ..... 2500, 2529
- \ptcrule ..... 2275, 3792
- \ptcSfont ..... 1107, 1118, 3744
- \ptcSPfont ..... 1107, 1118, 3744
- \ptcSSfont ..... 1107, 1118, 3744
- \ptcSSSfont ..... 1107, 1118, 3744
- \ptctitle . 1215, 1221, 1238, 3991, 4002,  
4027, 4046, 4065, 4081, 4097, 4113,  
4128, 4147, 4167, 4198, 4223, 4256,  
4302, 4319, 4334, 4349, 4364, 4379,  
4398, 4413, 4463, 4478, 4497, 4526,  
4541, 4556, 4572, 4591, 4631, 4671,  
4710, 4822, 4851, 4865, 4880, 4895,  
4983, 4998, 5017, 5036, 5051, 5066,  
5081, 5098, 5115, 5132, 5148, 5171,  
5187, 5207, 5218, 5237, 5256, 5271,  
5286, 5307, 5347, 5383, 5418, 5433,  
5448, 5463, 5482, 5503, 5518, 5533,  
5548, 5564, 5579, 5601, 5627, 5647,  
5663, 5682
- \ptctocrmarg ..... 2509, 2530
- \ptifont . . 1107, 1118, 1137, 1140, 1238,  
1307, 1385, 3744
- ## R
- \r@mti . . 475, 513, 584, 651, 796, 805, 814
- \r@pti 1130, 1154, 1221, 1295, 1369, 1454,  
1463, 1472
- \r@sti 1735, 1773, 1841, 1907, 2027, 2036,  
2045
- \raggedright . . 523, 594, 661, 1148, 1230,  
1304, 1378, 1783, 1850, 1916
- \read . 136, 224, 909, 971, 1026, 1574, 1631,  
1687, 2134, 2190, 2246
- README ..... **54**
- \refstepcounter ..... 1978, 3957
- \removelastskip ..... 11
- \renewcommand ..... 3935
- report ..... **80, 82**
- \RequirePackage ..... 3655, 3661
- \reset@font ... 265, 525, 532, 536, 596,  
603, 607, 663, 670, 674, 1050, 1055,  
1068, 1073, 1232, 1240, 1244, 1306,  
1313, 1317, 1380, 1387, 1391, 1785,  
1793, 1797, 1852, 1859, 1864, 1918,  
1925, 1930
- \resh . . 4824–4826, 4831–4833, 4838–4840
- \rightmargin ... 495, 541, 612, 679, 1198,  
1249, 1323, 1397, 1755, 1802, 1869,  
1935
- \rmfamily . 465, 466, 473, 474, 1108, 1109,  
1116, 1119–1121, 1128, 1727, 1728,  
1734
- rmk ..... **55**
- romanian.mld ..... **270**
- rubber@rubber ..... **46**
- \rule ..... 228, 232, 236, 567, 1103, 1725,  
2281–2283, 2290–2292, 2299–2301
- russian.mld ..... **271**
- russian2m.mld ..... **271**
- russian2o.mld ..... **272**
- russianb.mld ..... **273**
- russianc.mld ..... **274**
- ## S
- samepage ..... **123, 126, 129**
- samepage (environment) 513, 558, 584, 630,  
651, 697, 1221, 1269, 1295, 1342,  
1369, 1416, 1773, 1820, 1841, 1886,  
1907, 1953
- samin.mld ..... **275**

- \schapter ..... 1957
- scottish.mld ..... **275**
- \sectbegin . 2007, 2137, 2193, 2249, 3708
- \sectend . . . 2007, 2137, 2193, 2249, 3708
- section ..... **39, 73, 74**
- \section ..... 187, 203,  
714, 850, 1519, 1968, 2062, 2157,  
2213, 3215, 3245, 3674
- \sectlof . . . 1824, 2408, 3296, 3367, 3445,  
3529, 3727
- \sectlof@ ..... 769, 1824, 1825
- sectlofdepth ..... **141, 148**
- \sectlot . . . 1890, 2413, 3296, 3375, 3453,  
3538, 3727
- \sectlot@ ..... 769, 1890, 1891
- sectlotdepth ..... **141, 151**
- sectsty ..... **39, 52, 73, 76, 204**
- \secttoc 1757, 2403, 3359, 3437, 3520, 3727
- \secttoc@ ..... 769, 1757, 1758
- secttocdepth ..... **22, 142, 146**
- serbian.mld ..... **276**
- serbianc.mld ..... **276**
- \setcounter 1080, 1086, 1439, 1443, 1447,  
1708, 2781, 2801, 2813, 2843, 2984
- \sh ..... 4223, 4227, 4231
- \shin . . 4824, 4826, 4831, 4833, 4838, 4840
- shortext ..... **44**
- shorttext ..... **39**
- \sxt@@n ..... 224
- \SLF@ ..... 2191, 2193
- \SLF@contentsline ..... 2157, 2193
- \SLF@explist ..... 2155, 2179
- \SLF@line ..... 2189
- \SLF@list ..... 2152, 2155, 2179, 2193
- \SLF@lof ..... 2155, 2180, 2193
- \SLF@loop ..... 2152, 2180
- \SLF@next ..... 2016, 2152, 2179
- \slf@pgno ..... 344, 345, 1877
- \SLF@read ..... 2180, 2189, 2193
- \slf@rule 444, 1855, 1887, 2286, 2289, 2292
- \slf@setform ..... 1877, 2553
- \SLF@test ..... 2189, 2193
- \slfdotsep ..... 2518, 2556
- \slffont ..... 1727, 1850, 3763
- \slflotrmarg ..... 2509
- \slfname ..... 2157, 2162
- \slfpagenumbers . . . 323, 344, 355, 3810
- \slfpnumwidth ..... 2500, 2554
- \slfrule ..... 2275, 3792
- \slftitle . 1839, 1841, 1855, 3991, 4011,  
4036, 4055, 4074, 4090, 4106, 4122,  
4137, 4156, 4187, 4214, 4232, 4265,  
4311, 4328, 4343, 4358, 4373, 4388,  
4407, 4430, 4472, 4487, 4507, 4535,  
4550, 4565, 4581, 4616, 4620, 4656,  
4696, 4719, 4838, 4860, 4874, 4889,  
4904, 4992, 5007, 5026, 5045, 5060,  
5075, 5090, 5107, 5124, 5141, 5157,  
5180, 5196, 5212, 5227, 5246, 5265,  
5280, 5295, 5332, 5372, 5405, 5427,  
5442, 5457, 5472, 5496, 5512, 5527,  
5542, 5557, 5573, 5588, 5610, 5634,  
5656, 5672, 5691
- \slftocrmarg ..... 2514, 2555
- slovak.mld ..... **277**
- slovene.mld ..... **277**
- \slshape ..... 474
- \SLT@ ..... 2247, 2249
- \SLT@contentsline ..... 2213, 2249
- \SLT@explist ..... 2211, 2235
- \SLT@line ..... 2245
- \SLT@list ..... 2208, 2211, 2235, 2249
- \SLT@loop ..... 2208, 2236
- \SLT@lot ..... 2211, 2236, 2249
- \SLT@next ..... 2023, 2208, 2235
- \slt@pgno ..... 350, 351, 1943
- \SLT@read ..... 2236, 2245, 2249
- \slt@rule 444, 1923, 1954, 2295, 2298, 2301
- \slt@setform ..... 1943, 2568
- \SLT@test ..... 2245, 2249
- \sltdotsep ..... 2518
- \sltfont ..... 1727, 1916, 3763
- \sltlotrmarg ..... 2509
- \sltname ..... 2213, 2218
- \sltpagenumbers . . . 329, 350, 355, 3810
- \sltpnumwidth ..... 2500
- \sltrule ..... 2275, 3792
- \sltttitle . 1905, 1907, 1921, 3991, 4012,  
4037, 4056, 4075, 4091, 4107, 4123,  
4138, 4157, 4190, 4216, 4233, 4266,  
4312, 4329, 4344, 4359, 4374, 4389,  
4408, 4433, 4473, 4488, 4508, 4536,  
4551, 4566, 4582, 4660, 4700, 4720,  
4840, 4861, 4875, 4890, 4905, 4993,  
5008, 5027, 5046, 5061, 5076, 5091,  
5108, 5125, 5142, 5158, 5181, 5197,  
5213, 5228, 5247, 5266, 5281, 5296,  
5336, 5376, 5408, 5428, 5443, 5458,  
5473, 5498, 5513, 5528, 5543, 5558,  
5574, 5589, 5611, 5636, 5657, 5673,  
5692
- \slttocrmarg ..... 2517
- spanish.mld ..... **236, 237, 278**
- spanish2.mld ..... **278**
- spanish3.mld ..... **279**
- \starchapter ..... 780, 868, 1543
- \starparagraph ..... 882, 1543, 2110
- \starpert ..... 1504

- \starsection . . . . . 882, 1543, 1968, 2095
  - \starsubparagraph . . . . . 882, 1543, 2110
  - \starsubsection . . . . . 882, 1543, 2110
  - \starsubsubsection . . . . . 882, 1543, 2110
  - stc . . . . . **80, 81, 142**
  - \STC@ . . . . . 2135, 2137
  - \STC@contentsline . . . . . 2062, 2137
  - \STC@explist . . . . . 2060, 2123
  - \STC@line . . . . . 2133, 2135
  - \STC@list . . . . . 2057, 2060, 2123, 2137
  - \STC@loop . . . . . 2057, 2124
  - \STC@next . . . . . 2012, 2057, 2123
  - \stc@pgno . . . . . 338, 339, 1808
  - \STC@read . . . . . 2124, 2133, 2137
  - \stc@rule . . . . . 1716, 1788, 1820, 1886, 1921,  
1953, 2277, 2280, 2283
  - \stc@setform . . . . . 1808, 2538
  - \stc@ssect . . . . . 2007
  - \STC@test . . . . . 2135, 2137
  - \STC@toc . . . . . 2060, 2124, 2137
  - \stc@verse . . . . . 1746
  - stc@verse (environment) . . . . . 1746, 1805, 1808,  
1872, 1877, 1938, 1943, 2538, 2553,  
2568
  - \stcdotsep . . . . . 2518, 2541
  - \stcfont . . . . . 1727, 1783, 3763
  - \stcindent . . . . . 1716, 1754, 1800, 1867, 1933,  
3685
  - \stcclotrmarg . . . . . 2509
  - \stcname . . . . . 2062, 2067, 2095
  - \stcpagenumbers . . . . . 299, 338, 355, 3810
  - \stcPfont . . . . . 1727, 3763
  - \stcpnumwidth . . . . . 2500, 2539
  - \stcrule . . . . . 2275, 3792
  - \stcSPfont . . . . . 1727, 3763
  - \stcSSfont . . . . . 1727, 3763
  - \stcSSSfont . . . . . 1727, 3763
  - \stctitle . . . . . 1773, 1788, 3991, 4010, 4035,  
4054, 4073, 4089, 4105, 4121, 4136,  
4155, 4185, 4212, 4231, 4264, 4310,  
4327, 4342, 4357, 4372, 4387, 4406,  
4429, 4471, 4486, 4506, 4534, 4549,  
4564, 4580, 4613, 4653, 4693, 4718,  
4836, 4859, 4873, 4888, 4903, 4991,  
5006, 5025, 5044, 5059, 5074, 5089,  
5106, 5123, 5140, 5156, 5179, 5195,  
5211, 5226, 5245, 5264, 5279, 5294,  
5329, 5369, 5403, 5426, 5441, 5456,  
5471, 5494, 5511, 5526, 5541, 5556,  
5572, 5587, 5609, 5633, 5655, 5671,  
5690
  - \stctocrmarg . . . . . 2511, 2540
  - \stepcounter . . . . . 170, 203,  
215, 421, 836, 868, 943, 998, 1426,  
1491, 1505, 1544, 1602, 1658, 2065,  
2095, 2160, 2216
  - \stifont . . . . . 1727, 1788, 1855, 1921, 3763
  - subfig . . . . . **22, 179**
  - \subparagraph . . . . . 850, 1519, 2079
  - \subsection . . . . . 850, 1519, 2079
  - \subsubsection . . . . . 850, 1519, 2079
  - \sv@chapter . . . . . 704
  - \sv@minilof@ . . . . . 748
  - \sv@minilot@ . . . . . 748
  - \sv@minitoc@ . . . . . 748
  - \sv@part . . . . . 1426
  - \sv@partlof@ . . . . . 759
  - \sv@partlot@ . . . . . 759
  - \sv@parttoc@ . . . . . 759
  - \sv@sect . . . . . 1961, 1967
  - \sv@sectlof@ . . . . . 769
  - \sv@sectlot@ . . . . . 769
  - \sv@secttoc@ . . . . . 769
  - \sv@spart . . . . . 1426
  - swedish.mld . . . . . **279**
- T**
- T . . . . . **101**
  - \table . . . . . 1011, 1671, 2230
  - \tableofcontents . . . . . 197, 3265, 3463, 3492,  
3521
  - tabular (environment) . . . . . 513, 584, 651, 1238,  
1307, 1385, 1788, 1855, 1921
  - \tav . . . . . 4822, 4824, 4826, 4827, 4829, 4831,  
4833, 4834, 4836, 4838, 4840, 4841
  - \tet . . . . . 4827, 4834, 4841
  - \textbf . . . . . 1071
  - \tf@mtc . . . . . 7, 133, 238,  
823, 833, 837, 868, 913, 944, 975,  
999, 1030, 1046, 1066, 1481, 1489,  
1515, 1516, 1588, 1589, 1600, 1612,  
1613, 1644, 1645, 1656, 1668, 1669,  
1700, 1701, 2054, 2062, 2076, 2077,  
2107, 2108, 2145, 2146, 2157, 2171,  
2172, 2201, 2202, 2213, 2227, 2228,  
2257, 2258, 3733
  - thai.mld . . . . . **280**
  - thai.mlo . . . . . **280**
  - \The@mtc . . . . . 444, 501, 503, 572, 574, 639, 641
  - \thechapter . . . . . 3977, 3980
  - \themtc . . . . . 444, 833, 837, 868, 941, 944, 996,  
999
  - \Thepart . . . . . 1086, 1201, 1275, 1348
  - \theptc . . . . . 1086, 1489, 1508, 1509, 1512,  
1513, 1600, 1605, 1606, 1609, 1610,  
1656, 1661, 1662, 1665, 1666
  - \thestc . . . . . 1758, 1825, 1891,  
1947, 2062, 2069, 2070, 2073, 2074,

- 2095, 2157, 2164, 2165, 2168, 2169,  
2213, 2220, 2221, 2224, 2225
- \thispageminilofstyle . . . 390, 582, 3837
- \thispageminilotstyle . . . 390, 649, 3837
- \thispageminitocstyle . . . 390, 511, 3837
- \thispagepartlofsty . . . 390, 1289, 3828
- \thispagepartlotstyle . . . 390, 1363, 3828
- \thispageparttocstyle . . . 390, 1215, 3828
- \thispagesectlofsty . . . 390, 1838, 3846
- \thispagesectlotstyle . . . 390, 1904, 3846
- \thispagesecttocstyle . . . 390, 1771, 3846
- \thispagestyle . . . . . 390
- tight . . . . . **29, 47, 75, 94, 122**
- \tightmtcfalse . . . . . 63, 2989, 3856
- \tightmtctrue . . . . . 2988
- titlesec . . . . . **39**
- titletoc . . . . . **39**
- tocbibind . . . . . **37, 39**
- tocdepth . . . . . **96, 100, 102**
- \toclevel@chapter . . . . . 74
- \toclevel@paragraph . . . . . 78
- \toclevel@part . . . . . 73
- \toclevel@section . . . . . 75
- \toclevel@starchapter . . . . . 74
- \toclevel@starparagraph . . . . . 78
- \toclevel@starpart . . . . . 73
- \toclevel@starsection . . . . . 75
- \toclevel@starsubparagraph . . . . . 79
- \toclevel@starsubsection . . . . . 76
- \toclevel@starsubsubsection . . . . . 77
- \toclevel@subparagraph . . . . . 79
- \toclevel@subsection . . . . . 76
- \toclevel@subsubsection . . . . . 77
- \toclevel@xchapter . . . . . 71
- \toclevel@xpart . . . . . 70
- \toclevel@xssect . . . . . 72
- tocloft . . . . . **39, 49**
- TODO . . . . . **55**
- \topsep . . . . . 486, 1189
- turkish.mld . . . . . **280**
- U**
- UKenglish.mld . . . . . **280**
- ukraineb.mld . . . . . **281**
- undotted . . . . . **47**
- \undottedmtcfalse . . . . . 748, 2995, 3858
- \undottedmtctrue . . . . . 2994
- \unhcopy . . . . . 230, 234
- \unpenalty . . . . . 237
- unsrt . . . . . **38**
- \upshape . . . . . 465, 466, 473, 1108, 1109, 1116,  
1119–1121, 1128, 1727, 1728, 1734
- USenglish.mld . . . . . **281**
- \usepackage . . . . . 3869
- usorbian.mld . . . . . **282**
- V**
- \value . . . . . 3545
- \vav . . . . . 4822, 4825, 4827, 4829, 4832, 4834,  
4836, 4839, 4841
- vietnam.mld . . . . . **282, 283**
- vietnam.sty . . . . . **282**
- vietnamese.mld . . . . . **283**
- W**
- welsh.mld . . . . . **283**
- while.tip . . . . . 77
- \write . . . . . 238, 1046, 1066
- X**
- \xalx . . . . . 5418–5420, 5422–5424, 5426–5428
- xchapter . . . . . **82**
- \xchapter . . . . . 702, 940, 995, 3733
- xpart . . . . . **79**
- \xpart . . . . . 1086, 1601, 1657, 3733
- \xssect . . . . . 1957, 2159, 2215, 3733
- xsection . . . . . **81**
- Y**
- \yod . . . . . 4823–4826, 4830–4833, 4837–4840