The minitoc package*

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

9th January 2007

*This document corresponds to minitoc v51, dated 2007/01/09.

Summary

Table of Contents

List of Figures

List of Tables

About this document

This document is rather thick, but please, be not afraid: you do not need to read every page.

- The most useful chapters are in the first part ("User's Manual", page 22):
  - the chapter "The minitoc package", page 23, describes the essential commands to use the package;
  - the chapter "Frequently Asked Questions", page 53, may help you to solve some specific problems;
  - the "Memento" chapter, page 77, is a set of tables to be used as a remainder of the commands of this package;
  - the "Examples of documents" chapter, page 86, gives the code of some documents showing the basic usage of the minitoc package and some interesting situations;
  - the "Messages" chapter, page 141, is certainly boring, but it should be searched if you get some warning or error messages from the minitoc package, because it explains them and also the informative messages;
  - the "Jargon" chapter, page 191, attemps to explain most of the technical terms used here;
  - the "Installation" chapter, page 222, describes all the files included in the distribution of the package;
  - the "Postface" chapter, page 227, gives an abbreviated history of the package.
- The second part, "Implementation", page 238, is much more technical; you can read it if you are interested in the details of the coding of the package. The chapter "The language definition (`.mld`) and object (`.mlo`) files", page 430, may be useful if you are interested by some language.
- The third part, "Complements", page 523, contains a bibliography, a detailed history of the package, and an index.

For this document, I have used:

- a short table of contents (summary), with the `\shorttoc` command from my shorttoc package [105], displaying only parts and chapters;

- a main table of contents (`\tableofcontents`), with a maximum depth (6);
- a main list of figures (`\listoffigures`) and a main list of tables (`\listoftables`);
- for each part, a table of contents displaying only the chapters (`\parttoc` with `parttocdepth` equal to 1);
- for each chapter, a complete table of contents (`\minitoc` with `minitocdepth` equal to 6);
- for each chapter, a list of figures (`\minilof`) and a list of tables (`\minilot`) when useful;
- customized parameters for the layout of the mini-tables.

As the PDF version of the documentation uses hyperlinks, these tables should help you to navigate in the document.

Part I

User's Manual

Contents of the First Part

Chapter 1

The minitoc package

Contents

Figures

Tables

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). A summary of the evolution of this package is given in the chapter 8 on page 227. This package creates a mini-table of contents (a "minitoc" [1]) 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 [2]). 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

[1] 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.

[2] As the standard proc class, and the ltxdoc and ltxnews classes, load the standard article class, these classes will be just considered as variants of the article class.

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").

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

1.1.1 Important restrictions

Note: you cannot use chapter level and section level mini-tables in the same document. This restriction is intented 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`.

1.1.2 Version

The current version of this package is #51. You will find a resumed history of the package in the "Postface" chapter (chapter 8 on page 227) and a more detailed history in "Changes History", page 546.

1.2 License

This package must be distributed and/or may be modified under the conditions of the **LaTeX 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 LaTeX version 2003/12/01 or later.

But please don't bother me about hacked versions; they will not be supported. However, suggestions for corrections and improvements are welcome.

1.3 Using the minitoc package

1.3.1 Loading the package and creating the mini-tables

\usepackage
\minitoc
\chapter

To use the minitoc package, you must insert a command:

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

in the preamble of the document [3]. 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* [4] inside a chapter.

Of course, it is better to put 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 on the following page.

For each mini-table of contents, an auxiliary file will be created with a name of the form *document*`.mtc`⟨*N*⟩, where ⟨*N*⟩ is the absolute chapter number. "Absolute" means that this number is unique, and always increasing from the first chapter [5]. The suffix is `.mlf`⟨*N*⟩ for mini-lists of figures and is `.mlt`⟨*N*⟩ for mini-lists of tables. (If under MS-DOS or any operating system with short extensions to filenames, see section 1.9 on page 51 and section 2.5 on page 56). There are similar commands for mini-tables at the part or section level, depending of the document class.

1.3.2 Preparing the mini-tables

\dominitoc
\dominilof
\dominilot

The commands [6] `\dominitoc`, `\dominilof`, and `\dominilot` (for mini-tables at the chapter level), take respectively the *document*`.toc`, *document*`.lof`, and *document*`.lot` files, and cut slices from them to create the *document*`.mtc`⟨*N*⟩, *document*`.mlf`⟨*N*⟩, and *document*`.mlt`⟨*N*⟩ files.

---

[3] 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.

[4] "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 for details with a high value for `minitocdepth`.

[5] 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`.

[6] The code of these `\do...` commands is directly derived from that of the xr package [89], by David Carlisle, with his permission.

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
...
```

`\dosecttoc` `\dosectlof` `\dosectlot` `\doparttoc` `\dopartlof` `\dopartlot`

The commands `\dosecttoc`, `\dosectlof`, and `\dosectlot` (for mini-tables at the section level) and `\doparttoc`, `\dopartlof`, and `\dopartlot` (for mini-tables at the part level) are analog.

`\mtcprepare` `\tableofcontents` `\listoffigures` `\listoftables`

The `\mtcprepare` command invokes (and replaces) all these preparation commands when they are available with the document class and if the adequate contents file exists. This command accepts also an optional argument to set the default position of the title for *all* the mini-tables.

To obtain a satisfactory result (i.e., non empty), please note that all these commands must *imperatively be put before* any command analog to the `\tableofcontents`, `\listoffigures`, and `\listoftables` commands, or their `\fake...` siblings.

It is also *strongly* recommended to put these commands *before* any sectionning command producing an entry in the table of contents (for the `\do...toc` commands), and *before* any `\caption`-like command producing an entry in the list of figure (for the `\do...lof` commands) or in the list of tables (for the `\do...lot`) commands; else disorder in the mini-tables might result.

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
...
```

1.3.3 Placing the mini-tables

`\mtcskip` `\mtcskipamount` `\bigskipamount`

The `\mtcskip` command may be used to add a vertical skip between two mini-tables. Its height is `\mtcskipamount` (equal to `\bigskipamount` by default). `\mtcskip` eliminates any immediate previous vertical skip, to not accumulate vertical space when a mini-table is empty and skipped by the `checkfiles` option.

`\secttoc` `\section`

The section-level table of contents will be in the section, after the `\section` command, at the point of the `\secttoc` command. The `\secttoc` command may occur *almost anywhere* 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.

For each section-level table of contents, an auxiliary file will be created with a name of the form *document*`.stc`⟨*N*⟩, where ⟨*N*⟩ is the absolute section number. The suffix is `.slf`⟨*N*⟩ for section-level lists of figures and is `.slt`⟨*N*⟩ for section-level lists of tables. (If under MS-DOS or any operating system with short extensions to filenames, see section 1.9 on page 51 and section 2.5 on page 56).

\usepackage
\FloatBarrier

As floats (figures and tables) could drift[7] 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 `insection` option in the `\usepackage` command:

```
\usepackage[insection]{minitoc}
```

if you want more consistent sectlofs and sectlots. The `insection` option loads the placeins package [10] with its `verbose` and `section` options. Sometimes, it might be necessary to use the `\FloatBarrier` command of this package to correctly place the figure or table and have a correct mini-table. The options `above` or `below` options should not be used, because they allow floats to drift above or below a `\FloatBarrier` (or a section limit): the barrier becomes "porous" upwards[8] (↑) or downwards (↓), or both (↕). The `section` option makes a more "watertight" barrier (≡). This is illustrated by the figure 1.1 on the following page.

The placeins package, by Donald Arseneau, is available on CTAN archives; note that the file `placeins.sty` contains its own documentation, with a copy in `placeins.txt`. You need a version whose date is at least 2005/04/18.

Since version #45, this option also loads the flafter package (described in [162] and [189, page 286]) to force a float to appear *after* its reference. The `above` and `below` options of the placeins package are no more used, because they allowed the floats to move out of the section. In all cases, it is *strongly* recommended to verify the position of the floats and, if necessary, to look at the messages of the placeins package in the *document*`.log` file. The placement of floats is a very complex problem, so some manual intervention may be necessary, like the use of the float package [171] or, better, of the floatrow package [159].

If you want to add the partial table of contents for a part, you must use the sequence given in Table 1.3 on page 31. 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 1.9 on page 51 and section 2.5 on page 56).

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).

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

[7] *A float is like a ship in harbor. There is a place in the text which is the anchor location. The figure or "ship" can float around to various places relative to the anchor, but always downstream or downwind. A float with bad placement parameters is like a ship that slips its anchor and eventually crashes on the rocks at the end of a chapter.*
Donald Arseneau

[8] But a float can not drift upwards beyond the top of the current page.

With the `section` option

Command	**Action**
	× × × × × × × × × × × × × *Forbidden Area* × × × × × × × × × × × × ×
`\section`	═══
	Allowed Area
	Float inserted here
	Allowed Area
`\section`	═══
	× × × × × × × × × × × × × *Forbidden Area* × × × × × × × × × × × × ×

With the `above` option

Command	**Action**
	Allowed Area
`\section`	↑ ↑
	Allowed Area
	Float inserted here
	Allowed Area
`\section`	═══
	× × × × × × × × × × × × × *Forbidden Area* × × × × × × × × × × × × ×

With the `below` option

Command	**Action**
	× × × × × × × × × × × × × *Forbidden Area* × × × × × × × × × × × × ×
`\section`	═══
	Allowed Area
	Float inserted here
	Allowed Area
`\section`	↓ ↓
	Allowed Area

With the `above` and `below` options

Command	**Action**
	Allowed Area
`\section`	↕ ↕
	Allowed Area
	Float inserted here
	Allowed Area
`\section`	↕ ↕
	Allowed Area

Figure 1.1: Float barriers

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
...
```

in a chapter without sections and no \minilof or \minilot command in a chapter without figures or tables. The checkfiles (see section 1.3.3 on page 28) package option (default) skips empty mini-tables (with a note in the *document*.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 subfigure and subfig packages [94, 96] from Steven Douglas Cochran, new depth counters are created if necessary, with obvious names like partlofdepth, partlotdepth, minilofdepth, minilotdepth, sectlofdepth, and sectlotdepth.

1.3.4 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, a first method is to say:

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

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

\addcontentsline

```
\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 1.11 on page 52) knows what to do about \adjustmtc (and others), but can do nothing about \stepcounter or \addtocounter, as they are a standard basic commands of LaTeX, not minitoc specific commands. Syntax:

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

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

Table 1.4: Adding an entry in the table of contents for a starred part, chapter, or section

Level	With title
part	\mtcaddpart[*title*]
chapter	\mtcaddchapter[*title*]
section	\mtcaddsection[*title*]

\decrementptc
\decrementmtc
\decrementstc
\incrementptc
\incrementmtc
\incrementstc

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. These commands have no argument.

\mtcaddpart
\mtcaddchapter
\mtcaddsection

A more clever way to solve this problem is to use commands similar to:

\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. See also section 2.30 on page 68 for the problems with \mtcaddpart.

1.4 Typesetting of the mini-tables

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

1.4.1 Chapter-level mini-tables

\mtcfont
\mtcSfont
\mtcSSfont
\mtcSSSfont
\mtcPfont
\mtcSPfont
\mlffont
\mltfont
\mlfSfont
\mltSfont

The mini-table of contents is typeset in the \mtcfont font, which is \small\rmfamily by default. In fact, the font \mtcfont is selected at the beginning of a minitoc, minilof or minilot. More selective choices are made with the following fonts. Section entries are typeset in the \mtcSfont font, which is \small\bfseries by default. For subsections, subsubsections, paragraphs and subparagraphs, the commands \mtcSSfont, \mtcSSSfont, \mtcPfont and \mtcSPfont are available (by default, \small\rmfamily) to enable the use of various fonts. Mini lists of figures and tables are typeset in the fonts \mlffont and \mltfont, which are \small\rmfamily by default. There are also \mlfSfont and \mltSfont for sub-figures and sub-tables entries. Tables 1.5 on the following page and 1.6 on page 35 summarize these many commands [9].

[9] 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 `\part...` commands:			
`\parttoc`	`\ptcfont` `\normalsize\rmfamily`* `\small\rmfamily`**	`\ptctitle` Table of Contents†	`\ptifont` `\LARGE\bfseries`* `\Large\bfseries`**

`\partlof`	`\plffont`
	`\normalsize\rmfamily`*
	`\small\rmfamily`**
	`\plfSfont`
	`\normalsize\rmfamily`*
	`\small\rmfamily`**
		`\plftitle`
		List of Figures†
			`\ptifont`
			`\LARGE\bfseries`*
			`\Large\bfseries`**

`\partlot`	`\pltfont`
	`\normalsize\rmfamily`*
	`\small\rmfamily`**
	`\pltSfont`
	`\normalsize\rmfamily`*
	`\small\rmfamily`**
		`\plttitle`
		List of Tables†
			`\ptifont`
			`\LARGE\bfseries`*
			`\Large\bfseries`**

For the `\mini...` commands:*			
`\minitoc`	`\mtcfont` `\small\rmfamily`	`\mtctitle` Contents†	`\mtifont` `\large\bfseries`

`\minilof`	`\mlffont`
	`\small\rmfamily`
	`\mlfSfont`
	`\small\rmfamily`
		`\mlftitle`
		Figures†
			`\mtifont`
			`\large\bfseries`

`\minilot`	`\mltfont`
	`\small\rmfamily`
	`\mltSfont`
	`\small\rmfamily`
		`\mlttitle`
		Tables†
			`\mtifont`
			`\large\bfseries`

For the `\sect...` commands:**			
`\secttoc`	`\stcfont` `\small\rmfamily`	`\stctitle` Contents†	`\stifont` `\Large\bfseries`

`\sectlof`	`\slffont`
	`\small\rmfamily`
	`\slfSfont`
	`\small\rmfamily`
		`\slftitle`
		Figures†
			`\stifont`
			`\Large\bfseries`

`\sectlot`	`\sltfont`
	`\small\rmfamily`
	`\sltSfont`
	`\small\rmfamily`
		`\slttitle`
		Tables†
			`\stifont`
			`\Large\bfseries`

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

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 bold and non-bold fonts are mixed [10] (the true length of `1em` is not the same for the fonts). This can eventually be adjusted by changing some fonts.

[10] This appears, e.g., if you are using the Computer Modern Roman (CMR) fonts [142]. The symptom disappears if you do not use bold CMR fonts or if you use the TX fonts (txfonts package [220]), by example, like in this document. See also section 2.29 on page 67.

Table 1.6: Fonts for the mini-table entries

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

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

1.4.2 Titles for chapter-level mini-tables

\mtifont
\mtctitle
\mlftitle
\mlttitle
\mtcsettitle
\mtcselectlanguage

Titles are typeset in the \mtifont (\large\bfseries by default) font and the text strings of the titles are defined by \mtctitle, \mlftitle and \mlttitle, which are the strings "Contents", "Figures" and "Tables" by default. These title commands should be redefined by \renewcommand or \mtcsettitle for languages other than english.

The language definition files like french.mld and english.mld (the suffix .mld means "minitoc language definition (file)") (and many others, see the list in table 1.7 on the next page and section 1.4.12 on page 41) are available. You can easily prepare a similar file for a preferred language (see section 2.26 on page 67). You can change the language of these titles by using the \mtcselectlanguage{*language*} macro.

Table 1.7: Available languages

1. afrikaan (afrikaans)
2. albanian
3. arab (arabic) [c]
4. arab2 [a,c]
5. arabi [c,j]
6. armenian [c]
7. bahasai (bahasa, indon, indonesian) [c]
8. bahasam (malay, meyalu) [c]
9. bangla [c]
10. basque
11. bicig (uighur) [c,i]
12. bicig2 (uighur2) [c,i]
13. bicig3 (uighur3) [c,i]
14. bithe (manju) [c]
15. brazil (brazilian)
16. breton
17. bulgarian [c]
18. bulgarianb [c]
19. buryat [c]
20. buryat2 [c]
21. catalan
22. chinese1 [c,g]
23. chinese2 [c,g]
24. croatian
25. czech
26. danish
27. devanagari (hindi)[c]
28. dutch
29. english[†] (american, australian, british, canadian, newzealand, UKenglish, USenglish)
30. english1
31. english2
32. esperant (esperanto)
33. estonian
34. ethiopia (ethiopian) [c]
35. ethiopian2 [c,e,h]
36. farsi1 [c,f,g]
37. farsi2 [c,f,g]
38. farsi3 [c,j]
39. finnish
40. finnish2
41. french (frenchb, frenchle, frenchpro, francais, acadien, canadien)
42. french1
43. french2
44. galician
45. german (austrian)
46. germanb
47. germanb2
48. greek [c]
49. greek-mono [c,e]
50. greek-polydemo [c,e]
51. greek-polykatha [c,e]
52. guarani [h]
53. hangul1 [c,d,g]
54. hangul2 [c,d,g]
55. hangul3 [c,d,g]
56. hangul4 [c,d,g]
57. hangul-u8 [c,e,f,g,h]
58. hanja1 [c,d,g]
59. hanja2 [c,d,g]
60. hanja-u8 [c,e,f,g,h]
61. hebrew [c,h]
62. hebrew2 [c,h]
63. hindi-modern [c]
64. icelandic [f]
65. interlingua
66. irish
67. italian
68. italian2
69. japanese [c,d,g]
70. japanese2 [c,d,g]
71. japanese3 [c,d,g]
72. japanese4 [c,d,g]
73. japanese5 [c,d,g]
74. japanese6 [c,d,g]
75. kannada [c]
76. latin
77. latin2
78. latvian (letton) [e]
79. lithuanian
80. lowersorbian (lsorbian)
81. magyar (hungarian)
82. magyar2
83. magyar3
84. malayalam-keli [c]
85. malayalam-omega [c,e,g,h]
86. malayalam-rachana [c]
87. malayalam-rachana2 [c]
88. mongol [c]
89. ngermanb (ngerman, naustrian)
90. ngermanb2
91. norsk
92. norsk2
93. nynorsk
94. nynorsk2
95. polish
96. polish2 [c,e]
97. polski [c]
98. portuguese (portuges)
99. romanian
100. romanian2
101. romanian3
102. russian [b,c]
103. russianb [b,c]
104. russianc [b,c]
105. russian2m [c,e]
106. russian2o [c,e]
107. russian-cca [c,g,h]
108. russian-cca1 [c,g,h]
109. russian-lh [c,g,h]
110. russian-lhcyralt [c,g,h]
111. russian-lhcyrkoi [c,g,h]
112. russian-lhcyrwin [c,g,h]
113. samin
114. scottish
115. serbian
116. serbianc [c]
117. slovak
118. slovene
119. spanish (castillan, castillian)
120. spanish2
121. spanish3 [e,f]
122. spanish4
123. swedish
124. swedish2
125. thai [c,d,f,g]
126. turkish
127. ukrainian (ukraineb) [b,c]
128. uppersorbian (usorbian)
129. vietnam (vietnamese) [c,d]
130. welsh
131. xalx (khalkha) [c]
132. xalx2 [c]
133. xalx3 [c]

The languages in parentheses are aliases of a main language and their `.mld` files will load the `.mld` file of that main language.

[†] The presence of the `english.mld` file is mandatory.

[a] The `arab(ic)` and `arab2` languages require the use of the ArabTEX package [154, 155] (by Klaus Lagally).

[b] The `russian` language is not yet supported by the babel system [38, 39], but `russianb` [160] is supported if you use babel-3.6 or a higher version ; `russianc` is an extra. Look also at other `.mld` files for russian.

[c] Some languages may require specific fonts.

[d] Requires the CJK package [167, 168].

[e] Requires *Lambda* (**Λ**), the version of LaTeX for Omega (**Ω**).

[f] Requires a 8-bits input encoding.

[g] Uses also a `.mlo` file.

[h] Requires a specific input encoding.

[i] The `bicig` language is also known as `uighur`.

[j] The `arabi` and `farsi3` languages require the use of the Arabi package [135].

1.4.3 Part-level mini-tables

\ptcfont
\ptcCfont
\ptcSfont
\ptcSSfont
\ptcSSSfont
\ptcPfont
\ptcSPfont
\plffont
\pltfont
\plfSfont
\pltSfont

The partial table of contents is typeset in the \ptcfont font, which is defined as \normalsize\rmfamily by default. In fact, the font \ptcfont is selected at the beginning of a parttoc, partlof or partlot. More selective choices are made with the following fonts. Chapter entries are typeset in the \ptcCfont font, which is \normalsize\bfseries by default. Section entries are typeset in the \ptcSfont font, which is \normalsize\rmfamily by default. For subsections, subsubsections, paragraphs and subparagraphs, the commands \ptcSSfont, \ptcSSSfont, \ptcPfont, and \ptcSPfont are available (by default, \normalsize\rmfamily) if you want to use various fonts. Partial lists of figures and tables are typeset in the fonts \plffont and \pltfont, which are \normalsize\rmfamily by default. There are also \plfSfont and \pltSfont for sub-figures and sub-tables entries.

1.4.4 Titles for part-level mini-tables

\ptifont
\ptctitle
\plftitle
\plttitle
\mtcsettitle
\mtcselectlanguage

Titles are typeset in the \ptifont (\LARGE\bfseries by default) font and the text strings of the titles are defined by \ptctitle, \plftitle and \plttitle, which are the strings "Table of Contents", "List of Figures" and "List of Tables" by default. These title commands should be redefined by \renewcommand or \mtcsettitle for languages other than english. The language definition files like french.mld and english.mld (and many others; for a complete list, see table 1.7 on the preceding page) are available. Read also section 1.4.12 on page 41. You can easily prepare a similar file for a preferred language (see section 2.26 on page 67). You can change the language of these titles by using the \mtcselectlanguage{*language*} macro.

1.4.5 Section-level mini-tables

\stcfont
\stcSSfont
\stcSSSfont
\stcPfont
\stcPSfont
\slffont
\sltfont
\slfSfont
\sltSfont

The section-level table of contents is typeset in the \stcfont font, which is defined as \normalsize\rmfamily by default. In fact, the font \stcfont is selected at the beginning of a secttoc, sectlof or sectlot. More selective choices are made with the following fonts. Subsection entries are typeset in the \stcSSfont font, which is \normalsize\bfseries by default. Subsubsection entries are typeset in the \stcSSSfont font, which is \normalsize\rmfamily by default. For paragraphs and subparagraphs, the commands \stcPfont and \stcSPfont are available (by default, \normalsize\rmfamily) if you want to use various fonts. Section-level lists of figures and tables are typeset in the fonts \slffont and \sltfont, which are defined as \normalsize\rmfamily by default. There are also \slfSfont and \sltSfont for sub-figures and sub-tables entries.

1.4.6 Titles for section-level mini-tables

\stifont
\stctitle
\slftitle
\slttitle
\mtcsettitle
\mtcselectlanguage

Titles are typeset in the \stifont (\normalsize\bfseries by default) font and the text strings of the titles are defined by \stctitle, \slftitle and \slttitle, which are

the strings "Contents", "Figures" and "Tables" by default. These title commands should be redefined by `\renewcommand` or `\mtcsettitle` for languages other than english. The language definition files like `french.mld` and `english.mld` (and also many others, as listed in table 1.7 on page 36 and explained in section 1.4.12 on page 41) are available. You can easily prepare a similar file for your preferred language (see section 2.26 on page 67). You can change the language of these titles by using the `\mtcselectlanguage{`*language*`}` macro.

1.4.7 Position of the titles

1.4.7.1 For mini-tables at the part level

`\doparttoc`
`\dopartlof`
`\dopartlot`
`\parttoc`
`\partlof`
`\partlot`

By default, titles are on the left. The preparation commands `\doparttoc`, `\dopartlof` and `\dopartlot` 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 parttoc (or partlof or partlof), just use such an optional argument with the command `\parttoc` (or `\partlof` or `\partlot`).

1.4.7.2 For mini-tables at the chapter level

`\dominitoc`
`\dominilof`
`\dominilot`
`\minitoc`
`\minilof`
`\minilot`

By default, titles are on the left. The preparation commands `\dominitoc`, `\dominilof` and `\dominilot` 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 minitoc (or minilof or minilof), just use such an optional argument with the command `\minitoc` (or `\minilof` or `\minilot`).

1.4.7.3 For mini-tables at the section level

`\dosecttoc`
`\dosectlof`
`\dosectlot`
`\secttoc`
`\sectlof`
`\sectlot`

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 sectlof), just use such an optional argument with the command `\secttoc` (or `\sectlof` or `\sectlot`).

1.4.7.4 Summary of the positionning of titles

\doparttoc
\dopartlof
\dopartlot
\dominitoc
\dominilof
\dominilot
\dosecttoc
\dosectlof
\dosectlot
\mtcprepare

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

\doparttoc, \dopartlof, \dopartlot,
\dominitoc, \dominilof, \dominilot,
\dosecttoc, \dosectlof, \dosectlot,
\mtcprepare

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 (for all mini-tables in the case of \mtcprepare).

\parttoc
\partlof
\partlot
\minitoc
\minilof
\minilot
\secttoc
\sectlof
\sectlot

The following insertion commands:

\parttoc, \partlof, \partlot,
\minitoc, \minilof, \minilot,
\secttoc, \sectlof, \sectlot

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

1.4.8 Line spacing in the mini-tables

\iftightmtc
\tightmtctrue
\tightmtcfalse

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 [147, 195] (scrartcl, 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).

\ifktightmtc
\ktightmtctrue
\ktightmtcfalse

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.

1.4.9 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}
```

```
\mtcsetfont{minilof}{subfigure}%
           {\small\rmfamily\upshape\bfseries}
```

will redefine \mtcSSfont and \mlfSfont with the given font commands.

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

Moreover,

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

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

1.4.10 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.

1.4.11 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).

1.4.12 Languages for the titles

Most of the strings defined in the language definition files (`.mld`) were taken from the superb babel package [38, 39] of Johannes Braams, some were adapted, others were made available by gentle users or taken from specific packages, like ArabTEX [154, 155], Arabi [135], ArmTEX (armenian) [101], BangTEX (bangla) [202], CervanTEX (spanish) [30]. Devanāgarī for TEX [204], ethiop [29], guarani [32], malayalam [3] and omal [4], MonTEX (mongol) [97, 100], CJK (chinese, corean-hangûl/hanja, japanese, thai) [167, 168], PLATEX [199, 247] (polish), SLATEX [181] (swedish), FarsiTEX [109] (farsi or iranian), or vietnam — latvian (letton), `greek-mono`, `greek-polydemo`, `greek-polykatha`, `polish2`, `russian2m`, `russian2o` and `spanish3` need *Lambda* (**Λ**), i.e., the Omega (**Ω**) version of LATEX, (see [150]), 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 36.

But for some oriental languages [11], 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 [12], 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.

[11] Mainly for chinese, farsi, hangûl (korean), hanja (korean), japanese, `malayalam-omega`, thai and some variants of russian.

[12] The extension `.mlo` means *minitoc language object*.

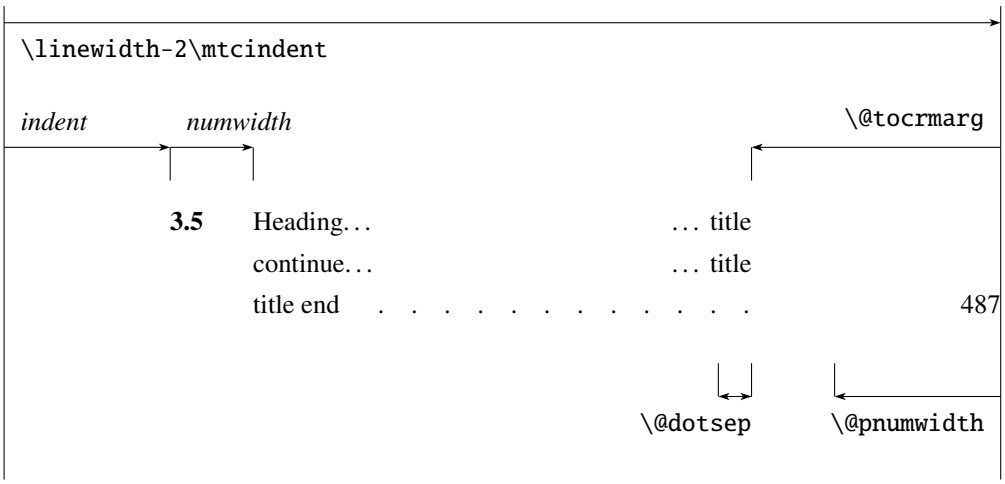


Figure 1.2: Layout of a ToC (LoF, LoT) entry

1.4.13 Altering the layout of the mini-tables

The layout of a mini-table is described in the figure 1.2 (this figure is adapted from [250]), which defines some internal commands (these are not *dimensions*, but LaTeX commands, created by `\newcommand`, modifiable via `\renewcommand`).

- `\@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. As the real size of `1em` is font dependent, the separation between the dots may vary if you use different fonts for different types of entries in the mini-tables.
- `\@pnumwidth`, is the width of the space reserved for the page number. It is a LaTeX command containing the representation of a length (e.g., `1.55em`).
- `\@tocrmarg`, is the distance (margin) between the right border of the table and the end of the dotted line. It should be larger than `\@pnumwidth`, and can be a rubber length (i.e., contain 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.

`\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 `\mtcsetformat` command:

`\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

Table 1.8: Horizontal rules

	rules in		no rules in	defaults for		
				`book`	`report`	`article`
`\ptcrule`	parttocs	`\noptcrule`	parttocs	**N**	**N**	**Y**
`\plfrule`	partlofs	`\noplfrule`	partlofs	**N**	**N**	**Y**
`\pltrule`	partlots	`\nopltrule`	partlots	**N**	**N**	**Y**
`\mtcrule`	minitocs	`\nomtcrule`	minitocs	**Y**	**Y**	**(NA)**
`\mlfrule`	minilofs	`\nomlfrule`	minilofs	**Y**	**Y**	**(NA)**
`\mltrule`	minilots	`\nomltrule`	minilots	**Y**	**Y**	**(NA)**
`\stcrule`	secttocs	`\nostcrule`	secttocs	**(NA)**	**(NA)**	**Y**
`\slfrule`	sectlofs	`\noslfrule`	sectlofs	**(NA)**	**(NA)**	**Y**
`\sltrule`	sectlots	`\nosltrule`	sectlots	**(NA)**	**(NA)**	**Y**

(NA) = not available.

Table 1.9: Page numbers

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

`\@dotsep`), `pagenumwidth` (for `\@pnumwidth`), or `tocrightmargin` (for `\@tocrmarg`) keywords; so:

```
\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 `tocrightmargin` (for `\@tocrmarg`) parameter should obviously be greater than the `pagenumwidth` parameter (this appears in the figure 1.2 on the preceding page).

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

1.5 Special Features

1.5.1 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 on the page before 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; if the first argument is a star (*), all mini-tables are affected; the keywords on and off have the following synonyms [13] :

- 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.

1.5.2 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 [13] :

- 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.

If the first argument is a star (*), all mini-tables are affected.

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 1.4.13 on page 42.

[13] O and o are the letter O, 0 is the zero digit.

Table 1.10: Features for mini-tables

Type	Command	Default
parttoc	\beforeparttoc	\cleardoublepage
parttoc	\afterparttoc	\cleardoublepage
parttoc	\thispageparttocstyle	\thispagestyle{empty}
partlof	\beforepartlof	\cleardoublepage
partlof	\afterpartlof	\cleardoublepage
partlof	\thispagepartlofstyle	\thispagestyle{empty}
partlot	\beforepartlot	\cleardoublepage
partlot	\afterpartlot	\cleardoublepage
partlot	\thispagepartlotstyle	\thispagestyle{empty}
minitoc	\beforeminitoc	\empty
minitoc	\afterminitoc	\empty
minitoc	\thispageminitocstyle	\empty
minilof	\beforeminilof	\empty
minilof	\afterminilof	\empty
minilof	\thispageminilofstyle	\empty
minilot	\beforeminilot	\empty
minilot	\afterminilot	\empty
minilot	\thispageminilotstyle	\empty
secttoc	\beforesecttoc	\empty
secttoc	\aftersecttoc	\empty
secttoc	\thispagesecttocstyle	\empty
sectlof	\beforesectlof	\empty
sectlof	\aftersectlof	\empty
sectlof	\thispagesectlofstyle	\empty
sectlot	\beforesectlot	\empty
sectlot	\aftersectlot	\empty
sectlot	\thispagesectlotstyle	\empty

\mtcsetfeature{*mini-table*}{before|after|pagestyle}{*command*}
Modifies the features for a mini-table.

1.5.3 Features for parttocs and other mini-tables

By default, a parttoc (or a partlof or a partlot), in a book- or report-class document, is preceded and followed by a \cleardoublepage (which acts like \clearpage in a one-side document), and has a page style of empty. Since version #32, you can modify this behaviour by redefining the commands of table 1.10, whose meaning is obvious. A feature defined as \empty does nothing.

`\mtcsetfeature` The command:

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

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).

1.5.3.1 Remark about page styles [14]

The default commands for part-level mini-tables page styles are defined as being simply a standard `\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 [15]:

```
\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`.

1.5.4 The "Chapter 0" Problem (solved)

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

[14] This remark is taken and adapted from a draft of the second edition of the JMPL [19], by Benjamin Bayart, where he comments the minitoc package.

[15] 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.

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

\firstpartis
\firstchapteris
\firstsectionis

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

```
\firstchapteris{⟨N⟩}
```

before the \dominitoc and analogous commands. ⟨*N*⟩ 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.

1.5.4.2 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 now they just produce harmless warnings.

1.5.5 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 package [253] 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).

Note that the same problems appear if you use one of the scrbook, scrreprt or scrartcl KOMA-Script classes [147, 195] with some options (liststotoc, liststotocnumbered, bibtotoc, bibtotocnumbered, and idxtotoc). The solutions are the same ones.

\mtcaddchapter
\tableofcontents
\listoffigures
\listoftables

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

\adjustmtc
\bibliography

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

\printindex
\addcontentsline
\mtcaddchapter
\mtcfixglossary

For the glossary, it is a bit more complicated, you should add the following commands just after the \printglossary command:

```
\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.

\printindex
\addcontentsline
\mtcaddchapter
\mtcfixindex

For the index, it is like for the glossary, you should add the following commands just after the \printindex command:

```
\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.

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

And proceed *with extreme care*, tracking in the *document*.log file the insertion of .mtc⟨*N*⟩ files (and siblings). They are some examples in the mtc-add.tex (see section 4.3 on page 89) and mtc-ads.tex (see section 4.4 on page 94) files distributed with minitoc. The mtc-ads.tex example shows how much that problem is difficult.

1.6 The notoccite option

\cite This option loads the notoccite package [9] (by Donald Arseneau). It avoids problems with \cite commands in sectionning commands or captions: if you then run BibTeX using the unsrt (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. See also http://www.tex.ac.uk/cgi-bin/texfaq2html?label=bibtocorder.

1.7 The listfiles and nolistfiles options

The listfiles package option creates a list of the minitoc auxiliary files into the file *document*.maf[16]. This feature can help you to remove these auxiliary files which are no more necessary after the LaTeX run. Under Unix or Linux, you can use something like:

```
cat document.maf | \tool{xargs} -i -t \rm {}
```

1.8 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 *document*.log file and emits a warning. The hints written in the *document*.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, but you can inhibit it via the nohints option. The following (potential) problems are currently detected:

\part
\@part
\spart
\@spart
\chapter
\@chapter
\schapter
\@schapter
\section
\@sect
\@ssect

- Alteration of some of the following commands [17]: \part, \@part, \@spart, \chapter, \@chapter, \@schapter, \section, \@sect, and \@ssect. Note that the hyperref (see section 2.17 on page 59) package alters these commands at \begin{document}, hence this problem might be reported if you use this package, but these alterations seem harmless. Note that the hyperref must be loaded *before* minitoc.
- Presence of the following packages or classes, which need some precautions: amsbook (class), memoir (class), appendix, placeins (beware to its options and its release date (2005/04/18 at least)), scrbook (class), scrreprt (class), scrartcl (class), tocbibind, and tocloft.

[16] This package option is now (since version #48) the default (list created).

[17] The commands containing the "@" character in their names are internal commands of LaTeX, of a package or of a class; they are sometimes altered by another packages; reconsider then the loading order of the packages.

- Presence of the following packages or classes, which, unfortunately, are incompatible with the minitoc package: amsart (class), amsproc (class), alphanum, jura (class), titlesec, and titletoc [18].

`\parttoc` `\doparttoc` `\sectlot` `\dosectlot` `\tableofcontents` `\listoftables` `\sectlof` `\sectlot`

- Usage of `\parttoc` without calling `\doparttoc`, ... , usage of `\sectlot` without calling `\dosectlot`; or the reverse.

  Usage of `\parttoc` without calling `\[fake]tableofcontents`, ... , of `\sectlot` without calling `\[fake]listoftables`.

  Usage of `\sectlof` and/or `\sectlot` without using the `insection` package option of minitoc (or the placeins package without its `section` option).

- If you are using short extensions (because of your operating system or the `shortext` package option, see section 1.9 on the next page) and go beyond the limit of 99 parts, chapters or sections, the `hints` package option displays a warning.

`abstract` `\mtcaddchapter` `\mtcaddsection`

- If the abstract package [251] (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 is detected and you should add a `\mtcaddchapter[]` or a `\mtcaddsection[]` command after your `abstract` environment.

- If the sectsty package [182] (by Rowland McDonnell) is used, it must be loaded *before* the minitoc package. The interaction has been pointed out by Bil Kleb.

- If the varsects package [228] (by Daniel Taupin†) is used, it must be loaded *before* the minitoc package.

- If the fncychap package [170] (by Ulf A. Lindgren) is used, it must be loaded *before* the minitoc package.

- If the hangcaption package [138] (by David M. Jones) is used, it must be loaded *before* the minitoc package.

- If the quotchap package [232] (by Karsten Tinnefeld) is used, it must be loaded *before* the minitoc package.

- If the romannum package [259] (by Peter R. Wilson) is used, it must be loaded *before* the minitoc package.

- If the sfheaders package [172] (by Maurizio Loreti) is used, it must be loaded *before* the minitoc package.

- If the alnumsec package [152] (by Frank Küster) is used, it must be loaded *before* the minitoc package.

- If the captcont package [95] (by Steven Douglas Cochran) is used, it must be loaded *before* the minitoc package.

[18] The titlesec package redefines the sectionning commands in a way completely alien to the standard LaTeX way; hence minitoc and titlesec-titletoc are fundamentaly 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`⟨*N*⟩	`.P`⟨*N*⟩
partlof	`.plf`⟨*N*⟩	`.G`⟨*N*⟩
partlot	`.plt`⟨*N*⟩	`.U`⟨*N*⟩
minitoc	`.mtc`⟨*N*⟩	`.M`⟨*N*⟩
minilof	`.mlf`⟨*N*⟩	`.F`⟨*N*⟩
minilot	`.mlt`⟨*N*⟩	`.T`⟨*N*⟩
secttoc	`.stc`⟨*N*⟩	`.S`⟨*N*⟩
sectlof	`.slf`⟨*N*⟩	`.H`⟨*N*⟩
sectlot	`.slt`⟨*N*⟩	`.V`⟨*N*⟩

- If one of the caption [224], caption2[19] [223], (both written by Axel Sommerfeldt), ccaption [255] (written by Peter R. Wilson), or mcaption [131] (written by Stephan Hennig), packages is used, it must be loaded *before* the minitoc package.
- If you try to insert empty mini-tables, the `hints` option gives a global warning (except if you used also the `nocheckfiles` option, see section 1.3.3 on page 28).

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

- If you use one of the obsolete commands (`\firstpartis`, `\firstchapteris`, or `\firstsectionis`), a warning is issued for each use, of course, but also a global hint as reminder.
- If you invoke a same preparation command more than once, an informative hint is issued for each spurious invocation.

1.9 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`⟨*N*⟩ extensions will become `.M`⟨*N*⟩, where ⟨*N*⟩ is the absolute chapter number. The extensions `.mlf`⟨*N*⟩ and `.mlt`⟨*N*⟩ become `.F`⟨*N*⟩ and `.T`⟨*N*⟩. The `.ptc`⟨*N*⟩ extensions become `.P`⟨*N*⟩, where ⟨*N*⟩ is the absolute part number. The extensions `.plf`⟨*N*⟩ and `.plt`⟨*N*⟩ become `.G`⟨*N*⟩ and `.U`⟨*N*⟩. The `.stc`⟨*N*⟩ extensions become `.S`⟨*N*⟩, where ⟨*N*⟩ is the absolute section number. The extensions `.slf`⟨*N*⟩ and `.slt`⟨*N*⟩ become `.H`⟨*N*⟩ and `.V`⟨*N*⟩. 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 1.8 on page 49) will report such situations. See also section 2.5 on page 56.

[19] This package is obsolete; now use the caption package.

1.10 Why several LaTeX 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 LaTeX 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 LaTeX run, the mini, part- and section-level tables and lists should be correct (see figure 2.1 on page 55).

1.11 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 *document*`.log` file. At least two 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 commands which are now spurious.

Chapter 2

Frequently Asked Questions

Contents

Figures

Tables

2.0 Introduction

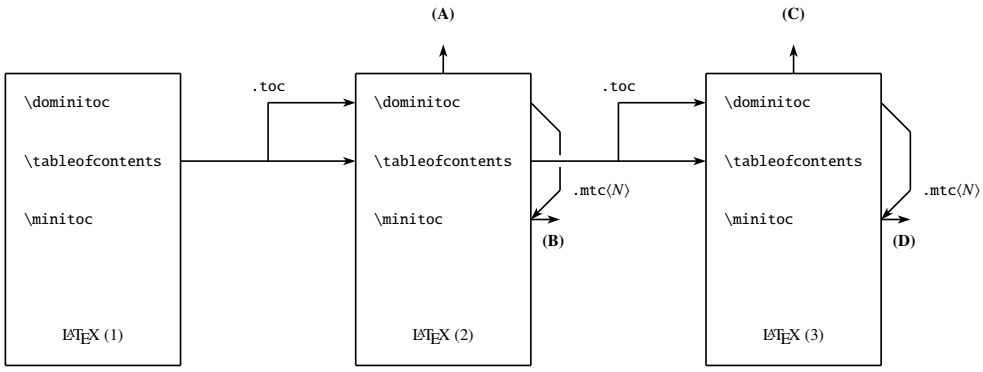
Here is a list of problems and frequently asked questions about the minitoc.sty package. If the version has a number less than 51, please upgrade to version #51. This list is also given in the `minitoc.bug` file, in pure text form. The numbering of this list is done by date of the first occurrence of the question.

If a problem arises, it is often wise to: a) use the `hints` option (see section 1.8 on page 49), which is activated by default, and b) read the *document*`.log` file, which may contain pertinent messages. If you do not find a solution, ask a question on an adequate news group, like `fr.comp.text.tex` (in french) or `comp.text.tex` (in english) preferably, groups which I try to follow, or send me a mail in last ressort (please join a minimal but complete example reproducing the problem).

2.1 Avoiding a page break near the rules before and after a mini-table

`\enlargethispage`

This problem seemed solved since version #8, but version #12 added better fixes. You may have to make some final tuning with `\enlargethispage`. See the LaTeX manual [156]. The needspace package [249] may also be useful.



(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

2.2 Implementing others layouts for a mini-table

Suggestions are welcome, but look at the section 1.4.13 on page 42.

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 LaTeX at least three times (yes, it is time consuming...). See figure 2.1. Running with the mtcoff package 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 at least 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 `shortext`. Then you will get first the autoconfiguration messages, then a message saying that you will use short extensions. But then 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 that number 1. And look at "Special Entries for TOC, LOF, LOT, Bibliography and Index", section 1.5.5 on page 47.

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 "part-tocs" 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 [1]. See also section 2.24 on page 63.

2.8 Compatibility with LaTeX versions

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

[1] 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.

```
Package minitoc Warning: <W0021>
Undefined command ... \@inputcheck ...
Your version of latex.tex is obsolete. Trying to continue...
```

or:

```
Package minitoc Warning: <W0022>
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 (or a recent distribution) 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 BibTeX's dealing with `.aux` files. Look at the chapterbib[12], bibunits [121], multibib [122], bibtopic [16], and splitbib [179] packages. Version #13 has implemented basic minilofs and minilots. Minibbls are not the aim of this package[2].

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 1.3.3 on page 28) package option impractical to implement. Note that the many files `*.mtc*`, etc., may be deleted after the LaTeX run. They are rebuilt by the preparation commands (like `\dominitoc` 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. Since version #44, the `listfiles` package option is available to create a list of these auxiliary files; see section 1.7 on page 49.

[2] See http://www.tex.ac.uk/cgi-bin/texfaq2html?label=multbib

These files contain the mini-tables extracted from the `.toc`, `.lof`, and `.lot` files. They are no more useful after the LaTeX run. If you run 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 51 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` tool [22] (written in Python) provided by Emmanuel Beffara:

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

2.11 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 LaTeX2.09

`\protect` `\contentsline`

The more recent version of LaTeX 2_ε adds `\protect` before `\contentsline` in the `.toc`, `.lof` and `.lof` files. The version #17 of minitoc attempts to be compatible with LaTeX 2_ε and LaTeX2.09. This will be the *last* version usable with LaTeX2.09. Versions #18 and later are LaTeX 2_ε specific, and no more compatible with LaTeX2.09, 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 [147, 195] (since version #43).

2.15 The secttocs are wrong

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

2.16 Removing 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 1.4.13 on page 42.

2.17 Using the hyperref package with minitoc

Since version #31, minitoc works correctly with the powerful hyperref package [214], thanks to Heiko Oberdiek, who used the work of Bernd Jaehne and Didier Verna. *Hence the minitoc-hyper package is now obsolete and should no more be used. It it still present on the CTAN archives for compatibility with old documents.* 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 LaTeX run, then re-LaTeX 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 the documents `minitoc.dtx` and `fminitoc.dtx` show (not so) basic examples 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 LaTeX run with version #31 or higher will produce a lot of errors (the next run should be ok). See also the section 2.17 on the preceding page.

2.19 A local table of contents for the set of appendices

`\doparttoc` `\tableofcontents` `\appendix` `\part` `\parttoc` `\addtocontents` `\protect` `\setcounter` `\chapter` `\partbegin`

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 `\part` subdivision of the document and ask for a table of contents at the `\part` level:

```
\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}
```

See also section 2.25 on page 63.

2.20 Use with the appendix package

`appendices` `\addcontentsline` `\adjustmtc` `\adjuststc`

If you use the appendix package [252] (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 respective appendices. In fact, the environnement opening `\begin{appendices}` hides a `\addcontentsline` command for a chapter or a section, putting trouble in the numbering of minitocs or secttocs. Several 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. An 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.

These two solutions may be modified by replacing `\adjustmtc` by the sequence:

```
\addtocontents{toc}{\chapterend}
OR
\addtocontents{toc}{\sectend}
```

when it is necessary to delimit the end of the preceding chapter or section.

A rather elegant solution is to add an entry into the TOC via the `\addappheadtotoc` command offered by the appendix package. As this entry is a chapter-level (or section-level) entry, it delimits correctly the end of the preceding chapter or section.

2.21 Use with the tocloft package

`\mtcsetfont` (This answer is given in the documentation of the tocloft package [250].) The tocloft (by Peter R. Wilson) and minitoc packages have an unfortunate interaction[3], 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}
```

in the preamble.

[3] Discovered by Lyndon Dudding.

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. See also section 2.22.

2.22 Use with the memoir class

The memoir class [257, 258] offers basically the functionnalities 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 (sections 2.20 on page 60, 1.5.5 on page 47, and 2.21 on the page before respectively). If your version of the memoir class is recent, the syntax of the `\chapter` command is different and the memoir class *could be no more compatible* with the minitoc package, but a patch is inserted to fix the problem. Hopefully, if your version of the memoir is more recent than 2005/09/25, the patch is no more necessary.

If you are using the memoir class (or the tocloft package), the `\mtcsetfont` command has no effect (`\mtcsettitlefont` works); you should use the font commands which are specific of the memoir class (or of the tocloft package).

If you still want to use the `\mtcsetfont` commands while using the memoir class (or of the tocloft package), you must disable the memoir/tocloft font commands. This is done by the following commands:

```
\let\cftpartfont\relax
\let\cftchapterfont\relax
\let\cftsectionfont\relax
\let\cftsubsectionfont\relax
\let\cftsubsubsectionfont\relax
\let\cftparagraphfont\relax
\let\cftsubparagraphfont\relax
\let\cftfigurefont\relax
\let\cftsubfigurefont\relax
\let\cfttablefont\relax
\let\cftsubtablefont\relax
```

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

\mtcsetfont
\mtcsettitlefont

Since version #41, the \mtcsetfont and \mtcsettitlefont commands are available. You do not need anymore to know \mtcSSSfont, \ptifont, etc.

\mtcsettitle

Since version #42, the \mtcsettitle command is available. You do not need anymore to know \mtctitle, \slttitle, etc.

\mtcsetdepth

Since version #43, the \mtcsetdepth command is available. You do not need anymore to know the counters minitocdepth, sectlotdepth, etc.

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

This problem has been pointed out by Henri Massias.

\mtcaddchapter

Unfortunately, the amsart and amsproc document classes are incompatible with minitoc. The amsbook 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 Hiding some entries from the main table of contents

mtchideinmaintoc
mtchideinmainlof
mtchideinmainlot

It is a problem similar to that of section 2.19 on page 60. 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). You can look at the `mtc-apx.tex` example file:

```
⟨*mtc – apx⟩
\documentclass[oneside]{book}
\ProvidesFile{mtc-apx.tex}%
  [2007/01/04]
\usepackage{lipsum} % provides filling text
\usepackage{tocbibind} % adds some entries in the main TOC.
\usepackage[tight,listfiles]{minitoc}
\setcounter{minitocdepth}{3}
\setcounter{parttocdepth}{3}
\begin{document}
\doparttoc
\dominitoc
\tableofcontents
\mtcaddchapter % because tocbibind adds a chapter entry in the TOC
\chapter{First}
\minitoc
First chapter
\section{First section}
\lipsum[1]
\section{Second section}
\lipsum[2]
\chapter{Second}
\minitoc
Second
\section{First section of second chapter}
\lipsum[3]
\section{Second section of second chapter}
\lipsum[4]
\appendix      % begins the appendices
\addcontentsline{toc}{part}{Appendices} % adds a part entry in the TOC
\adjustptc     % fixes the parttoc counter ptc
\parttoc       % adds a partial toc for the appendices
\begin{mtchideinmaintoc}[-1] % hides the details of the
%              % appendices in the main TOC, but chapter-level
%              % entries would be still visible if you use 0
%              % in place of -1 as optional argument.
\chapter{First appendix}
\minitoc
First appendix
\section{First section}
\lipsum[5]
\section{Second section}
\lipsum[6]
\chapter{Second appendix}
\minitoc
Second appendix
\section{First section of second appendix}
\lipsum[7]
\section{Second section of second appendix}
\lipsum[8]
\end{mtchideinmaintoc} % end of hiding
```

```
\end{document}
⟨/mtc – apx⟩
```

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

Note that the position of the end of these environments must be adjusted to include a page break (like the one done by a `\chapter` command), else the restore command might be inserted to early into the `.toc`, `.lof` or `.lot` file. There is an example file (`mtc-hi1.tex`):

```
⟨*mtc – hi1⟩
\documentclass{report}
\ProvidesFile{mtc-hi1.tex}%
  [2007/01/04]
\usepackage%
  [tight,listfiles]{minitoc}
\begin{document}
\dominilof
\dominilot
\listoffigures
\listoftables
\chapter{First}
\minilof
\minilot
\begin{figure}
\caption{AAAA1}
\end{figure}
\begin{figure}
\caption{AAAA2}
\end{figure}
\begin{table}
\caption{TAAAA1}
\end{table}
\begin{table}
\caption{TAAAA2}
\end{table}
\chapter{Second}
\minilof
\minilot
%%----------------------
```

We begin the hiding of figure entries in the list of figures and of table entries in the list of tables. In this document, we use the environment forms.

```
\begin{mtchideinmainlof}
\begin{mtchideinmainlot}
\begin{figure}
\caption{BBBB1}
\end{figure}
\begin{figure}
\caption{BBBB2}
\end{figure}
\begin{table}
\caption{TBBBB1}
\end{table}
\begin{table}
\caption{TBBBB2}
\end{table}
\chapter{Third}
```

We terminate the hiding of figure entries in the list of figures and of table entries in the list of tables. In this document, we use the environment forms.

```
\end{mtchideinmainlot}
\end{mtchideinmainlof}
%%----------------------
\minilof
\minilot
\begin{figure}
\caption{CCCC1}
\end{figure}
\begin{figure}
\caption{CCCC2}
\end{figure}
\begin{table}
\caption{TCCCC1}
\end{table}
\begin{table}
\caption{TCCCC2}
\end{table}
\end{document}
⟨/mtc – hi1⟩
```

But it is also possible to use *commands* in place of these environments: you place a `\mtchideinmainlof` (or `\mtchideinmainlot`) command in the first figure (or table) to

hide, *before* its caption and a `\endmtchideinmainlof` (or `\endmtchideinmainlot`) command at the end of the last figure (or table) to hide, *after* its caption, like in this example file (`mtc-hi2.tex`):

```
⟨*mtc – hi2⟩
\documentclass{report}
\ProvidesFile{mtc-hi2.tex}%
  [2007/01/04]
\usepackage%
  [tight,listfiles]{minitoc}
\begin{document}
\dominilof
\dominilot
\listoffigures
\listoftables
\chapter{First}
\minilof
\minilot
\begin{figure}
\caption{AAAA1}
\end{figure}
\begin{figure}
\caption{AAAA2}
\end{figure}
\begin{table}
\caption{TAAAA1}
\end{table}
\begin{table}
\caption{TAAAA2}
\end{table}
\chapter{Second}
\minilof
\minilot
```

We begin the hiding of figure entries in the list of figures and of table entries in the list of tables. In this document, we use the command forms: a command is inserted *before* the caption of the first "hidden" entry.

```
\begin{figure}
\mtchideinmainlof     % <--
\caption{BBBB1}
\end{figure}
\begin{figure}
\caption{BBBB2}
\endmtchideinmainlof % <--
\end{figure}
\begin{table}
\mtchideinmainlot     % <--
\caption{TBBBB1}
\end{table}
```

We terminate the hiding of figure entries in the list of figures and of table entries in the list of tables. In this document, we use the command forms: a command is inserted *after* the caption of the *last* "hidden" entry.

```
\begin{table}
\caption{TBBBB2}
\endmtchideinmainlot % <--
\end{table}
\chapter{Third}
\minilof
\minilot
\begin{figure}
\caption{CCCC1}
\end{figure}
\begin{figure}
\caption{CCCC2}
\end{figure}
\begin{table}
\caption{TCCCC1}
\end{table}
\begin{table}
\caption{TCCCC2}
\end{table}
\end{document}
⟨/mtc – hi2⟩
```

This method, recommended while more delicate to apply, is more reliable in delimiting the hiding domain: its solves the problem of the asynchronism between the writing of floats and the writing of the normal text.

2.26 Defining your own .mld file

\mtcsettitle
\mtcselectlanguage

First, you should not directly modify one of the distributed .mld and .mlo 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 [251] (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 [182] (by Rowland McDonnell) is used, it must be loaded *before* the minitoc package, because it alters (redefines) the sectionning commands. Of course, the 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 section/subsection/... 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 witten 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 (the default font choices are given in table 1.6 on page 35), hence one "em" has different sizes in these two fonts and the alignement 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 in a package or via a command defined by a package or between `\makeatletter` and `\makeatother`, because these commands have a `@` in their names; you must use `\renewcommand*` to redefine these commands.
- Use the tocloft package [250] to change the indentation, with font independent units. But then see section 2.21 on page 61.
- Use the same font for the section and subsection entries in the minitocs, using the `\mtcsetfont` command (see section 1.4.9 on page 39) or redefining the `\mtcSfont`, `\mtcSSfont`, `\mtcSSSfont`, `\mtcPfont` and `\mtcSPfont` commands (see table 1.6 on page 35), or similar.

2.30 Useful precautions with starred sectionning commands

- The headers are not modified by `\part*`, `\chapter*` or `\section*`; it is necessary to use `\markboth` or `\markright` to get correct page headers for the current and following pages.
- If you need an entry in the table of contents for a `\chapter*` or a `\section*` command, you must use `\mtcaddchapter[`*title*`]` or `\mtcaddsection[`*title*`]` *after* the starred sectionning command. If you need an entry in the table of contents for a `\part*` command, the page number in the table of contents would be wrong, because `\part*` implies a `\clearpage` or a `\cleardoublepage` before the first page of the part. Use the sequence

```
\cleardoublepage     % \clearpage if openany option.
\mtcaddpart[title]
\part*[title]
```

2.31 Use with packages for captions

If one of the caption [224], caption2[4] [223], (both written by Axel Sommerfeldt), ccaption [255] (by Peter R. Wilson), or mcaption [131] (by Stephan Hennig), packages is used, it must be loaded *before* the minitoc package, because such packages alter (redefine) the commands listing figures and tables. Of course, the `hints` option detects this problem.

2.32 Bad interaction minitoc/hyperref/memoir

When the minitoc and hyperref [214] packages are used in a document of class memoir [257, 258], the chapter header "Chapter" does not appear on the first page of the chapter.

This problem is fixed in version #44 of minitoc.

2.33 Use with the varsects package

If the varsects package [228] (by Daniel Taupin[†]) is used, it must be loaded *before* the minitoc package, because it alters (redefines) the sectionning commands. Of course, the `hints` option detects this problem.

2.34 Initial font settings

The setting of the fonts in the mini-tables is a rather complex problem. If we take the parttocs as an example, there is a `\ptcfont` font-command which is used for two purposes[5]:

- First, to be used as default value for some other font-commands (like `\ptcSPfont`). As its default value is used in the initialization of the minitoc package, the value of these other commands is *not altered* if you modify `\ptcfont`. You must modify these commands one at a time.
- Second, it is invoked at the beginning of each parttoc, partlof or partlot to set an initial font command. Then each entry of the mini-table calls its own font command (like

[4] This package is obsolete; now use a recent version of the caption package.
[5] The same remarks apply to the other mini-tables.

\ptcSPfont). Thus, if you modify \ptcfont, you can obtain a global effect on the fonts in the parttocs, partlofs, and partlots. So you can play with the various parameters of the fonts (family, shape, series, size), if you want fancy mini-tables; but it is rather difficult.

In the initialization of the minitoc package, we have a sequence of commands:

```
\let\ptcSSfont\ptcfont      % (subsections)
\let\ptcSSSfont\ptcfont     % (subsubsections)
\let\ptcPfont\ptcfont       % (paragraphs)
\let\ptcSPfont\ptcfont      % (subparagraphs)
\let\plffont\ptcfont        % (figures)
\let\plfSfont\ptcfont       % (subfigures)
\let\pltfont\ptcfont        % (tables)
\let\pltSfont\ptcfont       % (subtables)
```

to define some default fonts. But this sequence is executed only once. If you alter \ptcfont, the modification is not applied to these font commands. The command \ptcfont is invoked at the beginning of each parttoc. \ptcCfont is invoked for each chapter entry in a parttoc (\ptcSfont for each section entry, etc.). So \ptcfont can be used to define some global characteristics for the fonts in the parttocs, while \ptcCfont (etc.) can be used to customize the fonts for each level of entries.

Note that if you say:

```
\def\ptcSSfont{\ptcfont}    % (subsections)
\def\ptcSSSfont{\ptcfont}   % (subsubsections)
\def\ptcPfont{\ptcfont}     % (paragraphs)
\def\ptcSPfont{\ptcfont}    % (subparagraphs)
\def\plffont{\ptcfont}      % (figures)
\def\plfSfont{\ptcfont}     % (subfigures)
\def\pltfont{\ptcfont}      % (tables)
\def\pltSfont{\ptcfont}     % (subtables)
```

after loading the minitoc package, these font commands will be "associated" to \ptcfont, hence if you modify \ptcfont (by via \mtcsetfont{parttoc}{*}{...} or \renewcommand), they will follow the modification. But if you modify one of these commands via \renewcommand or \mtcsetfont{parttoc}{subsection}{...} (subsection is an example), the association is broken. But you could be more clever by saying something like

```
\mtcsetfont{parttoc}{subsection}{\ptcfont\itshape}
```

to preserve the association and modify only some parameters of a minitoc font command.

For levels above subsection (part, chapter and section), the fonts a more specific in general, but you can, of course, say something like `\def\ptcCfont{\ptcfont}` to make a similar association. You can even make other associations, like this:

```
% for high sectionning levels:
    \def\highlevelsfont{\rmfamily\bfseries\normalsize\upshape}
% for low sectionning levels:
    \def\lowlevelsfont{\rmfamily\mdseries\smallsize\upshape}
% then for each level:
    \def\ptcCfont{\highlevelsfont}
    \def\ptcSfont{\highlevelsfont}
    \def\ptcSSfont{\lowlevelsfont}
    \def\ptcSSSfont{\lowlevelsfont}
    \def\ptcPfont{\lowlevelsfont\itshape}
    \def\ptcSPfont{\lowlevelsfont\itshape}
```

Then you can redefine `\highlevelsfont` or `\lowlevelsfont` to act on several fonts in one step, but you must use `\renewcommand`. You cannot act on `\highlevelsfont` or `\lowlevelsfont` with `\mtcsetfont`.

Note that only the fonts for parttocs are used in the examples above; but, of course, the situation is the same for minitocs and secttocs. `\highlevelsfont` and `\lowlevelsfont` are macro names that you can choice, they are not part of the minitoc package.

2.35 Use with the KOMA-Script classes

If a KOMA-Script class [147, 195], compatible with minitoc (scrbook, scrreprt or scrartcl), is used, some class options may cause problems with the minitoc package, because these options add chapter or section entries in the table of contents. See section 1.5.5 on page 47. Of course, the `hints` option detects this problem.

2.36 Use with the jura class or the alphanum package

The jura class loads the alphanum package, which redefines the sectionning structure in a non-standard way, after the loading of the report class. This class and this package are incompatible with minitoc.

2.37 The `.mld` files and the babel package

If you are using the babel package [38, 39], you can automatize the loading of the `.mld` file by adding some code in the preamble of your document, like this:

```
\AtBeginDocument{%
  \addto\captionslanguage1{\mtcselectlanguage{language2}}}
```

where *language1* is the language name for babel and *language2* the language name for minitoc; there are often identical, but there are exceptions (when you use a locally customized `.mld` file, by example).

2.38 Use with the fncychap package

If the fncychap package [170] (by Ulf A. Lindgren) is used, it must be loaded *before* the minitoc package, because it alters (redefines) the sectionning commands. Of course, the `hints` option detects this problem.

2.39 Use with the quotchap package

If the quotchap package [232] (by Karsten Tinnefeld) is used, it must be loaded *before* the minitoc package, because it alters (redefines) the sectionning commands. Of course, the `hints` option detects this problem.

2.40 Use with the romannum package

If the romannum package [259] (by Peter R. Wilson) is used, it must be loaded *before* the minitoc package, because it alters (redefines) the numbering of the sectionning commands. Of course, the `hints` option detects this problem.

2.41 Use with the sfheaders package

If the sfheaders package [172] (by Maurizio Loreti) is used, it must be loaded *before* the minitoc package, because it alters (redefines) the sectionning commands. Of course, the `hints` option detects this problem.

2.42 Use with the alnumsec package

If the alnumsec package [152] (by Frank Küster) is used, it must be loaded *before* the minitoc package, because it alters (redefines) the numbering of the sectionning commands. Of course, the `hints` option detects this problem.

2.43 Use with the captcont package

If the captcont package [95] (by Steven Douglas Cochran) is used, it must be loaded *before* the minitoc package, because it alters (redefines) the caption commands. Of course, the `hints` option detects this problem.

2.44 Vertical space (gap) before or after parttocs, partlofs, and partlots titles

These vertical gaps were hard-coded like for the chapter heads in the book and report document classes. The values were 50pt and 40pt, but some users want to adjust them for the titles of the part-level mini-tables. Since version #45, these gaps are defined by `\mtcgapbeforeheads` and `\mtcgapafterheads`, with these defaults values. These commands apply globally to parttocs, partlofs and partlots. They are *commands*, *not* dimensions, so they must be modified via `\renewcommand` (but *not* via `\setlength`). An example of use is given in the `mtc-gap.tex` document file:

```
⟨*mtc – gap⟩
\documentclass[a4paper,oneside,12pt]{book}
\ProvidesFile{mtc-gap.tex}%
  [2007/01/04]
\usepackage{vruler} % to have a vertical graduation to note positions
%                   % (by Zhuhan Jiang)
\usepackage{txfonts}
\usepackage[english2,tight,listfiles]{minitoc}
\begin{document}
\setvruler[1cm][0][10][3][0][0pt][0pt][0pt][] % with vruler package
\doparttoc
\faketableofcontents
\part{First part}
```

A normal parttoc, with the normal gaps before and after it.

```
\parttoc
\chapter{First chapter of first part} \chapter{Second chapter of first part}
\part{Second part}
```

We set large gaps. Note the new position of the parttoc.

```
\renewcommand{\mtcgapbeforeheads}{100pt}
\renewcommand{\mtcgapafterheads}{80pt}
\parttoc
\chapter{First chapter of second part} \chapter{Second chapter of second part}
\part{Third part}
```

We set small gaps. Note the new position of the parttoc.

```
\renewcommand{\mtcgapbeforeheads}{20pt}
\renewcommand{\mtcgapafterheads}{10pt}
\parttoc
\chapter{First chapter of third part} \chapter{Second chapter of third part}
\end{document}
⟨/mtc – gap⟩
```

2.45 Vertical spacing before the bottom rule of a minitable

The little spacing between a minitable and its bottom rule is implemented as a vertical kern that should be sufficient to allow the descending parts of the letters of the last entry of the mini-table. The values should depend of the line spacing and of the font size. They are defined as macros that you can adjust by redefining them via `\renewcommand`. The (empirical) default values are given in table 2.1.

Table 2.1: Kernings before minitable bottom rules

Command	Default value
`\kernafterparttoc`	`\kern-1.\baselineskip\kern.5ex`
`\kernafterpartlof`	`\kern-1.\baselineskip\kern.5ex`
`\kernafterpartlot`	`\kern-1.\baselineskip\kern.5ex`
`\kernafterminitoc`	`\kern-.5\baselineskip\kern.5ex`
`\kernafterminilof`	`\kern-1.\baselineskip\kern0.ex`
`\kernafterminilot`	`\kern-1.\baselineskip\kern0.ex`
`\kernaftersecttoc`	`\kern-1.\baselineskip\kern.5ex`
`\kernaftersectlof`	`\kern-1.\baselineskip\kern.5ex`
`\kernaftersectlot`	`\kern-1.\baselineskip\kern.5ex`

2.46 Another interaction between the tocloft and minitoc packages

I encountered an interaction between tocloft *and* minitoc*. I want to force* minitoc *to not display the page numbers, but because of* tocloft *it doesn't. Here is an example code:*

```
\documentclass[12pt,a4paper]{book}
\usepackage{tocloft}
\usepackage{minitoc}
\begin{document}
\frontmatter
```

```
\dominitoc
\tableofcontents
\mainmatter
\chapter{Chapter}
\section{Section A}
\section{Section B}
\chapter{Second Chapter}
\mtcsetfont{minitoc}{section}{\normalfont\small}
\mtcsetpagenumbers{minitoc}{off}
\minitoc
\section{Section A}
\section{Section B}
\end{document}
```

If I comment the line loading the tocloft *package, I will get a minitoc without page numbers as I wanted.*

When using together tocloft and minitoc, the tocloft package must be loaded first, and its commands take precedence to format the entries in the TOC (and in minitocs). To suppress the page numbers, your should try the `\cftpagenumbersoff{`*XXX*`}` command (from tocloft), which is described in the `tocloft.pdf` documentation [250, pages 45-56]; *XXX* is the level of entry (`chapter`, `sec`, `subsec`, etc.). There are similar remarks about font related commands.

The tocloft package is more specialized in that job than minitoc, so if it is loaded, minitoc uses the tocloft tools. There is the corrected example (`mtc-tlo.tex`):

```
⟨*mtc – tlo⟩
\documentclass[12pt,a4paper]{book}
\ProvidesFile{mtc-tlo.tex}%
  [2007/01/04]
```

We must load tocloft *before* minitoc:

```
\usepackage{tocloft}
\usepackage[tight]{minitoc}
\begin{document}
```

We define the global font for the minitoc entries:

```
\mtcsetfont{minitoc}{*}{\normalfont\small}
\frontmatter
\dominitoc
\tableofcontents
```

For the section entries in the minitocs, we suppress the page numbers and change the font by using commands from the tocloft package:

```
\cftpagenumbersoff{sec}
\renewcommand{\cftsecfont}{\normalfont\small}
```

```
\mainmatter
\chapter{Chapter}
\section{Section A}
\section{Section B}
\chapter{Second Chapter}
\minitoc
\section{Section A}
\section{Section B}
\end{document}
⟨/mtc – tlo⟩
```

2.47 Use with the hangcaption package

If the hangcaption package [138] (by David M. Jones) is used, it must be loaded *before* the minitoc package, because it alters (redefines) the sectionning commands. Of course, the `hints` option detects this problem.

Chapter 3

Memento

Tables

Table 3.1: Package options

Options	Default	Meaning
`shortext`	`*NO*`	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`	Presence of leaders (dotted lines).
`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.
`listfiles`, `nolistfiles`	`listfiles`	Lists the minitoc auxiliary files into *document*`.maf`.
`hints`, `nohints`	`hints`	Adds hints in the *document*`.log` file. Useful to detect some problems.

Language options are listed in table 1.7 on page 36. Default: `english`.

Table 3.2: General commands

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

Table 3.3: Commands: part level

Command	Meaning
\doparttoc[*x*]	Before \[fake]tableofcontents if you use \parttoc*.
\dopartlof[*x*]	Before \[fake]listoffigures if you use \partlof*.
\dopartlot[*x*]	Before \[fake]listoftables if you use \partlot*.
\parttoc[*x*]	After each \part command for which a parttoc is needed*.
\partlof[*x*]	After each \part command for which a partlof is needed*.
\partlot[*x*]	After each \part command for which a partlot is needed*.
\setcounter{parttocdepth}{*depth*}	
	Depth of the following parttocs. Analog to tocdepth. Default: 2.
	Has no action on partlofs and partlots.
or:	
\mtcsetdepth{parttoc|partlof|partlot}{*depth*}	
	Idem, but can also act on partlofs and partlots.
\ptcindent	Left/right indentation of a partial table. Default: 24pt.
\ptcfont	Font command for parttoc.
	Default: \small\rmfamily\upshape\mdseries (article)
	or: \normalsize\rmfamily\upshape\mdseries (book, report).
\ptcCfont	Font command for parttoc, chapter entries.
	Default: \normalsize\rmfamily\upshape\bfseries.
\ptcSfont	Font command for parttoc, section entries.
	Default: \small\rmfamily\upshape\bfseries (article)
	or: \small\rmfamily\upshape\bfseries (book, report).
\ptcSSfont	Font command for parttoc, subsection entries**.
\ptcSSSfont	Font command for parttoc, subsubsection entries**.
\ptcPfont	Font command for parttoc, paragraph entries**.
\ptcSPfont	Font command for parttoc, subparagraph entries**.
\plffont	Font for partlof. Default: \small\rmfamily\upshape\mdseries.
\plfSfont	Font for partlof (subfigures). Default:
	\small\rmfamily\upshape\mdseries.
\pltfont	Font for partlot. Default: \small\rmfamily\upshape\mdseries.
\pltSfont	Font for partlot (subtables). Default:
	\small\rmfamily\upshape\mdseries.
\ptctitle	Title of parttocs. Default: Table of Contents.
\plftitle	Title of partlofs. Default: List of Figures.
\plttitle	Title of partlots. Default: List of Tables.
\ptifont	Font for partXXX titles.
	Default: \Large\rmfamily\upshape\bfseries (article)
	or: \LARGE\rmfamily\upshape\bfseries (book, report).
\mtcgapbeforeheads	Vertical gap before part-level mini-tables titles. Default: 50pt
\mtcgapafterheads	Vertical gap after part-level mini-tables titles. Default: 40pt

*: [*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 3.4: Commands: chapter level

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

*: [*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 3.5: Commands: section level

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

*: [*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 3.6: Commands for horizontal rules

Command	Meaning
\[no]ptcrule	Activates or inhibits rules in parttocs.
\[no]mtcrule	Activates or inhibits rules in minitocs.
\[no]stcrule	Activates or inhibits rules in secttocs.
\[no]plfrule	Activates or inhibits rules in partlofs.
\[no]mlfrule	Activates or inhibits rules in minilofs.
\[no]slfrule	Activates or inhibits rules in sectlofs.
\[no]pltrule	Activates or inhibits rules in partlots.
\[no]mltrule	Activates or inhibits rules in minilots.
\[no]sltrule	Activates or inhibits rules in sectlots.
\mtcsetrules{*mini-table*|*}{on|off}	Activates/inhibits horizontal rules in some or all mini-tables.
\kernafterparttoc	Vertical kerning between a parttoc and its bottom rule.
\kernafterpartlof	Vertical kerning between a partlof and its bottom rule.
\kernafterpartlot	Vertical kerning between a partlot and its bottom rule.
\kernafterminitoc	Vertical kerning between a minitoc and its bottom rule.
\kernafterminilof	Vertical kerning between a minilof and its bottom rule.
\kernafterminilot	Vertical kerning between a minilot and its bottom rule.
\kernaftersecttoc	Vertical kerning between a secttoc and its bottom rule.
\kernaftersectlof	Vertical kerning between a sectlof and its bottom rule.
\kernaftersectlot	Vertical kerning between a sectlot and its bottom rule.

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

Table 3.7: Commands for page numbers

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

By default, the page numbers are present.

Table 3.8: Commands for mini-tables features

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

Table 3.9: Preparation and insertion commands

Type	Phase	Level		
		part	chapter	section
table of contents	preparation	\doparttoc[*p*]	\dominitoc[*p*]	\dosecttoc[*p*]
	insertion	\parttoc[*p*]	\minitoc[*p*]	\secttoc[*p*]
list of figures	preparation	\dopartlof[*p*]	\dominilof[*p*]	\dosectlof[*p*]
	insertion	\partlof[*p*]	\minilof[*p*]	\sectlof[*p*]
list of tables	preparation	\dopartlot[*p*]	\dominilot[*p*]	\dosectlot[*p*]
	insertion	\partlot[*p*]	\minilot[*p*]	\sectlot[*p*]
all	preparation	.....	\mtcprepare[*p*]	.....

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 3.10: Adjustment commands

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

Table 3.11: Obsolete commands

Command	Meaning
\firstpartis{*N*}	*N* is the number of the first part.
\firstchapteris{*N*}	*N* is the number of the first chapter.
\firstsectionis{*N*}	*N* is the number of the first section.

These commands have no effect (except a harmless warning).

Table 3.12: Classes and packages needing some precautions with minitoc

	P/C	Names	Author(s)	Page(s)	Reference(s)
	P	abstract	Peter R. Wilson	50	[251]
	P	alnumsec	Frank Küster	50	[152]
∗	P	alphanum	Felix Braun	71	[85]
∗	C	amsart	$\mathcal{A}\mathcal{M}\mathcal{S}$	63	[6]
	C	amsbook	$\mathcal{A}\mathcal{M}\mathcal{S}$	63	[6]
∗	C	amsproc	$\mathcal{A}\mathcal{M}\mathcal{S}$	63	[6]
	P	appendix	Peter R. Wilson	60	[252]
	P	captcont	Steven Douglas Cochran	50	[95]
	P	caption	Axel Sommerfeldt	51	[224, 225]
	P	caption2	Axel Sommerfeldt	51	[223]
	P	ccaption	Peter R. Wilson	51	[255]
	P	fncychap	Ulf A. Lindgren	72	[170]
	P	hangcaption	David M. Jones	76	[138]
	P	hyperref	Sebastian Rahtz and Heiko Oberdiek	59	[196, 214, 215]
∗	C	jura	Felix Braun	71	[85]
	P	mcaption	Stephan Hennig	51	[131]
	C	memoir	Peter R. Wilson	62	[257, 258]
	P	notoccite	Donald Arseneau	49	[9]
	P	placeins	Donald Arseneau	29	[10]
	P	quotchap	Karsten Tinnefeld	50	[232]
	P	romannum	Peter R. Wilson	50	[259]
	C	scrartcl, scrbook and scrreprt	Frank Neukam, Markus Kohm, Axel Kielhorn, and Jens-Uwe Morawski	71	[147, 195]
	P	sectsty	Rowland McDonnell	67	[182]
	P	sfheaders	Maurizio Loreti	72	[172]
	P	subfig	Steven Douglas Cochran	31	[96]
	P	subfigure	Steven Douglas Cochran	31	[94]
∗	P	titlesec	Javier Bezos	50	[33]
∗	P	titletoc	Javier Bezos	50	[33]
	P	tocbibind	Peter R. Wilson	47	[253]
	P	tocloft	Peter R. Wilson	61, 74	[250]
	P	varsects	Daniel Taupin†	50	[228]

∗ Incompatible with minitoc.
C **C**lass.
P **P**ackage.

Chapter 4

Examples of documents

Contents

This chapter shows the code of some examples of documents. Some are extracted or derived from real documents, others are just demonstrations to illustrate problems or features. The code of specific example files has been shown earlier: `mtc-apx.tex` on page 64, `mtc-hi1.tex` on page 65, `mtc-hi2.tex` on page 66, `mtc-gap.tex` on page 73, and `mtc-tlo.tex` on page 75,

4.1 The mtc-2c.tex document file

This document shows the use of the minitoc package in a document with a two columns layout. The layout uses the multicol standard package [186] and its multicols environment. We set \mtcindent to zero. We test several combinations. If a minitoc is long enough to be splitted on both columns, the result may be funny.

```
⟨*mtc – 2c⟩
\documentclass[12pt,a4paper]%
    {report}
\ProvidesFile{mtc-2c.tex}%
  [2007/01/04]
\usepackage{multicol}
\usepackage{lipsum}
\usepackage%
  [tight,latin,listfiles]{minitoc}
\usepackage[latin]{babel}
\setlength{\mtcindent}{0pt}
\begin{document}
\dominitoc
\tableofcontents
```

Two first chapters with a standard chapter head, a minitoc on one full width column, then the body of the chapter on two columns:

```
\chapter{Primum capitulum}
\begin{multicols}{2}[\minitoc]
\section{Prima sectio}
\lipsum[1-2]
\section{Secunda sectio}
\lipsum[3-4]
\end{multicols}
\chapter{Secundum capitulum}
\begin{multicols}{2}[\minitoc]
\section{Prima sectio}
\lipsum[5-6]
\section{Secunda sectio}
\lipsum[7-8]
\section{Tertia sectio}
\lipsum[9-10]
\end{multicols}
```

A third chapter entirely on two columns, so the chapter head and the minitoc are in the first column:

```
\begin{multicols}{2}
\chapter{Tertium capitulum}
\minitoc
\section{Prima sectio}
\lipsum[11-12]
\section{Secunda sectio}
\lipsum[13-14]
\section{Tertia sectio}
\lipsum[15-16]
\end{multicols}
```

A fourth chapter, with the chapter head on one column, and the minitoc and the chapter body on two columns (the minitoc is in the first column):

```
\chapter{Quadrum capitulum}
\begin{multicols}{2}
\minitoc
\section{Prima sectio}
\lipsum[16-17]
\section{Secunda sectio}
\lipsum[18-19]
\section{Tertia sectio}
\lipsum[20-21]
\end{multicols}
\end{document}
⟨/mtc – 2c⟩
```

4.2 The mtc-2nd.tex document file

This document tests the french2.mld minitoc language definition file (section 13.60 on page 463) and its supporting code in the minitoc package. First, the preamble of the document uses the french2 minitoc package language option[1]:

```
⟨*mtc – 2nd⟩
%% Test de french2.mld:
%% « seconde » ou « deuxième » partie?
%% compilez 3 fois.
\documentclass{report}
\ProvidesFile{mtc-2nd.tex}%
  [2007/01/04]
\usepackage[french2,tight,listfiles]{minitoc}
\usepackage[french]{babel}
\usepackage{franc,frnew}
\usepackage[OT1,TS1,T1]{fontenc}
\usepackage[isolatin]{inputenc}
\usepackage{mypatches}
\begin{document}
```

We test if there is only two parts: if yes, we will use “seconde”. Else (zero, one, three or more parts), we will use “deuxième”.

```
\ifmtcsecondpart
Il n’y a que 2 parties: seconde.
\else
Il y a une partie ou plus de deux parties: deuxième.
\fi
\clearpage
```

The body of the document, with two parts:

```
\doparttoc
\faketableofcontents
\part{P1}
\parttoc
\chapter{P1C1}
\chapter{P1C2}
\part{P2}
\parttoc
\chapter{P2C1}
\chapter{P2C2}
```

If you want also a third part, comment out this line and recompile 3 times:

```
%% Commentez la ligne suivante puis recompilez 2 ou 3 fois.
\end{document}
```

[1] The franc, frnew and mypatches packages are local additions, provided with the minitoc package documentation sources.

```
\part{P3}
\parttoc
\chapter{P3C1}
\chapter{P3C2}
\end{document}
⟨/mtc – 2nd⟩
```

Observe the titles of the parttocs when the document has two then three parts. Note the changes after each compilation. You can play by adding and removing parts. It would be interesting to add a starred part, with starred chapters, at the beginning of the document, this part and these chapters having their entries in the TOC. By example, try to add the following lines after `\faketableofcontents`:

```
\part*{Partie étoilée}
\mtcaddpart[Partie étoilée]
\adjustptc[-2] % IMPORTANT <<<<<<<<<<<<<<<<<
\parttoc
\chapter*{Premier chapitre étoilé}
\mtcaddchapter[Premier chapitre étoilé]
\chapter*{Second chapitre étoilé}
\mtcaddchapter[Second chapitre étoilé]
```

and you will see how the parttoc of the new starred part is titled.

4.3 The `mtc-add.tex` document file

This document shows how to add special entries in the table of contents, and the interaction with the tocbibind package [253].

```
⟨*mtc – add⟩
\documentclass[12pt,a4paper]{report}
%%\documentclass[12pt,a4paper]{book}
\ProvidesFile{mtc-add.tex}%
  [2007/01/04]
\usepackage{url}
\usepackage{tocbibind}
\usepackage{makeidx}
\makeatletter
\newif\ifscan@allowed
\scan@allowedtrue
\makeatother
\def\dotfil{\leaders\hbox to.6em{\hss .\hss}\hfil}%
\def\pfill{\unskip~\dotfill\penalty500\strut\nobreak
           \dotfil~\ignorespaces}%
```

Load the minitoc package, or mtcoff.

```
\usepackage[tight,hints,listfiles]{minitoc}
%%\usepackage{mtcoff}

\makeindex
\begin{document}
```

We call the mini-table preparation commands:

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

We are using the tocbibind package to add special entries in the table of contents, so we must take the precautions specified in section 1.5.5 on page 47:

```
\tableofcontents
\mtcaddchapter
\listoffigures
\mtcaddchapter
\listoftables
\mtcaddchapter
```

For a chapter, we want a minitoc, a minilof and a minitoc:

```
\chapter{First chapter}\index{chapter!normal}
\minitoc
\mtcskip
\minilof
\mtcskip
\minilot
```

Then the text of the chapter, with sections, figures and tables:

```
\section{First section}

\begin{figure}[tp]
\caption{First figure}
\end{figure}
\begin{table}[tp]
\caption{First table}
\end{table}

\section{Second section}
A small nice citation from~\cite{dark}:\\
\index{small}\index{citation}\index{nice}\index{A}\index{a}%
\index{and}\index{bird}\index{But}\index{cannot}%
\index{claim}\index{great}\index{he}\index{I}%
\index{imagine}\index{it}\index{know}\index{land}%
\index{on}\index{once}\index{that}\index{to}\index{tree}%
\index{would}\index{yes}%
```

```
\textsf{A bird cannot land once on a great tree and claim to know it.
But I imagine that he would, yes.}\\
\hbox{}\hfill
Iain~M.~\textsc{Banks}~(1993),~\textsl{Against~a~dark~background.}%
\index{Iain}\index{Banks}\index{Against}\index{dark}\index{background}

\begin{figure}[tp]
\caption{Second figure}
\end{figure}
\begin{table}
\caption{Second table}
\end{table}
```

A starred chapter requires a special treatment; three solutions are possible. You can test variations on the `\mtcaddchapter` command. Just uncomment one (and only one) of the `\mtcaddchapter` commands after `\chapter*` in the source code of `mtc-add.tex`. For each case, look at the Table of Contents and the involved chapter.

```
\chapter*{Second chapter, starred}
\index{chapter!starred}
%% UNCOMMENT ONE AND ONLY ONE OF THE 3 FOLLOWING LINES
\mtcaddchapter[Second chapter, starred] % OK
%%\mtcaddchapter[~]    % produces a (strange) correct result. OK
%%\addcontentsline{toc}{xchapter}{}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%\mtcaddchapter[]     % BAD SOLUTION
%%\mtcaddchapter       % BAD SOLUTION
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\index{tests}

This is a starred chapter; you can test here variations on
the \verb|\mtcaddchapter| command. Just uncomment one (and
only one) of the \verb|\mtcaddchapter| commands after
\verb|\chapter*| in the source code of \texttt{mtc-add.tex}.
For each case, look at the \index{Table of Contents}Table of Contents
and at this chapter.
\index{a}\index{added}\index{after}\index{also}\index{and}%
\index{at}\index{can}\index{case}\index{chapter}\index{code}%
\index{command}\index{commands}\index{Contents}\index{each}%
\index{entries}\index{For}\index{here}\index{I}\index{in}%
\index{index}\index{is}\index{Just}\index{just}\index{look}%
\index{lot}\index{of}\index{on}\index{one}\index{only}%
\index{source}\index{starred}\index{Table}\index{test}%
\index{the}\index{This}\index{this}\index{to}\index{uncomment}%
\index{variations}\index{you}%
I also added a lot of index entries, just to test.

\chapter{Third chapter}
\index{chapter!normal}
\minitoc
\mtcskip
\minilof
\mtcskip
```

```
\minilot
\section{Third section}

\begin{figure}
\caption{Third figure}
\end{figure}
\begin{table}
\caption{Third table}
\end{table}

\section{Fourth section}

\begin{figure}
\caption{Fourth figure}
\end{figure}
\begin{table}
\caption{Fourth table}
\end{table}
```

As we want to add an entry for the bibliography in the table of contents, and we use the tocbibind package for that, we must add a correction with `\adjustmtc`:

```
\nocite*
\def\noopsort#1{\relax}
\bibliographystyle{plain}
\bibliography{mtc-add}
\adjustmtc
```

As we want to add an entry for the index in the table of contents, and we use the tocbibind package for that, we must add a correction; two solutions are available: use `\mtcfixindex` or the other given three lines:

```
\printindex
%%\mtcfixindex % use this OR the 3 following lines
\addcontentsline{lof}{xchapter}{}
\addcontentsline{lot}{xchapter}{}
\mtcaddchapter
%%

\appendix
\chapter{App.~1}
\index{chapter!appendix}
\minitoc
\mtcskip
\minilof
\mtcskip
\minilot
\section{Fifth section}

\begin{figure}
\caption{Fifth figure}
\end{figure}
```

```
\begin{table}
\caption{Fifth table}
\end{table}

\section{Sixth section}

\begin{figure}
\caption{Sixth figure}
\end{figure}
\begin{table}
\caption{Sixth table}
\end{table}
```

The next chapter asks for a minitoc, a minilof and a minilot, but contains no tables; hence the minitoc package will give some warnings.

```
\chapter{App.~2}
\index{chapter!appendix}
%% contains no tables but asks for a minilot! No minilot printed.
\minitoc
\mtcskip
\minilof
\mtcskip
\minilot
\section{Seventh section}
\begin{figure}
\caption{Seventh figure}
\end{figure}
\begin{figure}
\caption{Eighth figure}
\end{figure}

\section{Eighth section}

\begin{figure}
\caption{Ninth figure}
\end{figure}
\begin{figure}
\caption{Eleventh figure}
\end{figure}

\end{document}
⟨/mtc – add⟩
```

And we need also its small bibliographic data base:

- the english documentation of the minitoc package [107]:

```
⟨*mtc – addbib⟩
@MISC{minitoc,
      TITLE="The {\texttt{minitoc}} package",
      AUTHOR="Drucbert, Jean-Pierre F.",
      ADDRESS="{\texttt{drucbert@onera.fr}}",
```

```
505          NOTE="{\url{CTAN:macros/latex/contrib/minitoc/minitoc.pdf}}",
506          MONTH=jan,
507          YEAR=2007}
508
```

- the french documentation of the minitoc package [106]:

```
509     @MISC{fminitoc,
510          TITLE="Le paquetage {\texttt{minitoc}}",
511          AUTHOR="Drucbert, Jean-Pierre F.",
512          ADDRESS="{\texttt{drucbert@onera.fr}}",
513          NOTE="{\url{CTAN:macros/latex/contrib/minitoc/fminitoc.pdf}}",
514          MONTH=jan,
515          YEAR=2007}
516
```

- the documentation of the shorttoc package [105]:

```
517     @MISC{shorttoc,
518          TITLE="The {\texttt{shorttoc}} package",
519          AUTHOR="Drucbert, Jean-Pierre F.",
520          ADDRESS="{\texttt{drucbert@onera.fr}}",
521          NOTE="{\url{CTAN:macros/latex/contrib/shorttoc/shorttoc.dtx}}",
522          MONTH=aug,
523          YEAR=2002}
524
```

- a novel [15] from which a short citation is taken:

```
525     @BOOK{dark,
526          TITLE="{Against a Dark Background}",
527          AUTHOR="Banks, Iain Menzies",
528          PUBLISHER="Bantam Books",
529          ISBN="0553292240 (pb)",
530          YEAR=1993}
531     ⟨/mtc – addbib⟩
```

4.4 The mtc-ads.tex document file

This document uses the article class and shows some problems for adding special entries in the table of contents and some problems with floating objects. We need to use the tocbibind package [253] for the first ones and the minitoc insection package option to avoid the drift if floating objects outside of their section. That gives the following document preamble:

```
532 ⟨*mtc – ads⟩
533 \documentclass[oneside,12pt,a4paper]{article}
534 \ProvidesFile{mtc-ads.tex}%
535   [2007/01/04]
536 \usepackage{url,tocbibind,makeidx}
537 \makeatletter
538 \newif\ifscan@allowed
539 \scan@allowedtrue
```

```
\makeatother
\def\dotfil{\leaders\hbox to.6em{\hss .\hss}\hfil}%
\def\pfill{\unskip~\dotfill\penalty500\strut\nobreak
           \dotfil~\ignorespaces}%
\usepackage[tight,hints,insection]{minitoc}
%%\usepackage{mtcoff}
\makeindex
```

The preparation commands:

```
\begin{document}
\doparttoc \dopartlof \dopartlot
\dosecttoc \dosectlof \dosectlot
```

The commands to prepare the table of contents, the list of figures and the list of tables. As we use the tocbibind package, we must add some `\mtcaddsection` commands:

```
\setcounter{tocdepth}{6}
\setcounter{parttocdepth}{6}
\setcounter{secttocdepth}{6}
\tableofcontents
\mtcaddsection
\listoffigures
\mtcaddsection
\listoftables
\mtcaddsection
```

The body of the document: a part with its part-level mini-tables, some sections with their section-level mini-tables. The document has an index and contains figures and tables.

```
\part{Part~1}
\parttoc \mtcskip \partlof \mtcskip \partlot

\section{First section}
\index{section!normal}
\secttoc \mtcskip \sectlof \mtcskip \sectlot
\subsection{First subsection}

\begin{figure}[tp]
\caption{First figure}
\end{figure}
\begin{table}[tp]
\caption{First table}
\end{table}

\subsection{Second subsection}
A small nice citation from~\cite{dark}:\\
\index{small}\index{citation}\index{nice}\index{A}%
\index{a}\index{and}\index{bird}\index{But}%
\index{cannot}\index{claim}\index{great}\index{he}%
\index{I}\index{imagine}\index{it}\index{know}%
\index{land}\index{on}\index{once}\index{that}%
```

```
\index{to}\index{tree}\index{would}\index{yes}%
A bird cannot land once on a great tree and claim to know it.
But I imagine that he would, yes.\\
\hbox{}\hfill Iain M. Banks (1993), \textsl{Against a dark background.}%
\index{Iain}\index{Banks}\index{Against}\index{dark}%
\index{background}
\begin{figure}[tp]
\caption{Second figure}
\end{figure}
\begin{table}
\caption{Second table}
\end{table}
```

Here, we try a starred section, with its entry in the table of contents. You can try several solutions (good or bad).

```
\section*{Second section, starred}
\index{section!starred}
%% UNCOMMENT ONE AND ONLY ONE OF THE 4 FOLLOWING LINES
\mtcaddsection[Second section, starred] % OK
%%\mtcaddsection[]     % BAD
%%\mtcaddsection[~]    % produces a (strange) correct result.
%%\mtcaddsection       % BAD
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\index{tests}

This is a starred section; you can test here variations on
the \verb|\mtcaddsection| command. Just uncomment one (and
only one) of the \verb|\mtcaddsection| commands after
\verb|\section*| in the source code of \texttt{mtc-add.tex}.
For each case, look at the \index{Table of Contents}Table of Contents
and at this section.
\index{a}\index{added}\index{after}\index{also}\index{and}%
\index{at}\index{can}\index{case}\index{section}%
\index{code}\index{command}\index{commands}%
\index{Contents}\index{each}\index{entries}\index{For}%
\index{here}\index{I}\index{in}\index{index}\index{is}%
\index{Just}\index{just}\index{look}\index{lot}%
\index{of}\index{on}\index{one}\index{only}\index{source}%
\index{starred}\index{Table}\index{test}\index{the}%
\index{This}\index{this}\index{to}\index{uncomment}%
\index{variations}\index{you}%
I also added a lot of index entries, just to test.

\section{Third section}
\index{section!normal}
\secttoc \mtcskip \sectlof \mtcskip \sectlot
\subsection{Third subsection}

\begin{figure}
\caption{Third figure}
\end{figure}
\begin{table}
```

```
\caption{Third table}
\end{table}

\subsection{Fourth subsection}

\begin{figure}
\caption{Fourth figure}
\end{figure}
\begin{table}
\caption{Fourth table}
\end{table}

\subsubsubsection{Even a sub-sub-section!}
\subsubsubsection{And yet another one}

\part{Part~2}
\parttoc \mtcskip \partlof \mtcskip \partlot

\section{Fourth section}
\index{section!normal}
\secttoc \mtcskip \sectlof \mtcskip \sectlot
\subsection{Fifth subsection}

\begin{figure}[tp]
\caption{Fifth figure}
\end{figure}
\begin{table}[tp]
\caption{Fifth table}
\end{table}

\subsection{Sixth subsection}
A small nice citation from~\cite{dark}:\\
\index{small}\index{citation}\index{nice}\index{A}%
\index{a}\index{and}\index{bird}\index{But}%
\index{cannot}\index{claim}\index{great}\index{he}%
\index{I}\index{imagine}\index{it}\index{know}%
\index{land}\index{on}\index{once}\index{that}%
\index{to}\index{tree}\index{would}\index{yes}%
A bird cannot land once on a great tree and claim to know it.
But I imagine that he would, yes.\\
\hbox{}\hfill
Iain M. Banks (1993), \textsl{Against a dark background.}%
\index{Iain}\index{Banks}\index{Against}%
\index{dark}\index{background}

\begin{figure}[tp]
\caption{Sixth figure}
\end{figure}
\begin{table}
\caption{Sixth table}
\end{table}

\section*{Fifth section, starred}
\index{section!starred}
```

```
%% UNCOMMENT ONE AND ONLY ONE OF THE 4 FOLLOWING LINES
\mtcaddsection[Fifth section, starred] % OK
%%\mtcaddsection[]     % OK
%%\mtcaddsection[~]    % produces a (strange) correct result.
%%\mtcaddsection       % OK
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\index{tests}

This is a starred section; you can test here variations on
the \verb|\mtcaddsection| command. Just uncomment one (and
only one) of the \verb|\mtcaddsection| commands after
\verb|\section*| in the source code of \texttt{mtc-add.tex}.
For each case, look at the \index{Table of Contents}Table of Contents
and at this section.\index{a}%
\index{added}\index{after}\index{also}\index{and}%
\index{at}\index{can}\index{case}\index{section}%
\index{code}\index{command}\index{commands}\index{Contents}%
\index{each}\index{entries}\index{For}\index{here}%
\index{I}\index{in}\index{index}\index{is}%
\index{Just}\index{just}\index{look}\index{lot}%
\index{of}\index{on}\index{one}\index{only}%
\index{source}\index{starred}\index{Table}\index{test}%
\index{the}\index{This}\index{this}\index{to}%
\index{uncomment}\index{variations}\index{you}%
I also added a lot of index entries, just to test.

\section{Sixth section}
\index{section!normal}
\secttoc \mtcskip \sectlof \mtcskip \sectlot
\subsection{Seventh subsection}

\begin{figure}
\caption{Seventh figure}
\end{figure}
\begin{table}
\caption{Seventh table}
\end{table}

\subsection{Eighth subsection}

\begin{figure}
\caption{Eighth figure}
\end{figure}
\begin{table}
\caption{Eighth table}
\end{table}
```

The bibliography: as we want an entry for it in the table of contents, we use the tocbibind package [253] and a correction with `\adjuststc`:

```
\nocite*
\def\noopsort#1{\relax}
\bibliographystyle{plain}
```

```
\bibliography{mtc-add}
\adjuststc
```

The index: as we want an entry for it in the table of contents, we use the tocbibind package [253] and a correction with `\mtcfixindex`:

```
\printindex
\mtcfixindex % use this OR the 2 following lines
%%\addcontentsline{lof}{xsect}{}
%%\addcontentsline{lot}{xsect}{}
%%\mtcaddsection

\appendix
\section{App.~1}
\index{section!appendix}
\secttoc \mtcskip \sectlof \mtcskip \sectlot
\subsection{Ninth subsection}

\begin{figure}
\caption{Ninth figure}
\end{figure}
\begin{table}
\caption{Ninth table}
\end{table}

\subsection{Tenth subsection}

\begin{figure}
\caption{Tenth figure}
\end{figure}
\begin{table}
\caption{Tenth table}
\end{table}

\section{App.~2}
\index{section!appendix}
%% contains no tables but asks for a sectlot! No sectlot printed.
\secttoc \mtcskip \sectlof \mtcskip \sectlot
\subsection{Eleventh subsection}

\begin{figure}
\caption{Eleventh figure}
\end{figure}
\begin{figure}
\caption{Twelfth figure}
\end{figure}

\subsection{Twelfth subsection}

\begin{figure}
\caption{Thirdteenth figure}
\end{figure}
\begin{figure}
```

```
\caption{Fourteenth figure}
\end{figure}

\end{document}
⟨/mtc – ads⟩
```

4.5 The mtc-amm.tex document file

This example shows the use of the appendices environment in a memoir class document when the minitoc package is loaded. First, the preamble:

```
⟨*mtc – amm⟩
\documentclass[oneside]{memoir}
\ProvidesFile{mtc-amm.tex}%
  [2007/01/04]
\usepackage{lipsum} % filling text
\usepackage%
  [tight]{minitoc}
\begin{document}
\dominitoc
\tableofcontents
\adjustmtc
\chapter{First chapter}
\minitoc
\lipsum[1]
\section{First section}
\lipsum[2]
```

The appendices are set in an appendices environment; we can add an entry in the TOC with \addappheadtotoc (a command from the memoir class):

```
\begin{appendices}
\addappheadtotoc
\chapter{Afterthought}
\minitoc
\lipsum[3]
Afterthought appendix
\section{Further remarks}
\lipsum[4]
\end{appendices}
\chapter{Conclusion}
\minitoc
\section{Bye}
\lipsum[5]
\end{document}
⟨/mtc – amm⟩
```

4.6 The mtc-art.tex document file

This is a basic document using the minitoc package. It contains sections but no chapters, so it must use an article-like document class. You should work on a *copy* of this file and can alter its preamble and its contents to make experiments with parameters. A typical preamble follows:

```
⟨*mtc – art⟩
%% mtc-art.tex
%% This file contains a set of tests for the minitoc.sty version #51
%% package. You can alter most of parameters to test.
%% article (\section must be defined)
\documentclass[12pt,a4paper]{article}
\ProvidesFile{mtc-art.tex}%
  [2007/01/04]
\usepackage{lipsum} % provides filling text
```

```
%%\usepackage{hyperref}  % If used, load it BEFORE minitoc
\usepackage[tight,insection]{minitoc}
\setcounter{secnumdepth}{5}   % depth of numbering of sectionning commands
\setcounter{tocdepth}{3}      % depth of table of contents
\setlength{\stcindent}{24pt}  % indentation of secttocs, default
%%                            % font for secttocs, default
\renewcommand{\stcfont}{\small\rmfamily\upshape\mdseries}%
%%                            % font for secttocs, subsections
%%\renewcommand{\stcSSfont}{\small\sf}%
%% you can make experiments with \stcSSSfont, \stcPfont and \stcSPfont
%% but it is ``fontomania''...
\raggedbottom                   % or \flushbottom, at your choice
```

If you want to use sections numbered in each part (the section number restarts to 1 at the beginning of each part), uncomment the 3 lines of code below. This demonstrates that the numbering of the secttoc files is independent of the numbering of the sections (it is absolute).

```
%%% TEST: uncomment the next line to test
%%% resetting section number in each part
%%\makeatletter \@addtoreset{section}{part} \makeatother
%%% END TEST
```

We begin the body of the document. You can still alter some parameters (presence or absence of rules and page numbers in the mini-tables):

```
\begin{document}
```

The preparation commands, with their optional argument if necessary:

```
\dosecttoc
\dosectlof[c]                   % center titles of the sectlofs
\dosectlot
\doparttoc                      % test of parttoc/partlof stuff
\dopartlof                      % added in version #15
\dopartlot                      % added in version #15
```

It is necessary to create the contents files; use the “fake” version to not print.

```
\faketableofcontents            % or \tableofcontents
\fakelistoffigures              % to check compatibility
\fakelistoftables               % to check compatibility
```

There is the text of the document, with its sectionning commands; we define a part, with a parttoc, a partlof (with the title on the right) and a parttoc:

```
\part{First Part}
\parttoc
\partlof[r]
\partlot
```

A section, in two columns mode, with a secttoc (title on the right), and a sectlof; this section contains subsections to make a non-empty secttoc but no figures (to detect an empty sectlof).

```
\twocolumn\sloppy                % the secttoc in twocolumn layout is ugly,
                                 % but works. Ideas to make it better?
\section{AAAAA}                  % a section with a lot of sections
\secttoc[r]                      % secttoc title on the right
\mtcskip \sectlof %ADDED
\lipsum[1]
\subsection{S1}
\lipsum[2]
\subsection{S2}
\lipsum[3]
\subsection{S3}
\lipsum[4]
\subsection*{S4}
%%\addcontentsline{toc}{starsubsection}{*S4*}
\lipsum[5]
\subsection{S5}
\lipsum[6]
\subsection{S6}
\lipsum[7]
\subsection{S7}
\lipsum[8]
\subsection{S8}
\lipsum[9]
\subsection{S9}
\lipsum[10]
\subsection{S10}
\lipsum[11]
\subsection{S11}
\lipsum[12]
\subsection{S12}
\lipsum[13]
\subsection{S13}
\lipsum[14]
\subsection{S14}
\lipsum[15]
\subsection{S15}
\lipsum[16]
\subsection{S16}
\lipsum[17]
\subsection{S17}
\lipsum[18]
\subsection{S18}
\lipsum[19]
\subsection{S19}
\lipsum[20]
\subsection{S20}
\lipsum[21]
\subsection{S21}
\lipsum[22]
\subsection{S22}
\lipsum[23]
```

```
\subsection{S23}
\lipsum[24]
\subsection{S24}
\lipsum[25]
\subsection{S25}
\lipsum[26]
\subsection{S26}
\lipsum[27]
\subsection{S27}
\lipsum[28]
\subsection{S28}
\lipsum[29]
\subsection{S29}
\lipsum[30]
\subsection{S30}
\lipsum[31]
```

We return to the one column mode. Then a section with a secttoc and a sectlof (there are subsections and figures). The `insection` package option should ensure that floating objects (like figures) do not drift outside their section.

```
\onecolumn\fussy         % back to one column
\section{BBBBB}
\secttoc
\mtcskip                 % put some skip here
\sectlof                 % a sectlof
\lipsum[32]
\subsection{T1}
\lipsum[33]
\begin{figure}[t]        % tests compatibility with floating bodies
\setlength{\unitlength}{1mm}
\begin{picture}(100,50)
\end{picture}
\caption{F1}             % (I have not tested tables, but it is similar)
\end{figure}
\FloatBarrier
\subsubsection[tt1]{TT1} % tests optional arg. of a sectionning command
\lipsum[34]
\paragraph{TTT1}
\lipsum[35]
\subparagraph{TTTT1}
\lipsum[36]
\begin{figure}[t]
\setlength{\unitlength}{1mm}
\begin{picture}(100,50)
\end{picture}
\caption[f2]{F2}         % tests optional arg. of a caption
\end{figure}
\FloatBarrier
\subsection{T2}
\lipsum[37]
\section*{CCCCC}         % tests a pseudo-section. should have no secttoc
%%\addstarredsection{CCCCC}
```

```
\mtcaddsection[CCCCC]
\secttoc \mtcskip \sectlof %ADDED
\lipsum[38]
\subsection{U1}
\lipsum[39]
\subsubsection{UU1}
\lipsum[40]
\paragraph{UUU1}
\lipsum[41]
\subparagraph{UUUU1}
\lipsum[42]
\subsection{U2}
\lipsum[43]
\part{Second Part}
\parttoc
\partlof[c]
\partlot
%%                      % the following section should have no secttoc,
\section{DDDDD}         % but if you uncomment \secttoc,
%%\secttoc
\mtcskip \sectlof %ADDED
%                       % the secttoc appears
\lipsum[44]
\subsection{V1}
\lipsum[45]
\subsubsection{VV1}
\lipsum[46]
\paragraph{VVV1}
\lipsum[47]
\subparagraph{VVVV1}
\lipsum[48]
\begin{figure}[t]       % tests compatibility with floating bodies
\setlength{\unitlength}{1mm}
\begin{picture}(100,50)
\end{picture}
\caption{F3}            % (I have not tested tables, but it is similar)
\end{figure}
\FloatBarrier
\lipsum[49]
\subsection{V2}
\lipsum[50]
```

We change the depth of the secttocs, inside a local group (a pair of braces):

```
\section{EEEEE}                % this section should have a secttoc
{%                             % left brace, see below
\setcounter{secttocdepth}{3}   % depth of sect table of contents;
                               % try with different values.
\secttoc
\mtcskip \sectlof %ADDED
}                              % right brace
%% this pair of braces is used to keep local the change
%% on secttocdepth.
```

```
\lipsum[51]
\subsection{W1}                 % with the given depth
\lipsum[52]
\subsubsection{WW1}
\lipsum[53]
\paragraph{WWW1}
\lipsum[54]
\begin{figure}[t]            % tests compatibility with floating bodies
\setlength{\unitlength}{1mm}
\begin{picture}(100,50)
\end{picture}
\caption{F4}                 % (I have not tested tables, but it is similar)
\end{figure}
\FloatBarrier
bla bla bla bla bla bla bla bla bla bla bla
\subparagraph{WWWW1}
\lipsum[55]
\subsection{W2}
\lipsum[56]
\chapter*{}
\part{Appendices}
\parttoc
\mtcskip
\partlof
\mtcskip
\partlot
\FloatBarrier
\appendix
\section{Comments}
\lipsum[57]
\secttoc
\mtcskip \sectlof %ADDED
\subsection{C1}
\lipsum[58]
\subsection{C2}
\lipsum[59]
\subsection{C3}
\lipsum[60]
\begin{figure}[hb]        % tests compatibility with floating bodies
\setlength{\unitlength}{1mm}
\begin{picture}(100,50)
\end{picture}
\caption{F5}              % (I have not tested tables, but it is similar)
\end{figure}
\FloatBarrier
\subsection{C4}
\lipsum[61]
\FloatBarrier
\section{Evolution}
\secttoc
\sectlof % empty
\sectlot % empty
\lipsum[62]
\subsection{D1}
```

```
\lipsum[63]
\subsection{D2}
\lipsum[64]
\subsection{D3}
\lipsum[65]
\subsection{D4}
\lipsum[66]
\end{document}
⟨/mtc − art⟩
```

4.7 The `mtc-bk.tex` document file

This is a basic document using the minitoc package. It contains chapters, so it must use a book-like or report-like document class. You should work on a *copy* of this file and can alter its preamble and its contents to make experiments with parameters. A typical preamble follows:

```
⟨∗mtc − bk⟩
%%%%%%%%%%%%%%%%%%%%% A example file (differs from previous versions)
%% mtc-bk.tex
%% This file contains a set of tests for the minitoc.sty version #51
%% package file. You can alter most of parameters to test.
%% book/report (\chapter must be defined).
%% You can use a copy of this file to play with minitoc
%% commands and parameters.
\documentclass[12pt,a4paper]{report} % the report class uses less pages
%%\documentclass[12pt,a4paper]{book}
\ProvidesFile{mtc-bk.tex}%
  [2007/01/04]
\usepackage{lipsum} % provides filling text
%%\usepackage{hyperref} % if used, load it BEFORE minitoc
%%\usepackage{mtcoff}
\usepackage[tight]{minitoc} % tight option make shorter mini-tables
\setcounter{secnumdepth}{5} % depth of numbering of sectionning commands
\setcounter{tocdepth}{3}              % depth of table of contents
\setlength{\mtcindent}{24pt}          % indentation of minitocs, default
\renewcommand{\mtcfont}{\small\rm}     % font for minitocs, default
\renewcommand{\mtcSfont}{\small\bf}    % font for minitocs, sections, default
%%\renewcommand{\mtcSSfont}{\small\sf} % font for minitocs, subsections
%% you can make experiments with \mtcSSSfont, \mtcPfont and \mtcSPfont
%% but it is ``fontomania''...
\raggedbottom                  % or \flushbottom, at your choice
```

If you want to use chapters numbered in each part (the chapter number restarts to 1 at the beginning of each part), uncomment the 3 lines of code below. This demonstrates that the numbering of the minitoc files is independent of the numbering of the chapters (it is absolute).

```
%%% TEST: uncomment the 3 next lines to test
%%% resetting chapter number in each part
```

```
%%\makeatletter
%%\@addtoreset{chapter}{part}
%%\makeatother
%%% END TEST
```

We begin the body of the document. You can still alter some parameters (presence or absence of rules and page numbers in the mini-tables):

```
\begin{document}
\mtcpagenumbers
\noptcrule
%% \nomtcrule                    % suppresses minitoc rules
%% \nomtcpagenumbers             % suppresses minitoc page numbers
%% \nomlfpagenumbers             % ---------- minilof ---- -------
%% \nomltpagenumbers             % ---------- minilot ---- -------
```

The preparation commands, with their optional argument if necessary:

```
\dominitoc
\dominilof[c]                   % centers title of minilof's
\dominilot
\doparttoc                      % test of parttoc/partlof stuff
\dopartlof                      % added in version #15
\dopartlot                      % added in version #15
```

It is necessary to create the contents files; use the “fake” version to not print.

```
\tableofcontents            % or \faketableofcontents
\listoffigures              % or \fakelistoffigures
\fakelistoftables           % or \listoftables
```

Uncomment the following line if the first chapter must be numbered “0”:

```
%%\addtocounter{chapter}{-1} % to begin with Chapter 0
```

There is the text of the document, with its sectionning commands:

```
\part{First Part}
\parttoc
\partlof[r]
\partlot[r]
```

A chapter, in two column mode, with a minitoc (title on the right):

```
\twocolumn\sloppy               % the minitoc in twocolumn layout is ugly,
\chapter{AAAAA}                 % a chapter with a lot of sections
\minitoc[r]                     % minitoc title on the right
\lipsum[1]
\section{S1}
```

```
\lipsum[2]
\section{S2}
\lipsum[3]
\section{S3}
\lipsum[4]
```

A starred section; we want an entry in the TOC, so we add it the normal way:

```
\section*{S4}
\addcontentsline{toc}{section}{\protect\numberline{}{S4}}
\lipsum[5]
\section{S5}
\lipsum[6]
\section{S6}
\lipsum[6]
\section{S7}
\lipsum[7]
\section{S8}
\lipsum[9]
\section{S9}
\lipsum[10]
\section{S10}
\lipsum[11]
\section{S11}
\lipsum[12]
\section{S12}
\lipsum[13]
\section{S13}
\lipsum[14]
\section{S14}
\lipsum[15]
\section{S15}
\lipsum[16]
\section{S16}
\lipsum[17]
\section{S17}
\lipsum[18]
\section{S18}
\lipsum[19]
\section{S19}
\lipsum[20]
\section{S20}
\lipsum[21]
\section{S21}
\lipsum[22]
\section{S22}
\lipsum[23]
\section{S23}
\lipsum[24]
\section{S24}
\lipsum[25]
\section{S25}
\lipsum[26]
```

```
\section{S26}
\lipsum[27]
\section{S27}
\lipsum[28]
\section{S28}
\lipsum[29]
\section{S29}
\lipsum[30]
\section{S30}
\lipsum[31]
\subsection{SS1}
\lipsum[32]
\section{S31}
\lipsum[33]
```

We return to one column mode. A new chapter, with a minitoc, a minilof and a minilot:

```
\onecolumn\fussy        % back to one column
\chapter{BBBBB}
\minitoc
\mtcskip                % put some skip here
\minilof                % a minilof
\mtcskip                % put some skip here
\minilot                % a minilot
\lipsum[34]
\section{T1}
\lipsum[35]
\begin{figure}[t]       % tests compatibility with floating bodies
\setlength{\unitlength}{1mm}
\begin{picture}(100,50)
\end{picture}
\caption{F1}            % (tables are similar)
\end{figure}
\begin{table}[b]        % tests compatibility with floating bodies
\setlength{\unitlength}{1mm}
\begin{picture}(100,50)
\end{picture}
\caption{T1}            % (tables are similar)
\end{table}
\clearpage
\subsection[tt1]{TT1}   % tests optional arg. of a sectionning command
\lipsum[36]
\subsubsection{TTT1}
\lipsum[37]
\paragraph{TTTT1}
\lipsum[38]
\begin{figure}
\setlength{\unitlength}{1mm}
\begin{picture}(100,50)
\end{picture}
\caption[f2]{F2}        % tests optional arg. of a caption
\end{figure}
\section{T2}
```

```
\lipsum[39]
```

A starred chapter with an entry added in the TOC; all subordinate (lower) sectionning commands must also be starred.

```
\chapter*{CCCCC}        % tests a pseudo-chapter; could have a minitoc.
\addstarredchapter{CCCCC}
\lipsum[40]
\section*{U1}
\addcontentsline{toc}{section}{U1}
\lipsum[41]
\subsection*{UU1}
\addcontentsline{toc}{subsection}{UU1}
\lipsum[42]
\subsubsection*{UUU1}
\addcontentsline{toc}{subsubsection}{UUU1}
\lipsum[43]
\paragraph*{UUUU1}
\addcontentsline{toc}{paragraph}{UUUU1}
\lipsum[44]
\section*{U2}
\addcontentsline{toc}{section}{U2}
\lipsum[45]
\part{Second Part}
\parttoc
\partlof[c]
```

This chapter has no minitoc, but if you uncomment `\minitoc`, the minitoc will appear.

```
%%                      % the following chapter should have no minitoc,
\chapter{DDDDD}         % but if you uncomment \minitoc,
%%\minitoc               % the minitoc appears
\lipsum[46]
\section{V1}
\lipsum[47]
\subsection{VV1}
\lipsum[48]
\subsubsection{VVV1}
\lipsum[49]
\paragraph{VVVV1}
\lipsum[50]
\begin{figure}[t]       % tests compatibility with floating bodies
\setlength{\unitlength}{1mm}
\begin{picture}(100,50)
\end{picture}
\caption{F3}            % (I have not tested tables, but it is similar)
\end{figure}
\lipsum[51]
\section{V2}
\lipsum[52]
```

We change the depth of the minitocs, inside a local group (a pair of braces):

```
\chapter{EEEEE}                 % this chapter should have a minitoc
{%                              % left brace, see below
\setcounter{minitocdepth}{3}    % depth of mini table of contents;
                                % try with different values.
\minitoc
}                               % right brace
%% this pair of braces is used to keep local the change
%% on minitocdepth.
\lipsum[53]
\section{W1}                    % with the given depth
\lipsum[54]
\subsection{WW1}
\lipsum[55]
\subsubsection{WWW1}
\lipsum[56]
\begin{figure}[t]        % tests compatibility with floating bodies
\setlength{\unitlength}{1mm}
\begin{picture}(100,50)
\end{picture}
\caption{F4}             % (I have not tested tables here, but it is similar)
\end{figure}
\lipsum[57]
\paragraph{WWWW1}
\lipsum[58]
\subparagraph{WWWWW1}
\lipsum[59]
\section{W2}
\lipsum[60]
```

Here, we encounter a classical problem: to make a local table of contents for a set of appendices, while hiding these entries in the main table of contents. First, we create a part, with its parttoc:

```
\appendix
\part{Appendices}
\parttoc
```

Then, we begin a `mtchideinmaintoc` environment, with the hiding depth as optional argument:

```
\begin{mtchideinmaintoc}[-1]
\chapter{Comments}
\minitoc
\section{C1}
\lipsum[61]
\section{C2}
\lipsum[62]
\section{C3}
\lipsum[63]
\begin{figure}[t]        % tests compatibility with floating bodies
```

```
\setlength{\unitlength}{1mm}
\begin{picture}(100,50)
\end{picture}
\caption{F5}            % (I have not tested tables, but it is similar)
\end{figure}
\section{C4}
\chapter{Evolution}
\minitoc
\minilof %Empty => invisible
\minilot %Empty => invisible
\section{D1}
\lipsum[64]
\section{D2}
\lipsum[65]
\section{D3}
\lipsum[66]
\section{D4}
```

We terminate the part by adding a marker in the TOC file, then we must close this `mtchideinmaintoc` environment:

```
%% this line closes the omitted part
\addtocontents{toc}{\protect\partbegin}
%% this line restore the depth in the main TOC
\end{mtchideinmaintoc}
\lipsum[67]
\end{document}
⟨/mtc − bk⟩
```

4.8 The `mtc-bo.tex` document file

This document shows the use of the minitoc package in a document using a two column layout for some portions and the tocloft package [250]. The aim is to begin a chapter with a special head and a preliminary block containing a minitoc and some indications, on two columns. The preamble loads some useful packages for the french language, the geometry package [237], which defines the global page layout, the multicol package [186], the color package [91], because we want a colored background for the minitoc, the tocloft package [250], to change some parameters of the minitoc, and, at least, the minitoc package.

```
⟨∗mtc − bo⟩
\documentclass[10pt]{book}
\ProvidesFile{mtc-bo.tex}%
  [2007/01/04]
\usepackage[paperwidth=8.5in,paperheight=11in,%
      lmargin=1.25in,rmargin=1.25in,tmargin=1in,bmargin=1in]{geometry}
\usepackage[francais]{babel}
\usepackage{franc,frnew}
\usepackage[T1]{fontenc}
\usepackage[isolatin]{inputenc}
```

```
\usepackage{mypatches}
\usepackage{multicol}
\usepackage{color}
```

We use the tocloft package and its commands to set the indentations in the TOC and the minitoc:

```
\usepackage{tocloft}
\setlength{\cftsecindent}{0cm}
\setlength{\cftsecnumwidth}{15 pt}
\setlength{\cftsubsecindent}{\cftsecindent}
\addtolength{\cftsubsecindent}{\cftsecnumwidth}
\setlength{\cftsubsecnumwidth}{20 pt}
\setlength{\cftsubsubsecindent}{\cftsubsecindent}
\addtolength{\cftsubsubsecindent}{\cftsubsecnumwidth}
```

Note that if we want to suppress the page numbers in the minitoc, we must use the commands from tocloft:

```
\cftpagenumbersoff{sec}
\cftpagenumbersoff{subsec}
```

We load the minitoc package and change the indentation, suppress the rules and change the minitoc title. The hyperref package [215] can also be loaded (*after* minitoc).

```
\usepackage[francais,tight]{minitoc}
\usepackage{hyperref}
\setlength{\mtcindent}{0pt}
\nomtcrule        % pas de filets en haut et en bas de la mini-tdm
\nomtcpagenumbers % pas de numéro de pages
%                   (non fonctionnel avec tocloft)
\renewcommand{\mtctitle}{Contenu de la rencontre}
```

This code redefines the format of the chapter head:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  MACRO POUR AVOIR LE MOT RENCONTRE AU LIEU DE CHAPITRE
%  Sans saut de ligne
%  (modification du code qui se trouve dans la FAQ)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\makeatletter
\def\@makechapterhead#1{%
  \vspace*{10\p@}%
  {\parindent \z@ \raggedleft \normalfont
   \interlinepenalty\@M
   \ifnum \c@secnumdepth >\m@ne
     \Huge\bfseries\sffamily Rencontre \thechapter\\% \quad
   \fi
   \Huge\bfseries\sffamily #1\par\nobreak
   \vskip 10\p@
}}
```

```
\def\@makeschapterhead#1{%
  \vspace*{10\p@}%
  {\parindent \z@ \raggedright \normalfont
   \interlinepenalty\@M
   \Huge \sffamily #1\par\nobreak
   \vskip 10\p@
}} \makeatother
```

We define an environment (pageUn) for the block placed at the beginning of a chapter. This block contains a minitoc, then a sequence of informations given by the 6 parameters of the environment. The block uses a multicols environment to typeset on two columns. Some decorations are added: rules, colored background for the minitoc.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%: ENVIRONNEMENT POUR LA PAGE 1 DES RENCONTRES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% param1: date de la rencontre
% param2: nombre de périodes
% param3: liste des documents distribués
% param4: messages
% param5: lecture
% param6: exercices
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\newenvironment{pageUn}[6]{%
\parindent = 0.0in
\rule{\linewidth}{1pt}
\begin{multicols}{2}
        {\large \bfseries Math. pour médecine nucléaire\\
        \textit{(#2)}}
        \vfill\columnbreak
        \raggedleft\bfseries Automne 2003\\
        #1
\end{multicols}
\vspace{-18pt}
\rule{\linewidth}{1pt}

\setlength{\columnseprule}{.3pt}
\setlength{\columnsep}{1cm}
\begin{multicols}{2}%

%: TABLE DES MATIÈRES (col. gauche)
\colorbox[cmyk]{.1,0,0,0}{%
        \parbox{\linewidth}{%
        \setcounter{minitocdepth}{3}%
        \minitoc%
        }}
%\vfill \columnbreak ~ \vfill

\mtcskip

%: DOCUMENTS DISTRIBUÉS (début col. droite)

{\large \bfseries Documents distribués}
```

```
\begin{itemize} \renewcommand{\labelitemi}{$\star$} #3 \end{itemize}

%: MESSAGES AUX ÉTUDIANTS

\vspace{12pt}{\large \bfseries Messages}
\begin{itemize} \renewcommand{\labelitemi}{$\star$}#4\end{itemize}

%: LECTURE
\vspace{12pt}{\large \bfseries Lecture}\vspace{-6pt} \par#5 \par

% EXERCICES
\vspace{12pt}{\large \bfseries Exercices}\vspace{-6pt}\par#6\par

%\newpage
\end{multicols}%
}
%{\newpage}
{\hrule}

%\renewcommand{\baselinestretch}{1.2} %interligne

%\pagestyle{empty}  %pas de # de page
%\parindent = 0.0in
\parskip = 0.1in

%%%%%%%%%%%%%%%%%%%%%%%%
%: REDÉFINIR LES SECTIONS
%%%%%%%%%%%%%%%%%%%%%%%%
\renewcommand{\thesection}{\Alph{section}}
%\renewcommand{\thesection}{\thechapter-\Alph{section}}

```

The body of the document:

```
%%%%%%%%%%%%%%%%%%%%%%%%
%: BEGIN
%%%%%%%%%%%%%%%%%%%%%%%%
\begin{document}
\dominitoc
\faketableofcontents
\chapter{Première étape (début)}
%\minitoc
\begin{pageUn}
        {%1     Date du cours
                lundi 25 août
                }
        {%2     Durée du cours
                2 périodes
                }
        {%3     Liste des documents à distribuer
                \item Plan de cours
                \item Fiche d'identification
                \item Grille horaire
```

```
                \item Feuilles d'exercices supplémentaires
                }
        {%4     Ne pas oublier
                \item Acheter le livre de référence
                \item Apporter une disquette
                }
        {%5     Lecture
                Lire les pages ppp à ppp et ppp à ppp
                }
        {%6     Exercice
                Faire les exercices nnn de la page ppp
                }
\end{pageUn}

%%%%%%%%%%%%%%%%%%%%%%%%
\section{Titre de la section}
%%%%%%%%%%%%%%%%%%%%%%%%

Texte dans la section

%%
\subsection{Une sous-section}
%%
Bla bla bla

%%
\subsection{Une autre sous-section}
%%
Bla bla bla


%%%%%%%%%%%%%%%%%%%%%%%%
\section{Titre d'une autre section}
%%%%%%%%%%%%%%%%%%%%%%%%

Texte dans la section

%%
\subsection{Une sous-section}
%%
Bla bla bla

%%
\subsection{Une autre sous-section}
%%
Bla bla bla


%%%%%%%%%%%%%%%%%%%%%%%%
\section{Encore une autre section}
%%%%%%%%%%%%%%%%%%%%%%%%

Texte dans la section

```

```
\end{document}
⟨/mtc – bo⟩
```

4.9 The mtc-ch0.tex document file

This document shows the use of the minitoc package in a document using a starred first chapter, inducing the "Chapter Zero" problem.

```
⟨*mtc – ch0⟩
\documentclass[12pt,a4paper]{report}
\ProvidesFile{mtc-ch0.tex}%
  [2007/01/04]
\usepackage[tight,english]{minitoc}
\begin{document}
\dominitoc
\tableofcontents
```

The first chapter is starred, but contains real numbered sections. We add an entry in the TOC for this chapter and see that its sections are using "0" as chapter number:

```
\chapter*{Chapter One (starred)}
\mtcaddchapter[Fake chapter one]
\minitoc
\section{Chap 1, section 1}
That's right, folks -- we're close to the release of Firefox and
Thunderbird~1.0 and, just like our last 1.0~release, we want to organize
worldwide parties to celebrate.

Thanks to Dominik 'Aeneas' Schnitzer, we have an all-new and improved
Mozilla Party Webtool~2.0. You can create your own party, or sign up for
one already in progress -- and, in an improvement on Webtool~1.0,
organizers can now edit and update party details. The tool allows you to
organize a celebration in any of 243~countries, principalities,
dominions and islands around the world. Never let it be said that we do
things by halves around here.

\subsection{Chap 1, section 1, subsection 1}
That's right, folks -- we're close to the release of Firefox and
Thunderbird~1.0 and, just like our last 1.0~release, we want to organize
worldwide parties to celebrate.

Thanks to Dominik 'Aeneas' Schnitzer, we have an all-new and improved
Mozilla Party Webtool~2.0. You can create your own party, or sign up for
one already in progress -- and, in an improvement on Webtool~1.0,
organizers can now edit and update party details. The tool allows you to
organize a celebration in any of 243~countries, principalities,
dominions and islands around the world. Never let it be said that we do
things by halves around here.
```

The second chapter is normal:

```
\chapter{Chapter Two (numbered one)}
\minitoc
\section{Chapter 2, section 1}
That's right, folks -- we're close to the release of Firefox and
Thunderbird 1.0 and, just like our last 1.0~release, we want to organize
worldwide parties to celebrate.

Thanks to Dominik 'Aeneas' Schnitzer, we have an all-new and improved
Mozilla Party Webtool~2.0. You can create your own party, or sign up for
one already in progress -- and, in an improvement on Webtool~1.0,
organizers can now edit and update party details. The tool allows you to
organize a celebration in any of 243~countries, principalities,
dominions and islands around the world. Never let it be said that we do
things by halves around here.
\end{document}
⟨/mtc – ch0⟩
```

4.10 The `mtc-cri.tex` document file

This document shows the use of the minitoc package in a document with a starred part and starred chapters. Note the use of the adjustment commands. This example is not commented: just follow the insertion of the mini-tables in the `mtc-cri.log` file.

```
⟨*mtc – cri⟩
\documentclass[12pt,a4paper]%
  {report}
\ProvidesFile{mtc-cri.tex}%
  [2007/01/04]
\usepackage[francais]{babel}
\usepackage[T1]{fontenc}
\usepackage[latin1]{inputenc}
\usepackage%
  [french2,tight]%
  {minitoc}

\mtcsetdepth{parttoc}{2}

\begin{document}
\doparttoc
\dominitoc
\tableofcontents
\part*{Présentation générale}
\mtcaddpart[Présentation générale]
\adjustptc[-2]
\parttoc

Texte de la
présentation générale\ldots

\chapter*{Les auteurs}
\mtcaddchapter[Les auteurs]
Présentation des auteurs\ldots

\chapter*{Les lecteurs}
\mtcaddchapter[Les lecteurs]
Présentation des lecteurs\ldots

\part{Première partie}
\parttoc
\chapter*{Introduction}
\mtcaddchapter[Introduction]
\chapter{Premier chapitre}
\minitoc
\section{Première section~A}
\section{Deuxième section~A}
\chapter{Deuxième chapitre}
\minitoc
\section{Première section~B}
\section{Deuxième section~B}

\part{Deuxième partie}
\parttoc
```

```
\chapter{Premier chapitre}
\minitoc
\section{Première section~C}
\section{Deuxième section~C}
\chapter{Deuxième chapitre}
\minitoc
\section{Première section~D}
\section{Deuxième section~D}
\end{document}
⟨/mtc – cri⟩
```

4.11 The `mtc-fo1.tex` document file

This document creates several copies of the same parttoc, but with different fonts (for the chapter level entries); you can compare the results.

```
⟨*mtc – fo1⟩
\documentclass{report}
\ProvidesFile{mtc-fo1.tex}%
  [2007/01/04]
\usepackage{lipsum}
\usepackage{txfonts}
\usepackage[tight]{minitoc}
\begin{document}
\doparttoc
\dominitoc
\tableofcontents
\part{Introduction}
\clearpage
```

A first copy, with default fonts:

```
\parttoc
```

A second copy, roman bold font for chapter entries:

```
\clearpage
\mtcsetfont{parttoc}{chapter}{\normalsize\rmfamily\upshape\bfseries}
\parttoc
```

A third copy, typewriter bold font for chapter entries:

```
\clearpage
\mtcsetfont{parttoc}{chapter}{\normalsize\ttfamily\upshape\bfseries}
\parttoc
```

A fourth copy, not bold typewriter font for chapter entries:

```
\clearpage
\mtcsetfont{parttoc}{*}{\normalsize\ttfamily\upshape\mdseries}
\parttoc
```

```
\chapter{A very short chapter}
\minitoc
\lipsum[1]
\section{First section}
\lipsum[2]
\subsection{Alpha}
\lipsum[3]
\subsection{Beta}
\lipsum[4]
\section{Second section}
\lipsum[5]
\subsection{Gamma}
\lipsum[6]
\subsection{Delta}
\lipsum[7]
\end{document}
⟨/mtc – fo1⟩
```

4.12 The `mtc-fo2.tex` document file

This document creates several copies of the same parttoc, but with different fonts (for the chapter level entries); you can compare the results. As the fonts are not declared the same way, compare the results with those of `mtc-fo1.tex` (see section 4.11 on the preceding page).

```
⟨*mtc – fo2⟩
\documentclass{report}
\ProvidesFile{mtc-f02.tex}%
  [2007/01/04]
\usepackage{lipsum}
\usepackage{txfonts}
\usepackage[tight]{minitoc}
```

We declare the fonts with the old method:

```
   \def\ptcSSfont{\ptcfont}     % (subsections)
   \def\ptcSSSfont{\ptcfont}    % (subsubsections)
   \def\ptcPfont{\ptcfont}      % (paragraphs)
   \def\ptcSPfont{\ptcfont}     % (subparagraphs)
   \def\plffont{\ptcfont}       % (figures)
   \def\plfSfont{\ptcfont}      % (subfigures)
   \def\pltfont{\ptcfont}       % (tables)
   \def\pltSfont{\ptcfont}      % (subtables)
\begin{document}
\doparttoc
\dominitoc
\tableofcontents
\part{Introduction}
\clearpage
```

A first version of the parttoc, with the fonts defined above:

```
\parttoc
\clearpage
```

A second version of the parttoc, with chapter entries in a roman bold font:

```
\mtcsetfont{parttoc}{chapter}{\normalsize\rmfamily\upshape\bfseries}
\parttoc
\clearpage
```

A third version of the parttoc, with chapter entries in a typewriter bold font:

```
\mtcsetfont{parttoc}{chapter}{\normalsize\ttfamily\upshape\bfseries}
\parttoc
\clearpage
```

A fourth version of the parttoc, with chapter entries in a non bold typewriter font:

```
\mtcsetfont{parttoc}{*}{\normalsize\ttfamily\upshape\mdseries}
\parttoc
\chapter{A very short chapter}
\minitoc
\lipsum[1]
\section{First section}
\lipsum[2]
\subsection{Alpha}
\lipsum[3]
\subsection{Beta}
\lipsum[4]
\section{Second section}
\lipsum[5]
\subsection{Gamma}
\lipsum[6]
\subsection{Delta}
\lipsum[7]
\end{document}
```

⟨/mtc – fo2⟩

4.13 The `mtc-hia.tex` document file

This document shows the use of the minitoc package in a document where the entries for some tables must be hidden in the main list of tables. The document uses the article class.

```
⟨*mtc – hia⟩
\documentclass%
  [oneside,a4paper]{article}
\ProvidesFile{mtc-hia.tex}%
  [2007/01/04]
\usepackage{lipsum}
\usepackage%
```

```
  [tight,insection]%
            {minitoc}
\dosectlot
\begin{document}
\listoftables
\section{First section}
\sectlot
\lipsum[1]
\begin{table}[hb]
\caption{My first visible table}
\end{table}
\lipsum[2]
\begin{table}[ht]
\caption{A second visible table}
\end{table}
\lipsum[3]
```

For the *first* hidden table, we add `\mtchideinmainlot` *before* its caption:

```
\begin{table}[hb]
\mtchideinmainlot
\caption{My first hidden table}
\end{table}
\lipsum[4-6]
\begin{table}[ht]
\caption{A second hidden table}
\end{table}
\lipsum[7]
\section{Second section}
\sectlot
\lipsum[8]
```

For the *last* hidden table, we add `\endmtchideinmainlot` *after* its caption:

```
\begin{table}[hb]
\caption{My last hidden table}
\endmtchideinmainlot
\end{table}
\lipsum[9]
\begin{table}[ht]
\caption{A third visible table}
\end{table}
\lipsum[10]
\begin{table}[hb]
\caption{A fourth visible table}
\end{table}
\lipsum[11]
\end{document}
⟨/mtc – hia⟩
```

4.14 The `mtc-hir.tex` document file

This document shows the use of the minitoc package in a document where the entries for some tables must be hidden in the main list of tables. The document uses the report class.

```
⟨*mtc – hir⟩
\documentclass[a4paper]{report}
\ProvidesFile{mtc-hir.tex}%
  [2007/01/04]
\usepackage{lipsum}
\usepackage%
  [tight]{minitoc}
\dominilot
\begin{document}
\listoftables
\chapter{First chapter}
\minilot
\lipsum[1]
\begin{table}[hb]
\caption{My first visible table}
\end{table}
\lipsum[2]
\begin{table}[ht]
\caption{A second visible table}
\end{table}
\lipsum[3]
```

For the *first* hidden table, we add `\mtchideinmainlot` *before* its caption:

```
\begin{table}[hb]
\mtchideinmainlot     % <--
\caption{My first hidden table}
\end{table}
\lipsum[4-6]
\begin{table}[ht]
\caption{A second hidden table}
```

```
\end{table}
\lipsum[7]
\chapter{Second chapter}
\minilot
\lipsum[8]
```

For the *last* hidden table, we add `\endmtchideinmainlot` *after* its caption:

```
\begin{table}[hb]
\caption{My last hidden table}
\endmtchideinmainlot  %  <--
\end{table}
\lipsum[9]
\begin{table}[ht]
\caption{A third visible table}
\end{table}
\lipsum[10]
\begin{table}[hb]
\caption{A fourth visible table}
\end{table}
\lipsum[11]
\end{document}
⟨/mtc – hir⟩
```

4.15 The `mtc-hop.tex` document file

This document shows the use of the minitoc package in a document of class scrbook.

```
⟨*mtc – hop⟩
\documentclass[oneside,12pt]{scrbook}
\ProvidesFile{mtc-hop.tex}%
  [2007/01/04]
\usepackage{lipsum}
\usepackage[hints]{minitoc}
\begin{document}
```

We prepare the minitocs and the minilofs, we print the TOC but not the LOF (while the LOF file is prepared):

```
\dominitoc
\dominilof
\tableofcontents
\fakelistoffigures
```

A starred part with its entry in the TOC:

```
\part*{Part 1: Strategic Marketing}
\mtcaddpart[Part 1: Strategic Marketing]
```

Then two chapters with their minitocs and minitocs:

```
\chapter{Chapter 1}
\minitoc
\minilof
\section{Section one of first chapter}
\lipsum[1]
\begin{figure}
```

```
    \centering
    Test
    \caption{Picture one of first chapter}
\end{figure}

\section{Section two of first chapter}
\lipsum[2]
\begin{figure}
    \centering
    Test
    \caption{Picture two of first chapter}
\end{figure}

\chapter{Chapter 2}
\minitoc \minilof

\section{Section one of second chapter}
\lipsum[3]
\begin{figure}
    \centering
    Test
    \caption{Picture one of second chapter}
\end{figure}
\cleardoublepage
\section{Section two of second chapter}
\lipsum[4]
\begin{figure}
    \centering
    Test
    \caption{Picture two of second chapter}
\end{figure}
\end{document}
⟨/mtc – hop⟩
```

4.16 The `mtc-liv.tex` document file

This document shows the use of the minitoc package in a document of book class, with customized TOC and minitocs.

```
⟨*mtc – liv⟩
\documentclass[10pt,twoside,openright]{book}
\ProvidesFile{mtc-liv.tex}%
  [2007/01/04]
```

First, we want that empty pages be really empty, without page number nor headers, so we redefine `\cleardoublepage`:

```
\makeatletter
\def\ps@chapterverso{\ps@empty}%
```

```
\def\cleardoublepage{\clearpage
 \if@twoside
  \ifodd\c@page\else
   \null\thispagestyle{chapterverso}\newpage
   \if@twocolumn\null\newpage\fi
   \fi
  \fi
 }%
\def\ps@chapterverso{\ps@empty}%
\makeatother
```

We define the encodings, for input and output, because the document is in french and uses accented letters:

```
\usepackage[latin1]{inputenc}
\usepackage[TS1,T1]{fontenc}
```

We load two packages, tocloft [250], to customize the TOC and the minitocs, and sectsty [182], to customize the sectionning commands:

```
\usepackage{tocloft}
\usepackage{sectsty}
```

We load the minitoc package then some complementary local packages for the french language:

```
\usepackage[french,undotted,tight]{minitoc}
\usepackage[english,francais]{babel}
\usepackage{franc,frnew}
\usepackage{mypatches}
\providecommand{\fup}{\textsuperscript}
```

We make some customizations: indentation for the subsection entries in the TOC and the minitocs, depth of the TOC, numerotation depth, depth of the minitocs, some fonts:

```
\addtolength{\cftsubsecindent}{1em} % for tocloft
\cftsetrmarg{2.55em plus 1fil} % to avoid hyphenations in the ToC (tocloft).
\setcounter{tocdepth}{3}
\setcounter{secnumdepth}{1}
\setcounter{minitocdepth}{4}
\chapterfont{\huge\bfseries\sffamily} % for sectsty
\renewcommand{\thesection}{\arabic{section}}
\sectionfont{\Large\raggedright} % for sectsty (to avoid hyphenations in section titles)
```

Some informations for the title page:

```
\title{Systèmes d'occultation}
\author{Laurent~\textsc{Bloch}}
```

And the document body[2]:

```
\begin{document}

\maketitle

\dominitoc
\tableofcontents

\chapter{Définition et contrôle du travail à~faire}
\label{chap+controle}
\minitoc

\section{Le modèle de la grande industrie et le taylorisme}%
\index{taylorisme}
C'est au \textsc{xviii}\fup{e}~siècle que la vision du travail comme
marchandise est vraiment devenue dominante, pour s'imposer au
\textsc{xix}\fup{e}~siècle dans l'organisation type de la grande usine
industrielle.

\subsection*{Après l'usine, le centre d'appel}\index{centre d'appel}
\addcontentsline{toc}{subsection}{Après l'usine, le centre d'appel}
Aujourd'hui le taylorisme\index{taylorisme} au sens
strict est en déclin parce qu'il n'est plus guère adapté aux
besoins de la production industrielle contemporaine non plus qu'aux
nouvelles normes de comportement individuel et collectif.

\section{Tout travail émet de la pensée}
Le travail a vocation à~produire du sens, pour son auteur comme pour
son destinataire.

\section{Théorie et pratique de la commande publique}
En France, les prestations de service commandées par les
services publics à~des entreprises font l'objet de contrôles
de leur bonne réalisation selon des procédures et des règles
qui sont des cas particuliers d'un ensemble plus vaste, la
réglementation des marchés publics de l'État, dont nous
allons donner ci-dessous une brève description.

\subsection{Réglementation des marchés publics}
Le dispositif juridique, réglementaire et comptable qui encadre les
actes contractuels de la puissance publique en France est très~[...]

\subsubsection{Premier principe: séparation de l'ordonnateur et du comptable}
Le premier élément du dispositif est le principe de
séparation de l'ordonnateur et du comptable. Il a été
instauré en 1319 par l'ordonnance portant création de la~[...]

\subsubsection{Second principe: contrôle \emph{a~priori}}
Le second élément du dispositif est le principe du contrôle
\emph{a~priori}. Lorsque le directeur de l'organisme public
de recherche pris ici comme exemple (l'ordonnateur) décide~[...]
```

[2] The text has been shortened, so there is an undefined reference; do not worry.

```

\subsubsection{Le Code des Marchés Publics}
Le troisième pilier de la commande publique est le Code des Marchés
Publics (CMP), qui régit tous les contrats, conclus par des organismes
publics ou des collectivités territoriales, dont le montant excède un~[...]

\subsection{La pratique des marchés publics}
Lorsque l'administration française fait réaliser un système
informatique par un prestataire, elle est en position de maître
d'ouvrage\index{maîtrise d'ouvrage}.  Elle rédige (ou fait rédiger) un
cahier des charges\index{cahier des charges} qui décrit les
spécifications du système à~réaliser. Ce cahier des charges constitue~[...]

\subsection{Quels sont les services publics «~rentables~»?}
Pour parler comme les informaticiens, nous pouvons identifier un
« effet de bord », c'est-à-dire une conséquence non intentionnelle de
la réglementation des marchés publics: les administrations ne disposent
d'aucun moyen pour envisager la notion d'investissement. Le~[...]

\section{Projet et cahier des charges}\index{cahier des charges}
Jean-Pierre~\textsc{Boutinet} nous guidera ici pour ce qui concerne
l'histoire de la notion de~[...]

\subsection{La frontière entre conception et fabrication}
La vision classique de la conduite d'un projet informatique de gestion
est la suivante: le maître d'ouvrage\index{maîtrise d'ouvrage}~[...]

\subsection{Bâtiment, mécanique, programmation}

Nous y reviendrons au chapitre~\ref{chap+travail}, mais nous savons
déjà que la mise en {\oe}uvre de l'informatique s'est beaucoup
inspirée des procédures de travail les plus élaborées du
\textsc{xx}\fup{e}~siècle~[...]

\chapter{Le travail informatique}
\minitoc

\section{De la nature de l'informatique}

\subsection{Premières croyances}\label{sub+premcroyances}
Les premiers ordinateurs, qui entrèrent en fonction à~l'extrême fin
des années 1940 et durant les années 1950, étaient consacrés à~des
travaux militaires ou scientifiques puisque, à~cette époque,
on pensait~[...]

\subsection{Comment l'informatique diffère des mathématiques}
J'aimerai à~l'occasion de cette analyse attirer l'attention du
lecteur sur une question qui est une source constante de malentendus
au sujet de la programmation.

\subsubsection{Les preuves de programme}
L'écriture de programmes informatiques obéit à~de tout autres
principes. Il convient de préciser cette affirmation pour la
préserver~[...]
```

```

\section{Programmation dans le monde réel}
\subsection{La vraie nature de la programmation des ordinateurs}

Alors, comment s'écrivent les programmes informatiques? Et
d'ailleurs, qu'est-ce qu'une erreur\index{erreur} de programmation?
Ces questions sont liées et elles sont, bien sûr, au c{\oe}ur de notre
préoccupation.

\subsection{Méthodes de programmation}
Un processeur quelconque est caractérisé par le jeu des actions
élémentaires qu'il est capable d'effectuer. Ces actions élémentaires
sont appelées les \emph{primitives} du processeur, ou, si le
processeur est une machine, les «~instructions machine~». Un~[...]

\subsection{Méthodes de construction de programmes}
Nous avons décrit ci-dessus le processus élémentaire de la
programmation, celui qui consiste à~écrire les instructions ou les
expressions qui vont composer un programme.

\subsubsection{La programmation structurée}
Le premier courant de pensée qui associa la recherche d'une syntaxe
claire et expressive à~une organisation logique et commode des unités
de programme fut la \emph{programmation
structurée}\index{programmation!structurée} des années 1970, dont~[...]

\subsubsection{La programmation par objets}
Après la programmation structurée vint un autre courant significatif:
la programmation par objets\index{programmation!par objets}, inventée
en Norvège à~la fin des années 1960 par l'équipe de~[...]

\subsubsection{Excès dans la pensée}
Il y a eu beaucoup de verbiage autour de l'aptitude supposée du
modèle~[...]
\end{document}
⟨/mtc – liv⟩
```

4.17 The `mtc-mem.tex` document file

This example shows the use of the minitoc package in a memoir class document. First, the preamble:

```
⟨∗mtc – mem⟩
\documentclass%
    [oneside]{memoir}
\ProvidesFile{mtc-mem.tex}%
  [2007/01/04]
\usepackage{lipsum}
%%\usepackage{hyperref}
%%\usepackage{memhfixc}
\usepackage%
  [tight]{minitoc}
%%\usepackage{mtcoff}
\begin{document}
```

We use the starred form `\tableofcontents*` specific of the memoir class. Note that the the

\chapter command has *two* optional arguments in the memoir class.

```
\dominitoc
\tableofcontents*

\chapter[oneA][oneB]{oneC}
\minitoc
\lipsum[1]
\newpage
\lipsum[2]
\section{S-1-one}
\lipsum[3]
\section{S-1-two}
\lipsum[4]

\chapter[twoA][twoB]{twoC}
\minitoc
\lipsum[4]
\newpage
\lipsum[5]
\section{S-2-one}
\lipsum[6]
\section{S-2-two}
\lipsum[7]
\end{document}
⟨/mtc – mem⟩
```

4.18 The mtc-mm1.tex document file

This example shows the use of the minitoc package in a memoir class document and shows some of the adaptations necessary for fonts. First, the preamble:

```
⟨*mtc – mm1⟩
\documentclass[oneside]{memoir}
\ProvidesFile{mtc-mm1.tex}%
  [2007/01/04]
\usepackage{lipsum} % filling text
```

We inhibit some font commands of the memoir class:

```
\providecommand{\cftsecfont}{\empty}
\providecommand{\cftsubsecfont}{\empty}
```

Then we redefine them:

```
\renewcommand{\cftsubsecfont}{\normalsize\scshape}
\renewcommand{\cftsubsecfont}{\normalsize\scshape}
```

We load the minitoc package and try to use some minitoc font commands, without success:

```
%% hyperref before minitoc, optional
%%\usepackage[linktocpage=true]{hyperref}\usepackage{memhfixc}
\usepackage[tight]{minitoc}
\mtcsetfont{minitoc}{section}{\normalsize\scshape}   % <- no scshape
\mtcsetfont{minitoc}{subsection}{\normalsize\scshape}% <- no scshape
```

But if we use the font commands of the memoir class, it works!

```
\providecommand{\cftsecfont}{\empty}
\providecommand{\cftsubsecfont}{\empty}
\renewcommand{\cftsubsecfont}{\normalsize\rmfamily\scshape}
\renewcommand{\cftsubsecfont}{\normalsize\rmfamily\scshape}
```

But for mini-table titles (font and text), we can use the minitoc commands:

```
\mtcsettitlefont{minitoc}{\Large\scshape}
%%  this is working beautifully ->
\mtcsettitle{minitoc}{Chapter Contents}
```

The document body:

```
\begin{document}
\dominitoc
\tableofcontents*

\chapter[OneA][OneB]{OneC}
\minitoc
\section{This section}
\lipsum[1]
\section{Second section}
\lipsum[2]
\section{Third section}
\lipsum[3]
\end{document}
⟨/mtc – mm1⟩
```

4.19 The `mtc-mu.tex` document file

This document shows the use of the minitoc package in a document, the minitoc being inserted in the text with the wrapfig package [11].

```
⟨*mtc – mu⟩
\documentclass[12pt]{report}
\ProvidesFile{mtc-mu.tex}%
  [2007/01/04]
\usepackage[tight]{minitoc}
\setlength{\mtcindent}{0pt}
\usepackage{wrapfig}
\newcommand{\LangSig}[1]{\textsc{[#1]}} % smallcaps
\begin{document}
\dominitoc
\tableofcontents
```

A chapter, with its minitoc set in a `minipage`, included in a `wrapfigure` environment on the half of the text width, with some vertical adjustements:

```
\chapter{Mulspren}\label{chapter+mulspren}
\begin{wrapfigure}{r}{0.5\linewidth}
\begin{minipage}{\linewidth}
\vspace{-2.\baselineskip}
\minitoc
\vspace{-1.\baselineskip}
\end{minipage}
\end{wrapfigure}
```

The remaining of the text:

```
The previous chapter examined many end-user programming environments
and found that most contain cognitive programming gulfs.
These gulfs were often created when programing environments used
multiple notations, and could manifest themselves in a variety of
usability problems, ranging from users being unable to understand
a program representation, to not wanting to execute their programs.
Conversely, the previous chapter also found circumstances where multiple
notations helped users understand programs.
It concluded that there was a place for multiple notation programming
environments, but developers had to be very careful to avoid creating
programming gulfs.
It concluded that there was a place for multiple notation programming
environments, but developers had to be very careful to avoid creating
programming gulfs.

This chapter introduces our programming environment, Mulspren.
Mulspren was designed to avoid these gulfs and gain the potential
benefits of multiple notations.
Users program using two notations, one similar to English and one
similar to conventional code.
Changes in one notation are immediately reflected in the other notation,
and users can move rapidly and seamlessly between the notations.
This is programming using dual notations.
When the program is executed, both notations are animated.
Mulspren's language signature is \LangSig{Re/Wr/Wa + Re/Wr/Wa + Wa}.

Papers describing Mulspren have been published in~\cite{Wright02-2}
and~\cite{Wright03-3}.

\section{section 1}
\section{section 2 bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla}
\section{section 3}
\section{section 4}
\section{section 5 bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla}
```

I tried to find some articles of the net to fill the citations:

```
\begin{thebibliography}{1}
\bibitem{Wright02-2}
Tim Wright and Andy Cockburn.
```

```
\newblock Mulspren: a multiple language simulation programming
  environment.
\newblock In {\em HCC '02: Proceedings of the IEEE 2002 Symposia
  on Human Centric Computing Languages and Environments (HCC'02)},
  page 101, Washington, DC, USA, 2002. IEEE Computer Society.

\bibitem{Wright03-3}
Tim Wright and Andy Cockburn.
\newblock Evaluation of two textual programming notations for children.
\newblock In {\em AUIC '05: Proceedings of the Sixth Australasian
  conference on User interface}, pages 55--62, Darlinghurst, Australia,
  Australia, 2005.
  Australian Computer Society, Inc.
\end{thebibliography}
\end{document}
⟨/mtc − mu⟩
```

4.20 The `mtc-sbf.tex` document file

This document shows the use of the minitoc package with a document containing subfigures (here with the subfigure package [94]). We show how to use minilofs and to adjust their depth.

The preamble loads the subfigure package and redefines the format of subfigure entries in the list of figures:

```
⟨∗mtc − sbf⟩
\documentclass[12pt]{report}
\ProvidesFile{mtc-sbf.tex}%
  [2007/01/04]
\usepackage{subfigure}
\makeatletter
\renewcommand{\l@subfigure}{%
    \@dottedxxxline{\ext@subfigure}{2}{3.9em}{3.3em}}
\makeatother
```

We load the varioref package (to have nice cross-references) and the minitoc package:

```
\usepackage{varioref}
\usepackage[tight]{minitoc}
```

We define some features for the layout of the subfigures, then the depth of the list of figures:

```
\newcommand{\goodap}{%
  \hspace{\subfigtopskip}%
  \hspace{\subfigbottomskip}}
\setcounter{lofdepth}{2}
\begin{document}
```

We define the depth of the mini-lists of figures, then some fonts:

```
\mtcsetdepth{minilof}{2}
\mtcsetfont{minitoc}{section}{\small\rmfamily\upshape\bfseries}
\mtcsetfont{partlof}{subfigure}{\small\rmfamily\slshape\bfseries}
\mtcsetfont{partlof}{figure}{\small\rmfamily\upshape\bfseries}
\mtcsetfont{minilof}{subfigure}{\small\rmfamily\slshape\bfseries}
\mtcsetfont{minilof}{figure}{\small\rmfamily\upshape\bfseries}
%% no tables in this document
%% \mtcsetfont{partlot}{subtable}{\small\rmfamily\slshape\bfseries}
%% \mtcsetfont{partlot}{table}{\small\rmfamily\upshape\bfseries}
%% \mtcsetfont{minilot}{subtable}{\small\rmfamily\slshape\bfseries}
%% \mtcsetfont{minilot}{table}{\small\rmfamily\upshape\bfseries}
```

We prepare the minilofs, the table of contents and the list of figures:

```
\dominilof
\tableofcontents
\listoffigures
```

A chapter, with is minilof, twice but with different depths:

```
\chapter{First Chapter}
\minilof
\mtcskip
\mtcsetdepth{minilof}{1}
\minilof
```

A figure containing three subfigures and their captions:

```
\begin{figure}
\centering
   \fbox{%
     \begin{minipage}{3.5in}%
       \raggedright
       \begin{center}
         \subfigure[First]{%
           \fbox{\hbox to 20mm{\vbox to 15mm{\vfil\null}\hfil}}}%
           \hspace{\subfigtopskip}\hspace{\subfigbottomskip}%
         \subfigure[Second Figure]{%
           \fbox{\hbox to 20mm{\vbox to 10mm{\vfil\null}\hfil}}}\\
         \subfigure[Third]{\label{3figs-c}%
           \fbox{\hbox to 20mm{\vbox to 10mm{\vfil\null}\hfil}}}\\
         \caption{Three subfigures.}%
         \label{3figs}%
       \end{center}
       \vspace{4pt}%
       This figure contains two top ‘subfigures’ and
       Figure~\ref{3figs-c}.
       \end{minipage}}
\end{figure}
Figure~\vref{3figs} contains two top ‘‘subfigures’’ and
```

```
Figure~\vref{3figs-c}.
\end{document}
⟨/mtc – sbf⟩
```

4.21 The mtc-scr.tex document file

This document shows the use of the minitoc package with a KOMA-Script document class [147, 195], scrreprt. Some precautions are needed, because these classes have specific interfaces with the TOC (class options and commands).

```
⟨*mtc – scr⟩
\documentclass[12pt,halfparskip,liststotoc,bibtotoc]{scrreprt}
\ProvidesFile{mtc-scr.tex}%
  [2007/01/04]
\setcounter{secnumdepth}{4}
\setcounter{tocdepth}{4}
\usepackage[latin1]{inputenc}
\usepackage{longtable}
```

The hyperref package, if used, must be loaded *before* minitoc:

```
\usepackage{hyperref}
```

With a KOMA-Script class [147, 195], use the k-tight package option in place of tight; as it is a document in german, use also a language package option:

```
\usepackage[k-tight,germanb]{minitoc}
\usepackage[germanb]{babel}
\begin{document}
\pagenumbering{Roman} % page number in Roman, reset to 1 (I)
\dominitoc
\tableofcontents
```

In this document class, with the liststotoc class option, the \listoftables macro adds a chapter entry in the TOC, so we must add \mtcaddchapter after \listoftables:

```
\listoftables
\mtcaddchapter
```

With the bibtotoc class option, it is necessary to add a \adjustmtc command after \bibliography. This problem is similar to the one of compatibility with the tocbibind package.

```
\clearpage
\pagenumbering{arabic} % page number in arabic digits
\setcounter{page}{1} % forced to 1
```

```
\chapter{Test 1}\label{cha:test-1}

Text.

\section{Tabelle}\label{sec:tabelle}

\begin{table}
  \centering
  \begin{longtable}[l]{ll}
  \underline{Ausbildungsbetrieb}  \hspace{10mm} & Kommanditgesellschaft
\\
  \underline{Ausbildender} & Hammer \\
  Ausbildungsstätte & XXXXX Ort
  \end{longtable}
  \caption{Ausbildungsbetrieb}
\end{table}

Text.

\appendix
\chapter{Dokumente}
\minitoc % Aufruf Minitoc

\section{Anhang}\label{sec:anhang}
Text.\\
Hier sollte nun der Anhang sein, davor das Verzeichnis dafür.
\end{document}
⟨/mtc – scr⟩
```

4.22 The `mtc-syn.tex` document file

This document shows the use of the minitoc package when the table of contents is not at the beginning of the document, but is preceded by some starred chapters.

We have the preamble, then we invoke `\dominitoc` to prepare the minitocs:

```
⟨∗mtc – syn⟩
\documentclass[a4paper,twoside,12pt]{book}
\ProvidesFile{mtc-syn.tex}%
  [2007/01/04]
\usepackage{minitoc}
\begin{document}
\dominitoc
```

Some starred chapters, with calls to `\mtcaddchapter` to synchronize. The optional argument is used when an entry in the ToC is wanted:

```
\chapter*{ } % Dedication chapter, no title
\mtcaddchapter           % Added for a starred chapter
```

```
                        % without entry in the ToC
Dedication goes here

\chapter*{Abstract}
\mtcaddchapter[Abstract] % Added for a starred chapter
                         % with an entry in the ToC
Abstract goes here

\chapter*{Declaration}
\mtcaddchapter[Declaration] % Added for a starred chapter
                            % with an entry in the ToC
Declaration goes here
```

The table of contents comes here. Looking at the *document*`.log` file shows that the minitoc files inserted after here are from one chapter to far: we add a correction.

```
\tableofcontents
\decrementmtc % Synchro added (look at the .log file)
```

The following chapters are normal (unstarred):

```
\chapter{Chapter One}
\minitoc

\section{Section 1}
Some text.

\section{Section 2}
Some text.

\chapter{Chapter Two}
\minitoc

\section{Another Section 1}
Some more text.

\section{Another Section 2}
Some more text.

\end{document}
⟨/mtc – syn⟩
```

4.23 The `mtc-tbi.tex` document file

This document shows the use of the minitoc package with a document using the tocbibind package [253].

```
⟨*mtc – tbi⟩
\documentclass[a4paper]{report}
\ProvidesFile{mtc-tbi.tex}%
  [2007/01/04]
```

We dont want an entry for the TOC in the TOC: option nottoc for the tocbibind package:

```
\usepackage[nottoc]{tocbibind}
\usepackage[tight]{minitoc}
\begin{document}
\dominitoc
\tableofcontents
```

As there is no entry for the TOC in the TOC, no correction is necessary; we comment out the usual correction[3]:

```
%% tocbibind compatibility
%% not used if nottoc option :
%% \mtcaddchapter[]
```

A starred chapter with an entry in the TOC; we add it;

```
\chapter*{Introduction}
\mtcaddchapter[Introduction]
```

Some normal chapters:

```
\chapter{Title of chapter~1}
\minitoc
\section{as1}
\section{as2}
\chapter{Title of chapter~2}
\minitoc
\section{bs1}
\section{bs2}
\chapter{Title of chapter~3}
\minitoc
\section{cs1}
\section{cs2}
\chapter{Title of chapter~4}
\minitoc
\section{ds1}
\section{ds2}
```

Another starred chapter, with an entry in the TOC:

```
\chapter*{Conclusion}
\mtcaddchapter[Conclusion]
```

Yet another starred chapter, with an entry in the TOC, but with starred sections, also listed in the TOC:

```
\chapter*{Appendices}
\mtcaddchapter[Appendices]
\minitoc
\section*{first appendix}
\addcontentsline{toc}{section}%
  {first appendix}
\section*{second appendix}
\addcontentsline{toc}{section}%
  {second appendix}
```

The list of figures has an entry in the TOC (via the tocbibind package), so a correction must be applied:

```
\listoffigures
%% tocbibind compatibility
\mtcaddchapter
```

The tocbibind package adds an entry in the TOC for the bibliography, so we must add the recommended correction:

```
\begin{thebibliography}{3}
   \bibitem {s1}{title ...}
\end{thebibliography}
%% tocbibind compatibility
\adjustmtc
\end{document}
```

⟨/mtc – tbi⟩

[3] It is recommended to keep this comment; you could change your mind.

4.24 The mtc-tlc.tex document file

This document shows the use of the minitoc package in a document of the article class. It is the example of [189, page 58], modernized.

```
⟨*mtc – tlc⟩
\documentclass{article}
\ProvidesFile{mtc-tlc.tex}%
  [2007/01/04]
```

Dimensions of the text on the page:

```
\setlength{\textwidth}%
   {124.20126pt}
\setlength{\textheight}%
   {19\baselineskip}
```

We load the minitoc package and set some parameters (indentation, base font and depth) for the secttocs:

```
\usepackage{minitoc}
\setlength{\stcindent}{0pt}
\mtcsetfont{secttoc}{*}%
   {\footnotesize}
\mtcsetdepth{secttoc}{3}
```

We prepare the secttocs, without title, and the table of contents which is not printed:

```
\begin{document}
\dosecttoc[e]
\faketableofcontents

\section{Afghanistan}
\secttoc
\subsection{Geography}
\subsubsection{Total area}
   647,500 km2
\subsubsection{Land area}
   647,500 km2
\subsection{History} \ldots

\section{Albania}
\secttoc
\subsection{Geography}
\subsubsection{Total area}
   28,750 km2
\subsubsection{Land area}
   27,400 km2
\subsection{History} \ldots
\end{document}
⟨/mtc – tlc⟩
```

4.25 The mtc-tsf.tex document file

This document[4] shows the use of the minitoc package with a document containing subfigures (here with the subfig package [96]). We show how to use minilofs and to adjust their depth.

The preamble loads the subfig package and redefines the format of subfigure entries in the list of figures:

```
⟨*mtc – tsf⟩
\documentclass{report}
\ProvidesFile{mtc-tsf.tex}%
  [2007/01/04]
\usepackage{fullpage}
\usepackage[config=altsf]{subfig}
\usepackage[tight]{minitoc}
```

[4] It is derived from one of the examples distributed with the subfig package [96].

This is utility code to make graduated rules and a box around a figure.

```
\newdimen\testtemp
\newcommand{\ru}[1]{%
  \testtemp #1%
  \advance\testtemp .5pt%
  \divide\testtemp 2%
  \hbox to \testtemp{\leaders\hbox to 1mm{%
    \vrule height1mm depth0pt width.25pt\hfil}\hfil}%
  \hbox to 0pt{\hss\vrule height3mm depth0pt width.25pt\hss}%
  \hbox to \testtemp{\leaders\hbox to 1mm{%
    \hfil\vrule height1mm depth0pt width.25pt}\hfil}}
%%
\fboxsep=-\fboxrule
\newcommand{\figbox}[1]{%
  \fbox{%
    \vbox to 1in{%
      \vfil
      \hbox to 2in{%
        \parbox{2in}{%
          \centering
          #1}}%
      \vfil
      \vbox to 0pt{%
        \vss
        \hbox to 2in{%
          \hfil
          \ru{1.1in}%
          \hfil}}}}}
```

The body of the document. We set the depth of the list of figures and prepare the minilofs and the list of figures:

```
\begin{document}
\setcounter{lofdepth}{2}
\dominilof
\listoffigures
\newpage
```

A chapter containing a figure with subfigures. We print a minilof twice, with different depths:

```
\chapter{Reference Test}
\minilof
\mtcskip
\mtcsetdepth{minilof}{1}
\minilof

\begin{figure}[ht]%
  \centering
  \subfigure{%
    \label{fig+A}%
    \figbox{SUBFIGURE ONE:\\
```

```
            (no opt)}}
  \quad
  \subfigure[]{%
    \label{fig+B}%
    \figbox{SUBFIGURE TWO:\\
            (empty opt)}}\\

  \subfigure[Subfigure Three.]{%
    \label{fig+C}%
    \figbox{SUBFIGURE THREE:\\
            (opt)}}
  \quad
  \subfigure[][Subfigure Four.]{%
    \label{fig+D}%
    \figbox{SUBFIGURE FOUR:\\
            (empty opt and opt)}}
  \quad
  \subfigure[][]{%
    \label{fig+E}%
    \figbox{SUBFIGURE FIVE:\\
            (both empty opt)}}\\

  \subfigure[The Sixth Subfigure.][Subfigure Six.]{%
    \label{fig+F}%
    \figbox{SUBFIGURE SIX:\\
            (both opt)}}
  \quad
  \subfigure[The Seventh Subfigure][]{%
    \label{fig+G}%
    \figbox{SUBFIGURE SEVEN:\\
            (opt and empty opt)}}

  \caption{Optional argument test.}%
  \label{fig+main}%
\end{figure}

The figure~\ref{fig+main} on page~\pageref{fig+main} is composed
of the seven subfigures~\subref{fig+A} (aka: \ref{fig+A}),
\subref{fig+B} (aka: \ref{fig+B}), \subref{fig+C} (aka: \ref{fig+C}),
\subref{fig+D} (aka: \ref{fig+D}), \subref{fig+E} (aka: \ref{fig+E}),
\subref{fig+F} (aka: \ref{fig+F}), and \subref{fig+G} (aka: \ref{fig+G}).

\section{Centering Test}
Note that figures~\ref{fig+B}, \ref{fig+E} and \ref{fig+G} are centered.
This means that \verb|\subfigcapskip| has been set to zero and is not
offsetting the simple label to the left.  Also the remaining captioned
subfigures (figures~\ref{fig+C}, \ref{fig+D}, and \ref{fig+F}) should
have centered labels.
\end{document}
```

⟨/mtc – tsf⟩

Chapter 5

Messages

Contents

5.1 Introduction

This chapter lists and comments the messages given by the minitoc package, and is associates, mtcoff and mtcpatchmem[1].

- The first line of each message contains usually the name of the package and an unique identifier (this identifier may be useful to search in this chapter of the documentation, but has no special meaning, except the leading letter: `I` for informative, `W` for warning, and `E` for error).
- Informative messages are written only in the *document*`.log` file; the prefix is `F` for the warning messages from the mtcoff package and `M` for the informative message from the mtcpatchmem package.
- Warning messages are shown on the screen (but often too quickly to be seen, because normally LaTeX does not stop for warnings) and written in the *document*`.log` file.

[1] The texts of the messages given in this chapter may slighty differ from the real text, because some messages contain variable elements and the layout may differ.

- Error messages are shown on the screen and written in the *document*`.log` file, but LaTeX stops, so you can ask for help by typing "h".
- In this chapter, some words in the messages are typeset in italic characters; they represent the variable parts of the message:
  - *ARG1* The first argument of the command.
  - *ARG2* The second argument of the command.
  - *ARG3* The third argument of the command.
  - *CLASS* The name of the class of your document.
  - *command* The name of the command.
  - *counter* The name of a LaTeX counter.
  - *document* The name of your document, without its `.tex` extension.
  - *extension* The extension part of the name of a file.
  - *file* The name of a file (often a minitoc auxiliary file, the *document*`.toc` file, the *document*`.lof` file, or the *document*`.lot` file).
  - *internal_name* The name of an internal macro redefined by a `\mtcset...` command.
  - *LANGUAGE* The name of the language (for `\mtcselectlanguage`).
  - *LINE* The number of the line in the source file.
  - *macro* The name of a minitoc font command redefined by `\mtcsetfont`.
  - *new_title* The new value of a title redefined by `\mtcsettitle`.
  - *NUMBER* The number of mini-tables of the given type in your document (when you are using short extensions and that number is greater than 99).
  - *PREPARATION* The name of a minitoc preparation command.
  - *sequence* A sequence of commands used to redefine an internal macro via a `\mtcset...` command.
  - *STRING* A string of characters, often part of a command name.
  - *value* The new value of a counter.

The messages are produced using macros from the mtcmess package, described in chapter 11 on page 427.

5.2 Messages from the minitoc package

5.2.1 Informative messages

```
Package: minitoc 2007/01/09 v51 Package minitoc (JPFD)
```

This is the announce message of the minitoc package, with its name, date and version. "JPFD" are my initials. This message has no identifier because it is emitted by `\ProvidesPackage`.

```
Package minitoc Info: <I0001>
(minitoc)             *** minitoc package, version 51 ***.
```

Remember the version of the package.

```
Package minitoc Info: <I0002>
(minitoc)             Autoconfiguration of extensions.
```

The minitoc package tries to determine if short or long extensions for file names are used by the operating system.

```
Package minitoc Info: <I0003>
(minitoc)             chapter level macros available.
```

The `\chapter` sectionning command is available, so you can use the mini-table commands at the chapter level, but *not* the mini-table commands at the section level.

```
Package minitoc Info: <I0004>
(minitoc)             chapter level macros NOT available.
```

The `\chapter` sectionning command is *not* available, so you cannot use the mini-table commands at the chapter level, but, if the `\section` sectionning command is available, you can use mini-table commands at the section level.

```
Package minitoc Info: <I0005>
(minitoc)       compatible with hyperref.
```

This version of minitoc is compatible with the hyperref package.

```
Package minitoc Info: <I0006>
(minitoc)            document.extension is empty on input line LINE.
```

The auxiliary file for a mini-table is found empty (or inexistent) when minitoc tries to insert it. If the `checkfiles` option is active, it is skipped.

```
Package minitoc Info: <I0007>
(minitoc)             Horizontal rules are activated
(minitoc)             for the ARG1s on input line LINE.
```

The horizontal rules will be present in the mini-tables of type *ARG1*.

```
Package minitoc Info: <I0008>
(minitoc)             Horizontal rules are inhibited
(minitoc)             for the ARG1s on input line LINE.
```

The horizontal rules will be omitted in the mini-tables of type *ARG1*.

```
Package minitoc Info: <I0009>
(minitoc)             Listing minitoc auxiliary files.
(minitoc)             Creating the document.maf file.
```

You have used the `listfiles` package option. A list of the minitoc auxiliary files is written in the *document*`.maf` file. It may be helpful to delete these files. See section 1.7 on page 49. This option is the default since version #48.

```
Package minitoc Info: <I0010>
Package minitoc Info: The LANGUAGE language is selected.
(minitoc)             on input line LINE.
```

The *LANGUAGE*.mld file has been successfully loaded for the *LANGUAGE* language[2] by the \mtcselectlanguage command at line *LINE*. The titles for the mini-tables are changed.

```
Package minitoc Info: <I0011>
(minitoc)             LANGUAGE language object selected.
(minitoc)              on input line LINE.
```

The \mtcselectlanguage macro has successfully (indirectly) loaded the *LANGUAGE*.mlo minitoc object file.

```
Package minitoc Info: <I0012>
(minitoc)             Long extensions (Unix-like) will be used.
```

The autoconfiguration has detected that your operating system is able to use long extensions; this will be the default.

```
Package minitoc Info: <I0013>
(minitoc)             \mtcsetdepth redefines the counter
(minitoc)             "counter" as "value" on input line LINE.
```

The \mtcsetdepth macro changes the value of the specified depth counter and forces it to *value*.

```
Package minitoc Info: <I0014>
(minitoc)             \mtcsetfeature redefines the macro
(minitoc)             "\internal_name" as
(minitoc)             "\sequence" on input line LINE.
```

[2] The english.mld file is always loaded first, to have english as default language.

The \mtcsetfeature macro has redefined the internal macro *internal_name* with the given *sequence*.

```
Package minitoc Info: <I0015>
(minitoc)             \mtcsetfont redefines the macro
(minitoc)             "macro" as "sequence" on input line LINE.
```

The \mtcsetfont command redefines the (old style) *macro* by the given *sequence* of font commands.

```
Package minitoc Info: <I0016>
(minitoc)             \mtcsetformat redefines the macro
(minitoc)             "\internal_name" as "ARG3" on input line LINE.
```

The macro \mtcsetformat redefines an internal macro with the value given by its third argument.

```
Package minitoc Info: <I0017>
(minitoc)             \mtcsettitle redefines the macro
(minitoc)             "internal_name" as
(minitoc)             "new_title" on input line LINE.
```

A mini-table title is redefined via the \mtcsettitle macro.

```
Package minitoc Info: <I0018>
(minitoc)             \mtcsettitlefont redefines the macro
(minitoc)             "\internal_name" as
(minitoc)             "sequence" on input line LINE.
```

The \mtcsettitlefont macro redefines the (old style) *internal_name* macro which the given sequence *sequence*.

```
Package minitoc(hints) Info: <I0019>
(minitoc(hints))             No hints have been written
(minitoc(hints))             in the document.log file.
```

The hints package option has detected no potential problem.

```
Package minitoc Info: <I0020>
(minitoc)             old version of the memoir class.
```

The version of the memoir class is old. The minitoc package does not need to patch this class.

```
Package minitoc Info: <I0021>
(minitoc)             Page numbers are activated
(minitoc)             for the ARG1s on input line LINE.
```

The page numbers will be present in the mini-tables of type *ARG1*.

```
Package minitoc Info: <I0022>
(minitoc)             Page numbers are inhibited
(minitoc)             for the ARG1s on input line LINE.
```

The page numbers will be omitted in the mini-tables of type *ARG1*.

```
Package minitoc Info: <I0023>
(minitoc)             part level macros available.
```

The \part sectionning command is available, so you can use the mini-table commands at the part level.

```
Package minitoc Info: <I0024>
(minitoc)             PREPARING MINITOCS FROM file on input line LINE.
```

A \dominitoc command prepares the minitoc auxiliary files for minitocs from *file*.

```
Package minitoc Info: <I0025>
(minitoc)             PREPARING PARTTOCS FROM file on input line LINE.
```

A \doparttoc command prepares the parttoc auxiliary files for parttocs from *file*.

```
Package minitoc Info: <I0026>
(minitoc)             PREPARING SECTTOCS FROM file on input line LINE.
```

A \dosecttoc command prepares the secttoc auxiliary files for secttocs from *file*.

```
Package minitoc Info: <I0027>
(minitoc)             recent version of the memoir class.
```

The version of the memoir class is recent. The minitoc package will try to patch it.

```
Package minitoc Info: <I0028>
(minitoc)             section level macros available.
```

The \section sectionning command is available but the \chapter sectionning command is *not* available, so you can use the mini-table commands at the section level.

```
Package minitoc Info: <I0029>
(minitoc)             section level macros NOT available.
```

The \section sectionning command is not defined (by the document class), so the section level commands of the minitoc package are not available.

```
Package minitoc Info: <I0030>
(minitoc)             the memoir class is loaded:
(minitoc)             compatibility attempted.
```

The memoir document class is used. The minitoc package tries to ensure compatibility.

```
Package minitoc Info: <I0031>
(minitoc)             ==> this version is configured for UNIX-like
(minitoc)             (long extensions) file names.
```

The autoconfiguration has detected that your operating uses UNIX-like (long extensions) file names.

```
Package minitoc Info: <I0032>
(minitoc)             This version of the memoir class uses
(minitoc)             a version of \chapter which is
(minitoc)             incompatible with the minitoc package.
(minitoc)             We try to patch.
```

The memoir class uses a version of the `\chapter` command which needs to be corrected because its syntax has been changed. A patch is loaded.

```
Package minitoc Info: <I0033>
(minitoc)             Writing document.extension.
```

An auxiliary file for a mini-table is written by a minitoc preparation command (like `\dominitoc`).

```
Package minitoc Info: <I0034>
(minitoc)             PREPARING MINILOFS FROM file on input line LINE.
```

A `\dominilof` command prepares the minilof auxiliary files for minilofs from *file*.

```
Package minitoc Info: <I0035>
(minitoc)             PREPARING PARTLOFS FROM file on input line LINE.
```

A \dopartlof command prepares the partlof auxiliary files for partlofs from *file*.

```
Package minitoc Info: <I0036>
(minitoc)             PREPARING SECTLOFS FROM file on input line LINE.
```

A \dosectlof command prepares the sectlof auxiliary files for sectlofs from *file*.

```
Package minitoc Info: <I0037>
(minitoc)             PREPARING MINILOTS FROM file on input line LINE.
```

A \dominilot command prepares the minilot auxiliary files for minilots from *file*.

```
Package minitoc Info: <I0038>
(minitoc)             PREPARING PARTLOTS FROM file on input line LINE.
```

A \dopartlot command prepares the partlot auxiliary files for partlots from *file*.

```
Package minitoc Info: <I0039>
(minitoc)             PREPARING SECTLOTS FROM file on input line LINE.
```

A \dosectlot command prepares the sectlot auxiliary files for sectlots from *file*.

5.2.1.1 Informative messages for hints

```
Package minitoc(hints) Info: <I0040>
(minitoc(hints))             The ‘‘abstract’’ package has been
(minitoc(hints))             loaded with the ‘‘addtotoc’’ option.
(minitoc(hints))             You need to look at the
(minitoc(hints))             documentation to adjust.
```

As you are using the abstract package with its addtotoc option, you should look at the minitoc package documentation for specific precautions. See section 2.27 on page 67.

```
Package minitoc(hints) Info: <I0041>
(minitoc(hints))             --- The amsbook class is loaded.
(minitoc(hints))             See the minitoc package documentation
(minitoc(hints))             for specific precautions.
```

As you are using the amsbook class, you should look at the minitoc package documentation for specific precautions. See section 2.24 on page 63.

```
Package minitoc(hints) Info: <I0042>
(minitoc(hints))             --- The appendix package is loaded.
(minitoc(hints))             See the minitoc package documentation
(minitoc(hints))             for specific precautions.
```

As you are using also the appendix package, you should look at the minitoc package documentation for specific precautions. See section 2.20 on page 60.

```
Package minitoc(hints) Info: <I0043>
(minitoc(hints))             --- The KOMAScript CLASS class is loaded.
(minitoc(hints))             See the minitoc package documentation
(minitoc(hints))             for specific precautions.
```

As you are using also the *CLASS* class, you should look at the minitoc package documentation for specific precautions. See section 1.5.5 on page 47. The classes involved here are scrbook, scrreprt, and scrartcl, i.e., the KOMA-Script classes [147, 195] compatible with minitoc.

```
Package minitoc(hints) Info: <I0044>
(minitoc(hints))             --- The memoir class is loaded.
(minitoc(hints))             See the minitoc package documentation
(minitoc(hints))             for specific precautions.
```

As you are using the memoir class, you should look at the minitoc package documentation for specific precautions. See section 2.22 on page 62.

```
Package minitoc(hints) Info: <I0045>
(minitoc(hints))             The \PREPARATION command
(minitoc(hints))             has been invoked more than once
(minitoc(hints))             on input line LINE.
```

A minitoc preparation command has been invoked more than once.

```
Package minitoc(hints) Info: <I0046>
(minitoc(hints))             --- The tocbibind package is loaded.
(minitoc(hints))             See the minitoc package documentation
(minitoc(hints))             for specific precautions.
```

As you are using also the tocbibind package, you should look at the minitoc package documentation for specific precautions. See section 1.5.5 on page 47.

```
Package minitoc(hints) Info: <I0047>
(minitoc(hints))             --- The tocloft package is loaded.
(minitoc(hints))             See the minitoc package documentation
(minitoc(hints))             for specific precautions.
```

As you are using also the tocloft package, you should look at the minitoc package documentation for specific precautions. See section 2.21 on page 61.

```
Package minitoc(hints) Info: <I0048>
(minitoc(hints))             Using \mtcprepare may induce some
(minitoc(hints))             hints about the preparation commands,
(minitoc(hints))             because it invokes ALL the preparation
(minitoc(hints))             commands allowed by the document class,
(minitoc(hints))             without any previous check.
```

The `\mtcprepare` command invoke all the possible preparation commands, depending only on the document class and the available contents files. It does not known exactly what you want, so it can prepare too many mini-tables files.

```
Package minitoc(hints) Info: <I0049>
(minitoc(hints))             ==> You requested the hints option
(minitoc(hints))             Some hints are eventually given below.
```

As you have requested the `hints` package option (which is set by default), some "hints" are eventually given in the *document*`.log` file. You can find them easily by searching for the string "`minitoc(hints)`" with a text editor.

```
Package minitoc Warning: <I0050>
(minitoc)                The required "LANGUAGE.mld" file is missing.
(minitoc)                The "LANGUAGE" language option
(minitoc)                will not be available.
(minitoc)                Please install it from a recent distribution
(minitoc)                or from the CTAN archives.
```

The *LANGUAGE*`.mld` has not been installed on your system. You should take it from a recent distribution or from the CTAN archives to complete your installation, else the *LANGUAGE* language option will not be available.

```
Package minitoc Warning: <I0051>
(minitoc)                The required "LANGUAGE.mlo" file is missing.
(minitoc)                The "LANGUAGE" language option
(minitoc)                will not be available.
(minitoc)                Please install it from a recent distribution
(minitoc)                or from the CTAN archives.
```

The *LANGUAGE*.mlo has not been installed on your system. You should take it from a recent distribution or from the CTAN archives to complete your installation, else the *LANGUAGE* language option will not be available.

5.2.2 Warning messages

```
Package minitoc Warning: <W0001>
(minitoc)                \chapter and \section are undefined.
(minitoc)                Cannot use \mtcfixglossary without
(minitoc)                optional argument [part].
```

The sectionning commands \chapter and \section are not defined (by the document class), hence the \mtcfixglossary macro cannot be used without an optional argument (try \part). This situation is very unlikely to happen, so also verify your document class.

```
Package minitoc Warning: <W0002>
(minitoc)                \chapter and \section are undefined.
(minitoc)                Cannot use \mtcfixindex without
(minitoc)                optional argument [part].
```

The sectionning commands \chapter and \section are not defined (by the document class), hence the \mtcfixindex macro cannot be used without an optional argument (try \part). This situation is very unlikely to happen, so also verify your document class.

```
Package minitoc Warning: <W0003>
(minitoc)                \firstchapteris is an obsolete
(minitoc)                command on input line LINE.
```

You have used an obsolete command (\firstchapteris). You should remove it.

```
Package minitoc Warning: <W0004>
(minitoc)                \firstpartis is an obsolete
(minitoc)                command on input line LINE.
```

You have used an obsolete command (\firstpartis). You should remove it.

```
Package minitoc Warning: <W0005>
(minitoc)                \firstsectionis is an obsolete
(minitoc)                command on input line LINE.
```

You have used an obsolete command (`\firstsectionis`). You should remove it.

```
Package minitoc Warning: <W0006>
(minitoc)                \mtcfixglossary can only be used
(minitoc)                with the [part] optional argument,
(minitoc)                which becomes the default.
```

The `\mtcfixglossary` macro can only use `[part]` as optional argument (which becomes the default), because `\chapter` and `\section` are not defined.

```
Package minitoc Warning: <W0007>
(minitoc)                \mtcfixindex can only be used
(minitoc)                with the [part] optional argument,
(minitoc)                which becomes the default.
```

The `\mtcfixindex` macro can only use `[part]` as optional argument (which becomes the default), because `\chapter` and `\section` are not defined.

```
Package minitoc Warning: <W0008>
(minitoc)                No file file.
(minitoc)                MINILOFS NOT PREPARED on input line LINE.
```

The *file* cannot be found, because it has not been created by a `\dominilof` command. Please check if you have called `\dominilof` in the correct sequence of commands.

```
Package minitoc Warning: <W0009>
(minitoc)                No file file.
(minitoc)                MINILOTS NOT PREPARED on input line LINE.
```

The *file* cannot be found, because it has not been created by a `\dominilot` command. Please check if you have called `\dominilot` in the correct sequence of commands.

```
Package minitoc Warning: <W0010>
(minitoc)                No file file.
(minitoc)                MINITOCS NOT PREPARED on input line LINE.
```

The *file* cannot be found, because it has not been created by a `\dominitoc` command. Please check if you have called `\dominitoc` in the correct sequence of commands.

```
Package minitoc Warning: <W0011>
(minitoc)                No file file.
(minitoc)                PARTLOFS NOT PREPARED on input line LINE.
```

The *file* cannot be found, because it has not been created by a `\dopartlof` command. Please check if you have called `\dopartlof` in the correct sequence of commands.

```
Package minitoc Warning: <W0012>
(minitoc)                No file file.
(minitoc)                PARTLOTS NOT PREPARED on input line LINE.
```

The *file* cannot be found, because it has not been created by a `\dopartlot` command. Please check if you have called `\dopartlot` in the correct sequence of commands.

```
Package minitoc Warning: <W0013>
(minitoc)                No file file.
(minitoc)                PARTTOCS NOT PREPARED on input line LINE.
```

The *file* cannot be found, because it has not been created by a `\doparttoc` command. Please check if you have called `\doparttoc` in the correct sequence of commands.

```
Package minitoc Warning: <W0014>
(minitoc)                No file file.
(minitoc)                SECTLOFS NOT PREPARED on input line LINE.
```

The *file* cannot be found, because it has not been created by a `\dosectlof` command. Please check if you have called `\dosectlof` in the correct sequence of commands.

```
Package minitoc Warning: <W0015>
(minitoc)                No file file.
(minitoc)                SECTLOTS NOT PREPARED on input line LINE.
```

The *file* cannot be found, because it has not been created by a `\dosectlot` command. Please check if you have called `\dosectlot` in the correct sequence of commands.

```
Package minitoc Warning: <W0016>
(minitoc)                No file file.
(minitoc)                SECTTOCS NOT PREPARED on input line LINE.
```

The *file* cannot be found, because it has not been created by a `\dosecttoc` command. Please check if you have called `\dosecttoc` in the correct sequence of commands.

```
Package minitoc Warning: <W0017>
(minitoc)                no section or chapter level macros available
(minitoc)                PLEASE VERIFY YOUR MAIN DOCUMENT CLASS.
```

The `\chapter` and `\section` sectionning commands are not defined. Your document class is likely without any sectionning command, so the minitoc package is pointless. *Verify your main document class.*

```
Package minitoc Warning: <W0018>
Package minitoc Warning: part level macros NOT available.
```

The `\part` sectionning command is not defined (by the document class), so the part level commands of the minitoc package are not available. It is a warning message because most classes with sectionning commands define the `\part` command, so you should verify which class you are using.

```
Package minitoc Warning: <W0019>
(minitoc)                Short extensions (MSDOS-like) will be used.
(minitoc)                ==> this version is configured for MSDOS-like
(minitoc)                    (8+3) file names.
```

The autofiguration has found that the operating system uses file names with short extensions (8+3 scheme).

```
Package minitoc Warning: <W0020>
(minitoc)                You have forced the use of short extensions.
```

You have used the `shortext` package option to force the use of short extensions (8+3 scheme). This action limits the number of usable mini-tables of each kind and may be problematic if you have more that 99 mini-tables of the same kind. If your operating system allows for long extensions, do not use the `shortext` package option, except for testing purposes.

```
Package minitoc Warning: <W0021>
(minitoc)                Your version of latex.tex is obsolete.
(minitoc)                Trying to continue...
```

You are using an obsolete version of LaTeX, but the minitoc package will try to continue. It would be better to update your LaTeX installation.

```
Package minitoc Warning: <W0022>
Package minitoc Warning: Your version of latex.tex is very obsolete.
(minitoc)                Trying to continue... crossing fingers.
```

Your version of LaTeX is very obsolete, and almost unusable with the minitoc package. You can try to continue the compilation, but you are urged to update your LaTeX installation as soon as possible.

```
Package minitoc Warning: <W0093>
(minitoc)                Some "*.mld" or "*.mlo" files are missing
(minitoc)                in your installation.
(minitoc)                Search for the I0050 and I0051 info messages
(minitoc)                in the \jobname.log file.
(minitoc)                Please install the missing files from
(minitoc)                a recent distribution
(minitoc)                or from the CTAN archives.
```

Some .mld or .mlo files have not been installed on your system. Search for the I0050 and I0051 info messages in the *document*.log file to find which files are missing. You can retrieve them from a recent distribution or from the CTAN archives to complete your installation, else some language options will not be available. The list of the missing files is given in the W0094 warning message.

```
Package minitoc Warning: <W0094>
(minitoc)                Missing minitoc language file(s):
(minitoc)                ...
```

Some .mld or .mlo files have not been installed on your system. The list is given in the message.

5.2.2.1 Warning messages for hints

```
Package minitoc(hints) Warning: <W0023>
(minitoc(hints))                --- It may be the consequence
(minitoc(hints))                of loading the ‘‘hyperref’’ package.
```

Some sectionning commands have been altered *after* the loading of the minitoc package. The hyperref package does that, but it is harmless. For other packages or user-made alterations, it is recommended to alter the sectionning commands only *before* loading the minitoc package. See section 2.17 on page 59.

```
Package minitoc(hints) Warning: <W0024>
(minitoc(hints))                Some hints have been written
(minitoc(hints))                in the document.log file.
```

The hints package option has detected some potential problems and written hints into the *document*.log file. You can search it for the “minitoc(hints)” string with a text editor.

```
Package minitoc(hints) Warning: <W0025>
(minitoc(hints))                --- The alphanum package is loaded.
(minitoc(hints))                It is incompatible
(minitoc(hints))                with the minitoc package.
```

You are using the alphanum package which is incompatible with the minitoc package. The compilation can continue, but the result could be unsatisfactory.

```
Package minitoc(hints) Warning: <W0026>
(minitoc(hints))                --- The amsart class is loaded.
(minitoc(hints))                It is incompatible
(minitoc(hints))                with the minitoc package.
```

You are using the amsart document class which is incompatible with the minitoc package. The compilation can continue, but the result could be unsatisfactory.

```
Package minitoc(hints) Warning: <W0027>
(minitoc(hints))                --- The amsproc class is loaded.
(minitoc(hints))                It is incompatible
(minitoc(hints))                with the minitoc package.
```

You are using the amsproc document class which is incompatible with the minitoc package. The compilation can continue, but the result could be unsatisfactory.

```
Package minitoc(hints) Warning: <W0028>
(minitoc(hints))                --- The \chapter command is altered
                                after minitoc.
```

Some packages alter the sectionning commands, like `\chapter`. Most of them should be loaded *before* the minitoc package. The hyperref package, even if it is loaded *before* the minitoc package (as recommended), alters the sectionning commands in an `\AtBeginDocument`, so this message is always printed when you use the hyperref package with minitoc, but then it is harmless.

```
Package minitoc(hints) Warning: <W0029>
Package minitoc(hints) Warning: --- The jura class is loaded.
(minitoc(hints))                It is incompatible
(minitoc(hints))                with the minitoc package.
```

You are using the jura document class which is incompatible with the minitoc package. The compilation can continue, but the result could be unsatisfactory.

```
Package minitoc(hints) Warning: <W0030>
(minitoc(hints))                --- The \part command is altered
                                after minitoc.
```

Some packages alter the sectionning commands, like `\part`. Most of them should be loaded *before* the minitoc package. The hyperref package, even if it is loaded *before* the minitoc package (as recommended), alters the sectionning commands in an `\AtBeginDocument`, so this message is always printed when you use the hyperref package with minitoc, but then it is harmless.

```
Package minitoc(hints) Warning: <W0031>
(minitoc(hints))                --- The placeins package is loaded
(minitoc(hints))                without the section option,
(minitoc(hints))                but minitoc used the insection option
(minitoc(hints))                which implies it. Try to inverse the
(minitoc(hints))                loading order and use consistent options.
(minitoc(hints))                You may have got a message
(minitoc(hints))        ! LaTeX Error: Option clash for package placeins.
```

You are using the placeins package, but without its `section` option, while minitoc is called with its `insection` option which implies it. See page 29, near a "dangerous bend" symbol like the one shown in the margin.

```
Package minitoc(hints) Warning: <W0032>
(minitoc(hints))                --- The placeins package loaded is
(minitoc(hints))                too old. You should use a version
(minitoc(hints))                dated of 2005/04/18 at least.
```

You are using an obsolete version of the placeins package. Please update it from the CTAN archives or a recent distribution.

```
Package minitoc(hints) Warning: <W0033>
(minitoc(hints))                The caption package should be
(minitoc(hints))                loaded BEFORE the minitoc package.
```

The caption package alters some commands and must be loaded *before* the minitoc package. See section 2.31 on page 69.

```
Package minitoc(hints) Warning: <W0034>
(minitoc(hints))                The caption2 package should be
(minitoc(hints))                loaded BEFORE the minitoc package.
```

The caption2 package alters some commands and must be loaded *before* the minitoc package. See section 2.31 on page 69. Note that the caption2 package is now obsolete; please use a recent version of the caption package.

```
Package minitoc(hints) Warning: <W0035>
(minitoc(hints))                The ccaption package should be
(minitoc(hints))                loaded BEFORE the minitoc package.
```

The ccaption package alters some commands and must be loaded *before* the minitoc package. See section 2.31 on page 69.

```
Package minitoc(hints) Warning: <W0036>
(minitoc(hints))                The mcaption package should be
(minitoc(hints))                loaded BEFORE the minitoc package.
```

The mcaption package alters some commands and must be loaded *before* the minitoc package. See section 2.31 on page 69.

```
Package minitoc(hints) Warning: <W0037>
(minitoc(hints))                The sectsty package should be
(minitoc(hints))                loaded BEFORE the minitoc package.
```

The sectsty package alters some commands and must be loaded *before* the minitoc package. See section 2.28 on page 67.

```
Package minitoc(hints) Warning: <W0038>
(minitoc(hints))                The varsects package should be
(minitoc(hints))                loaded BEFORE the minitoc package.
```

The varsects package alters some commands and must be loaded *before* the minitoc package. See section 2.33 on page 69.

```
Package minitoc(hints) Warning: <W0039>
(minitoc(hints))                --- The \section command is altered
                                after minitoc.
```

Some packages alter the sectionning commands, like `\section`. Most of them should be loaded *before* the minitoc package. The hyperref package, even if it is loaded *before* the minitoc package (as recommended), alters the sectionning commands in an `\AtBeginDocument`, so this message is always printed when you use the hyperref package with minitoc, but then it is harmless.

```
Package minitoc(hints) Warning: <W0040>
(minitoc(hints))                --- The titletoc package is loaded.
(minitoc(hints))                It is incompatible
(minitoc(hints))                with the minitoc package.
```

You are trying to use also the titletoc package, but it is incompatible with the minitoc package. See note 18 on page 50.

```
Package minitoc(hints) Warning: <W0041>
(minitoc(hints))                You have attempted to insert
                                empty minilofs.
```

You have attempted to insert empty minilofs. If you have used the `nocheckfiles` package option, you will get some ugly empty mini-tables, with only a title and two horizontal rules. By default (`checkfiles` package option), you will only get this harmless message.

```
Package minitoc(hints) Warning: <W0042>
(minitoc(hints))                You have attempted to insert
                                empty minilots.
```

You have attempted to insert empty minilots. If you have used the `nocheckfiles` package option, you will get some ugly empty mini-tables, with only a title and two horizontal rules. By default (`checkfiles` package option), you will only get this harmless message.

```
Package minitoc(hints) Warning: <W0043>
(minitoc(hints))                You have attempted to insert
                                empty minitocs.
```

You have attempted to insert empty minitocs. If you have used the `nocheckfiles` package option, you will get some ugly empty mini-tables, with only a title and two horizontal rules. By default (`checkfiles` package option), you will only get this harmless message.

```
Package minitoc(hints) Warning: <W0044>
(minitoc(hints))                You have attempted to insert
                                empty partlofs.
```

You have attempted to insert empty partlofs. If you have used the `nocheckfiles` package option, you will get some ugly empty mini-tables, with only a title and two horizontal rules. By default (`checkfiles` package option), you will only get this harmless message.

```
Package minitoc(hints) Warning: <W0045>
(minitoc(hints))                You have attempted to insert
                                empty partlots.
```

You have attempted to insert empty partlots. If you have used the `nocheckfiles` package option, you will get some ugly empty mini-tables, with only a title and two horizontal rules. By default (`checkfiles` package option), you will only get this harmless message.

```
Package minitoc(hints) Warning: <W0046>
(minitoc(hints))                You have attempted to insert
                                empty parttocs.
```

You have attempted to insert empty parttocs. If you have used the `nocheckfiles` package option, you will get some ugly empty mini-tables, with only a title and two horizontal rules. By default (`checkfiles` package option), you will only get this harmless message.

```
Package minitoc(hints) Warning: <W0047>
(minitoc(hints))                You have attempted to insert
                                empty sectlofs.
```

You have attempted to insert empty sectlofs. If you have used the `nocheckfiles` package option, you will get some ugly empty mini-tables, with only a title and two horizontal rules. By default (`checkfiles` package option), you will only get this harmless message.

```
Package minitoc(hints) Warning: <W0048>
(minitoc(hints))                You have attempted to insert
                                empty sectlots.
```

You have attempted to insert empty sectlots. If you have used the `nocheckfiles` package option, you will get some ugly empty mini-tables, with only a title and two horizontal rules. By default (`checkfiles` package option), you will only get this harmless message.

```
Package minitoc(hints) Warning: <W0049>
(minitoc(hints))                You have attempted to insert
                                empty secttocs.
```

You have attempted to insert empty secttocs. If you have used the `nocheckfiles` package option, you will get some ugly empty mini-tables, with only a title and two horizontal rules. By default (`checkfiles` package option), you will only get this harmless message.

```
Package minitoc(hints) Warning: <W0050>
(minitoc(hints))                You have invoked an obsolete
                                command: \firstchapteris.
```

You have used an obsolete command (`\firstchapteris`). You should remove it.

```
Package minitoc(hints) Warning: <W0051>
(minitoc(hints))                You have invoked an obsolete
                                command: \firstpartis.
```

You have used an obsolete command (`\firstpartis`). You should remove it.

```
Package minitoc(hints) Warning: <W0052>
(minitoc(hints))                You have invoked an obsolete
                                command: \firstsectionis.
```

You have used an obsolete command (`\firstsectionis`). You should remove it.

```
Package minitoc(hints) Warning: <W0053>
(minitoc(hints))                You have used short extensions
(minitoc(hints))                and more than 99 chapters (NUMBER).
```

You have used short extensions (limited to 3 characters) and more than 99 chapters, so the number of the auxiliary file does not fit in the extension. *NUMBER* is the number of effective chapters in your document. See section 1.9 on page 51.

```
Package minitoc(hints) Warning: <W0054>
(minitoc(hints))                You have used short extensions
(minitoc(hints))                and more than 99 parts (NUMBER).
```

You have used short extensions (limited to 3 characters) and more than 99 parts, so the number of the auxiliary file does not fit in the extension. *NUMBER* is the number of effective parts in your document. See section 1.9 on page 51.

```
Package minitoc(hints) Warning: <W0055>
(minitoc(hints))                You have used short extensions
(minitoc(hints))                and more than 99 sections (NUMBER).
```

You have used short extensions (limited to 3 characters) and more than 99 sections, so the number of the auxiliary file does not fit in the extension. *NUMBER* is the number of effective sections in your document. See section 1.9 on page 51.

```
Package minitoc(hints) Warning: <W0056>
(minitoc(hints))                You are using \dosectlof and/or
(minitoc(hints))                \dosectlot, \sectlof and/or \sectlot,
(minitoc(hints))                hence the ``insection'' package
(minitoc(hints))                option is recommended.
```

You are asking for mini-lists of figures or tables at the section level. But 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 `insection` package option. See page 29.

```
Package minitoc(hints) Warning: <W0057>
(minitoc(hints))                You have used \minilof,
(minitoc(hints))                but not \dominilof.
```

You have attempted to insert some minilofs (via `\minilof`), but the minilofs have not been prepared (via `\dominilof`).

```
Package minitoc(hints) Warning: <W0058>
(minitoc(hints))                You have used \minilot,
(minitoc(hints))                but not \dominilot.
```

You have attempted to insert some minilots (via `\minilot`), but the minilots have not been prepared (via `\dominilot`).

```
Package minitoc(hints) Warning: <W0059>
(minitoc(hints))                You have used \minitoc,
(minitoc(hints))                but not \dominitoc.
```

You have attempted to insert some minitocs (via `\minitoc`), but the minitocs have not been prepared (via `\dominitoc`).

```
Package minitoc(hints) Warning: <W0060>
(minitoc(hints))                You have used \partlof,
(minitoc(hints))                but not \dopartlof.
```

You have attempted to insert some partlofs (via `\partlof`), but the partlofs have not been prepared (via `\dopartlof`).

```
Package minitoc(hints) Warning: <W0061>
(minitoc(hints))                You have used \partlot,
(minitoc(hints))                but not \dopartlot.
```

You have attempted to insert some partlots (via `\partlot`), but the partlots have not been prepared (via `\dopartlot`).

```
Package minitoc(hints) Warning: <W0062>
(minitoc(hints))                You have used \parttoc,
(minitoc(hints))                but not \doparttoc.
```

You have attempted to insert some parttocs (via `\parttoc`), but the parttocs have not been prepared (via `\doparttoc`).

```
Package minitoc(hints) Warning: <W0063>
(minitoc(hints))                You have used \sectlof,
(minitoc(hints))                but not \dosectlof.
```

You have attempted to insert some sectlofs (via `\sectlof`), but the sectlofs have not been prepared (via `\dosectlof`).

```
Package minitoc(hints) Warning: <W0064>
(minitoc(hints))                You have used \sectlot,
(minitoc(hints))                but not \dosectlot.
```

You have attempted to insert some sectlots (via `\sectlot`), but the sectlots have not been prepared (via `\dosectlot`).

```
Package minitoc(hints) Warning: <W0065>
(minitoc(hints))                You have used \secttoc,
(minitoc(hints))                but not \dosecttoc.
```

You have attempted to insert some secttocs (via `\secttoc`), but the secttocs have not been prepared (via `\dosecttoc`).

```
Package minitoc(hints) Warning: <W0066>
(minitoc(hints))                You have used \minilof,
(minitoc(hints))                but not \listoffigures nor
(minitoc(hints))                \fakelistoffigures.
```

You have tried to insert some minilofs (via `\minilof`), but the *document*`.lof` file is not available because you have not invoked `\listoffigures` nor `\fakelistoffigures`.

```
Package minitoc(hints) Warning: <W0067>
(minitoc(hints))                You have used \minilot but not
(minitoc(hints))                \listoftables nor
(minitoc(hints))                \fakelistoftables.
```

You have tried to insert some minilots (via `\minilot`), but the *document*`.lot` file is not available because you have not invoked `\listoftables` nor `\fakelistoftables`.

```
Package minitoc(hints) Warning: <W0068>
(minitoc(hints))                You have used \minitoc but not
(minitoc(hints))                \tableofcontents
(minitoc(hints))                nor \faketableofcontents.
```

You have tried to insert some minitocs (via `\minitoc`), but the *document*`.toc` file is not available because you have not invoked `\tableofcontents` nor `\faketableofcontents`.

```
Package minitoc(hints) Warning: <W0069>
(minitoc(hints))                You have used \partlof but not
(minitoc(hints))                \listoffigures
(minitoc(hints))                nor \fakelistoffigures.
```

You have tried to insert some partlofs (via `\partlof`), but the *document*`.lof` file is not available because you have not invoked `\listoffigures` nor `\fakelistoffigures`.

```
Package minitoc(hints) Warning: <W0070>
(minitoc(hints))                You have used \partlot but not
(minitoc(hints))                \listoftables
(minitoc(hints))                nor \fakelistoftables.
```

You have tried to insert some partlots (via `\partlot`), but the *document*`.lot` file is not available because you have not invoked `\listoftables` nor `\fakelistoftables`.

```
Package minitoc(hints) Warning: <W0071>
(minitoc(hints))                You have used \parttoc but not
(minitoc(hints))                \tableofcontents
(minitoc(hints))                nor \faketableofcontents.
```

You have tried to insert some parttocs (via `\parttoc`), but the *document*`.toc` file is not available because you have not invoked `\tableofcontents` nor `\faketableofcontents`.

```
Package minitoc(hints) Warning: <W0072>
(minitoc(hints))                You have used \sectlof but not
(minitoc(hints))                \listoffigures
(minitoc(hints))                nor \fakelistoffigures.
```

You have tried to insert some sectlofs (via `\sectlof`), but the *document*`.lof` file is not available because you have not invoked `\listoffigures` nor `\fakelistoffigures`.

```
Package minitoc(hints) Warning: <W0073>
(minitoc(hints))                You have used \sectlot but not
(minitoc(hints))                \listoftables
(minitoc(hints))                nor \fakelistoftables.
```

You have tried to insert some sectlots (via `\sectlot`), but the *document*`.lot` file is not available because you have not invoked `\listoftables` nor `\fakelistoftables`.

```
Package minitoc(hints) Warning: <W0074>
(minitoc(hints))                You have used \secttoc but not
(minitoc(hints))                \tableofcontents
(minitoc(hints))                nor \faketableofcontents.
```

You have tried to insert some secttocs (via `\secttoc`), but the *document*`.toc` file is not available because you have not invoked `\tableofcontents` nor `\faketableofcontents`.

```
Package minitoc(hints) Warning: <W0075>
(minitoc(hints))                You have used \doparttoc
(minitoc(hints))                but not \parttoc.
```

You have prepared some parttocs (via `\doparttoc`), but you never used one of them.

```
Package minitoc(hints) Warning: <W0076>
(minitoc(hints))                You have used \dopartlof
(minitoc(hints))                but not \partlof.
```

You have prepared some partlofs (via `\dopartlof`), but you never used one of them.

```
Package minitoc(hints) Warning: <W0077>
(minitoc(hints))                You have used \dopartlot
(minitoc(hints))                but not \partlot.
```

You have prepared some partlots (via `\dopartlot`), but you never used one of them.

```
Package minitoc(hints) Warning: <W0078>
(minitoc(hints))                You have used \dominitoc
(minitoc(hints))                but not \minitoc.
```

You have prepared some minitocs (via `\dominitoc`), but you never used one of them.

```
Package minitoc(hints) Warning: <W0079>
(minitoc(hints))                You have used \dominilof
(minitoc(hints))                but not \minilof.
```

You have prepared some minilofs (via \dominilof), but you never used one of them.

```
Package minitoc(hints) Warning: <W0080>
(minitoc(hints))                You have used \dominilot
(minitoc(hints))                but not \minilot.
```

You have prepared some minilots (via \dominilot), but you never used one of them.

```
Package minitoc(hints) Warning: <W0081>
(minitoc(hints))                You have used \dosecttoc
(minitoc(hints))                but not \secttoc.
```

You have prepared some secttocs (via \dosecttoc), but you never used one of them.

```
Package minitoc(hints) Warning: <W0082>
(minitoc(hints))                You have used \dosectlof
(minitoc(hints))                but not \sectlof.
```

You have prepared some sectlofs (via \dosectlof), but you never used one of them.

```
Package minitoc(hints) Warning: <W0083>
(minitoc(hints))                You have used \dosectlot
(minitoc(hints))                but not \sectlot.
```

You have prepared some sectlots (via \dosectlot), but you never used one of them.

```
Package minitoc(hints) Warning: <W0084>
(minitoc(hints))                --- The placeins package is loaded
(minitoc(hints))                with the above option,
(minitoc(hints))                but minitoc used the insection option
(minitoc(hints))                which is incompatible with it.
(minitoc(hints))                Try to remove the above option and
(minitoc(hints))                use consistent options.
```

You are using the placeins package, but with its `above` option, while minitoc is called with its `insection` option which is incompatible with it. See page 29, near a "dangerous bend" symbol like the one shown in the margin.

```
Package minitoc(hints) Warning: <W0085>
(minitoc(hints))                --- The placeins package is loaded
(minitoc(hints))                with the below option,
(minitoc(hints))                but minitoc used the insection option
(minitoc(hints))                which is incompatible with it.
(minitoc(hints))                Try to remove the below option
(minitoc(hints))                and use consistent options.
```

You are using the placeins package, but with its `below` option, while minitoc is called with its `insection` option which is incompatible with it. See page 29, near a "dangerous bend" symbol like the one shown in the margin.

```
Package minitoc(hints) Warning: <W0086>
(minitoc(hints))                The fncychap package should be
(minitoc(hints))                loaded BEFORE the minitoc package.
```

The fncychap package alters some commands and must be loaded *before* the minitoc package. See section 2.38 on page 72.

```
Package minitoc(hints) Warning: <W0087>
(minitoc(hints))                The quotchap package should be
(minitoc(hints))                loaded BEFORE the minitoc package.
```

The quotchap package alters some commands and must be loaded *before* the minitoc package. See section 2.39 on page 72.

```
Package minitoc(hints) Warning: <W0088>
(minitoc(hints))                The romannum package should be
(minitoc(hints))                loaded BEFORE the minitoc package.
```

The romannum package alters the numbering of some sectionning commands and must be loaded *before* the minitoc package. See section 2.40 on page 72.

```
Package minitoc(hints) Warning: <W0089>
(minitoc(hints))                The sfheaders package should be
(minitoc(hints))                loaded BEFORE the minitoc package.
```

The sfheaders package alters the sectionning commands and must be loaded *before* the minitoc package. See section 2.41 on page 72.

```
Package minitoc(hints) Warning: <W0090>
(minitoc(hints))                The alnumsec package should be
(minitoc(hints))                loaded BEFORE the minitoc package.
```

The alnumsec package alters the sectionning commands and must be loaded *before* the minitoc package. See section 2.42 on page 72.

```
Package minitoc(hints) Warning: <W0091>
(minitoc(hints))                The captcont package should be
(minitoc(hints))                loaded BEFORE the minitoc package.
```

The captcont package alters the caption commands and must be loaded *before* the minitoc package. See section 2.43 on page 73.

```
Package minitoc(hints) Warning: <W0092>
(minitoc(hints))                The hangcaption package should be
(minitoc(hints))                loaded BEFORE the minitoc package.
```

The hangcaption package alters some commands and must be loaded *before* the minitoc package. See section 2.47 on page 76.

5.2.3 Error messages

```
! Package minitoc Error: <E0001>
(minitoc)                But \part is undefined.
See the minitoc package documentation for explanation.
Type  H <return>  for immediate help.
? h
\mtcfixglossary not usable
```

There are no adequate sectionning command available to use the `\mtcfixglossary` macro; even `\part` is undefined. Verify your document class.

```
! Package minitoc Error: <E0002>
(minitoc)                But \part is undefined.
See the minitoc package documentation for explanation.
Type  H <return>  for immediate help.
? h
\mtcfixindex not usable
```

There are no adequate sectionning command available to use the `\mtcfixindex` macro; even `\part` is undefined. Verify your document class.

```
! Package minitoc Error: <E0003>
(minitoc)                Imbrication of mtchideinmainlof environments.
See the minitoc package documentation for explanation.
Type  H <return>  for immediate help.
? h
The hiding in main LoF could be incorrect
```

Some `mtchideinmainlof` environments are incorrectly imbricated (overlapping), so the hiding in the main list of figures will be strange.

```
! Package minitoc Error: <E0004>
(minitoc)               Imbrication of mtchideinmainlot environments.
See the minitoc package documentation for explanation.
Type  H <return>  for immediate help.
? h
The hiding in main LoT could be incorrect
```

Some `mtchideinmainlot` environments are incorrectly imbricated (overlapping), so the hiding in the main list of tables will be strange.

```
! Package minitoc Error: <E0005>
(minitoc)               Imbrication of mtchideinmaintoc environments.
See the minitoc package documentation for explanation.
Type  H <return>  for immediate help.
? h
The hiding in main ToC could be incorrect
```

Some `mtchideinmaintoc` environments are incorrectly imbricated (overlapping), so the hiding in the main table of contents will be strange.

```
! Package minitoc Error: <E0006>
(minitoc)               LANGUAGE is not a known language,
(minitoc)               LANGUAGE.mld not found.
(minitoc)               Command ignored.
See the minitoc package documentation for explanation.
Type  H <return>  for immediate help.
? h
See the minitoc documentation.
Correct the source using a valid language name.
Press RETURN
```

The `\mtcselectlanguage` macro has attempted to load the *LANGUAGE*`.mld` minitoc language definition file, but has not found it. First, verify the name of the language (likely to be misspelt), then check if your installation contains *all* the many distributed `.mld` files of the minitoc package, at the right place. If it is a local `.mld` file, it should be installed in the right place (in a local hierarchy) or be in the working directory.

```
! Package minitoc Error: <E0007>
(minitoc)               LANGUAGE is not a known minitoc
(minitoc)               language object file (.mlo),
(minitoc)               LANGUAGE.mlo not found.
(minitoc)               Command ignored.
See the minitoc package documentation for explanation.
Type  H <return>  for immediate help.
? h
See the minitoc documentation.
Correct the source using a valid language name.
Press RETURN
```

The `\mtcselectlanguage` macro has attempted to load indirectly the *LANGUAGE*`.mlo` minitoc language object file, but has not found it. First, verify the name of the language (likely to be misspelt), then check if your installation contains *all* the many distributed `.mlo` files of the minitoc package, at the right place. If it is a local `.mlo` file, it should be installed in the right place (in a local hierarchy) or be in the working directory.

```
! Package minitoc Error: <E0008>
(minitoc)               \mtcsetdepth attempts to use
(minitoc)               an undefined counter (ARG1depth).
See the minitoc package documentation for explanation.
Type  H <return>  for immediate help.
Correct the source code.
Type <return> and rerun LaTeX
```

You are trying to set the depth for an inexistent or undefined type of mini-table. Verify the type given and the document class, and the loaded packages.

```
! Package minitoc Error: <E0009>
(minitoc)               \mtcsetdepth has a wrong first argument
(minitoc)               (ARG1).
(minitoc)               It should be a mini-table type
(minitoc)               (parttoc...sectlot).
See the minitoc package documentation for explanation.
Type  H <return>  for immediate help.
? h
Correct the source code.
Type <return> and rerun LaTeX
```

The first argument of the \mtcsetdepth macro is incorrect. It should be a type of mini-table (parttoc, ..., sectlot).

```
! Package minitoc Error: <E0010>
(minitoc)               \mtcsetdepth: Illegal type of table (ARG1).
See the minitoc package documentation for explanation.
Type  H <return>  for immediate help.
? h
Correct the source code.
Type <return> and rerun LaTeX
```

The first argument of the \mtcsetdepth macro is incorrect. It should be a mini-table type (parttoc, ..., sectlot).

```
! Package minitoc Error: <E0011>
(minitoc)               \mtcsetfeature has a wrong first argument
(minitoc)               (ARG1).
(minitoc)               It should be a mini-table type
(minitoc)               (parttoc...sectlot).
See the minitoc package documentation for explanation.
Type  H <return>  for immediate help.
? h
Correct the source code.
Type <return> and rerun LaTeX
```

The first argument of the \mtcsetfeature macro is incorrect. It should be a mini-table type (parttoc, ..., sectlot).

```
! Package minitoc Error: <E0012>
(minitoc) \mtcsetfeature has a wrong second argument
(minitoc)               (ARG2).
(minitoc)               It should be a feature param
(minitoc)               (before, after, pagestyle).
See the minitoc package documentation for explanation.
Type  H <return>  for immediate help.
? h
Correct the source code.
Type <return> and rerun LaTeX
```

The second argument of the \mtcsetfeature macro is incorrect. It should be before, after, or thispagestyle.

```
! Package minitoc Error: <E0013>
(minitoc)               \mtcsetfont has a wrong first argument
(minitoc)               (arg1).
(minitoc)               It should be a mini-table type
(minitoc)               (parttoc...sectlot).
See the minitoc package documentation for explanation.
Type  H <return>  for immediate help.
? h
Correct the source code.
Type <return> and rerun LaTeX
```

The first argument of \mtcsetfont is incorrect; it should be the type of a mini-table (parttoc ..., sectlot).

```
! Package minitoc Error: <E0014>
(minitoc)               \mtcsetfont has a wrong second argument
(minitoc)               (ARG2).
(minitoc)               It should be a sectionning level
(minitoc)               (part...subparagraph) or * .
See the minitoc package documentation for explanation.
Type  H <return>  for immediate help.
? h
Correct the source code.
Type <return> and rerun LaTeX
```

The second argument of \mtcsetfont is incorrect; it should be a sectionning level (i.e., a sectionning command without its backslash), like part ..., subparagraph.

```
! Package minitoc Error: <E0015>
(minitoc)               \mtcsetformat has a wrong first argument
(minitoc)               (ARG1).
(minitoc)               It should be a mini-table type
(minitoc)               (parttoc...sectlot).
See the minitoc package documentation for explanation.
Type  H <return>  for immediate help.
? h
Correct the source code.
Type <return> and rerun LaTeX
```

The first argument of a \mtcsetformat macro is incorrect. It should be a mini-table type (parttoc, ..., sectlot).

```
! Package minitoc Error: <E0016>
(minitoc)               \mtcsetformat has a wrong second argument
(minitoc)               (ARG2).
(minitoc)               It should be a formatting param choosen from:
(minitoc)               pagenumwidth, tocrightmargin, dotinterval.
See the minitoc package documentation for explanation.
Type  H <return>  for immediate help.
? h
Correct the source code.
Type <return> and rerun LaTeX
```

The second argument of the `\mtcsetformat` macro is wrong. It should be one of the following keywords: `pagenumwidth`, `tocrightmargin`, or `dotinterval`.

```
! Package minitoc Error: <E0017>
(minitoc)               \mtcsetpagenumbers has a wrong first
(minitoc)               argument (ARG1)..
See the minitoc package documentation for explanation.
Type  H <return>  for immediate help.
? h
It should be a mini-table type
(minitoc)               (parttoc...sectlot)
Correct the source code.
Type <return> and rerun LaTeX
```

The first argument of the `\mtcsetpagenumbers` macro must be a type of minitable (`parttoc`, ..., `sectlot`).

```
! Package minitoc Error: <E0018>
(minitoc)               \mtcsetpagenumbers has a wrong second
(minitoc)               argument (ARG2).
See the minitoc package documentation for explanation.
Type  H <return>  for immediate help.
? h
It should be a boolean value (0/1, yes/no, on/off, ...)
Correct the source code.
Type <return> and rerun LaTeX
```

The second argument of the `\mtcsetpagenumbers` must be a keyword chosen in the following lists[3] :

- `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`.

```
! Package minitoc Error: <E0019>
(minitoc)               \mtcsetrules has a wrong first argument
(minitoc)               (ARG1).
See the minitoc package documentation for explanation.
Type  H <return>  for immediate help.
? h
It should be a mini-table type
(minitoc)               (parttoc...sectlot)
Correct the source code.
Type <return> and rerun LaTeX
```

The first argument of the `\mtcsetrules` is incorrect. It should be a mini-table type (`parttoc`, ..., `sectlot`).

```
! Package minitoc Error: <E0020>
(minitoc)               \mtcsetrules has a wrong second argument
(minitoc)               (ARG2).
See the minitoc package documentation for explanation.
Type  H <return>  for immediate help.
? h
It should be a boolean value (0/1, yes/no, on/off, ...)
Correct the source code.
Type <return> and rerun LaTeX
```

The second argument of the `\mtcsetrules` must be a keyword chosen in the following lists[3] :

- `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`.

[3] `O` and `o` are the letter `O`, `0` is the zero digit.

```
! Package minitoc Error: <E0021>
(minitoc)               \mtcsettitle has a wrong first argument
(minitoc)               (ARG1).
(minitoc)               It should be a mini-table type
(minitoc)               (parttoc...sectlot).
See the minitoc package documentation for explanation.
Type  H <return>  for immediate help.
? h
Correct the source code.
Type <return> and rerun LaTeX
```

The first argument of a `\mtcsettitle` macro is incorrect; it should be a mini-table type (`parttoc`, ..., `sectlot`).

```
! Package minitoc Error: <E0022>
(minitoc)               \mtcsettitlefont has a wrong first argument
(minitoc)               (ARG1).
(minitoc)               It should be a mini-table type
(minitoc)               (parttoc...sectlot).
See the minitoc package documentation for explanation.
Type  H <return>  for immediate help.
? h
Correct the source code.
Type <return> and rerun LaTeX
```

The first argument of the `\mtcsettitlefont` must be a mini-table type. You likely misspelt it.

```
! Package minitoc Error: <E0023>
(minitoc)               The macro \mtcsetfeature has incompatible
(minitoc)               first (ARG1) and second (ARG2) arguments.
See the minitoc package documentation for explanation.
Type  H <return>  for immediate help.
? h
Correct the source code.
Type <return> and rerun LaTeX
```

The first and second arguments of the `\mtcsetfeature` macro are incompatible. You should verify them.

```
! Package minitoc Error: <E0024>
(minitoc)               The macro \mtcsetfont has incompatible
(minitoc)               first (ARG1) and second (ARG2) arguments.
See the minitoc package documentation for explanation.
Type  H <return>  for immediate help.
? h
Correct the source code.
Type <return> and rerun LaTeX
```

The `\mtcsetfont` macro takes a mini-table type as first argument, a sectionning level as second argument (or a star), and a sequence of font commands as third argument. The second argument must have a lower level than the first one (i.e., it is meaningless to specify the font for the chapter level entries for a minitoc or a secttoc).

```
! Package minitoc Error: <E0025>
(minitoc)               The macro \mtcsetformat has incompatible
(minitoc)               first (ARG1) and second (ARG2) arguments.
See the minitoc package documentation for explanation.
Type  H <return>  for immediate help.
? h
Correct the source code.
Type <return> and rerun LaTeX
```

The first and second arguments of a `\mtcsetformat` macro are incompatible. One is likely to be misspelt.

```
! Package minitoc Error: <E0026>
(minitoc)               The optional argument of \mtcfixglossary
(minitoc)               is wrong.
See the minitoc package documentation for explanation.
Type  H <return>  for immediate help.
? h
It must be omitted (chapter), or be part, chapter or section
```

The optional argument of the `\mtcfixglossary` macro is incorrect: it should be omitted (then it defaults to `chapter`) or be `part`, `chapter`, or `section`.

```
! Package minitoc Error: <E0027>
(minitoc)               The optional argument of \mtcfixindex
(minitoc)               is wrong.
See the minitoc package documentation for explanation.
Type  H <return>  for immediate help.
? h
It must be omitted (chapter), or be part, chapter or section
```

The optional argument of the `\mtcfixindex` macro is incorrect: it should be omitted (then it defaults to `chapter`) or be `part`, `chapter`, or `section`.

```
! Package minitoc Error: <E0028>
(minitoc)               Unable to patch the memoir class.
See the minitoc package documentation for explanation.
Type  H <return>  for immediate help.
? h
So it remains incompatible. Sorry.
```

Your version of the memoir class is really incompatible with the minitoc package and cannot be automatically patched. Please update the memoir class and/or the minitoc package from the CTAN archives or a recent distribution.

```
! Package minitoc Error: <E0029>
(minitoc)               Unbalanced mtchideinmainlof environment.
See the minitoc package documentation for explanation.
Type  H <return>  for immediate help.
? h
The hiding in main LoF could be incorrect
```

A `mtchideinmainlof` environment is unbalanced, so the hiding in the main list of figures could be incorrect.

```
! Package minitoc Error: <E0030>
(minitoc)               Unbalanced mtchideinmainlot environment.
See the minitoc package documentation for explanation.
Type  H <return>  for immediate help.
? h
The hiding in main LoT could be incorrect
```

A mtchideinmainlot environment is unbalanced, so the hiding in the main list of tables could be incorrect.

```
! Package minitoc Error: <E0031>
(minitoc)                Unbalanced mtchideinmaintoc environment.
See the minitoc package documentation for explanation.
Type  H <return>  for immediate help.
? h
The hiding in main ToC could be incorrect
```

A mtchideinmaintoc environment is unbalanced, so the hiding in the main table of contents could be incorrect.

```
! Package minitoc Error: <E0032>
(minitoc)                You are using the \mtcloadmlo command
(minitoc)                outside of a .mld file.
See the minitoc package documentation for explanation.
Type  H <return>  for immediate help.
? h
It will be ignored
```

It is *forbidden* to use the \mtcloadmlo macro outside of a .mld file (which is loaded via \mtcselectlanguage). The command is ignored.

```
! Package minitoc Error: <E0033>
(minitoc)                The macro \mtcsettitle uses
(minitoc)                an illegal type of table (ARG1).
See the minitoc package documentation for explanation.
Type  H <return>  for immediate help.
? h
Correct the source code.
Type <return> and rerun LaTeX
```

The first argument of a minitoc macro is incorrect. It should be a type of mini-table, like parttoc, partlof, partlot, minitoc, minilof, minilot, secttoc, sectlof, or sectlot.

```
! Package minitoc Error: <E0034>
(minitoc)               The macro \mtcsettitlefont uses
(minitoc)               an illegal type of table (ARG1).
See the minitoc package documentation for explanation.
Type  H <return>  for immediate help.
? h
Correct the source code.
Type <return> and rerun LaTeX
```

The first argument of a minitoc macro is incorrect. It should be a type of mini-table, like `parttoc`, `partlof`, `partlot`, `minitoc`, `minilof`, `minilot`, `secttoc`, `sectlof`, or `sectlot`.

```
! Package minitoc Error: <E0035>
(minitoc)               You have used the 'insection' option in
(minitoc)               a document where chapters are defined.
(minitoc)               This is not compatible: option ignored.
Type  H <return>  for immediate help.
? h
Remove this option.
Type <return> and rerun LaTeX
```

The `insection` package option is intended for article-like document classes, to prevent floats from drifting out of their section. It is pointless for book-like or report-like document classes, where floats are contained in their chapter.

```
! Package minitoc Error: <E0036>
(minitoc)               Your minitoc installation is incomplete.
(minitoc)               The minitoc language object file (.mld),
(minitoc)               english.mld is not found.
(minitoc)               We will try to continue with default values.
Type  H <return>  for immediate help.
? h
See the minitoc documentation.
Please fix your minitoc installation.
Type <return> to continue
```

The `english.mld` language definition file can not be found. You should verify your installation of the minitoc package. As an interim solution, we provide the missing english titles.

```
! Package minitoc Error: <E0037>
(minitoc)               The \command command is incompatible
(minitoc)               with the document class.

See the minitoc package documentation for explanation.
Type  H <return>  for immediate help.
 ...

l.39 \dominitoc[r]

? h
Correct the source code.
Type <return> and rerun LaTeX
```

You have used a preparation or insertion command (*command*) which is not available for the document class you are using. Please verify that the document class is compatible with minitoc and if the level of the mini-table is available in the document class (section-level mini-tables are not available in book- or report-like classes, chapter-level mini-tables are not available in article-like classes, mini-tables are not available in letter-like classes, etc.).

```
! Package minitoc Error: <E0038>
(minitoc)               Your minitoc installation is incomplete.
(minitoc)               A mandatory minitoc language object file,
(minitoc)               LANGUAGE.mld, is not found.
(minitoc)               We will try to continue with
(minitoc)               current/default values.
Type  H <return>  for immediate help.
? h
See the minitoc documentation.
Please fix your minitoc installation.
Type <return> to continue
```

The mandatory *LANGUAGE*.mld language definition file can not be found. You should verify your installation of the minitoc package. As an interim solution, we provide the default english titles.

5.3 Messages from the mtcoff package

The mtcoff package gives only warning messages; their numbers begin with F.

5.3.1 Warning messages

```
Package mtcoff Warning: <F0001>
(mtcoff)                \addstarredchapter{...} should be replaced
(mtcoff)                by \addcontentsline{toc}{chapter}{...}
(mtcoff)                 on input line LINE.
```

The \addstarredchapter command is specific of the minitoc package and simulated by the mtcoff package. If necessary, it should be replaced by the equivalent \addcontentsline{toc}{chapter}{...} command.

```
Package mtcoff Warning: <F0002>
(mtcoff)                \addstarredpart{...} should be replaced
(mtcoff)                by \addcontentsline{toc}{part}{...}
(mtcoff)                 on input line LINE.
```

The \addstarredpart command is specific of the minitoc package and simulated by the mtcoff package. If necessary, it should be replaced by the equivalent \addcontentsline{toc}{part}{...} command.

```
Package mtcoff Warning: <F0003>
(mtcoff)                \addstarredsection{...} should be replaced
(mtcoff)                by \addcontentsline{toc}{section}{...}
(mtcoff)                 on input line LINE.
```

The \addstarredsection command is specific of the minitoc package and simulated by the mtcoff package. If necessary, it should be replaced by the equivalent \addcontentsline{toc}{section}{...} command.

```
Package mtcoff Warning: <F0004>
(mtcoff)                \mtcaddchapter{...} should be replaced
(mtcoff)                by \addcontentsline{toc}{chapter}{...}
(mtcoff)                 on input line LINE.
```

The `\mtcaddchapter` command is specific of the minitoc package and simulated by the mtcoff package. If necessary, it should be replaced by the equivalent `\addcontentsline{toc}{chapter}{...}` command.

```
Package mtcoff Warning: <F0005>
(mtcoff)                \mtcaddpart{...} should be replaced
(mtcoff)                by \addcontentsline{toc}{part}{...}
(mtcoff)                 on input line LINE.
```

The `\mtcaddpart` command is specific of the minitoc package and simulated by the mtcoff package. If necessary, it should be replaced by the equivalent `\addcontentsline{toc}{part}{...}` command.

```
Package mtcoff Warning: <F0006>
(mtcoff)                \mtcaddsection{...} should be replaced
(mtcoff)                by \addcontentsline{toc}{section}{...}
(mtcoff)                 on input line LINE.
```

The `\mtcaddsection` command is specific of the minitoc package and simulated by the mtcoff package. If necessary, it should be replaced by the equivalent `\addcontentsline{toc}{section}{...}` command.

```
Package mtcoff Warning: <F0007>
(mtcoff)                You should scan (backwards) your .log
(mtcoff)                file to find some commands needing
(mtcoff)                to be replaced if you decide to
(mtcoff)                DEFINITELY stop using minitoc for this
(mtcoff)                document. It is more wise to keep the
(mtcoff)                \usepackage lines for minitoc and mtcoff
(mtcoff)                and to comment out only one of them.
```

You have replaced the use of the minitoc package by its substitute mtcoff. It is recommended to keep the \usepackage lines for both minitoc and mtcoff and to comment out only one of them. If you decide to *definitely* stop using minitoc for this document, it is wise to scan (backwards) the *document*.log file (after a compilation using mtcoff) to locate some commands needing to be replaced.

```
Package mtcoff Warning: <F0008>
(mtcoff)                The macro \string\kernafterSTRING
(mtcoff)                should not be used out of context
(mtcoff)                on line LINE.
```

You are using one of the \kernafter*STRING* macros with the mtcoff package. The result may be impredictible. You can only redefine these macros to adjust the position of the bottom rule of a type of minitables. Any other usage is meaningless without the minitoc package.

5.4 Message from the mtcpatchmem package

```
Package mtcpatchmem Info: <M0001>
Package mtcpatchmem Info: mtcpatchmem package to patch the memoir class.
```

You are using a version of the memoir class which needs a correction. This correction has been automatically loaded if necessary. Very recent versions should not need it anymore. See chapter 12 on page 428.

Chapter 6

Jargon

Tables

This chapter attempts to explain some terms used in this documentation, and describes some useful files and suffixes. Many data come from the documentations of various cited packages, and from [3, 4, 97, 100, 146, 147, 157, 163, 164, 165, 176, 177, 180, 189, 195, 197, 198, 244].

`.aux` The suffix of the name for an *auxiliary* file of a LATEX document. It carries some information from a LATEX run to the next.

`.cls` The suffix for the name of a document class file, loaded via `\documentclass`.

`.dtx` The suffix of the name for a documented source file of a LATEX package or class. This file is often associated with a `.ins` file to generate the package or class. Compiling a `.dtx` file with LATEX generates the documentation.

`.F` The base suffix of the name for a minilof file when short extensions (suffixes) are used. The full suffix is `.F`*nn* where *nn* is the absolute number of the minilof.

`.G` The base suffix of the name for a partlof file when short extensions (suffixes) are used. The full suffix is `.G`*nn* where *nn* is the absolute number of the partlof.

.H The base suffix of the name for a sectlof file when short extensions (suffixes) are used. The full suffix is `.H`*nn* where *nn* is the absolute number of the sectlof.

.ins The suffix of the name for an installation file of a LaTeX package or class. When compiled with LaTeX, it extracts the files of the package or class from an `.dtx` file.

.lof The suffix of the name of the "list of figures" file.

.log The suffix of the name of the log file (compilation report).

.lot The suffix of the name of the "list of tables" file.

.M The base suffix of the name for a minitoc file when short extensions (suffixes) are used. The full suffix is `.M`*nn* where *nn* is the absolute number of the minitoc.

.maf The suffix of the name of the file generated by the `listfiles` package option. This file contains the list of the minitoc auxiliary files.

.mld The suffix for the name of a minitoc language definition file. A minitoc language definition file contains the definitions for the mini-table titles in a given language.

.mlf The base suffix of the name for a minilof file when long extensions (suffixes) are used. The full suffix is `.mlf`*nn* where *nn* is the absolute number of the minilof.

.mlo The suffix for the name of a minitoc language object file. For some exotic languages, the encoding makes not easy to put direcly the titles in a `.mld` file; hence the `.mld` file must load a `.mlo` file.

.mlt The base suffix of the name for a minilot file when long extensions (suffixes) are used. The full suffix is `.mlt`*nn* where *nn* is the absolute number of the minilot.

.mtc The base suffix of the name for a minitoc file when long extensions (suffixes) are used. The full suffix is `.mtc`*nn* where *nn* is the absolute number of the minitoc.

.P The base suffix of the name for a parttoc file when short extensions (suffixes) are used. The full suffix is `.P`*nn* where *nn* is the absolute number of the parttoc.

.plf The base suffix of the name for a partlof file when long extensions (suffixes) are used. The full suffix is `.plf`*nn* where *nn* is the absolute number of the partlof.

.plt The base suffix of the name for a partlot file when long extensions (suffixes) are used. The full suffix is `.plt`*nn* where *nn* is the absolute number of the partlot.

.ptc The base suffix of the name for a parttoc file when long extensions (suffixes) are used. The full suffix is `.ptc`*nn* where *nn* is the absolute number of the parttoc.

.S The base suffix of the name for a secttoc file when short extensions (suffixes) are used. The full suffix is `.S`*nn* where *nn* is the absolute number of the secttoc.

.slf The base suffix of the name for a sectlof file when long extensions (suffixes) are used. The full suffix is `.slf`*nn* where *nn* is the absolute number of the sectlof.

.slt The base suffix of the name for a sectlot file when long extensions (suffixes) are used. The full suffix is `.slt`*nn* where *nn* is the absolute number of the sectlot.

.stc The base suffix of the name for a secttoc file when long extensions (suffixes) are used. The full suffix is `.stc`*nn* where *nn* is the absolute number of the secttoc.

.sty The suffix for the name of a package file, loaded via `\usepackage`.

.T The base suffix of the name for a minilot file when short extensions (suffixes) are used. The full suffix is `.T`*nn* where *nn* is the absolute number of the minilot.

.tex The suffix of the name of a TEX or LATEX normal source file.

.toc The suffix of the name of the "table of contents" file.

.U The base suffix of the name for a partlot file when short extensions (suffixes) are used. The full suffix is `.U`*nn* where *nn* is the absolute number of the partlot.

.V The base suffix of the name for a sectlot file when short extensions (suffixes) are used. The full suffix is `.V`*nn* where *nn* is the absolute number of the sectlot.

absolute numbering The auxiliary files for the mini-tables have a suffix containing an *absolute number*, i.e., the number is unique and always increasing from the first part, chapter or section; this has solved some obscure problems, and also made obsolete some commands, like `\firstpartis`, `\firstchapteris`, and `\firstsectionis`. The absolute numbering has been introduced in version #23.

abstract The abstract package [251] (by Peter R. Wilson) needs some precautions if used with its `addtotoc` option.

adjustment Some minitoc commands and environments are known as "adjustment commands" because they are used in some circumstances to "adjust" a counter or to alter the displaying of contents files. These commands and environments are `\adjustptc`, `\adjustmtc`, `\adjuststc`, `\decrementptc`, `\decrementmtc`, `\decrementstc`, `\incrementptc`, `\incrementmtc`, `\incrementstc`, `\mtcaddpart`, `\mtcaddchapter`, `\mtcaddsection`, `\mtcfixglossary`, `\mtcfixindex`, `mtchideinmaintoc`, `mtchideinmainlof`, and `mtchideinmainlot`. It is *strongly recommended* that the user verifies the result of such adjustments in the final document.

after A type of *feature* (see this term) which is executed *after* a given type of mini-table. Look at the documentation of the `\mtcsetfeature` command, in section 1.10 on page 45.

alnumsec The alnumsec package [152] allows you to use alphanumeric section numbering, e.g. A. Introduction; III. International Law. It's output is similar to the alphanum package (part of the jura class [85]), but you can use the standard LATEX sectionning commands. Thus it is possible to switch numbering schemes easily. Greek letters, double letters (bb) and different delimiters around them are supported. It must be loaded *before* the minitoc package (see point 1.8 on page 50 and section 2.42 on page 72).

alphanum The alphanum package, which is part of the specialized jura class [85], by Felix Braun, is incompatible with the minitoc package.

$\mathcal{AMS}$ The American Mathematical Society. This society has developped some document classes: unfortunately, amsart and amsproc are incompatible with the minitoc package; amsbook is compatible but needs precautions.

amsart, `amsart.cls` A document class for articles, provided by the American Mathematical Society ($\mathcal{A}\mathcal{M}\mathcal{S}$). Unfortunately, this class is incompatible with the minitoc package.

amsbook, `amsbook.cls` A document class for books, provided by the American Mathematical Society ($\mathcal{A}\mathcal{M}\mathcal{S}$). This class is compatible with the minitoc package, but needs some precautions. See section 2.24 on page 63.

amsproc, `amsproc.cls` A document class for conference proceedings, provided by the American Mathematical Society ($\mathcal{A}\mathcal{M}\mathcal{S}$). Unfortunately, this class is incompatible with the minitoc package.

Antomega Antomega [150] (by Alexej M. Kryokov and Dmitry Ivanov) is a language support package for *Lambda* ($\mathbf{\Lambda}$), based on the original `omega.sty` file of the Omega project ($\mathbf{\Omega}$). However, it provides some additional useful functionalities. Some languages definition files (`.mld`) use titles taken from Antomega: `greek-mono.mld`, `greek-polydemo.mld`, `greek-polykatha.mld`, `latvian.mld`, `polish2.mld`, `russian2m.mld`, `russian2o.mld`, and `spanish3.mld`.

`appendices` See appendix below.

appendix The appendix package [252] (by Peter R. Wilson) provides various ways of formatting the titles of appendices. Also `(sub)appendices` environments are provided that can be used, for example, for per chapter/section appendices. If this package is used with minitoc, some precautions are needed (see section 2.20 on page 60).

Arabi Arabi [135] is a system (by Youssef Jabri) to prepare LaTeX documents in the `arabic` or `farsi` languages. The titles in `arabi.mld` and `farsi3.mld` come from the `arabic.ldf` and `farsi.ldf` files of this system.

ArabTeX ArabTeX [154, 155] is a package (by Klaus Lagally) to prepare LaTeX documents in the `arabic` or `hebrew` languages. The titles in `arab.mld` (or `arabic.mld`), `arab2.mld` and `hebrew.mld` come from ArabTeX, while those of `hebrew2.mld` come from babel [38, 39].

ArmTeX ArmTeX [101] is a package (prepared by Serguei D'Achian, Arnak Dalalyan and Vardan Akopian) to prepare LaTeX documents in the `armenian` language. The titles in `armenian.mld` come from ArmTeX.

article A standard LaTeX document class. It has sectionning commands: `\part` and `\section` (and below), but not `\chapter`. It is compatible with the minitoc package and you can make mini-tables at the part and section levels (but, of course, not at the chapter level).

`\AtBeginDocument` This standard macro allows to add code to be executed at the beginning of the document (if fact, at the very end of its preamble, but inside it, which implies some restrictions), at the point where `\begin{document}` is processed. This allows a package (or a class) to add code without creating any conflicts with other packages trying to do the same.

\AtEndDocument This standard macro allows to add code to be executed at the end of the document, at the point where `\end{document}` is processed. This allows a package (or a class) to add code without creating any conflicts with other packages trying to do the same.

autoconfiguration Since version #28, minitoc detects automaticaly if the extensions (suffixes) of the file names are limited to 3 characters (like under MS-DOS) or not. This process is named autoconfiguration. The package option `shortext` forces the limitation to 3 characters.

auxiliary During the preparation of a document, the LaTeX system uses some **auxiliary** files to store information. The standard auxiliary files are *document*`.aux` (for cross-reference labels, counters, etc.), *document*`.toc` for the table of contents, *document*`.lof` for the list of figures, and *document*`.lot` for the list of tables. The minitoc package creates its own auxiliary files, to store the contents of each mini-table. These files are the *minitoc auxiliary files*, whose names are *document*`.`*extension*, the table 1.11 on page 51 lists the possible extensions.

babel The babel package [38, 39] (by Johannes Braams and others) is a large package useful to write LaTeX documents in many languages, not only english. Many titles for mini-tables come from the babel package.

BangTeX A package for typesetting documents in the `bangla` language using the TeX/LaTeX systems; see [202].

before A type of *feature* (see this term) which is executed *before* a given type of mini-table. Look at the documentation of the `\mtcsetfeature` command, in section 1.10 on page 45.

BibTeX A program by Oren Patashnik to make bibliographies in LaTeX documents. Distributed with LaTeX. See [180, 206, 207].

bibtopic A LaTeX package [16] for including several bibliographies in a document. These bibliographies might be considered to cover different topics (hence the name) or bibliographic material (e.g., primary and secondary literature) and the like.

bibunits The bibunits package [121] allows separate bibliographies for different units or parts of the text. The units can be chapters, sections or `bibunit` environments. This package is compatible with a wide variety of packages, including, but not limited to, , and KOMA-Script classes [147, 195].

book A standard LaTeX document class. It has sectionning commands: `\part`, `\chapter`, and `\section` (and below). It is compatible with the minitoc package and you can make mini-tables at the part and chapter levels (but not at the section level, to avoid too many auxiliary files).

cappuccino See "minutes" below.

captcont The captcont package [95] provides support for retaining a figure or caption number across several float environments — usually over several pages. It allows control over the contents of the List-of-Figures and the List-of-Tables pages. It should be compatible with all other packages that modify or extend the float environment and with the subfig package [96] in particular.

caption The caption package [224] provides many ways to customize the captions in floating environments such `figure` and `table` and cooperates with many other packages.

caption2 The[1] caption2 package [223] used to be an experimental side-version of the regular caption package [224] and has been superseed by the new release of the regular caption package version 3.0 in December 2003. caption2 is still some kind of supported, that means it will be part of future releases and bugs will still be fixed, so existing documents using this package will still compile. But Axel Sommerfeldt will *not* answer questions about this package anymore except questions on migrating to the regular version of the caption package. And it will *not* be adapted or enhanced in the future.

So please don't use this package for new documents. It's old, it's obsolete and it starts to begin smell bad!

Please ignore all hints in books or other documents which try to tell you that the caption2 package should be used instead of the caption package – these hints are outdated since December 2003.

CATALOG A plain text file which contains basic infos about the minitoc package (abstract, date, author, version, licence).

catcode Short for "category code". In[2] the first place, it's wise to have a precise idea of what your keyboard sends to the machine. There are 256 characters that TEX might encounter at each step, in a file or in a line of text typed directly on your terminal. These 256 characters are classified into 16 categories numbered 0 to 15. See table 6.1 on the following page. It's not necessary for you to learn these code numbers; the point is only that TEX responds to 16 different types of characters. At first, "*The TEXbook*" led you to believe that there were just two types—the escape character and the others—and then you were told about two more types, the grouping symbols `{` and `}`. The category code for any character can be changed at any time, but it is usually wise to stick to a particular scheme.

ccaption The ccaption package [255] provides commands for "continuation captions", unnumbered captions, and a legend heading for any environment. Methods are provided to define captions for use outside float environments, and to define new float environments and subfloats. Tools are provided for defining your own captioning styles.

Chapter 0 Some documents do not begin with chapter number one, but with chapter number zero (or even a weirder number). This caused a serious problem in old versions of the minitoc package: the minitocs appeared in the wrong chapters, and a first correction was the introduction of specific commands (`\firstchapteris` and co.). With the addition of the absolute numbering of the mini-table auxiliary files (see *absolute* above), the problem was solved in minitoc version #23, and these commands became obsolete. See section 1.5.4 on page 46.

chapterbib The chapterbib package [12] allows multiple bibliographies in a LATEX document, including items `\cite`'d in more than one bibliography. Despite the name

[1] This text comes from the documentation of the caption package. The caption and caption2 packages have the same author, Axel Sommerfeldt.

[2] This definition is taken from "*The TEXbook*" [143].

Table 6.1: Category codes

	Category	Meaning
0	Escape character	(\ usually)
1	Beginning of group	({ usually)
2	End of group	(} usually)
3	Math shift	($ usually)
4	Alignment tab	(& usually)
5	End of line	(*return* usually)
6	Parameter	(# usually)
7	Superscript	(^ usually)
8	Subscript	(_ usually)
9	Ignored character	(*null* usually)
10	Space	(␣ usually)
11	Letter	(A, . . . , Z and a, . . . , z)
12	Other character	(none of the above or below)
13	Active character	(~ usually)
14	Comment character	(% usually)
15	Invalid character	(*delete* usually)

Table 6.2: Encoding schemes implemented in CJK

Encoding	1 byte	2 bytes	3 bytes
GB	0xA1–0xF7	0xA1–0xFE	—
Big 5	0xA1–0xF9	0x40–0xFE	—
JIS	0xA1–0xF4	0xA1–0xFE	—
SJIS	0xA1–0xFE	0x40–0xFC	—
KS	0xA1–0xFD	0xA1–0xFE	—
UTF 8	0xC0–0xEF	0x80–0xBF	0x80–0xBF
CNS	0xA1–0xFE	0xA1–0xFE	—

"chapterbib", *the bibliographies are for each included file*, not necessarily for each chapter.

checkfiles A package option of minitoc. It checks every mini-table to look if it is empty; then empty mini-tables are *not* printed. This is the default. The opposite package option (nocheckfiles) prints even the empty mini-tables, which look ugly. See section 9.74.2 on page 374.

CJK The CJK system [167, 168] (by Werner Lemberg and others), is a set of packages and fonts to prepare LaTeX documents in some oriental language, like chinese, japanese, korean (with Hangûl or Hanja fonts), and thai, plus some variants of russian. The titles of mini-tables for these languages come from some CJK files and were inserted in .mld files when possible, or in .mlo files when the encoding is incompatible with the .ins/.dtx mechanism; then the .mld file must input the corresponding .mlo file. CJK implements the GB, Big 5, JIS, SJIS, KS, UTF 8, and CNS encodings (on 16 bits, except UTF 8 on 24 bits). See table 6.2.

Some encoding schemes (Big 5, SJIS) have gaps in the range of the second byte. It is difficult to input Big 5 and SJIS encoding directly into TeX since some of the

Table 6.3: Standard document classes

Class	Usage
article	For articles in scientific journals, presentations, short reports, program documentation, invitations, ...
proc	For preparing conference proceedings; analog to the article class.
ltxdoc	For preparing the documentation of a package or of a class; analog to the article class.
ltxnews	For preparing the announcement of a LaTeX release; analog to the article class.
report	For longer reports containing several chapters, small books, PhD theses, ...
book	For real books.
letter	For letters; as this class has no sectionning commands, do not use minitoc with this class.
slides	For slides; the class uses big sans serif letters. You might want to consider using BeamerTeX[a] instead. Do not use minitoc with these classes.

[a] http://www.dante.de/tex-archive/macros/latex/contrib/beamer/doc/beameruserguide.pdf

values used for the encodings' second bytes are reserved for control characters: '{', '}', and '\'. Redefining them breaks a lot of things in LaTeX; to avoid this, preprocessors are normally used which convert the second byte into a number followed by a delimiter character. For further details, please refer to [176, 177]; Ken Lunde discusses in great detail all CJK encodings which are or have been in use. Please note that the minitoc package uses the `.mlo` files as a workaround for this problem; see section 1.4.12 on page 41.

class The **class**[3] is the first information LaTeX needs to know when processing an input file; it is the type of document the author wants to create. This is specified with the `\documentclass` command.

`\documentclass[`*options*`]{`*class*`}`

Here *class* specifies the type of document to be created. Table 6.3 lists the standard document classes. The LaTeX 2_ε distribution provides additional classes for other documents, including letters and slides, but the minitoc package has not been tested with all these classes. The *options* parameters customize the behaviour of the document class. The options have to be separated by commas. The standard classes supported by the minitoc package are listed in section 2.7 on page 56.

CMR For "Computer Modern Roman". The roman subset of the Computer Modern fonts. See "Computer Modern" below.

`comp.text.tex` The Usenet news group about TeX and LaTeX, in english.

Computer Modern A set of fonts [142] designed by Donald Knuth for TeX. Initially they were built with METAFONT [102, 144] (a program also created by Knuth), but PostScript type 1 (vector) versions exist today, with extensions (for accented characters, mainly): the EC-fonts (European Computer Modern), the cm-super fonts, etc.

[3] This note is extracted from [198], then adapted.

counter A TEX register containing an integer value. There are 256 counters (from 0 to 255) in TEX, but LATEX uses some of them, and many packages need some counters for their own usage. An extended version derived from TEX, ε-TEX [87], allows more counters. Omega (**Ω**) also offers more counters.

CTAN The *Comprehensive TEX Archive Network*, a set of computer archives containing most of the TEX related resources (like fonts, sofware, documentations, packages). They are accessible via Internet. The participating hosts in the Comprehensive TeX Archive Network are:

- `ftp.dante.de` (Koeln, Germany)
  - anonymous ftp `/tex-archive` (`/pub/tex` `/pub/archive`)
  - rsync access on rsync://rsync.dante.de/CTAN/
  - World Wide Web access on http://www.dante.de/
  - Administrator: mailto:ftpmaint@dante.de
- `ftp.tex.ac.uk` (Cambridge, England, UK)
  - anonymous ftp `/tex-archive` (`/pub/tex` `/pub/archive`)
  - rsync access on rsync://rsync.tex.ac.uk/CTAN/
  - NFS mountable from nfs.tex.ac.uk:/tex-archive
  - World Wide Web access on http://www.tex.ac.uk/
  - Administrator: mailto:ctan-uk@tex.ac.uk
- `tug.ctan.org` (Colchester, Vermont, USA)
  - anonymous ftp `/tex-archive` (`/pub/archive`)
  - World Wide Web access on http://www.ctan.org/
  - Administrator: mailto:tex@ctan.tug.org

There are also many sites mirroring these main archives.

`de.comp.text.tex` The Usenet news group about TEX and LATEX, in german.

depth In the standard documents classes (and in most classes) with sectionning commands, we have a notion of **depth**. The depth of a sectionning command determines the numbering level in its title (from the value of the `secnumdepth` counter), and the entries for a given sectionning command appear in the main table of contents if the depth of this sectionning command is lower than or equal to the value of the `tocdepth` counter; see table 6.4 on the next page for the depths of the sectionning commands in the main document classes.

The mechanism is analog for the parttocs, minitocs, and secttocs, using the values of the `parttocdepth`, `minitocdepth`, and `secttocdepth` counters. If you use sub-figures or sub-tables, the corresponding mini-tables use counters like `partlofdepth`, `partlotdepth`, `minilofdepth`, `minilotdepth`, `sectlofdepth`, and `sectlotdepth`.

descriptor (file descriptor). A software entity describing the interface between a program and a file. For most programs and operating systems, the number of file descriptors is limited. For TEX (and LATEX), there are 16 file descriptors for writing and 16 file descriptors for reading.

Table 6.4: Depths for sectionning commands

Class:	book	report	article
secnumdepth	2	2	3
\part	−1	−1	0
\chapter	0	0	×
\section	1	1	1
\subsection	2	2	2
\subsubsection	3	3	3
\paragraph	4	4	4
\subparagraph	5	5	5

devanagari.sty The *Devanāgarī for TEX* (Devanāgarī) package [204] provides a way to typeset high-quality Devanāgarī text with TEX. Devanāgarī is a script used for writing and printing Sanskrit and a number of languages in Northern and Central India such as Hindi and Marathi, as well as Nepali. The Devanāgarī package was originally developed in May 1991 by Frans Velthuis for the University of Groningen, The Netherlands, and it was the first system to provide support for the Devanāgarī script for TEX.

duplex2v.pro A PostScript header file, which tries to activate recto-verso printing. It works with some HP and Canon PostScript printers.

em A length unit equal (approximatively) to the width of a "m" letter in the current font.

emk An example of shell script to prepare the english documentation of the minitoc package. The script imk must have be run previously. See item 9 on page 224.

en-bst.bst A bibliographic style derived from the plain.bst standard style, but modified with the urlbst tool [118] to add an URL field. Family names of authors and editors are in small caps, years are in old style digits.

encoding This specifies the order that characters appear in the font (e.g., whether the 65th character is "A"). The most common value for TeX font encoding is OT1. The other predefined option is T1 (extended TEX). There's also US ASCII (7 bit), ISO Latin-1 (8 bit), Adobe Standard Encoding, UTF8 (Unicode [93, 104, 238]), etc. See table 6.5 on the next page and [166].

environment An environment is a delimited domain in a document, where special rules apply. Such a domain is delimited by \begin{*env*} ... \end{*env*} and may take arguments, like this:

```
\begin{minipage}[t]{.5\textwidth}
\end{minipage}
```

ethiop A LATEX package [29] giving the ethiopian language support for the babel package [38, 39].

extension The name of a file is often made of 2 parts: a *base name* and an *entension*, separated by a dot. On some old operating systems, the base name is limited to

Table 6.5: Various encodings

Encoding	Comment
`ansinew`	Windows 3.1 ANSI encoding, extension of Latin-1.
`applemac`	Macintosh encoding.
`ascii`	ASCII encoding for the range 32–127.
`cp1250`	Windows 1250 (Central and Eastern Europe) code page.
`cp1251`	Windows 1251 (Cyrillic) code page.
`cp1252`	Synonym for `ansinew`.
`cp437`	IBM 437 code page, which is the original American code page and contains letters, digits, mathematical symbols, and some characters useful in the construction of pseudographics.
`cp437de`	IBM 437 code page (German version).
`cp850`	IBM 850 code page, almost the same as ISO Latin 1, but character arrangement is not the same.
`cp852`	IBM 852 code page.
`cp855`	IBM 855 code page (Cyrillic).
`cp865]`	IBM 865 code page.
`cp866]`	IBM 866 code page (MS-DOS Cyrillic).
`decmulti`	DEC Multinational Character Set encoding.
`latin1`	ASCII encoding plus the characters needed for most Western European languages, including Danish, Dutch, English, Faroese, Finnish, Flemish, French, German, Icelandic, Italian, Norwegian, Portuguese, Spanish, and Swedish. Some non-European languages, such as Hawaiian and Indonesian, are also written in this character set.
`latin2`	ASCII encoding plus the characters needed for most Central European languages, including Croatian, Czech, Hungarian, Polish, Romanian, Slovak, and Slovenian.
`latin3`	ASCII encoding plus the characters needed for Esperanto, Maltese, Turkish, and Galicean. However, `latin5` is the prefered character set for Turkish.
`latin4`	ASCII encoding plus the characters needed for the Baltic languages (Latvian, Estonian, and Lithuanian), Greenlandic, and Lappish (Sámi).
`latin5`	Is essentially the same as `latin1`, except that some Turkish characters replace less commonly used Icelandic letters.
`next`	Next encoding.

8 characters and the extension to 3 characters (the "8+3" scheme). See also sections 1.9 on page 51 and 2.5 on page 56. It is strongly recommended to not have more than one dot in a file name.

`farsi.sty` See FarsiTEX below.

FarsiTEX A package [109] to typeset a document in the `farsi` (iranian, persian) language. See http://www.farsitex.org. But this package is today available only for LaTeX2.09. See also sections 13.52 on page 459 and 13.53 on page 459.

features A feature (for the minitoc package) is a set of actions executed at each occurrence of a mini-table of a given type. Three features are associated to each mini-table type: a "before" feature (executed before the mini-table), an "after" feature (executed after the mini-table), and a "thispagestyle" feature, which is exe-

cuted with the mini-table to set its page style. Look at the documentation of the `\mtcsetfeature` command, in section 1.10 on page 45.

filecontents A special LaTeX environment. It allows to create a file (whose name is passed as an argument of the environment) by writing the contents of the environment into that file:

```
\begin{filecontents}{file}
...contents  ...
\end{filecontents}
```

This environment should be used *before* `\documentclass`. It is used in `minitoc.ins` to prepare the `.mlo` files (see section 1.4.12 on page 41) and some files used in the compilation of the documentation.

\firstchapteris An obsolete command, temporarily used as a workaround for the Chapter 0 problem; see **Chapter 0** and **absolute numbering** above, and section 1.5.4 on page 46.

\firstpartis Analog to `\firstchapteris` above.

\firstsectionis Analog to `\firstchapteris` above.

\FloatBarrier A macro from the placeins package [10]. It sets up a "barrier" against the drift of floats (like figures or tables).

fminitoc.bib A bibliographic data base for the french documentation of the minitoc package.

fminitoc.dtx The source file for the french documentation of the minitoc package. In fact, it just sets `\jobname` then loads `minitoc.dtx`, which itself loads `\jobname.lan` to select the language used in `minitoc.dtx`; `minitoc.dtx` contains both english and french documentation fragments, selected by `\ifcase` constructs with the `\LANG` variable, set to 0 by `minitoc.lan` or to 1 by `fminitoc.lan` (i.e., by `\jobname.lan`). `fminitoc.dtx` is generated when compiling `minitoc.ins`.

fminitoc.ist This file contains a style for formating the index in the french documentation. It is generated when compiling `minitoc.ins`.

fminitoc.lan A file used to force the french language in the documentation. It is generated when compiling `minitoc.ins`.

fminitoc.pdf The french documentation in PDF format.

fminitoc.ps The french documentation in PostScript format. No more distributed (but look at the `fmk` and `pmk` scripts).

fmk An example of shell script to prepare the french documentation of the minitoc package. The script `imk` must have be run previously. See item 9 on page 224.

fncychap The fncychap package [170] provides a set of commands for changing the format used for some headings (chapters) in the standard LaTeX 2_ε document classes: book and report. It must be loaded *before* the minitoc package (see point 1.8 on page 50 and section 2.38 on page 72).

fr.comp.text.tex The Usenet newsgroup about TEX and LATEX, in french.

franc.sty A small package file used to prepare the french documentation. It is generated when compiling `minitoc.ins`.

frbib.sty A small package file used to prepare the bibliography of the french documentation. It is generated when compiling `minitoc.ins`.

fr-mtc.bst A bibliographic style file used to prepare the bibliography of the french documentation. It has been updated from the standard `plain.bst` for french by Ronan KERYELL, then I added some adaptations for french (like last names in small caps for authors and editors, years in old style digits), then modified with the `urlbst` [118] tool to add an `URL` field.

frnew.sty A small package file used to prepare the french documentation. It is generated when compiling `minitoc.ins`.

guarani A LATEX package to compose text in Guarani, the main language spoken in Paraguay. The file `guarani.ldf`, included in this package, defines the titles. See [32] and section 13.72 on page 470.

hangcaption The hangcaption package [138] defines a variant of the `\caption` command to produce captions with hanging indentation. This package is likely obsolete (1992, LATEX2.09).

Hindi For the Hindi language, see the Devanāgarī package [204] above. The minitoc package accepts the `devanagari` and `hindi` language options, which are synonyms. A `hindi-modern` language option is also available.

hint An indication, a clue to detect a problem. It is also a message written (into the *document*`.log` file) by the `hints` option (see below).

hints An option of the minitoc package. It verifies the loading order of some packages, the invocation order of some minitoc commands, the consistency between main minitoc commands, etc., and gaves warnings and other useful hints (mainly in the *document*`.log` file). This is a default option (use the `nohints` option to skip these checks).

HLATEX A system to write documents in the Korean language, using *Lambda* (**Λ**) (see below). Written by KOAUNGHI Un [146, in korean]. It uses special Hangûl or Hanja fonts and the UTF-8 input encoding.

hyperlink In a document, a reference to another object which is dynamicaly found (via a click with the mouse). This requires a special type of document (PDF, PostScript with hypertext features) and a suitable viewer (PDF viewer, recent PS viewer). This is useful to navigate in a document or in many documents, which can be remote documents.

hyperref The hyperref package [214] is used to emend cross-referencing commands in LATEX to produce some sort of `\special` commands; there are backends for the `\special` set defined for HyperTEX `dvi` processors, for embedded pdfmark commands for processing by Acrobat Distiller (**dvips** and **dvipsone**), for **dviwindo**, for **pdfTEX**, for **TEX4ht**, and for **VTEX**'s pdf and HTML backends.

This package derives from, and builds on, the work of the HyperTEX project, described at http://xxx.lanl.gov/hypertex/. It extends the functionality of all the LATEX cross-referencing commands (including the table of contents, bibliographies, etc.) to produce `\special` commands which a driver can turn into hypertext links; it also provides new commands to allow the user to write *ad hoc* hypertext links (hyperlinks), including those to external documents and URLs.

ifmtarg The ifmtarg package [260] provides an if-then-else command for testing if a macro argument is empty ("empty" meaning zero or more spaces only).

`imk` An example of shell script, which prepares the minitoc package from `minitoc.ins` and `minitoc.dtx`; note that `imk` must be run before running `emk` or `fmk`. See item 9 on page 224.

`insection` The `insection` package option loads the placeins package [10] with adequate options to avoid the floats (like figures and tables) to drift outside of their sections. This package option is recommended if you use sectlofs or sectlots in your document. See section 1.2 on page 28.

insertion The insertion commands of the minitoc package insert a mini-table in the document. A corresponding *preparation* command must have been invoked (only once) before. The insertion commands are:

`\parttoc`, `\partlof`, `\partlot`,
`\minitoc`, `\minilof`, `\minilot`,
`\secttoc`, `\sectlof`, `\sectlot`,
`\mtcprepare`

`INSTALL` A text file describing the installation of the minitoc package. See chapter 7 on page 222.

`\jobname` A TEX primitive containing the name of the document in preparation, i.e., the name of the file read first by TEX (or LATEX), without its extension. Very useful to build the names of other files.

jura The jura class [85], by Felix Braun, is incompatible with the minitoc package. It is used for german judicial documents.

`k-loose` A minitoc package option useful if your document is written with one of the KOMA-Script classes [147, 195]. This option tries to set a loose line spacing in the mini-tables. Analog to the `loose` package option for standard classes.

`k-tight` A minitoc package option useful if your document is written with one of the KOMA-Script classes [147, 195]. This option tries to set a tight line spacing in the mini-tables. Analog to the `tight` package option for standard classes.

KOMA-Script KOMA-Script [147, 195] is a very complex bundle. You may see this, because it is not only one class or one package but a bundle of many classes and packages. The classes (scrartcl, scrbook, scrlettr, scrlttr2, and scrreprt) are counterparts to the standard classes but never they come with only the same commands, environments, options and optional possibilities like the standard classes nor they result in the same look-a-like.

The scrbook, scrreprt, and scrartcl classes are compatible with the minitoc package, with some precautions (see section 1.5.5 on page 47). The scrlettr and scrlttr2 have no sectionning commands, so the minitoc package is pointless with them.

KOMA-Script comes with a lot of classes, packages, commands, environments and possibilities. Some of these you may find also at the standard classes, many of them you wouldn't. Some are even supplements to the LaTeX kernel.

The main classes of the KOMA-Script bundle are designed as counterparts to the standard LaTeX classes. This means that the KOMA-Script bundle contains replacements for the three standard classes book, report, and article. There is also a replacement for the standard class letter.

Lambda The LaTeX format (in the TeX meaning of that word) adapted to the special features of Omega (**Ω**) is called "*Lambda*" (**Λ**).

Lamed The LaTeX format (in the TeX meaning of that word) adapted to the special features of *Aleph* (**א**) is called "*Lamed*" (**ל**).

LaTeX LaTeX [156] is a typesetting system that is very suitable for producing scientific and mathematical documents of high typographical quality. It is also suitable for producing all sorts of other documents, from simple letters to complete books. LaTeX uses TeX [143, 145] as its formatting engine (from [198]).

In fact, LaTeX is a macro package that enables authors to typeset and print their work at the highest typographical quality, using a predefined, professional layout. LaTeX was originally written by Leslie Lamport [156]. It uses the TeX formatter as its typesetting engine. These days LaTeX is maintained by Frank Mittelbach and his team.

In 1994 the LaTeX package was (deeply) updated by the LaTeX3 team, led by Frank Mittelbach, to include some long-requested improvements, and to reunify all the patched versions which had cropped up since the release of LaTeX2.09 some years earlier. To distinguish the new version from the old, it is called LaTeX 2_ε.

LaTeX is pronounced "Lay-tech" or "Lah-tech." If you refer to LaTeX in an `ASCII` environment, you type `LaTeX`. LaTeX 2_ε is pronounced "Lay-tech two e" and typed `LaTeX2e`.

LaTeX2.09 An obsolete version of the LaTeX program, before 1994; it is no more supported. Do not use it[4]. Use the current version of LaTeX 2_ε, which is supported and much more efficient.

LaTeX 2_ε The current version of the LaTeX program, after 1994; it is supported.

LaTeX3 The future version of LaTeX, whose development is still in progress.

leaders A repetitive sequence of dots (or of one another small character), regulary spaced, used to link two objects on the same line (leading from a title to a page number in a table of contents or the like).

letter A standard document class to prepare letters for postal mail (mail on paper). As such documents have no sectionning commands nor structure, the minitoc package is pointless (hence incompatible) with them.

[4] Except in the case of a very old document; if possible, try to convert it.

lipsum The lipsum package [123] allows to easily insert sentences in a test file with a minimum of typing. The sentences are in latin but are modified and made nearly senseless. I have used this package in some of the examples of documents. See also http://lipsum.com for the origin of this text (pieces of *De Finibus Bonorum et Malorum* by Marcus Tullius Cicero).

`listfiles` An option of the minitoc package. It creates a list of the minitoc auxiliary files (these files contains the mini-tables and may be removed after the LaTeX run) in the *document*`.maf` file. Default. See section 1.7 on page 49.

LOF An acronym for "list of figures".

`lofdepth` This counter, if it exists, contains the depth of the list of figures.

`loose` An option of the minitoc package. It gives a loose line spacing in the mini-tables. It is the default. The opposite option is `tight`.

LOT An acronym for "list of tables".

`lotdepth` This counter, if it exists, contains the depth of the list of tables.

LPPL The *LaTeX Project Public License*, available at

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

Its current version is 1.3 (2003-12-01). The minitoc package is distributed under this license.

ltxdoc A standard LaTeX document class, for preparing the documentation of a package or of a class. For the minitoc package, it is very similar to the article document class; see above.

ltxnews A standard LaTeX document class, for preparing the announcement of a LaTeX release. For the minitoc package, it is very similar to the article document class; see above.

`\makeatletter` and **`\makeatother`** Many[5] internal commands of LaTeX, of packages and classes contain the `@` character in their names. This effectively prevents such names from being used in documents for user-defined commands. However, it also means that they cannot appear in a document, even in the preamble, without taking special precautions. As it is sometimes necessary to have such bits of "internal code" in the preamble, the commands `\makeatletter` and `\makeatother` make it easy to do: the difficult bit is to remember to add them, failure to do so can result in some strange errors. And these two commands should never be used in a package or class file.

makefile A special text file containing instructions describing the creation and the installation of a piece of software, using the "**make**" utility; **make** is a nice tool coming from the Unix operating system, but variants exists.

mcaption The mcaption package [131] provides a `margincap` environment for putting captions in the outer document margin with either a top or bottom alignment.

[5] Informations from [189, page 843].

memoir, `memoir.cls` A very general and powerful document class (by Peter R. Wilson, described in [257, 258]); this class is compatible with the minitoc package (with some precautions) if you use a recent version. See section 2.22 on page 62.

mini-bibliography See **minibbl** below.

mini-list Synonym for "mini-table" below.

mini-lof See "minilof" below.

mini-lot See "minilot" below.

mini-table This term refers to a local table of contents (like a table of contents, a list of figures or a list of tables) for a sectionning unit (part, chapter or section), by opposition to a global table (the table of contents, the list of figures or the list of tables for the whole document). The main aim of the minitoc package is the creation of such mini-tables. But the term "minitoc" is also used to refer to such mini-table, as a generic term, because the first versions of the package allowed only table of contents for chapters.

mini-toc See "minitoc" below.

minibbl Short for "mini-bibliography", i.e., to have a bibliography per part, chapter or section, or even by theme or subject. This is out of the domain of the minitoc package. See section 2.9 on page 57.

minilof A list of figures for a chapter.

`minilofdepth` This counter, if defined, contains the depth of the minilofs.

minilot A list of tables for a chapter.

`minilotdepth` This counter, if defined, contains the depth of the minilots.

minitoc A table of contents for a chapter. Also used as a generic term for any mini-table (see "mini-table" above).

`minitoc.bib` A bibliographic data base for the english documentation of the minitoc package.

`minitoc.bug` A plain text file containing a list of problems and questions about the minitoc package. See chapter 2 on page 53.

`minitocdepth` This counter contains the depth of the minitocs.

`minitoc.dtx` The file containing the documentation and the commented code of the minitoc package.

`minitoc-hyper.sty` A special version of the minitoc package which has been prepared by Bernd Jaehne and Didier Verna to work with the powerful hyperref package [214]; Heiko Oberdiek has integrated their work so since version #31, minitoc is compatible with hyperref. *Hence the minitoc-hyper package is now obsolete and should no more be used. It it still present on the CTAN archives for compatibility with old documents.*

minitoc.ins The installation file for the minitoc package. Compiling it with LaTeX produces most of the files of the minitoc package.

minitoc.ist This file contains a style for formating the index in the english documentation. It is generated when compiling `minitoc.ins`.

minitoc.l A text file containing the list of all the files being included in the minitoc package. Files not listed in `minitoc.l` are files used only to install the package or to produce its documentation.

minitoc.lan A file used to force the english language in the documentation. It is generated when compiling `minitoc.ins`.

minitoc.pdf The english documentation in PDF format.

minitoc.pre This file contains a LaTeX preamble for the documentation. It is generated when compiling `minitoc.ins`.

minitoc.ps The english documentation in PostScript format. No more distributed (but look at the `emk` and `pmk` scripts).

minitoc.sty This file contains the main part of the minitoc package, with comments removed. It is generated when compiling `minitoc.ins`.

minitoc.sum A plain text file containing a commented list of the minitoc commands and environments. See chapter 3 on page 77.

minitoc-texmf.zip A ZIP archive of a TDS-compliant hierarchy containing all files in the minitoc package.

minutes The minutes package [169] (by Knut Lickert) is used to prepare conference proceedings. The minitoc package allows to add "coffee breaks" in the table of contents via commands like `\addcoffeeline` and `\coffeeline` (and internal commands) whose names contain the string "`coffee`", hence the footnote about "cappuccino" ☕ in the installation chapter!

MonTeX MonTeX [97, 100] is a large package to prepare documents in various dialects of the Mongol language (Bicig and Bicig2, Mongol, Bithe and Manju, Buryat, Xalx and Khalkha) and in a dialect of Russian used in Mongolia (Russianc). Bicig is another name for Uighur. You can find many things about Mongolia and Mongolian at the web site [99]. The following description is extracted from [100].

MonTeX is a package which offers support for writing documents in Mongolian, Manju, Buryat and Russian. Mongolian can be represented in traditional Uighur script (also known as Classical or Traditional Script) and Cyrillic. Manju resembles the Traditional Mongolian script (from which it is derived) but uses a rich choice of diacritics in order to eliminate numerous ambiguities of the Mongolian script ancestor. Modern Buryat, like Mongolian in its present form, is written with a Cyrillic alphabet, but both Mongolian (35 letters) and Buryat (36 letters) use more letters than Russian (33 letters).

Mongolian The word *Mongolian* is actually an umbrella term for several languages rather than the precise name of a single language. Things become more complicated when names of ethnic groups, languages and writing systems are mixed.

Xalx or Khalkha is the name of the Mongolian nationality residing in Mongolia proper. Their dialect forms the basis of Mongolian written with Cyrillic letters. Throughout this text, *Modern Mongolian* is used as a synonym.

Buryat is the name of the Mongolian nationality residing in Buryatia, north of Mongolia, east of Lake Baikal, being a part of the Russian Federation. The Buryat call themselves *Buryaad* while Xalx Mongolians call them *Buriad*. The English name follows the Russian orthography. Linguistically, Xalx and Buryat Mongol are fairly close languages; Buryat has a slightly different sound system in which the phoneme /s/ partially shifted to /h/; the modern Buryat Cyrillic alphabet (virtually identical with the Cyrillic alphabet used for writing Modern Mongolian) has one additional letter (H/h, `\xalx{H/h}`) for marking the difference to /s/.

Bicig (literally *script* in Mongolian) denotes text written in the traditional Mongolian script which is also referred to as Uighur. Throughout this document, the term *Bicig* will be used on an equal footing with *Classical* and *Traditional* Mongolian. The latter term is used in the names of the Unicode/ISO10646 character plane U1800 which contains Mongolian, Manju, Sibe and sets of special characters called Ali Gali or Galig. In order to identify Mongolian script related commands distinct for Mongolian and Manju, the Mongolian commands have the name root `bicig` whereas the Manju commands have the name root `bithe`.

Xalx Mongolian, or Modern Colloquial Mongolian, is about as different from the form written in Classical script as modern English in phonetical spelling (assume it be written in Shavian letters) from the highly historical orthography of Standard English. Beyond these differences, Mongolian written in Classical Script usually preserves a substantial amount of historical grammatical features which make it look a bit like Elizabethan English.

Manju Manju is a Tungusic language closely related to Mongolian. Though Manju is virtually not spoken anymore, it has been the official language during 300 years of Manju government in Qing Dynasty China. Vast amounts of official documents survive, as well as some of the finest multilingual dictionaries ever compiled, e. g. the Pentaglot, or Mirror in Five Languages, a dictionary with 18671 entries in five languages (Manju, Tibetan, Mongolian, Uighur and Chinese). See [98] for more details. Manju writing is derived from Uighur Mongolian by adding diacritics in the form of dots and circles (*tongki fuka sindaha hergen*, script with dots and circles).

MS-DOS (Microsoft® Disk Operating System) An old operating system for personnal computers (PCs). From the minitoc point of view, its main drawback is the use of filenames with short extensions (the "8+3" scheme), which limits to 99 the number of mini-tables for each kind.

`mtc-2c.tex` An example file showing the use of the minitoc package with a two columns page layout.

`mtc-2nd.tex` An example of document using the minitoc package and its `french2` language option.

`mtc-add.bib` A small bibliographic data base for the `mtc-add.tex` and `mtc-ads.tex` example documents.

mtc-add.tex An example document showing how to use \mtcaddchapter and the tocbibind package [253] with minitoc. See section 4.3 on page 89.

mtc-ads.tex An example document showing how to use \mtcaddsection and the tocbibind package [253] with minitoc. See section 4.4 on page 94. It also shows how it is challenging to manage the mini-lists of floats at the section level.

mtc-amm.tex An example file showing the use of the appendices environment in a memoir class document with the minitoc package.

mtc-apx.tex An example file showing the use of the mtchideinmaintoc environment to hide the entries of the appendices in the main TOC and to create a part-level TOC for the appendices.

mtc-art.tex An example of document (article class) using the minitoc package.

mtc-bk.tex An example of document (book or report class) using the minitoc package.

mtc-bo.tex An example file showing the use of the minitoc package with a two columns page layout and using the tocloft package [250].

mtc-ch0.tex An example file showing the use of the minitoc package in a document with a starred first chapter.

mtc-cri.tex An example file showing the use of the minitoc package with starred parts and chapters.

mtc-fo1.tex An example file showing the use of the minitoc package with changing some fonts.

mtc-fo2.tex Another example file showing the use of the minitoc package with changing some fonts.

mtc-gap.tex An example file showing the use of the \mtcgapbeforeheads and \mtcgapafterheads commands.

mtc-hi1.tex An example file showing the use of the mtchideinmainlof and mtchideinmainlot specialized environments.

mtc-hi2.tex An example file showing the use of the following pairs of commands:

- \mtchideinmainlof and \endmtchideinmainlof,
- \mtchideinmainlot and \endmtchideinmainlot.

mtc-hia.tex An example file showing the use of the minitoc package to hide the entries for some tables in the main list of tables of an article class document.

mtc-hir.tex An example file showing the use of the minitoc package to hide the entries for some tables in the main list of tables of a report class document.

mtc-hop.tex An example file showing the use of the minitoc package with the scrbook document class.

mtc-liv.tex An example file showing the use of the minitoc package in a book with customized table of contents an minitocs.

mtc-mem.tex An example file showing the use of the minitoc package with the memoir class.

mtc-mm1.tex An example file showing the use of the minitoc package with the memoir class, if you want to change some fonts.

mtc-mu.tex A document using a minitoc set in a `wrapfigure` environment with the wrapfig package [11].

mtc-sbf.tex An example file showing the use of the minitoc package with the subfigure package [94].

mtc-scr.tex An example file showing the use of the minitoc package with a KOMA-Script class [147, 195], scrreprt.

mtc-syn.tex An example file showing the use of the minitoc package when the table of contents is prededed by some starred chapters.

mtc-tbi.tex An example file showing the use of the minitoc package with the tocbibind package [253].

mtc-tlc.tex An example file showing the use of the minitoc package in a document of article class. It is the example of [189, page 58], modernized.

mtc-tlo.tex An example file showing the use of the minitoc package with the tocloft package [250] and their interaction about the page numbers in the mini-tables.

mtc-tsf.tex An example file showing the use of the minitoc package with the subfig package [96].

mtcmess A package used to provide variants of the standard commands `\PackageInfo`, `\PackageWarning`, `\PackageWarningNoLine`, and `\PackageError` by adding an optional argument for an unique message identifier.

mtcoff A package which is used in place of the minitoc package to ignore all the commands and environments of the minitoc package. In fact, it defines them to do nothing. Useful if you want a version of your document without any mini-table.

mtcpatchmem A small package which is automatically loaded if necessary when you use the memoir document class with a version incompatible with the minitoc package, but correctible. It is generated when compiling `minitoc.ins`.

mu A length unit normaly used in math mode (`mu` means "math unit"); 18 math units make `1em` (one quad), which is about the width of a "m" in the current font. So the size of `1mu` is font dependent. The separation between dots in the dotted lines in the mini-tables is expressed in math units.

multibib The multibib package [122] allows to create references to multiple bibliographies within one document. It thus provides a complementary functionality to packages like bibunits [121] or chapterbib [12], which allow to create one bibliography for multiple, but different parts of the document.

needspace The needspace package [249] provides commands to reserve space at the bottom of a page. If there is not enough space on the current page (column) a new page (column) is started.

Table 6.6: Most common font encodings

Encoding	Description
T1	LaTeX extended text ("Cork")
TS1	LaTeX symbols (Latin)
T2A, T2B, T2C	LaTeX text (Cyrillic)
T3	LaTeX phonetic alphabet
TS3	LaTeX phonetic alphabet (extra symbols)
T4	LaTeX text (African languages)
T5	LaTeX text (Vietnamese)
T7	LaTeX text (reserved for Greek)
OT1	TeX text (as defined by Donald Knuth)
OT2	TeX text for Cyrillic languages (obsolete)
OT3	International phonetic alphabet (obsolete)
OT4	TeX text with extensions for the Polish language
OT6	TeX text with extensions for the Armenian language
OML	TeX math italic (Donald Knuth)
OMS	TeX math symbols (Donald Knuth)
OMX	TeX math large symbols (Donald Knuth)
X2	LaTeX extended text (Cyrillic)
U	Unknown
L⟨*xx*⟩	A local encoding
LTH	Encoding used for the Thai language
LV1	Encoding used with some VTeX fonts
LY1	Alternative to T1 encoding, for Y&Y software

NFSS The *New Font Selection Scheme*. The LaTeX 2_ε font selection system [165] was first released as the "New Font Selection Scheme" (NFSS) in 1989, and then in release 2 in 1993. LaTeX 2_ε includes NFSS release 2 as standard.

Every text font in LaTeX has five *attributes*:

encoding This specifies the order that characters appear in the font. The two most common text encodings used in LaTeX are Knuth's "TeX text" encoding (OT1), and the "TeX text extended" encoding (T1) developed by the TeX Users Group members during a TeX Conference at Cork in 1990 (hence its informal name "Cork encoding"). See [166].

family The name for a collection of fonts, usually grouped under a common name by the font foundry. For example, "Adobe Times", "ITC Garamond", and Knuth's "Computer Modern Roman" are all font families.

series How heavy or expanded a font is. For example, "medium weight", "narrow" and "bold extended" are all series.

shape The form of the letters within a font family. For example, "italic", "oblique" and "upright" (sometimes called "roman") are all font shapes.

size The design size of the font, for example "10pt".

The possible values for these attributes are given short acronyms by LaTeX. The most common values for the font encoding are given in table 6.6.

Table 6.7: Most common font families

Family	Description
cmr	Computer Modern Roman
cmss	Computer Modern Sans
cmtt	Computer Modern Typewriter
cmm	Computer Modern Math Italic
cmsy	Computer Modern Math Symbols
cmex	Computer Modern Math Extensions
ptm	Adobe Times
phv	Adobe Helvetica
pcr	Adobe Courier

Table 6.8: Most common font series

Series	Description
m	Medium
b	Bold
bx	Bold extended
sb	Semi-bold
c	Condensed

Table 6.9: Most common font shapes

Shape	Description
n	Normal (that is "upright" or "roman")
it	Italic
sl	Slanted (or "oblique")
sc	Caps and small caps

Table 6.10: The five font parameters for some fonts

LaTeX specification					Font	TeX name
OT1	cmr	m	n	10	Computer Modern Roman 10 point	cmr10
OT1	cmss	m	sl	1pc	Computer Modern Sans Oblique 1 pica	cmssi12
OML	cmm	m	it	10pt	Computer Modern Math Italic 10 point	cmmi10
T1	ptm	b	it	1in	Adobe Times Bold Italic 1 inch	ptmb8t at 1in

The "local" encodings are intended for font encodings which are only locally available, for example a font containing an organisation's logo in various sizes.

There are far too many font families to list them all, but some common ones are listed in table 6.7. The most common values for the font series are listed in table 6.8. The most common values for the font shape are listed in table 6.9.

The font size is specified as a dimension, for example `10pt` or `1.5in` or `3mm`; if no unit is specified, `pt` is assumed. These five parameters specify every LaTeX font, see table 6.10, for example.

These five parameters are displayed whenever LaTeX gives an overfull box warning, for example:

```
Overfull \hbox (3.80855pt too wide) in paragraph at lines
   314--318
[]\OT1/cmr/m/n/10 Normally [] and [] will be iden-ti-cal,
```

Table 6.11: Author commands for fonts

Author command	Attribute	Value in article class
\textrm... or \rmfamily	family	cmr
\textsf... or \sffamily	family	cmss
\texttt... or \ttfamily	family	cmtt
\textmd... or \mdseries	series	m
\textbf... or \bfseries	series	bx
\textup... or \upshape	shape	n
\textit... or \itshape	shape	it
\textsl... or \slshape	shape	sl
\textsc... or \scshape	shape	sc
\tiny	size	5pt
\scriptsize	size	7pt
\footnotesize	size	8pt
\small	size	9pt
\normalsize	size	10pt
\large	size	12pt
\Large	size	14.4pt
\LARGE	size	17.28pt
\huge	size	20.74pt
\Huge	size	24.88pt

The table 6.11 lists the author commands for fonts which set these five attributes[6].

nocheckfiles A package option of minitoc. The opposite of the `checkfiles` package option (see above).

nohints A package option of minitoc. The opposite of the `hints` package option (see above).

nolistfiles An option of the minitoc package. It is the opposite of the `listfiles` above. See section 1.7 on page 49.

notoccite This option of the minitoc package loads the notoccite package [9] (by Donald Arseneau). It avoids problems with `\cite` commands in sectionning commands or captions. See section 1.6 on page 49.

Omega The Omega typesetting system[7] (**Ω**) (by Yannis Haralambous and John Plaice) is an extension of TEX that is aimed primarily at improving TEX's multilingual abilities.

When the TEX program was originally developed in the mid seventies [*circa* 1975] by Professor Donald Knuth it was mainly aimed at typesetting mathematical texts in the english language. Since then TEX has made inroads in broader and broader areas of scientific, literary and other scholarly activities in many countries all over the world. In 1991, Knuth froze TEX, mainly in the interest of stability. However, he allows the TEX code to be used as the basis for further developments, so long as the resulting system is distributed under a different name.

[6] The values used by these commands are determined by the document class.

[7] Most but not all of this note is taken in the Omega documentation [125, 126, 129, 212, 213].

Table 6.12: Some systems derived from TEX and LATEX

TEX	$\longrightarrow$	$\mathbf{\Omega}$	+	LATEX	$\longrightarrow$	$\mathbf{\Lambda}$
		+				+
TEX	$\longrightarrow$	ε-TEX	+	LATEX	$\longrightarrow$	ε-LATEX
		אל	+	LATEX	$\longrightarrow$	לל

In Omega all characters and pointers into data-structures are 31-bit wide, instead of 8-bit, thereby eliminating many of the trivial limitations of TEX. Omega also allows multiple input and output character sets, and uses programmable filters to translate from one encoding to another, to perform contextual analysis, etc. Internally, Omega uses the universal Unicode/ISO-10646 character set. Omega also includes support for multiple writing directions.

These improvements not only make it a lot easier for TEX users to cope with multiple or complex languages, like Arabic, Indic, Khmer, Chinese, Japanese or Korean, in one document, but also form the basis for future developments in other areas, such as native color support and hypertext features.

The LATEX format (in the TEX meaning of that word) adapted to the special features of Omega is called "*Lambda*" ($\mathbf{\Lambda}$). Extending Omega with the ε-TEX extensions is a separate project, known as "*Aleph*" (אל) [34, 119] and led by Giuseppe BILOTTA. The LATEX for *Aleph* is known as "*Lamed*" (לל).

package Packages[8] are a very important feature of LATEX. These are extensions to the basic LATEX commands that are written to files with names that end with `.sty` and are loaded with the command `\usepackage` in the preamble. Packages can be classified by they origin.

- **Core** packages (in fact, **base** and **required** packages) are an integral part of the LATEX basic installation and are therefore fully standard.
- **Tools** packages are a set written by members of the LATEX3 Team and should always be in the installation.
- **Graphics** packages are a standardized set for including pictures generated by other programs and for handling colors; they are at the same level as the tools packages.
- **$\mathcal{A}\mathcal{M}\mathcal{S}$-LATEX** packages, published by the American Mathematical Society, should be in any installation[9].
- **Contributed** packages have been submitted by actual users; certain of these have established themselves as "essential" to standard LATEX usage, but all are useful.

partlof A list of figures for a part.

`partlofdepth` This counter, if defined, contains the depth of the partlofs.

[8] This info is taken from [148, page 12–13] and adapted.
[9] They are indispensable if you use a lot of mathematics.

partlot A list of tables for a part.

partlotdepth This counter, if defined, contains the depth of the partlots.

parttoc A table of contents for a part.

parttocdepth This counter contains the depth of the parttocs.

PDF Portable Document Format [115]. A descendant of the Postscript language from Adobe, optimized for navigation on the Internet. It adds hypertext, font substitution, and compression features.

placeins The placeins package [10] keeps floats "in their place", preventing them from floating past a `\FloatBarrier` command into another section. To use it, declare `\usepackage{placeins}` in the preamble and insert `\FloatBarrier` commands at places that floats should not move past, perhaps at every `\section`. The `insection` package option of the minitoc package does that with adequate options, and loads also the flafter package (described in [162] and [189, page 286]); see section 1.3.3 on page 28.

placeins.txt A plain text file containing the documentation of the placeins package [10].

PLaTeX A version of LaTeX customized for the polish (`polski`) language. See [199, 247]. But the *same* name refers also to a version of LaTeX customized for the japanese language, PLaTeX 2_ε [133].

pmk An example of shell script to prepare the minitoc package and its documentation; you should adapt it to your needs. See item 9 on page 224.

PostScript A page description language, by Adobe. It describes the appearance of a page, including elements such as text, graphics, and scanned images, to a printer or visualization device. Introduced by Adobe in 1985, it has become the language of choice in high quality printing.

preamble In the main file of a LaTeX source document, the part of it between the commands `\documentclass[...]{...}` and `\begin{document}`. In the preamble, you can insert global declarations and the loading of packages via `\usepackage` commands.

preparation The preparation commands of the minitoc package prepare the auxiliary files for the mini-tables of a given type. A *preparation* command must have been invoked (only once) before any insertion command for the mini-table type. The preparation commands are:

`\doparttoc`, `\dopartlof`, `\dopartlot`,	(part level)
`\dominitoc`, `\dominilof`, `\dominilot`,	(chapter level)
`\dosecttoc`, `\dosectlof`, `\dosectlot`,	(section level)
`\mtcprepare&(all levels)`	

proc A standard LaTeX document class, for preparing conference proceedings. For the minitoc package, it is very similar to the article document class; see above.

pseudo-chapter Or starred chapter. A chapter introduced by a `\chapter*` command. By default, it has no entry in the table of contents. `\chapter*` needs some precautions with the minitoc package. See section 1.3.4 on page 32.

quotchap The quotchap package [232] provides a set of commands for adding quotations to some headings (chapters) in the standard LaTeX 2_ε document classes: book, and report. It must be loaded *before* the minitoc package (see point 1.8 on page 50 and section 2.39 on page 72).

README is a plain text file (english) describing briefly the minitoc package, plus some useful infos.

report A standard LaTeX document class. It has sectionning commands: `\part`, `\chapter`, and `\section` (and below). It is compatible with the minitoc package and you can make mini-tables at the part and chapter levels (but not at the section level, to avoid too many auxiliary files).

`rmk` An example of shell script, which sorts the files of the minitoc package into classes (one directory for each class). It should be run after the scripts `imk` (mandatory) and `emk` and/or `fmk`, in that sequence. See item 9 on page 224.

romannum The romannum package [259] changes the numbers (for sectionning commands) generated by LaTeX from arabic digits to roman numerals. This package uses the stdclsdv package [248]. It must be loaded *before* the minitoc package (see point 1.8 on page 50 and section 2.40 on page 72).

rubber `rubber` [22] is a wrapper for LaTeX and companion programs. Its purpose is, given a LaTeX source to process, to compile it enough times to resolve all references, possibly running satellite programs such as **BibTeX**, **makeindex**, **Metapost**, etc., to produce appropriate data files. It has facilities to make some post-processing cleanup actions, like deleting the auxiliary files created by minitoc.

scrartcl See KOMA-Script above.

scrbook See KOMA-Script above.

scrreprt See KOMA-Script above.

sectionning commands These are the LaTeX commands which specify the logical structure of your document. The main sectionning commands are `\part`, `\chapter`, `\section`, `\subsection`, `\subsubsection`, `\paragraph`, or `\subparagraph`. Some standard document classes have not the `\chapter` command (like the article and proc classes), some have no sectionning commands (like the letter class). In the later case, the minitoc package is pointless. If some of the `\part`, `\chapter`, or `\section` commands are not defined, the minitoc commands for that level are unavailable. If `\chapter` is defined, the minitoc commands at the section level are not defined in the current and older versions of the minitoc package, but if `\chapter` is not defined and `\section` is defined, then the minitoc commands at the section level are defined. See section 1.1.1 on page 25. In non-standard document classes, sectionning commands with non-standard names cannot be recognized by the minitoc package.

sectlof A list of figures for a section.

`sectlofdepth` This counter, if defined, contains the depth of the sectlofs.

sectlot A list of tables for a section.

sectlotdepth This counter, if defined, contains the depth of the sectlots.

sectsty The sectsty package [182] provides a set of commands for changing the font used for the various sectional headings in the standard LaTeX 2_ε document classes: article, book, and report. This package also works with the KOMA-Script classes [147, 195] scrartcl, scrbook, and scrreprt. It must be loaded *before* the minitoc package (see point 1.8 on page 50 and section 2.28 on page 67).

secttoc A table of contents for a section.

secttocdepth This counter contains the depth of the secttocs.

sfheaders The sfheaders package [172] (for LaTeX 2_ε) borrows some definitions from the standard article/report/book classes and modifies them in order to print the part, chapter, section, subsection... headers with the Sans-Serif variant of the current font. It must be loaded *before* the minitoc package (see point 1.8 on page 50 and section 2.41 on page 72).

shell In the Unix, Unix-like and Linux operating systems, the **shell** is a program used as an interface between the operating system and the user. It can also be used as a scripting language to write programs or scripts to prepare routinely used sequences of tasks. The main shells are the Bourne shell (**sh**), the C shell (**csh**), the Korn shell (**ksh**), and their many successors (like **bash**, **tcsh**, etc.).

shortext An option of the minitoc package. It forces the use of short extensions (3 characters) in the names of the minitoc auxiliary files. This option is inactive by default, but is automatically activated if your operating system needs short extensions. See **autoconfiguration** above and the section 2.5 on page 56.

shorttoc The shorttoc package [105] allows to create an other table of contents in a document, with an other title and an other depth than the main table of contents.

SJIS The SJIS character encoding (for the japanese language), also known as MS-Kanji (Kanji for Microsoft®), consists of two overlaid character sets: the so-called halfwidth Katakana (JIS X0201-1976, 1-byte characters encoded in the range `0xA1` to `0xDF`) and the (fullwidth) JIS character set (JIS X0208-1990, mapped to the remaining code points). This information is taken from [167, 168].

SLaTeX A version of LaTeX customized for the swedish language. See [181].

splitbib A LaTeX package [179] which allows for sorting a bibliography into categories and subcategories; this is interesting for lists of publications, for grouping references by subject, by year, ...

stdclsdv The stdclsdv package [248] is intended to be used by the authors of LaTeX packages that need to know about the sectional divisions provided by the document class.

strut A vertical invisible rule used to force a minimal separation between two lines of text.

subfig The subfig package [96] provides support for the inclusion of small, "sub-figures" and "sub-tables". It simplifies the positioning, captioning and labeling of them within a single `figure` or `table` environment. In addition, this package allows such sub-captions to be written to the List of Figures or List of Tables if desired.

subfigure The subfigure package [94] is an obsolete version (by the same author) of the subfig package [96].

suffix See "extension" above.

TDS The TEX Directory Structure [235, 236]; a directory structure highly recommended to store macros, fonts, and the other implementation-independent TEX system files; it also suggests how to incorporate the rest of the TEX files in a single structure; the TDS has been designed to work on all modern systems.

TEX TEX is a computer program created by Donald KNUTH [143]. It is aimed at typesetting text and mathematical formulae. KNUTH started writing the TEX typesetting engine in 1977 to explore the potential of the digital printing equipment that was beginning to infiltrate the publishing industry at that time, especially in the hope that he could reverse the trend of deteriorating typographical quality that he saw affecting his own books and articles. TEX as we use it today was released in 1982, with some slight enhancements added in 1989 to better support 8-bit characters and multiple languages. TEX is renowned for being extremely stable, for running on many different kinds of computers, and for being virtually bug free. The version number of TEX is converging to π and is now at 3.141592.

TEX is pronounced "Tech," with a "ch" as in the German word "Ach" or in the Scottish "Loch." In an ASCII environment, TEX becomes `TeX`.

thailatex The thailatex package [183] allows to typeset documents in the Thai language. You can also use the CJK system [167, 168].

thispagestyle A type of *feature* (see this term) which is executed at each occurrence of a given type of mini-table, to force the page style to use for the current page. Look at the documentation of the `\mtcsetfeature` command, in section 1.10 on page 45.

`tight` An option of the minitoc package. It gives a tight line spacing in the mini-tables. The opposite option is `loose`.

titlesec The titlesec package [33] allows to change the sectioning titles. Amongst its many features it provides margin titles, different format in left and right pages, rules above and below the title, etc. Unfortunately, it is *incompatible* with the minitoc package.

titletoc The titletoc package is useful for toc entries formatting, providing the possibility of changing the format in the middle of a document, grouping the entries in a single paragraph, pretty free-forms entries, partial tocs, etc. Unfortunately, it is *incompatible* with the minitoc package.

The `titletoc.sty` file is not part of the titlesec package; it's an independent package, but it's described in the titlesec package documentation [33].

`tmk` A script file which creates a TDS hierarchy [235, 236] (to be ajusted to your system).

TOC Acronym for "table of contents".

tocbibind The tocbibind package [253] can be used to add the ToC and/or bibliography and/or the index etc., to the Table of Contents listing. But it needs some precautions when used with the minitoc package. See section 1.5.5 on page 47.

tocdepth This counter contains the depth of the table of contents.

tocloft The tocloft package [250] provides means of controlling the typographic design of the Table of Contents, List of Figures and List of Tables. New kinds of "List of ...' can be defined. If you use the tocloft package and the minitoc package, see section 2.21 on page 61 about fixing some minor compatibility issues.

TODO is a plain text file (english) which lists some suggested developments of the package, not yet implemented. Comments and suggestions are welcome.

token A token[10] is either (a) a single character with an attached category code (see "catcode" above), or (b) a control sequence. You *should* remember two chief things about TEX's tokens: (1) A control sequence is considered to be a single object that is no longer composed of a sequence of symbols. Therefore long control sequence names are no harder for TEX to deal with than short ones, after they have been replaced by tokens. Furthermore, spaces are not ignored after control sequences inside a token list; the ignore-space rule applies only in an input file, during the time that strings of characters are being tokenized. (2) Once a category code has been attached to a character token, the attachment is permanent. For example, if character '{' were suddenly declared to be of category 12 instead of category 1, the characters '$\{_1$' already inside token lists of TEX would still remain of category 1; only newly made lists would contain '$\{_{12}$' tokens. In other words, individual characters receive a fixed interpretation as soon as they have been read from a file, based on the category they have at the time of reading. Control sequences are different, since they can change their interpretation at any time. TEX's digestive processes always know exactly what a character token signifies, because the category code appears in the token itself; but when the digestive processes encounter a control sequence token, they must look up the current definition of that control sequence in order to figure out what it means.

UNIX A modern operating system, available on many computers and in various flavors. From the minitoc point of view, it has the advantage of using filenames with long extensions (the length limit is too high to be a problem with the number of mini-tables).

UNIX-like Operating systems analog to Unix, with the same advantages. Linux is a good example, but others exist.

urlbst A PERL script, by Norman Gray [118], to add a `webpage` BibTEX entry type, and add support for general `url` and `lastchecked` fields, to (most) BibTEX `.bst` files. Optionally adds basic support for `eprint` and `doi` fields, and hypertex/hyperref support, too.

UTF 8 UTF 8 (Unicode Transformation Format 8), also called UTF 2 or FSS-UTF, is a special representation of Unicode (resp. ISO 10 646). It uses multibyte sequences of various lengths, but only 2-byte and 3-byte sequences are implemented in CJK. ASCII characters will be used as-is — without this property it would be impossible to use UTF 8 with TEX. See table 6.2 on page 197.

varsects The varsects package [228] provides a set of commands for changing the font used for the various sectional headings in the standard LaTEX 2_ε document classes:

[10] This definition is taken from "*The TEXbook*" [143].

article, book, and report. It must be loaded *before* the minitoc package (see point 1.8 on page 50 and section 2.33 on page 69).

wrapfig The wrapfig package [11] provides the `wrapfigure` and `wraptable` environments to place a figure or table at the side of the page and wrap text around it.

xmk An example of shell script, which typesets the example document files into PDF documents.

xr The xr package [89] implements a system for eXternal References. I wrote the first version of this package, but it had severe problems. David Carlisle rewrote it in a much better and more robust way. With his permission, I used some of his code in the minitoc package to implement the preparation commands (like `\dominitoc`). If you use also the hyperref package [214], use xr-hyper [90] in place of the xr package.

Chapter 7

Installation

Tables

This chapter describes the installation of the minitoc package (version #51).

This package contains a lot of files. The list of all files is given in `minitoc.l`. See table 7.1 on the following page. 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`, `mtcoff.sty`, `mtcmess.sty`, and *all* `*.mld` and `*.mlo` files are the package itself[1].

The table 1.7 on page 36 lists the available languages; for each of these languages, a *language*`.mld` file is available; the languages in parentheses are aliases of a main language and their `.mld` files will load the `.mld` file of that main language.

`mtcpatchmem.sty` is a temporary fix for compatibility with the memoir class.

The files of this class must be *all* installed in a directory where LaTeX 2_ε finds the `.sty` files.

(2) Informative text files:

[1] The large number of `*.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 may load another one) and, if necessary, a `.mlo` file; 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). *Note that the presence of the* `english.mld` *file is mandatory.* Since version #50, the minitoc package signals the missing `.mld` or `.mlo` files and gives their list in a warning message.

Table 7.1: List of files (minitoc.l)

class (0) :
- minitoc.ins
- minitoc.dtx
- fminitoc.dtx

class (1) :
- minitoc.sty
- mtcoff.sty
- mtcmess.sty
- mtcpatchmem.sty
- acadian.mld
- acadien.mld
- afrikaan.mld
- afrikaans.mld
- albanian.mld
- american.mld
- arab.mld
- arab2.mld
- arabi.mld
- arabic.mld
- armenian.mld
- australian.mld
- austrian.mld
- bahasa.mld
- bahasai.mld
- bahasam.mld
- bangla.mld
- basque.mld
- bicig.mld
- bicig2.mld
- bicig3.mld
- bithe.mld
- brazil.mld
- brazilian.mld
- breton.mld
- british.mld
- bulgarian.mld
- bulgarianb.mld
- buryat.mld
- buryat2.mld
- canadian.mld
- canadien.mld
- castillan.mld
- castillian.mld
- catalan.mld
- chinese1.mld, chinese1.mlo
- chinese2.mld, chinese2.mlo
- croatian.mld
- czech.mld
- danish.mld
- devanagari.mld
- dutch.mld
- english.mld
- english1.mld
- english2.mld
- esperant.mld
- esperanto.mld
- estonian.mld
- ethiopia.mld
- ethiopian.mld
- ethiopian2.mld
- farsi1.mld, farsi1.mlo
- farsi2.mld, farsi2.mlo
- farsi3.mld
- finnish.mld
- finnish2.mld
- francais.mld
- french.mld
- french1.mld
- french2.mld
- frenchb.mld
- frenchle.mld
- frenchpro.mld
- galician.mld
- german.mld
- germanb.mld
- germanb2.mld
- greek.mld
- greek-mono.mld
- greek-polydemo.mld
- greek-polykatha.mld
- guarani.mld
- hangul1.mld, hangul1.mlo
- hangul2.mld, hangul2.mlo
- hangul3.mld, hangul3.mlo
- hangul4.mld, hangul4.mlo
- hangul-u8.mld, hangul-u8.mlo
- hanja1.mld, hanja1.mlo
- hanja2.mld, hanja2.mlo
- hanja-u8.mld, hanja-u8.mlo
- hebrew.mld
- hebrew2.mld
- hindi.mld
- hindi-modern.mld
- hungarian.mld
- icelandic.mld
- indon.mld
- indonesian.mld
- interlingua.mld
- irish.mld
- italian.mld
- italian2.mld
- japanese.mld, japanese.mlo
- japanese2.mld, japanese2.mlo
- japanese3.mld, japanese3.mlo
- japanese4.mld, japanese4.mlo
- japanese5.mld, japanese5.mlo
- japanese6.mld, japanese6.mlo
- kannada.mld
- khalkha.mld
- latin.mld
- latin2.mld
- latvian.mld
- letton.mld
- lithuanian.mld
- lowersorbian.mld
- lsorbian.mld
- magyar.mld
- magyar2.mld
- magyar3.mld
- malay.mld
- malayalam-keli.mld
- malayalam-omega.mld, malayalam-omega.mlo
- malayalam-rachana.mld
- malayalam-rachana2.mld
- manju.mld
- meyalu.mld
- mongol.mld
- naustrian.mld
- newzealand.mld
- ngerman.mld
- ngermanb.mld
- ngermanb2.mld
- norsk.mld
- norsk2.mld
- nynorsk.mld
- nynorsk2.mld
- polish.mld
- polish2.mld
- polski.mld
- portuges.mld
- portuguese.mld
- romanian.mld
- romanian2.mld
- romanian3.mld
- russian.mld
- russianb.mld
- russianc.mld
- russian2m.mld
- russian2o.mld
- russian-cca.mld, russian-cca.mlo
- russian-cca1.mld, russian-cca1.mlo
- russian-lh.mld, russian-lh.mlo
- russian-lhcyralt.mld, russian-lhcyralt.mlo
- russian-lhcyrkoi.mld, russian-lhcyrkoi.mlo
- russian-lhcyrwin.mld, russian-lhcyrwin.mlo
- samin.mld
- scottish.mld
- serbian.mld
- serbianc.mld
- slovak.mld
- slovene.mld
- spanish.mld
- spanish2.mld
- spanish3.mld
- spanish4.mld
- swedish.mld
- swedish2.mld
- thai.mld, thai.mlo
- turkish.mld
- uighur.mld
- uighur2.mld
- uighur3.mld
- UKenglish.mld
- ukraineb.mld
- ukrainian.mld
- uppersorbian.mld
- USenglish.mld
- usorbian.mld
- vietnam.mld
- vietnamese.mld
- welsh.mld
- xalx.mld
- xalx2.mld
- xalx3.mld

class (2) :
- INSTALL, README, TODO, CATALOG
- minitoc.l

class (3) :
- mtc-2c.tex, mtc-2nd.tex, mtc-add.bib, mtc-add.tex, mtc-ads.tex, mtc-amm.tex, mtc-apx.tex, mtc-art.tex, mtc-bo.tex, mtc-bk.tex, mtc-ch0.tex, mtc-cri.tex, mtc-fo1.tex, mtc-fo2.tex, mtc-gap.tex, mtc-hi1.tex, mtc-hi2.tex, mtc-hia.tex, mtc-hir.tex, mtc-hop.tex, mtc-liv.tex, mtc-mem.tex, mtc-mm1.tex, mtc-mu.tex, mtc-sbf.tex, mtc-scr.tex, mtc-syn.tex, mtc-tbi.tex, mtc-tlc.tex, mtc-tlo.tex, mtc-tsf.tex

class (4) :
- minitoc.bug
- minitoc.sum

class (5) :
- minitoc.ins
- minitoc.dtx
- minitoc.bib
- minitoc.ist
- minitoc.lan
- minitoc.pre
- en-mtc.bst
- lamed.eps, lamed.pdf, lamed.tex

class (6) :
- minitoc.dtx
- fminitoc.dtx
- fminitoc.bib
- fminitoc.ist
- fminitoc.lan
- minitoc.pre
- franc.sty, frbib.sty, frnew.sty
- fr-mtc.bst

class (7) :
- minitoc.pdf

class (8) :
- fminitoc.pdf

class (9) :
- imk, emk, fmk, pmk, xmk, rmk, tmk
- duplex2v.pro

class (10) :
- minitoc-texmk.zip

- `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 7.1 on the preceding page.
- `README` is a file describing briefly the minitoc package, plus some useful infos.
- `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) Examples of documents: `mtc-2c.tex`, `mtc-2nd.tex`, `mtc-add.tex`, `mtc-ads.tex`, `mtc-amm.tex`, `mtc-apx.tex`, `mtc-art.tex`, `mtc-bk.tex`, `mtc-bo.tex`, `mtc-ch0.tex`, `mtc-cri.tex`, `mtc-fo1.tex`, `mtc-fo2.tex`, `mtc-gap.tex`, `mtc-hi1.tex`, `mtc-hi2.tex`, `mtc-hia.tex`, `mtc-hir.tex`, `mtc-hop.tex`, `mtc-liv.tex`, `mtc-mem.tex`, `mtc-mm1.tex`, `mtc-mu.tex`, `mtc-sbf.tex`, `mtc-scr.tex`, `mtc-syn.tex`, `mtc-tbi.tex`, `mtc-tlc.tex`, `mtc-tlo.tex`, `mtc-tsf.tex`, are example files, to play with. The associated `.pdf` files are provided. Another (*short*) examples are welcome.

(4) `minitoc.bug`, `minitoc.sum` are plain text documentation: list of problems (faq, see chapter 2 on page 53) and summary of commands (see chapter 3 on page 77).

(5) `minitoc.ins`, `minitoc.ist`, `minitoc.pre`, `minitoc.lan`, `en-mtc.bst`, `minitoc.dtx`, and `minitoc.bib` are the source of the documentation in (non perfect) english, `lamed.eps` and `lamed.pdf` are images to include, `lamed.tex` is the source code[2]. `minitoc.pre` is the common preambule code for the documentation.

(6) `minitoc.dtx`, `fminitoc.dtx`, `fminitoc.bib`, `fminitoc.ist`, `fminitoc.lan`, `minitoc.pre`, franc.sty, frbib.sty, frnew.sty, `fr-mtc.bst` are the source (and tools) of the documentation in french[3].

(7) `minitoc.pdf`, is the documentation in (non perfect) english, in PDF format.

(8) `fminitoc.pdf` is the documentation in french, in PDF format. The french documentation and its source files must not be left out.

(9) pmk is a shell script[4] to prepare the package and its documentation; the pmk script uses the `/tmp/‘whoami‘.imk` and `/tmp/‘whoami‘.tmk` directories to not waste disk space under your home directory; there are also six partial scripts[5] (to be adapted):

- `imk`, which prepares the package from `minitoc.ins` and `minitoc.dtx`; note that `imk` must be run before running `emk` or `fmk`; it creates also some `.sty` files

[2] Compiling `lamed.tex` requires a specific hebrew font and a specific package (both obsolete), plus conversion into EPS or PDF; hence I prefer to provide also the image files.

[3] This seems rather strange. In fact, the english and french documentations are both contained 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 utilitary files which are automatically created (some `.sty` files, `minitoc.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 there is no more any `fminitoc.ins` file.

[4] You can sip a big cappuccino ☕ while this script is running! Be patient.

[5] In fact, pmk assembles the scripts `imk`, `emk`, `fmk`, `xmk`, `rmk`, and `tmk`.

necessary to prepare the documentation but that are to be installed with it; the `imk` script uses the `/tmp/`\`whoami\``.imk` directory to not waste disk space under your home directory;

- `emk`, which prepares the english documentation from `minitoc.dtx`;
- `fmk`, which prepares the french documentation from `fminitoc.dtx` and `minitoc.dtx`;
- `xmk`, which typesets the example files (in PDF format);
- `rmk`, which sorts the files into classes (one directory for each class);
- `tmk`, which creates a TDS-conformant hierarchy [235, 236] (to be tailored to your system); see table 7.2 on the next page; this hierarchy is saved in `minitoc-texmf.zip`.

These scripts are currently written in C-shell, but they are very simple, and should be easy to convert in another classic shell. The documentation in PostScript format is no more distributed on the CTAN archives, but these scripts prepare it (using the `duplex2v.pro` PostScript header file to try printing recto-verso).

(10) `minitoc-texmf.zip` is a ZIP-archive file containing a TDS-compliant hierarchy with all the files of the minitoc package.

Some remarks about the `rmk`, `tmk` and `pmk` scripts (which you should tailor to your needs):

- **with `rmk`:**
  1. the hierarchy is *not* TDS-compliant;
  2. the files of (0) must be installed in a directory where LaTeX 2_ε finds `.dtx` and `.ins` files;
  3. the files of (1) must be installed in a directory where LaTeX 2_ε finds `.sty` files;
  4. the files of (2), (3), (4), (5), (6) and (9) must be installed in a separate directory, but must not be left out;
  5. the files of (7) and (8) must be installed as on-line documentation;
  6. the directories created by the `rmk` script are under `/tmp/`\`whoami\``.rmk` to not waste disk space under your home directory.
- **with `tmk`:**
  1. the hierarchy is TDS-compliant;
  2. each file appears only once in the hierarchy;
  3. the installation is much easier: you only need a `.zip` or a `.tar` (or `.tgz`) dump file[6] of the hierarchy to be deployed into the installed TDS hierarchy; you should examine *very carefully* (by comparison with your TDS installation) and tailor the `tmk` script before using it;
  4. the directories created by the `tmk` script are under `/tmp/`\`whoami\``.tmk` to not waste disk space under your home directory;

[6] The `tmk` script creates the `minitoc-texmf.zip` file.

Table 7.2: A TDS-conformant hierarchy for the minitoc files

texmf/						
bibtex/		doc/	dvips/	makeindex/	scripts/	source/	tex/
bib/	bst/	latex/	minitoc/	minitoc/	minitoc/	latex/	latex/
minitoc/	minitoc/	minitoc/	(D)	(E)	minitoc/	minitoc/	(H)
(A)	(B)	(C)			(F)	(G)	
		examples/					
		(I)					

(A) fminitoc.bib, minitoc.bib;	(F) emk, fmk, imk, pmk, rmk, tmk, xmk;	mtc-art.pdf, mtc-art.tex, mtc-bk.pdf, mtc-bk.tex, mtc-bo.pdf, mtc-bo.tex, mtc-ch0.pdf, mtc-ch0.tex, mtc-cri.pdf, mtc-cri.tex, mtc-fo1.pdf, mtc-fo1.tex, mtc-fo2.pdf, mtc-fo2.tex, mtc-gap.pdf, mtc-gap.tex, mtc-hi1.pdf, mtc-hi1.tex, mtc-hi2.pdf, mtc-hi2.tex, mtc-hia.pdf, mtc-hia.tex, mtc-hir.pdf, mtc-hir.tex,	mtc-hop.pdf, mtc-hop.tex, mtc-liv.pdf, mtc-liv.tex, mtc-mem.pdf, mtc-mem.tex, mtc-mm1.pdf, mtc-mm1.tex, mtc-mu.pdf, mtc-mu.tex, mtc-sbf.pdf, mtc-sbf.tex, mtc-scr.pdf, mtc-scr.tex, mtc-syn.pdf, mtc-syn.tex, mtc-tbi.pdf, mtc-tbi.tex, mtc-tlc.pdf, mtc-tlc.tex, mtc-tlo.pdf, mtc-tlo.tex, mtc-tsf.pdf, mtc-tsf.tex;
(B) en-mtc.bst, fr-mtc.bst;	(G) fminitoc.dtx, minitoc.dtx, minitoc.ins;		
(C) INSTALL, README, TODO, CATALOG, fminitoc.lan, fminitoc.pdf, lamed.eps, lamed.pdf, lamed.tex, minitoc.bug, minitoc.l, minitoc.lan, minitoc.pdf, minitoc.pre, minitoc.sum, franc.sty, frbib.sty, frnew.sty;	(H) minitoc.sty, mtcmess.sty, mtcpatchmem.sty, mtcoff.sty, *.mld, *.mlo.		
(D) duplex2v.pro;	(I) mtc-2c.pdf, mtc-2c.tex, mtc-2nd.pdf, mtc-2nd.tex, mtc-add.pdf, mtc-add.tex, mtc-ads.pdf, mtc-ads.tex, mtc-amm.pdf, mtc-amm.tex, mtc-apx.pdf, mtc-apx.tex,		
(E) fminitoc.ist, minitoc.ist;			

5. the file `minitoc-texmf.zip` (10) should not be installed; it is just a method to help making a TDS-compliant installation.

- **with pmk:**

  1. the pmk script performs the actions of `imk` (preparation of the basic files), `emk` and `fmk` (preparation of the english and french documentation), `xmk` (preparation of the examples of documents), `rmk` and `tmk` (repartition of files into classes and in a TDS-compliant hierarchy);
  2. the directories created by the pmk script are under `/tmp/‘whoami‘.rmk` and `/tmp/‘whoami‘.tmk` to not waste disk space under your home directory;
  3. the *same precautions* as for `tmk` are needed.

The file `minitoc-texmf.zip` contains a ZIP archive of a TDS-compliant hierarchy of all files of the minitoc package. It has been prepared by the pmk or `tmk` scripts.

Note that `minitoc.dtx` and hence `fminitoc.dtx` are (not so trivial) examples of using minitoc with hyperref. They show how the combinaison of these two packages may be useful.

Chapter 8

Postface

Contents

This chapter summarizes the evolution of the minitoc package, year by year. A more detailed history is available in “Changes History”, page 546. Many minor changes are skipped here.

In fact, this chapter is for the average user of the package, who wants to have an overview ot its evolution; the chapter “Changes History” is more oriented towards peoples interested in the code of the package and the problems encountered during its development.

8.1 The origins

The minitoc package was initially written by Nigel WARD in 1990 and 1991, with major contributions by Dan JURAFSKY. But minitoc suffered of a major weakness: when the number of chapters exceeded 9 or 10, you got a rather mysterious error message:

```
No room for a new \write.
```

As I needed the basic fonctionnality of this package (printing mini-tables of contents for each chapter), I looked further in its code and finally found the culprit: minitoc used a `\newwrite` command to create a new file for the contents table of each chapter, allocating a new file descriptor each time. But the number of file descriptors for writing is limited to 16 under LaTeX (in fact, by the underlying TeX program itself). As some descriptors are already used by LaTeX, writing more than 9 or 10 chapters was too much. Such errors are difficult to find when testing on too small documents: with few chapters, everything goes fine. But on a real document, with many chapters, the mysterious error happens.

8.2 New design in 1993

So I decided (June 1993, when I took the maintenance of the package) to change the allocation method to always use the same file descriptor for all the mini-table of contents files. Some major improvements happened in 1993: the addition of the mtcoff (minitocoff at this time) package and a rewrite of minitoc to extract the data from the *document*`.toc` file, with a selection mechanism. Then a first solution for the short extension problem was added (still manual). An elementary system for the fonts in the mini-tables was added. In December 1993, the minilofs and minilots were added.

8.3 Developments in 1994

The first improvements in 1994 were about the formating of the mini-tables: spacing was improved and the position of the title became ajustable (with the optional parameter of `\minitoc` or `\dominitoc`). But a major addition was done: part-level mini-tables (parttocs, partlofs, partlots) and, for articles, section-level mini-tables (secttocs, sectlofs, sectlots).

With the emergence of LaTeX 2_ε, replacing the ageing LaTeX2.09, some work was necessary to support the compatibility with this new version. This was not easy, but Denis B. ROEGEL and Frank MITTELBACH gave me many helpful hints.

Another major addition is the language option feature, with the concept of the *minitoc language definition file* (or `.mld` file), coming from the babel package [38, 39], by Johannes BRAAMS. But at this time, these files were still named as `.sty` files.

An important simplification is introduced by the notion of “absolute numbering”, with avoids many problems when the chapters are not numbered the standard way (consecutively, starting from 1, with arabic digits).

8.4 No developments in 1995

Sorry, I was busy with an other project.

8.5 Developments en 1996

The very annoying problem with the starred sectionning commands received some (rather primitive) solutions, but it is very complex, so manual interventions are often required.

The names of the minitoc language definition files take now the extension `.mld`, more specific, and english is the default language. Some new languages are added.

The work on the starred sectionning commands continues.

The minitocoff companion package is renamed mtcoff to keep its name short.

8.6 Developments in 1997

The problem of short extensions for files names under some operating systems is addressed via the new `shortext` package option and by the new autoconfiguration feature.

For the starred sectionning commands, the `\addstarredchapter` commande is added (with analog commands for starred parts and sections).

8.7 Developments in 1998

The `tight` and `loose` package options are added to improve the line spacing in the mini-tables.

8.8 Developments in 1999

The `dotted` and `undotted` package options are added to add or suppress some lines of dots (leaders) in the mini-tables.

8.9 Developments in 2000

A major addition is the compatibility with the hyperref package [214], and I ought to thank loudly Heiko Oberdiek, Didier Verna and Bernd Jaehne.

Some corrections about the starred sectionning commands are added by Heiko Oberdiek.

The mini-tables features (`\beforeparttoc` and co.) commands are added.

Some adjustment commands, like `\mtcaddchapter`, are added, again about the problem with the starred sectionning commands.

The compatibility with the tocbibind package [253] is documented.

Aliases for some languages are added.

8.10 Developments in 2001

Added the `checkfiles` and `nocheckfiles` package options, to avoid the insertion of (ugly) empty mini-tables.

Added the `\mtcselectlanguage` command to change more easily the language of the mini-tables titles.

8.11 Developments in 2002

Correction of an interaction between `\tableofcontents` (creating a hidden `\chapter*` or `\section*` command) and the numbering of the mini-table files.

Added the `\mtcskip` and `\mtcskipamount` commands.

8.12 Developments in 2003

Added the `insection` package option (which was temporarily done by the `flsection` and `flsectionb` package options) to deal with floats drifting out of their section. The placeins package [10] (by Donald Arseneau) is used.

The font commands are made compatible with the memoir class [257, 258].

Added compatibility with the notoccite package [9].

8.13 Developments in 2004

Added comments in the `.mld` files needing special fonts. Better documentation about languages.

Added an explanation about making a local table of contents for an appendix, eventually masking it in the main table of contents. Compatibility with the appendix package [252].

A major addition is the `hints` package option, to detect some programming and compatibility problems.

8.14 Developments in 2005

All messages are now written via the standard interface commands (`\PackageInfo`, `\PackageWarning`, and `\PackageError`), so the minitoc package is less verbose on the terminal.

Added the `\mtcsetfont` and `\mtcsettitlefont` commands (from a suggestion by Benjamin Bayart) to replace many font commands by only one command with a better user interface.

Comments about the $\mathcal{AMS}$ classes (some ones are incompatible with minitoc).

Added the `\mtcsetformat` and `\mtcsettitle` commands, again to have a simpler user interface.

Added various hints (`insection` package option, order of minitoc basic commands, short extensions).

Added the `\mtcsetpagenumbers` and `\mtcsetrules` commands, again to have a simpler user interface.

Added the `mtchideinmaintoc` environment, to hide a group of entries in the main table of contents; added also the `mtchideinmainlof` and `mtchideinmainlot` environments.

Added the `\mtcfixindex` and `\mtcsettitle` commands.

Added the description of the installation of the package (a new chapter and the file `INSTALL`).

Improved and added hints about consistency of `\dominitoc`/`\minitoc` and co.

Added the `\mtcsetfeature` command (very complex).

Added a hint about the abstract package [251].

The minitoc package is now written using the `.dtx-.ins` system. Some cleanup is done in the code.

Added the `\mtcfixglossary` command, like `\mtcfixindex`.

Some improvements are made to print the documentation.

Some new hints are added (sectsty package [182], empty mini-tables, obsolete commands).

Added the notion of depth for mini-tables of figures/tables. Added the `\mtcsetdepth` command.

The `hints` package option is now the default.

Added a method for making a bilingual documentation in one file (the `minitoc.dtx` file). This method could be used for more languages.

Added or improved some adjustement commands (`\adjustptc`, `\incrementptc`, `\decrementptc`, etc.).

Added the `k-tight` and `k-loose` package options, for the KOMA-Script [147, 195] document classes.

Added a patch for the recent version of the memoir class [257, 258].

Use `\mtcselectlanguage` in language options and in "secondary" `.mld` files.

Added the `\mtcloadmlo` command to be used in some `.mld` files to load a `.mlo` file. The extension `.mlo` means *minitoc language object*; such files contains characters not easily manipulated in a `.dtx` file.

The history of changes is now displayed in a much simpler way (using a glossary was too cumbersome).

Added the `listfiles` package option, to create a list of the minitoc auxiliary files, which can be removed after the LaTeX compilation of the document. It is the *document*`.maf` file.

Added a remark in the FAQ chapter (and `minitoc.bug`) about precautions to take with the starred sectionning commands.

Added hints about the caption, caption2, ccaption, and mcaption packages (they must be loaded *before* minitoc).

A "Jargon" chapter is added. It will grow slowly.

Added a note about a problem with minitoc, hyperref and memoir used together.

Some bugs in the `\mtcset...` commands are fixed.

Added a hint about the varsects package [228].

Added a hint on the number of mini-tables when short extensions are used.

Added a chapter with all the (explained) messages.

8.15 Developments in 2006

Added the "*" keyword as first argument of the `\mtcsetpagenumbers` and `\mtcsetrules` command, to get an action on all kinds of mini-tables.

Corrections in the `\mtcaddsection`, `\mtcfixglossary`, and `\mtcfixindex` commands.

In the PDF documentation, the panel of bookmarks shows initialy only the bookmarks for parts and chapters, but you can open them to show deeper entries.

Added a comment about the initialization of fonts in the FAQ (point 34). It is still an open domain and I am working on it.

Added a hint about the KOMA-Script classes [147, 195], and an entry in the FAQ chapter (and in `minitoc.bug`).

Added the "Postface" chapter.

Added the `\mtcprepare` command.

Added an URL field in the bibliography (the styles are modified with the `urlbst` tool [118]).

Added the mtcmess package to add unique identifiers to the messages.

Suppressed the PostScript documentation files from the distribution (no more accepted on CTAN archives), but the scripts still creates them.

Corrections in the `insection` package option.

Reordering of the chapters in the user's manual (part I).

Added hints about the fncychap [170], quotchap [232], romannum [259], sfheaders [172], alnumsec [152], and captcont [95] packages.

Added FAQ 44 and the `\mtcgapbeforeheads` and `\mtcgapafterheads` formatting commands.

Added the chapter 4, "Examples of documents", page 86.

Added FAQ 45 and the `\kernafterparttoc` and co. commands for the vertical space between a minitable and its bottom rule.

Increased the text width and adjusted the format of the entries in the TOC in the documentation.

Correction of the preamble in the generated files (spurious lines have been eliminated, at least).

Added `devanagari.mld` and `hindi.mld`. Added `hindi-modern.mld`.

The bibliographic styles `plainurl.bst` and `frplain1.bst` are renamed `en-mtc.bst` and `fr-mtc.bst`.

All example documents are renamed with names beginning with "`mtc-`".

The "`listfiles`" package option is active by default.

Better error messages about undefined preparation and insertion commands.

Added `japanese6.mld` and `japanase6.mlo`.

Added a hint about the hangcaption package [138].

Added (in the memento) a table of the classes and packages which are incompatible or need precautions with minitoc.

Added a validation of the language options with the presence of the `.mld` and `.mlo` files.

Added the `tmk` script and a table describing a TDS structure for minitoc.

Updated the `INSTALL` file and the "Installation" chapter.

Added the file `minitoc-texmf.zip` (a ZIP archive of a TDS-compliant hierarchy of all files of the package) to the distribution.

Improving the index (entries for: packages and classes, scripts, tools, names, examples, extensions, option, language options).

8.16 Developments in 2007

Corrections of minor bugs and in the documentation.

The names of some internal macros are shortened to fit into the margin.

Indexing the environments and the files.

Indexing the counters and depth counters.

The example files are in their own directory in the (proposed) TDS hierarchy.

Acknowledgments

I ought to thank the following peoples [1], for their help, their questions, their interventions in the news groups [2], and/or for their packages, classes, documents, and tools:
Hassan Abolhassani, Nabil Abu El-Ata, Tommaso Addabbo, Juan M. Aguirregabiria, Alex AJ, Vardan Akopian, Jérôme Andrieux, Ralf Angeli, Achod André Aradian, Einar Árnason, Tim Arnold, Donald Arseneau, David Aspinall, Ivar Åssen, Philipp Bachmann, Gonvcal Badenes, Marin Balgarensky, Jean-Yves Baudais, David Bausum, Benjamin Bayart, Thierry Bayet, Claudio Beccari, Emmanuel Beffara, József Bérces, Karl Berry, Javier Bezos, Giuseppe Bilotta, Árpád Bíró, Denis Bitouzé, Laurent Bloch, Georgi Boshnakov, Patrice Bougette, Daniel Bourbonnais, Victor Boyko, Johannes Braams, Felix Braun, Mustafa Burc, Olivier Cardi, David Carella, David Carlisle, Kevin Carmody, Manuel Carriba, Régis Caspar, Jean-Côme Charpentier, Jana Chlebikova, Marian Clegg, Steven Douglas Cochran, Maurizio Codogno, David B. Cook, Oliver Corff, Prakash Countcham, Serguei D'Achian, Adrian Daerr, Arnak Dalalyan, Patrick W. Daly, Arnaldo Viegas de Lima, Sébastien Demoustier, Éric Depardieu, Ben De Rydt, Dorjpalam Dorj, Lyndon Dudding, Marko Èehaja, Victor Eijkhout, Danie Els, Behdad Esfahbod, Thomas Esser, Karl F. Everitt, Mike Fabian, Robin Fairbairns, Simon Fear, Jürgen Fenn, Jeff Fessler, Ulrike Fischer, Daniel Flipo, Peter Flynn, Jim Fox, Danny M. Fürniss, Diego García Morate, Hubert Gässlein, Mohammad Ghodsi, Markus Gleiszner, Vitali Gontsharuk, Michel Goossens, Mathieu Goutelle, Fraser Grant, Norman Gray, Loïc Grenon, Sébastien Grot, Micael Guignard, Marion Gunn, Hans Hagen, Boumediene Hamzi, Patrick Happel, Yannis Haralambous, Alexander Harin, Russel L. Harris, Danny Heap, André Heider, Thorsten Hein, Sten Hellman, Haavard Helstrup, Yvon Henel, Stephan Hennig, Florence Henry, Stephen Herborn, Morten Høgholm, Alv Kjetil Holme, Don Hosek, Yufan Hu, Dave W. Huseby, Dmitry Ivanov, Per Steinar Iversen, Zunbeltz Izaola, Youssef Jabri, Bernd Jaehne, Radwan Jalam, Michael Janich, Regnor Jernsletten, Zhuhan Jiang, Loïc Joly, David M. Jones, Christophe Jorssen, Robert Juhasz, Stefan Junge, Dan Jurafsky, Mikko Kanerva, Theppitak Karoonboonyanan, David Kastrup, Ronan Keryell, Axel Kielhorn, Ki-Joo Kim, Bil Kleb, Peter Kleiweg, Rune Kleveland, Jörg Knappen, Donald Knuth, Markus Kohm, Helmut Kopka, Attila Koppanyi, Adamantios Korais, Vincent Krakoviack, Thankmar Kronzucker, Alexej M. Kryokov, Markus G. Kuhn, Florian Kulzer, Toshiki Kumazawa, Frank Küster, Stéphane Laborde, Klaus Lagally, Leslie Lamport, Fabio Lanari, Robert Lange, Dag Langmyhr,

[1] And I apologize to all whose I forgot.
[2] Mainly, `fr.comp.text.tex` (in french) and `comp.text.tex` (in english), but also `de.comp.text.tex` (in german, but I do not read it well: send me also a mail in french or english).

Olga G. Lapko, Henning Larsen, Jean-Marc Lasgouttes, Jean-Philippe Lauffenburger, Arnaud Launay, Claire Lauvernet, Boris Lavva, André Lebacq, Werner Lemberg, Thomas Leonhardt, Erwan Le Pennec, Stéphane Lepolozec, Julien Le Thuaut, Adam Lewenberg, Knut Lickert, Ulf A. Lindgren, Anselm Lingnau, Pierre Lobel, Milos Lokajicek, Maurizio Loreti, Tristan Lorino, Tim Love, Vincent Lozano, Dan Luecking, Ken Lunde, Anders Lyhne, Pierre Mackay, Lars Madsen, Richard Mahoney, Irina A. Makhovaya, Bob Margolis, Nicolas Markey, Marcus Marr, Françoise Marre-Fournier, Terry Mart, Vadim Maslov, Henri Massias, Andreas Matthias, Sven Mattisson, Rowland McDonnell, Ben McKay, Surapant Meknavin, Sébastien Mengin, Yanick Michou, Frank Mittelbach, Jens-Uwe Morawski, Lapo Filippo Mori, Michael A. Morrison, Dejan Muhamedagić, Sergei O. Naumov, Frank Neukam, Cuong Nguyen, Julien Nicolas, Rolf Nieprasch, Josselin Noirel, Heiko Oberdiek, Mariusz Olko, Piet van Oostrum, Erik Östhols, Jörg Ott, Alan Paić, Scott Pakin, Palash Baran Pal, Anshuman Pandey, Oren Patashnik, Matthias Pelger, Terje Engeset Petterst, Bruno Piguet, John Plaice, Veerathanabutr Poonlap, Nico Poppelier, Rama Porrat, Camille-Aimé Possamaï, Roozbeh Pournader, CV Radhakrishnan, Sebastian Rahtz, Bernd Raichle, Jose Pedro Ramalhete, Keranen Reino, Adrian Rezuş, A.J. "Tony" Roberts Denis B. Roegel, Christian Rolland, Marti Ruiz-Altaba, Jan Michel Rynning, Young Ryu, Enn Saar, David Samsoen, Julio Sánchez, Morgan Sangeux, Eddie Saudrais, Elmar Schalueck, Tobias Schlemmer, Uwe Schneider, Martin Schröder, Jungshik Shin, Ali Shoukat, Andrij M. Shvaika, Chanop Silpa-Anan, Jankovic Slobodan, John Smith, Robin S. Socha, Axel Sommerfeldt, Arjen Steiner, D.P. Story, Éric Streit, Ralf Stubner, Chris Swoyer, Apostolos Syropoulos, Dung Ta Quang, Daniel Taupin†, Hàn Thế Thành, Aurélien Thureau, Karsten Tinnefeld, Ton 't Lam, Sigitas Tolusis, Laurent Tordella, Mark Trettin, Ahto Truu, Stefan Ulrich, Horst Umstatter, Koaunghi Un, Turgut Uyar, Jari Vaario, Thomas van Oudenhove de Saint Géry, Vincent Vaquin, Didier Verna, Sylvain Vesco, Alexandre Vial, Vladimir Volovich, Stephan P. von Bechtolsheim, Herbert Voss, Zdeněk Wagner, Nigel Ward, Staszek Wawrykiewicz, Stephan B. Webanck, Eduard Werner, Peter R. Wilson, Marcin Woliński, Élisabeth Wolkowski, Tim Wright, Dominik Wujastyk, Damien Wyart, C.S. Yogananda, Adi Zaimi, Lejzer Ludwig Zamenof, Danilo Zavrtanik, Krzysztof Konrad Żelechowski, Charlie S. Zender, Vadim V. Zhytnikov, Uwe Ziegenhagen, and Leon Žlajpah.

Part II

Implementation

Contents of the Second Part

Chapter 9

Commented code of `minitoc.sty`

`minitoc.sty`

Contents

9.1 Introduction

This long chapter presents the code of the minitoc package and attempts to explain it. Some comments of the original source file [1] 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.

[1] 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.

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.

9.2 Identification code

The code of minitoc.sty starts here:

```
⟨*minitoc⟩
```

\NeedsTeXFormat
\ProvidesPackage
\mtcPackageInfo
\RequirePackage

This code section identifies the package with its name, version number and date. A trace is written in the *document*.log file. This package will not work with LaTeX2.09.

```
\NeedsTeXFormat{LaTeX2e}[1996/06/02]%
\ProvidesPackage{minitoc}%
   [2007/01/09 v51 Package minitoc (JPFD)]
\RequirePackage{mtcmess}[2006/03/14]
\mtcPackageInfo[<I0001>]{minitoc}%
   {*** minitoc package, version 51 ***\@gobble}
```

9.3 A file descriptor to write

\tf@mtc
\newwrite

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

```
\newwrite\tf@mtc
```

9.4 Indentation and skip

\mtcindent
\mtcskip
\mtcskipamount
\parskip

We define the indentation \mtcindent (both sides) of the mini-tables and the command \mtcskip to make a vertical skip before a mini-table, its value is \mtcskipamount (default: \bigskipamount).

```
\newlength\mtcindent
\newskip\mtcskipamount
\setlength{\mtcskipamount}{\bigskipamount}
% %%%%\def\mtcskip{\leavevmode\unskip\removelastskip%
% %%%%       {\parskip=\z@\addvspace{\mtcskipamount}}}
\def\mtcskip{{\parskip=\z@\addvspace{\mtcskipamount}}}
```

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

\mtcgapbeforeheads
\mtcgapafterheads

We define the default values for the vertical gaps before and after titles part level mini-tables.

```
\def\mtcgapbeforeheads{50\p@}
\def\mtcgapafterheads{40\p@}
```

\@ifundefined
\kernafterparttoc
\kernafterpartlof
\kernafterpartlot
\kernaftersecttoc
\kernaftersectlof
\kernaftersectlot
\kernafterminitoc
\kernafterminilof
\kernafterminilot

We define the vertical kernings between the minitables and their before the bottom rule. The names of these macros is rather explicit. The values are empirical and can be changed via \renewcommand.

```
   \@ifundefined{part}{}{%
      \def\kernafterparttoc{\kern-1.\baselineskip\kern.5ex}%
      \def\kernafterpartlof{\kern-1.\baselineskip\kern.5ex}%
      \def\kernafterpartlot{\kern-1.\baselineskip\kern.5ex}%
                        }%
   \@ifundefined{chapter}{%
      \@ifundefined{section}{}%
                            {%
         \def\kernaftersecttoc{\kern-1.\baselineskip\kern.5ex}%
         \def\kernaftersectlof{\kern-1.\baselineskip\kern.5ex}%
         \def\kernaftersectlot{\kern-1.\baselineskip\kern.5ex}%
                            }%
                         }%
                         {%
         \def\kernafterminitoc{\kern-.5\baselineskip\kern.5ex}%
         \def\kernafterminilof{\kern-1.\baselineskip\kern0.ex}%
         \def\kernafterminilot{\kern-1.\baselineskip\kern0.ex}%
                         }%
```

9.5 Tests and flags

We need to declare some flags [2] (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 9.78 on page 382) or to allow the definition of some minitoc commands).

\if@mtc@hints@
\if@mtc@hints@w@
\if@mtc@hints@giben@

9.5.1 Flags for the hints option

But first, we define some flags for the hints option:

- The flag \if@mtc@hints@ is true if the hints option is required (default).

[2] 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 TEX engine (much more with ε-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.

- The flag \if@mtc@hints@w@ is set true if we detect that some sectionning 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 *document*.log file. It will be used at the end of the document to signal that you should look for hints in the *document*.log file.

```
\newif\if@mtc@hints@ \@mtc@hints@true
\newif\if@mtc@hints@w@ \@mtc@hints@w@false
\newif\if@mtc@hints@given@ \@mtc@hints@given@false
```

9.5.2 Use of section-level mini-lists of floats

\if@mtc@sect@floats@
\dosectlof
\dosectlot

We will check if the commands \dosectlof and \dosectlot are used:

```
\newif\if@mtc@sect@floats@ \@mtc@sect@floats@false
```

9.5.3 Presence of some packages and classes

\if@mtc@placeinsLoaded@
\if@mtc@memoirLoaded@
\if@mtc@memoirnew@
\if@mtc@sectstyLoaded@
\if@mtc@sectstyLoaded@a@
\if@mtc@captionLoaded@
\if@mtc@captionLoaded@a@
\if@mtc@captionIILoaded@
\if@mtc@captionIILoaded@a@
\if@mtc@ccaptionLoaded@
\if@mtc@ccaptionLoaded@a@
\if@mtc@mcaptionLoaded@
\if@mtc@mcaptionLoaded@a@
\if@mtc@varsectsLoaded@
\if@mtc@varsectsLoaded@a@
\if@mtc@fncychapLoaded@
\if@mtc@fncychapLoaded@a@
\if@mtc@HgcLoaded@
\if@mtc@HgcLoaded@a@
\if@mtc@quotchapLoaded@
\if@mtc@quotchapLoaded@a@
\if@mtc@romannumLoaded@
\if@mtc@romannumLoaded@a@
\if@mtc@sfheadersLoaded@
\if@mtc@sfheadersLoaded@a@
\if@mtc@alnumsecLoaded@
\if@mtc@alnumsecLoaded@a@
\if@mtc@captcontLoaded@
\if@mtc@captcontLoaded@a@

We will check if the placeins package is loaded, then if the memoir is loaded (and if it is an enough recent version), then if the sectsty package is loaded (before or after minitoc). We do the same for some caption-related packages and the varsects package and other packages altering the sectionning commands.

```
\newif\if@mtc@placeinsLoaded@ \@mtc@placeinsLoaded@false
\newif\if@mtc@memoirLoaded@ \@mtc@memoirLoaded@false
\newif\if@mtc@memoirnew@ \@mtc@memoirnew@false
\newif\if@mtc@sectstyLoaded@ \@mtc@sectstyLoaded@false
\newif\if@mtc@sectstyLoaded@a@ \@mtc@sectstyLoaded@a@false
\newif\if@mtc@captionLoaded@ \@mtc@captionLoaded@false
\newif\if@mtc@captionLoaded@a@ \@mtc@captionLoaded@a@false
\newif\if@mtc@captionIILoaded@ \@mtc@captionIILoaded@false
\newif\if@mtc@captionIILoaded@a@ \@mtc@captionIILoaded@a@false
\newif\if@mtc@ccaptionLoaded@ \@mtc@ccaptionLoaded@false
\newif\if@mtc@ccaptionLoaded@a@ \@mtc@ccaptionLoaded@a@false
\newif\if@mtc@mcaptionLoaded@ \@mtc@mcaptionLoaded@false
\newif\if@mtc@mcaptionLoaded@a@ \@mtc@mcaptionLoaded@a@false
\newif\if@mtc@varsectsLoaded@ \@mtc@varsectsLoaded@false
\newif\if@mtc@varsectsLoaded@a@ \@mtc@varsectsLoaded@a@false
\newif\if@mtc@fncychapLoaded@ \@mtc@fncychapLoaded@false
\newif\if@mtc@fncychapLoaded@a@ \@mtc@fncychapLoaded@a@false
\newif\if@mtc@HgcLoaded@ \@mtc@HgcLoaded@false
\newif\if@mtc@HgcLoaded@a@ \@mtc@HgcLoaded@a@false
\newif\if@mtc@quotchapLoaded@ \@mtc@quotchapLoaded@false
\newif\if@mtc@quotchapLoaded@a@ \@mtc@quotchapLoaded@a@false
\newif\if@mtc@romannumLoaded@ \@mtc@romannumLoaded@false
\newif\if@mtc@romannumLoaded@a@ \@mtc@romannumLoaded@a@false
```

```
\newif\if@mtc@sfheadersLoaded@ \@mtc@sfheadersLoaded@false
\newif\if@mtc@sfheadersLoaded@a@ \@mtc@sfheadersLoaded@a@false
\newif\if@mtc@alnumsecLoaded@ \@mtc@alnumsecLoaded@false
\newif\if@mtc@alnumsecLoaded@a@ \@mtc@alnumsecLoaded@a@false
\newif\if@mtc@captcontLoaded@ \@mtc@captcontLoaded@false
\newif\if@mtc@captcontLoaded@a@ \@mtc@captcontLoaded@a@false
```

`\if@mtc@empty@parttoc@`
`\if@mtc@empty@partlof@`
`\if@mtc@empty@partlot@`
`\if@mtc@empty@minitoc@`
`\if@mtc@empty@minitoc@`
`\if@mtc@empty@minilof@`
`\if@mtc@empty@sectlot@`
`\if@mtc@empty@sectlof@`
`\if@mtc@empty@sectlot@`

We will check if you have attempted to insert some empty mini-tables:

```
\newif\if@mtc@empty@parttoc@ \@mtc@empty@parttoc@false
\newif\if@mtc@empty@partlof@ \@mtc@empty@partlof@false
\newif\if@mtc@empty@partlot@ \@mtc@empty@partlot@false
\newif\if@mtc@empty@minitoc@ \@mtc@empty@minitoc@false
\newif\if@mtc@empty@minilof@ \@mtc@empty@minilof@false
\newif\if@mtc@empty@minilot@ \@mtc@empty@minilot@false
\newif\if@mtc@empty@secttoc@ \@mtc@empty@secttoc@false
\newif\if@mtc@empty@sectlof@ \@mtc@empty@sectlof@false
\newif\if@mtc@empty@sectlot@ \@mtc@empty@sectlot@false
```

9.5.4 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:

```
\newif\if@mtc@part@def@ \@mtc@part@def@false
  \@ifundefined{part}{\@mtc@part@def@false}{\@mtc@part@def@true}
```

`\if@mtc@chapter@def@`

The `chapter` counter:

```
\newif\if@mtc@chapter@def@ \@mtc@chapter@def@false
  \@ifundefined{chapter}{\@mtc@chapter@def@false}{\@mtc@chapter@def@true}
```

`\if@mtc@section@def@`

The `section` counter:

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

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

\if@mtc@chapter@undef@

The chapter counter:

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

\if@mtc@section@undef@

The section counter:

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

9.5.5 Flags to check 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 9.78 on page 382).

\if@parttoc@used@
\if@partlof@used@
\if@partlot@used@
\if@doparttoc@used@
\if@dopartlof@used@
\if@dopartlot@used@

For the part level:

```
\newif\if@parttoc@used@ \global\@parttoc@used@false
\newif\if@partlof@used@ \global\@partlof@used@false
\newif\if@partlot@used@ \global\@partlot@used@false
\newif\if@doparttoc@used@ \global\@doparttoc@used@false
\newif\if@dopartlof@used@ \global\@dopartlof@used@false
\newif\if@dopartlot@used@ \global\@dopartlot@used@false
```

\if@minitoc@used@
\if@minilof@used@
\if@minilot@used@
\if@dominitoc@used@
\if@dominilof@used@
\if@dominilot@used@

For the chapter level:

```
\newif\if@minitoc@used@ \global\@minitoc@used@false
\newif\if@minilof@used@ \global\@minilof@used@false
\newif\if@minilot@used@ \global\@minilot@used@false
\newif\if@dominitoc@used@ \global\@dominitoc@used@false
\newif\if@dominilof@used@ \global\@dominilof@used@false
\newif\if@dominilot@used@ \global\@dominilot@used@false
```

\if@secttoc@used@
\if@sectlof@used@
\if@sectlot@used@
\if@dosecttoc@used@
\if@dosectlof@used@
\if@dosectlot@used@

For the section level:

```
\newif\if@secttoc@used@ \global\@secttoc@used@false
\newif\if@sectlof@used@ \global\@sectlof@used@false
\newif\if@sectlot@used@ \global\@sectlot@used@false
\newif\if@dosecttoc@used@ \global\@dosecttoc@used@false
\newif\if@dosectlof@used@ \global\@dosectlof@used@false
\newif\if@dosectlot@used@ \global\@dosectlot@used@false
```

\if@firstpartis@used@
\if@firstchapteris@used@
\if@firstsectionis@used@

We also detect the use of some obsolete commands:

```
\newif\if@firstpartis@used@ \global\@firstpartis@used@false
\newif\if@firstchapteris@used@ \global\@firstchapteris@used@false
\newif\if@firstsectionis@used@ \global\@firstsectionis@used@false
```

9.5.6 Check if the document has exactly 2 parts

\ifmtcsecondpart

In french, the ordinal adjective is “deuxième” if the second object is not the last object, but “second” (masculine) or “seconde” (feminine) when it is also the last one. So we define a specific flag:

```
\newif\ifmtcsecondpart \mtcsecondpartfalse
```

\AtBeginDocument
\ifmtcsecondpart

At the beginning of the document, we test this flag and make it global:

```
\AtBeginDocument{%
  \ifmtcsecondpart
    \global\mtcsecondparttrue
  \else
    \global\mtcsecondpartfalse
  \fi}
```

\AtEndDocument
\ifmtcsecondpart
\\@mainaux

At the end of the document, we set and memorize the value of the flag in the `.aux` file:

```
\AtEndDocument{%
\ifnum\value{part}=2\relax
  \mtcsecondparttrue
\else
  \mtcsecondpartfalse
\fi
\if@filesw
  \ifmtcsecondpart
    \immediate\write\@mainaux
      {\string\global\string\mtcsecondparttrue}%
```

```
  \else
    \immediate\write\@mainaux
      {\string\global\string\mtcsecondpartfalse}%
  \fi
\fi}
```

So we need two LaTeX runs to get a correct result. The `french2.mld` language definition file (see section 13.60 on page 463) uses this trick to form the titles of part level mini-tables. See the `mtc-2nd.tex` example file in section 4.2 on page 88.

9.6 Preparation for the `notoccite` option

`\mtc@hook@beforeinputfile`
`\if@mtc@notoccite@`

We declare a flag for the presence of this option and the new internal "hook" command (redefinable command) `\mtc@hook@beforeinputfile`, used by this option (this has been requested by Donald Arseneau for his notoccite package [9]). See section 1.6 on page 49.

```
\newif\if@mtc@notoccite@ \@mtc@notoccite@false
\@ifundefined{mtc@hook@beforeinputfile}%
   {\let\mtc@hook@beforeinputfile\relax}{}
```

9.7 Preparation for the `tight` and `k-tight` options

`\iftightmtc`
`\ifktightmtc`

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

```
\newif\iftightmtc \tightmtcfalse
\newif\ifktightmtc \ktightmtcfalse
```

9.8 Preparation to work with hyperref

`\AtBeginDocument`
`\if@mtc@hyper@used@`
`\@ifpackageloaded`

This code prepares the interface with the hyperref package [214]. A flag is defined, then this preparation is performed in a `\AtBeginDocument` block if this package is loaded. This action defines some commands for the hyperref package.

```
\mtcPackageInfo[<I0005>]{minitoc}{compatible with hyperref\@gobble}
\newif\if@mtc@hyper@used@ \global\@mtc@hyper@used@false
\AtBeginDocument{%
  \@ifpackageloaded{hyperref}{%
    \global\@mtc@hyper@used@true
    \def\toclevel@xpart{1000}%
```

```
  \def\toclevel@xchapter{1000}%
  \def\toclevel@xsect{1000}%
  \def\toclevel@xsection{1000}%
  \let\toclevel@starpart\toclevel@part
  \let\toclevel@starchapter\toclevel@chapter
  \let\toclevel@starsection\toclevel@section
  \let\toclevel@starsubsection\toclevel@subsection
  \let\toclevel@starsubsubsection\toclevel@subsubsection
  \let\toclevel@starparagraph\toclevel@paragraph
  \let\toclevel@starsubparagraph\toclevel@subparagraph
}{}}
```

9.9 Checking the presence of some packages

9.9.1 Check if the sectsty package is loaded, and when

\AtBeginDocument
\if@mtc@sectstyLoaded@
\if@mtc@sectstyLoaded@a@
\@ifpackageloaded

We must test if the sectsty package [182] is loaded before or after minitoc, so we test when minitoc is loaded and also in a \AtBeginDocument block, when all packages have been loaded. See section 9.78.2.6 on page 399.

```
\@ifpackageloaded{sectsty}{\@mtc@sectstyLoaded@true}{}
\AtBeginDocument{\@ifpackageloaded{sectsty}{\@mtc@sectstyLoaded@a@true}{}}
```

9.9.2 Check if the varsects package is loaded, and when

\@ifpackageloaded
\AtBeginDocument
\if@mtc@varsectsLoaded@
\if@mtc@varsectsLoaded@a@

We must test if the varsects package [228] is loaded before or after minitoc, so we test when minitoc is loaded and also in a \AtBeginDocument block, when all packages have been loaded. See section 9.78.2.7 on page 399.

```
\@ifpackageloaded{varsects}{\@mtc@varsectsLoaded@true}{}
\AtBeginDocument{\@ifpackageloaded{varsects}{\@mtc@varsectsLoaded@a@true}{}}
```

9.9.3 Check if the fncychap package is loaded, and when

\@ifpackageloaded
\AtBeginDocument
\if@mtc@fncychapLoaded@
\if@mtc@fncychapLoaded@a@

We must test if the fncychap package [170] is loaded before or after minitoc, so we test when minitoc is loaded and also in a \AtBeginDocument block, when all packages have been loaded. See section 9.78.2.8 on page 399.

```
\@ifpackageloaded{fncychap}{\@mtc@fncychapLoaded@true}{}
\AtBeginDocument{\@ifpackageloaded{fncychap}{\@mtc@fncychapLoaded@a@true}{}}
```

9.9.4 Check if the hangcaption package is loaded, and when

\@ifpackageloaded
\AtBeginDocument
\if@mtc@HgcLoaded@
\if@mtc@HgcLoaded@a@

We must test if the hangcaption package [138] is loaded before or after minitoc, so we test when minitoc is loaded and also in a \AtBeginDocument block, when all packages have been loaded. See section 9.78.2.9 on page 400.

```
\@ifpackageloaded{hangcaption}{\@mtc@HgcLoaded@true}{}
\AtBeginDocument{\@ifpackageloaded{hangcaption}{\@mtc@HgcLoaded@a@true}{}}
```

9.9.5 Check if the quotchap package is loaded, and when

\@ifpackageloaded
\AtBeginDocument
\if@mtc@quotchapLoaded@
\if@mtc@quotchapLoaded@a@

We must test if the quotchap package [232] is loaded before or after minitoc, so we test when minitoc is loaded and also in a \AtBeginDocument block, when all packages have been loaded. See section 9.78.2.10 on page 400.

```
\@ifpackageloaded{quotchap}{\@mtc@quotchapLoaded@true}{}
\AtBeginDocument{\@ifpackageloaded{quotchap}{\@mtc@quotchapLoaded@a@true}{}}
```

9.9.6 Check if the romannum package is loaded, and when

\@ifpackageloaded
\AtBeginDocument
\if@mtc@romannumLoaded@
\if@mtc@romannumLoaded@a@

We must test if the romannum package [259] is loaded before or after minitoc, so we test when minitoc is loaded and also in a \AtBeginDocument block, when all packages have been loaded. See section 9.78.2.11 on page 400.

```
\@ifpackageloaded{romannum}{\@mtc@romannumLoaded@true}{}
\AtBeginDocument{\@ifpackageloaded{romannum}{\@mtc@romannumLoaded@a@true}{}}
```

9.9.7 Check if the sfheaders package is loaded, and when

\@ifpackageloaded
\AtBeginDocument
\if@mtc@sfheadersLoaded@
\if@mtc@sfheadersLoaded@a@

We must test if the sfheaders package [172] is loaded before or after minitoc, so we test when minitoc is loaded and also in a \AtBeginDocument block, when all packages have been loaded. See section 9.78.2.12 on page 401.

```
\@ifpackageloaded{sfheaders}{\@mtc@sfheadersLoaded@true}{}
\AtBeginDocument{\@ifpackageloaded{sfheaders}{\@mtc@sfheadersLoaded@a@true}{}}
```

9.9.8 Check if the alnumsec package is loaded, and when

\@ifpackageloaded
\AtBeginDocument
\if@mtc@alnumsecLoaded@
\if@mtc@alnumsecLoaded@a@

We must test if the alnumsec package [152] is loaded before or after minitoc, so we test when minitoc is loaded and also in a \AtBeginDocument block, when all packages have been loaded. See section 9.78.2.13 on page 401.

```
\@ifpackageloaded{alnumsec}{\@mtc@alnumsecLoaded@true}{}
\AtBeginDocument{\@ifpackageloaded{alnumsec}{\@mtc@alnumsecLoaded@a@true}{}}
```

9.9.9 Check if the captcont package is loaded, and when

\@ifpackageloaded
\AtBeginDocument
\if@mtc@captcontLoaded@
\if@mtc@captcontLoaded@a@

We must test if the captcont package [95] is loaded before or after minitoc, so we test when minitoc is loaded and also in a \AtBeginDocument block, when all packages have been loaded. See section 9.78.2.14 on page 401.

```
\@ifpackageloaded{captcont}{\@mtc@captcontLoaded@true}{}
\AtBeginDocument{\@ifpackageloaded{captcont}{\@mtc@captcontLoaded@a@true}{}}
```

9.9.10 Check if the caption package is loaded, and when

\@ifpackageloaded
\AtBeginDocument
\if@mtc@captionLoaded@
\if@mtc@captionLoaded@a@

We must test if the caption package [224] is loaded before or after minitoc, so we test when minitoc is loaded and also in a \AtBeginDocument block, when all packages have been loaded. See section 9.78.2.15 on page 402.

```
\@ifpackageloaded{caption}{\@mtc@captionLoaded@true}{}
\AtBeginDocument{\@ifpackageloaded{caption}{\@mtc@captionLoaded@a@true}{}}
```

9.9.11 Check if the caption2 package is loaded, and when

\@ifpackageloaded
\AtBeginDocument
\if@mtc@captionIILoaded@
\if@mtc@captionIILoaded@a@

We must test if the caption2 package [223] is loaded before or after minitoc, so we test when minitoc is loaded and also in a \AtBeginDocument block, when all packages have been loaded. See section 9.78.2.16 on page 402.

```
\@ifpackageloaded{caption2}{\@mtc@captionIILoaded@true}{}
\AtBeginDocument{\@ifpackageloaded{caption2}{\@mtc@captionIILoaded@a@true}{}}
```

9.9.12 Check if the ccaption package is loaded, and when

\@ifpackageloaded
\AtBeginDocument
\if@mtc@ccaptionLoaded@
\if@mtc@ccaptionLoaded@a@

We must test if the ccaption package [255] is loaded before or after minitoc, so we test when minitoc is loaded and also in a \AtBeginDocument block, when all packages have been loaded. See section 9.78.2.17 on page 402.

```
\@ifpackageloaded{ccaption}{\@mtc@ccaptionLoaded@true}{}
\AtBeginDocument{\@ifpackageloaded{ccaption}{\@mtc@ccaptionLoaded@a@true}{}}
```

9.9.13 Check if the mcaption package is loaded, and when

\@ifpackageloaded
\AtBeginDocument
\if@mtc@mcaptionLoaded@
\if@mtc@mcaptionLoaded@a@

We must test if the mcaption package [131] is loaded before or after minitoc, so we test when minitoc is loaded and also in a \AtBeginDocument block, when all packages have been loaded. See section 9.78.2.18 on page 403.

```
\@ifpackageloaded{mcaption}{\@mtc@mcaptionLoaded@true}{}
\AtBeginDocument{\@ifpackageloaded{mcaption}{\@mtc@mcaptionLoaded@a@true}{}}
```

9.10 Is the memoir class loaded?

\@ifclassloaded
\if@mtc@memoirLoaded@
\if@mtc@memoirnew@
\if@mtcpatchmemoir@

We test if the memoir [257, 258] class is loaded. This class needs some compatibility adjustments or may be incompatible if too recent. In the later case, a patch is inserted (see chapter 12 on page 428). This correction is no more necessary after the 2005/09/25 version of memoir.cls.

```
\newif\if@mtcpatchmemoir@ \@mtcpatchmemoir@false
\@ifclassloaded{memoir}{\@mtc@memoirLoaded@true\relax%
                        \mtcPackageInfo[<I0030>]{minitoc}%
                        {the memoir class is loaded:
                         \MessageBreak
                         compatibility attempted\@gobble}}%
                       {\@mtc@memoirLoaded@false}
\if@mtc@memoirLoaded@
 \@ifundefined{@m@mchapter}%
   {\@mtc@memoirnew@false\mtcPackageInfo[<I0020>]{minitoc}%
      {old version of the memoir class\@gobble}}
   {\@mtc@memoirnew@true\mtcPackageInfo[<I0027>]{minitoc}%
      {recent version of the memoir class\@gobble}
      \mtcPackageInfo[<I0032>]{minitoc}%
         {This version of the memoir class uses
          \MessageBreak
          a version of \string\chapter\space which is
          \MessageBreak
          incompatible with the minitoc package.
```

```
          \MessageBreak
          We try to patch\@gobble}%
   \@mtcpatchmemoir@true}
\fi
```

\if@mtcpatchmemoir@

And now the patch:

```
\if@mtcpatchmemoir@
\IfFileExists{mtcpatchmem.sty}{%
   \@ifclasslater{memoir}{2005/09/25}{}{\RequirePackage{mtcpatchmem}}}{%
   \mtcPackageError[<E0028>]{minitoc}%
     {Unable to patch the memoir class}%
     {So it remains incompatible. Sorry.}}
\fi
```

9.11 Testing the emptiness of a file

\mtc@ifmtarg
\mtc@xifmtarg
\mtc@EndWhile
\mtc@WhilePreCondition
\mtc@WhileCondition
\mtc@WhileBody
\mtc@While
\mtc@WhileNext

Some macros for testing if an argument of a macro is empty (taken from the package ifmtarg [260], by Peter R. Wilson and Donald Arseneau, and from while.tip, by Stephan P. von Bechtolsheim [246]). The group is necessary to keep local the catcode change of "Q", hence a \gdef is needed for \mtc@ifmtarg.

```
\begingroup
\catcode`\Q=3
\long\gdef\mtc@ifmtarg#1{%
\mtc@xifmtarg#1QQ\@secondoftwo\@firstoftwo\@nil}
\long\gdef\mtc@xifmtarg#1#2Q#3#4#5\@nil{#4}
\endgroup
\let\mtc@EndWhile = \fi
\def\mtc@While #1#2#3\mtc@EndWhile{%
   \def\mtc@WhilePreCondition{#1}%
   \def\mtc@WhileCondition{#2}%
   \def\mtc@WhileBody{#3}%
   \mtc@@While
}
\def\mtc@@While{%
   \mtc@WhilePreCondition
   \mtc@WhileCondition
      \def\mtc@WhileNext{%
          \mtc@WhileBody
          \mtc@@While
      }%
   \else
      \def\mtc@WhileNext{}%
   \fi
   \mtc@WhileNext
}
```

\if@mtc@checkfiles
\if@mtc@FE
\if@mtc@LI
\mtc@While
\mtc@Body
\mtc@EndWhile
\mtc@CkFile
\mtc@CkStr
\mtc@Rline
\tf@mtc
\@inputcheck

Some macros to test if a file is empty or not: `\mtc@CkFile{file}` returns `\@mtc@FEtrue` if the file is empty, `\@mtc@FEfalse` if the is file not empty. An inexistent file is empty. A file full of white space (space, tabulation, newline) is empty. Comments are empty.

Note: on a big empty file, the `\mtc@While` loop may be time consuming, but not an eternity (33 s for 10^6 lines on my computer), and the first non-empty line stops the loop. `\jobname.mtc` is used as scratch file. Its contents is erased after use.

```
\newif\if@mtc@LI\@mtc@LItrue
\newif\if@mtc@FE\@mtc@FEtrue
\newif\if@mtc@checkfiles\@mtc@checkfilestrue
\def\mtc@Body{\immediate\read\@inputcheck to
                                    \mtc@Rline\relax
  \ifeof\@inputcheck\relax\@mtc@LIfalse\fi
  \expandafter\ifx\mtc@Rline\par\relax
     \def\mtc@Rline{}
  \else
     \ifeof\@inputcheck\relax\global\@mtc@LIfalse\fi
     \mtc@ifmtarg{\mtc@Rline}{\relax}%
              {\@mtc@FEfalse\@mtc@LIfalse}
  \fi
}
\def\mtc@CkFile#1{%
\@mtc@LItrue\@mtc@FEtrue
\if@mtc@checkfiles
\IfFileExists{#1}{%
\immediate\openin\@inputcheck #1\relax
\mtc@While{}{\if@mtc@LI\relax}%
   {\mtc@Body}%
\mtc@EndWhile}%
{\@mtc@FEtrue}%
\else
\@mtc@FEfalse%
\fi}
\closein\@inputcheck\relax
\def\mtc@CkStr#1{%
\immediate\openout\tf@mtc \jobname.mtc
\immediate\write\tf@mtc{#1}%
\immediate\closeout\tf@mtc
\mtc@CkFile{\jobname.mtc}%
\immediate\openout\tf@mtc \jobname.mtc
\immediate\closeout\tf@mtc}
```

9.12 Internal macros to decrement minitoc counters

\mtc@onebackpart
\mtc@onebackchapter
\mtc@onebacksection
\addtocounter

It is sometimes necessary to decrement a minitoc counter (`ptc`, `mtc` or `stc`) by one. These macros are:

```
\def\mtc@onebackpart{\addtocounter{ptc}{-1}}
```

```
\def\mtc@onebackchap{\addtocounter{mtc}{-1}}
\def\mtc@onebacksect{\addtocounter{stc}{-1}}
```

9.13 Patching the \part command

\part
\mtc@svspart
\mtc@svpart
\@spart
\@part
\stepcounter

If the \part command is not defined (by the document class, usually), we cannot patch it and a warning is displayed[3]. Else, we patch its two branches, \@part (for the unstarred version) or \@spart (for the starred version, \part*): we add \stepcounter{ptc} to increment the parttoc counter ptc. See also section 9.51 on page 310.

The code of the next section (section 9.14) is also skipped if \part is not defined.

```
\@ifundefined{part}{%
  \mtcPackageWarningNoLine[<W0018>]{minitoc}%
     {part level macros NOT available}
}{%% else undefined part (\part defined)
  \mtcPackageInfo[<I0023>]{minitoc}%
     {part level macros available\@gobble}
  \let\mtc@svspart\@spart
  \def\@spart{\stepcounter{ptc}\mtc@svspart}
  \let\mtc@svpart\@part
  \def\@part{\stepcounter{ptc}\mtc@svpart}
```

9.14 Adding an entry in the TOC for a starred part

\mtcaddpart
\mtc@ifmtarg
\contentsline
\addcontentsline
\adjustptc
\l@xpart
\l@part

To add an entry in the TOC for a starred part, we need the \mtcaddpart macro, which has an optional argument, the title of the part as if should appear in the TOC.

By default, this argument is empty. If it is empty (tested via \mtc@ifmtarg) or omitted, we add a \contentsline{xpart}{}... line in the .toc file. If it is not empty, we add a \contentsline{part}{title...}... line in the .toc file. We always add a \contentsline{xpart}{}... line in the .lof and .lot files. Then we increment the ptc counter, via \adjustptc (defined in section 9.45 on page 297). 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 (10 000) for this entry, hence it will never be really printed (except if you cheat).

```
\newcommand{\mtcaddpart}[1][]{%
    \mtc@ifmtarg{#1}{\addcontentsline{toc}{xpart}{}}%
                     {\addcontentsline{toc}{part}{#1}}%
    \addcontentsline{lof}{xpart}{}%
    \addcontentsline{lot}{xpart}{}%
```

[3] Document classes with sectionning commands but no \part command are likely non standard, hence the warning displayed on the terminal.

```
    \adjustptc}
```

This code terminates (temporarily) the part level commands.

```
}%
```

9.15 Section level macros

\chapter
\section

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. 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.

```
\@ifundefined{chapter}{\mtcPackageInfo[<I0004>]{minitoc}%
          {chapter level macros NOT available\@gobble}%
  \@ifundefined{section}{\mtcPackageInfo[<I0029>]{minitoc}%
          {section level macros NOT available\@gobble}%
    \mtcPackageWarningNoLine[<W0017>]{minitoc}%
       {no section or chapter level macros available
        \MessageBreak
        PLEASE VERIFY YOUR MAIN DOCUMENT CLASS}}%
  {\mtcPackageInfo[<I0028>]{minitoc}%
       {section level macros available\@gobble}%
```

9.16 Corrections for numbering

\mtc@onebacksect
\tableofcontents
\listoffigures
\listoftables
\mtcsv@tableofcontents
\mtcsv@listoffigures
\mtcsv@listoftables

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

```
    \let\mtcsv@tableofcontents\tableofcontents
    \let\mtcsv@listoffigures\listoffigures
    \let\mtcsv@listoftables\listoftables
    \def\tableofcontents{\mtcsv@tableofcontents\mtc@onebacksect}
    \def\listoffigures{\mtcsv@listoffigures\mtc@onebacksect}
    \def\listoftables{\mtcsv@listoftables\mtc@onebacksect}
```

9.17 Patching the \section command

\mtc@svsection
\mtc@svss
\@ssect
\@sect
\section
\stepcounter

If the \section command is not defined (by the document class, usually), we cannot patch it and a warning is displayed. Else, we patch its two branches, \@sect (for the unstarred version)

or \@ssect (for the starred version, \section*): we add \stepcounter{stc} to increment the secttoc counter stc, only in the unstarred case (the version #25 has removed a spurious decrementation of this counter).

```
    \let\mtc@svsection\section
    \def\section{\stepcounter{stc}\mtc@svsection}
    \let\mtc@svss\@ssect
```

9.18 Adding an entry in the TOC for a starred section

\mtcaddsection
\mtc@ifmtarg
\contentsline
\adjuststc
\l@xsect
\l@xsection
\l@section

To add an entry in the TOC for a starred section, we need the \mtcaddsection macro, which has an optional argument, the title of the section as it should appear in the TOC. By default, this argument is empty. If it is empty (tested via \mtc@ifmtarg) or omitted, we add a \contentsline{xsect}{}... line in the .toc file. If it is not empty, we add a \contentsline{section}{title...}... line in the .toc file. We always add a \contentsline{xsect}{}... line in the .lof and .lot files. Then we increment the stc counter, via \adjuststc (this command is defined in section 9.54 on page 322). Using xsect as first argument of \contentsline means that \l@xsect will be invoked in place of \l@section to print the entry in the TOC, but \l@xsect uses a huge depth (10 000) for this entry, hence it will never be really printed (except if you cheat).

```
\newcommand{\mtcaddsection}[1][]{%
    \mtc@ifmtarg{#1}{\addcontentsline{toc}{xsect}{}}%
                    {\addcontentsline{toc}{section}{#1}}%
    \addcontentsline{lof}{xsect}{}%
    \addcontentsline{lot}{xsect}{}%
    \adjuststc}
```

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

```
  }}%
   {%
```

9.19 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}.

```
  \mtcPackageInfo[<I0003>]{minitoc}{chapter level macros available\@gobble}
```

9.20 Patching the \chapter command

\chapter
\@chapter
\mtc@svchapter
\stepcounter

The \chapter command is defined (by the document class, usually). We patch its two branches, \@chapter (for the unstarred version) or \@schapter (for the starred version, \chapter*): we add call to \stepcounter{mtc} to increment the minitoc counter mtc. Only the unstarred branch (\@chapter) is patched here. The other branch is patched later (section 9.36 on page 281).

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

9.21 Adding an entry in the TOC for a starred chapter

\mtcaddchapter
\mtc@ifmtarg
\contentsline
\adjustmtc
\l@xchapter
\l@chapter

To add an entry in the TOC for a starred chapter, we need the \mtcaddchapter macro, which has an optional argument, the title of the chapter as if should appear in the TOC. By default, this argument is empty. If it is empty (tested via \mtc@ifmtarg) or omitted, we add a \contentsline{xchapter}{}... line in the .toc file. If it is not empty, we add a \contentsline{chapter}{title...}... line in the .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 9.31 on page 268). 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 (10 000) for this entry, hence it will never be really printed (except if you cheat).

```
\newcommand{\mtcaddchapter}[1][]{%
    \mtc@ifmtarg{#1}{\addcontentsline{toc}{xchapter}{}}%
                    {\addcontentsline{toc}{chapter}{#1}}%
    \addcontentsline{lof}{xchapter}{}%
    \addcontentsline{lot}{xchapter}{}%
    \adjustmtc}
```

This code terminates (temporarily) the chapter level commands, i.e., terminates the \@ifundefined{chapter} at the beginning of section 9.15 on page 257.

```
}%
```

9.22 Miscellaneous declarations

\newread
\newtoks
\mtc@toks
\mtc@string
\mtc@strut
\mtc@strutbox
\mtc@hstrut
\mtc@hstrutbox
\mtc@v
\mtc@zrule
\mtc@BBR

The \newread command must be redeclared as being \outer (as Donald Arseneau told me). We need a token register (\mtc@toks), a temporary string (\mtc@string), struts (two kinds, each one using a box containing an invisible vertical rule), a rule with all dimensions equal to zero (\mtc@zrule) and a command discouraging page breaks (\mtc@BBR, for “bad break”).

Table 9.1: Hack to detect the limitation to short extensions

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

For the struts, which are boxes containing an invisible vertical rule, we use "ex" units, to follow the current font.

```
\def\newread{\alloc@6\read\chardef\sixt@@n}
\newtoks\mtc@toks
\def\mtc@string{\relax}
\newbox\mtc@strutbox
\setbox\mtc@strutbox=\hbox{\rule[1.8ex]{\z@}{2.5ex}}
\def\mtc@strut{\relax\ifmmode\copy\mtc@strutbox
                      \else\unhcopy\mtc@strutbox\fi}
\newbox\mtc@hstrutbox
\setbox\mtc@hstrutbox=\hbox{\rule[1.ex]{\z@}{1.ex}}
\def\mtc@hstrut{\relax\ifmmode\copy\mtc@hstrutbox
                       \else\unhcopy\mtc@hstrutbox\fi}
\def\mtc@v{\leavevmode\mtc@strut}
\def\mtc@zrule{\rule[\z@]{\z@}{\z@}}
\def\mtc@BBR{\unpenalty\nopagebreak[4]}
```

9.23 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[4]. This code is verbose if long extensions cannot be used, else the messages are only written in the *document*.log file. The sequencing of these operations is vital. The table 9.1 shows this sequence. A star (*) denotes which file is read in phase 3.

[4] 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 *document*.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 *document*.log file, and in the standard contents files, of course), we can reuse the same file descriptor and avoid this serious problem (which was present in the original version of the package). The minitoc package writes in the contents files when it encounters a major sectionning 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.

\if@longextensions@
\tf@mtc
\openout
\write
\closeout
\input
\jobname

(0) First, a message and a new flag:

```
\mtcPackageInfo[<I0002>]{minitoc}%
   {Autoconfiguration of extensions\@gobble}
\newif\if@longextensions@\@longextensions@false
```

(1) We write "\@longextensions@true" in \jobname.mtc1. But if the OS has short extensions, the real name of the file will be truncated to \jobname.mtc.

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

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

```
\immediate\openout\tf@mtc \jobname.mtc
\immediate\write\tf@mtc{\string\@longextensions@false}
\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.

```
\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.

```
\if@longextensions@
 \mtcPackageInfo[<I0012>]{minitoc}%
   {Long extensions (Unix-like) will be used\@gobble}
 \mtcPackageInfo[<I0031>]{minitoc}%
 {==> this version is configured for UNIX-like
  \MessageBreak
  \space\space\space\space(long extensions) file names\@gobble}%
\else
 \mtcPackageWarningNoLine[<W0019>]{minitoc}%
   {Short extensions (MSDOS-like) will be used
    \MessageBreak
    ==> this version is configured for MSDOS-like
    \MessageBreak
    \space\space\space\space(8+3) file names}
\fi
```

(5) We erase the contents of the two files (because \jobname.mtc is also used later as a scratch file, see section 9.11 on page 254).

```
\immediate\openout\tf@mtc \jobname.mtc
\immediate\closeout\tf@mtc
\immediate\openout\tf@mtc \jobname.mtc1
\immediate\closeout\tf@mtc
```

9.24 Detecting obsolete versions of LaTeX

\@inputcheck
\reset@font

This code detects old versions of the LaTeX 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 LaTeX

commands, \@inputcheck and \reset@font. If you get one of these messages, your are in bad luck and should *urgently* update your LaTeX installation, which is rusting since... many years!

```
\@ifundefined{@inputcheck}%
  {\mtcPackageWarningNoLine[<W0021>]{minitoc}%
     {Your version of latex.tex is obsolete.
      \MessageBreak
      Trying to continue..}\newread\@inputcheck\relax}{}
\@ifundefined{reset@font}%
  {\mtcPackageWarningNoLine[<W0022>]{minitoc}%
     {Your version of latex.tex is very obsolete.
      \MessageBreak
      Trying to continue... crossing fingers}%
  \let\reset@font\relax}{}
```

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

\@undottedtocline
\ifundottedmtc
\undottedmtcfalse

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

```
\newif\ifundottedmtc\undottedmtcfalse
\def\@undottedtocline#1#2#3#4#5{%
  \ifnum #1>\c@tocdepth\relax \else
  \vskip \z@ plus.2\p@
  {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
   \parindent #2\relax\@afterindenttrue
   \interlinepenalty\@M
   \leavevmode
   \@tempdima #3\relax \advance\leftskip \@tempdima \hbox{}%
   \hskip -\leftskip
    #4\nobreak\hfill \nobreak
           \null\par}%
  \fi}
```

9.26 Default values for the page-number customizations

\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 [257, 258] is loaded, it does the job. So, we test first \if@mtc@memoirLoaded@ to use the commands of memoir when they are available.

```
\if@mtc@memoirLoaded@
```

\mtcpagenumbers
\nomtcpagenumbers

For entries in minitocs:

```
\def\mtcpagenumbers{%
  \cftpagenumberson{section}
  \cftpagenumberson{subsection}
  \cftpagenumberson{subsubsection}
  \cftpagenumberson{paragraph}
  \cftpagenumberson{subparagraph}}
\def\nomtcpagenumbers{%
  \cftpagenumbersoff{section}
  \cftpagenumbersoff{subsection}
  \cftpagenumbersoff{subsubsection}
  \cftpagenumbersoff{paragraph}
  \cftpagenumbersoff{subparagraph}}
```

\stcpagenumbers
\nostcpagenumbers

For entries in secttocs:

```
\def\stcpagenumbers{%
  \cftpagenumberson{subsection}
  \cftpagenumberson{subsubsection}
  \cftpagenumberson{paragraph}
  \cftpagenumberson{subparagraph}}
\def\nostcpagenumbers{%
  \cftpagenumbersoff{subsection}
  \cftpagenumbersoff{subsubsection}
  \cftpagenumbersoff{paragraph}
  \cftpagenumbersoff{subparagraph}}
```

\ptcpagenumbers
\noptcpagenumbers

For entries in parttocs:

```
\def\ptcpagenumbers{%
  \cftpagenumberson{chapter}
  \cftpagenumberson{section}
  \cftpagenumberson{subsection}
  \cftpagenumberson{subsubsection}
  \cftpagenumberson{paragraph}
  \cftpagenumberson{subparagraph}}
\def\noptcpagenumbers{%
  \cftpagenumbersoff{chapter}
  \cftpagenumbersoff{section}
  \cftpagenumbersoff{subsection}
  \cftpagenumbersoff{subsubsection}
  \cftpagenumbersoff{paragraph}
  \cftpagenumbersoff{subparagraph}}
```

\mlfpagenumbers
\nomlfpagenumbers
\slfpagenumbers
\noslfpagenumbers
\plfpagenumbers
\noplfpagenumbers

For entries in minilofs, sectlofs, and partlofs:

```
\def\mlfpagenumbers{\cftpagenumberson{figure}}
\def\nomlfpagenumbers{\cftpagenumbersoff{figure}}
\def\slfpagenumbers{\cftpagenumberson{figure}}
\def\noslfpagenumbers{\cftpagenumbersoff{figure}}
\def\plfpagenumbers{\cftpagenumberson{figure}}
\def\noplfpagenumbers{\cftpagenumbersoff{figure}}
```

\mltpagenumbers
\nomltpagenumbers
\sltpagenumbers
\nosltpagenumbers
\pltpagenumbers
\nopltpagenumbers

For entries in minilots, sectlots, and partlots:

```
\def\mltpagenumbers{\cftpagenumberson{table}}
\def\nomltpagenumbers{\cftpagenumbersoff{table}}
\def\sltpagenumbers{\cftpagenumberson{table}}
\def\nosltpagenumbers{\cftpagenumbersoff{table}}
\def\pltpagenumbers{\cftpagenumberson{table}}
\def\nopltpagenumbers{\cftpagenumbersoff{table}}
```

Else, minitoc will use its own commands.

```
\else
```

\mtcpagenumbers
\nomtcpagenumbers
\mlfpagenumbers
\nomlfpagenumbers
\mltpagenumbers
\nomltpagenumbers

First, for minitocs, secttocs and parttocs:

```
\def\mtcpagenumbers{\let\mtc@pgno\null}
\def\nomtcpagenumbers{\let\mtc@pgno\relax}
\def\stcpagenumbers{\let\stc@pgno\null}
\def\nostcpagenumbers{\let\stc@pgno\relax}
\def\ptcpagenumbers{\let\ptc@pgno\null}
\def\noptcpagenumbers{\let\ptc@pgno\relax}
```

\mlfpagenumbers
\nomlfpagenumbers
\mlfpagenumbers
\nomlfpagenumbers
\mltpagenumbers
\nomltpagenumbers

Then, for minilofs, sectlofs and partlofs:

```
\def\mlfpagenumbers{\let\mlf@pgno\null}
\def\nomlfpagenumbers{\let\mlf@pgno\relax}
\def\slfpagenumbers{\let\slf@pgno\null}
\def\noslfpagenumbers{\let\slf@pgno\relax}
\def\plfpagenumbers{\let\plf@pgno\null}
\def\noplfpagenumbers{\let\plf@pgno\relax}
```

\mltpagenumbers
\nomltpagenumbers
\mltpagenumbers
\nomltpagenumbers
\mltpagenumbers
\nomltpagenumbers

Then, for minilots, sectlots and partlots:

```
\def\mltpagenumbers{\let\mlt@pgno\null}
\def\nomltpagenumbers{\let\mlt@pgno\relax}
\def\sltpagenumbers{\let\slt@pgno\null}
```

```
\def\nosltpagenumbers{\let\slt@pgno\relax}
\def\pltpagenumbers{\let\plt@pgno\null}
\def\nopltpagenumbers{\let\plt@pgno\relax}
\fi
```

\ptcpagenumbers
\plfpagenumbers
\pltpagenumbers
\mtcpagenumbers
\mlfpagenumbers
\mltpagenumbers
\stcpagenumbers
\slfpagenumbers
\sltpagenumbers

Then the default values are set; page numbers are present:

```
\ptcpagenumbers
\plfpagenumbers
\pltpagenumbers
\mtcpagenumbers
\mlfpagenumbers
\mltpagenumbers
\stcpagenumbers
\slfpagenumbers
\sltpagenumbers
```

9.27 "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 `\thispageparttocstyle` which contains code to define the page style implied by mini-tables of a given type: the command `\thispageparttocstyle` 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
\beforeparttoc
\beforepartlof
\beforepartlot

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

```
\@ifundefined{chapter}{%
   \let\beforeparttoc\empty
   \let\beforepartlof\empty
   \let\beforepartlot\empty}%
```

\cleardoublepage

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

```
{\let\beforeparttoc\cleardoublepage
 \let\beforepartlof\cleardoublepage
 \let\beforepartlot\cleardoublepage}
```

\beforeminitoc
\beforeminilof
\beforeminilot

Chapter level mini-tables have no "before" feature (by default):

```
\let\beforeminitoc\empty
\let\beforeminilof\empty
\let\beforeminilot\empty
```

\beforesecttoc
\beforesectlof
\beforesectlot

Section level mini-tables have no "before" feature (by default):

```
\let\beforesecttoc\empty
\let\beforesectlof\empty
\let\beforesectlot\empty
```

\chapter
\afterparttoc
\afterpartlof
\afterpartlot

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

```
\@ifundefined{chapter}{%
  \let\afterparttoc\empty
  \let\afterpartlof\empty
  \let\afterpartlot\empty}%
```

\cleardoublepage

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

```
{\let\afterparttoc\cleardoublepage
 \let\afterpartlof\cleardoublepage
 \let\afterpartlot\cleardoublepage}
```

\afterminitoc
\afterminilof
\afterminilot

Chapter level mini-tables have no "after" feature (by default):

```
\let\afterminitoc\empty
\let\afterminilof\empty
\let\afterminilot\empty
```

\aftersecttoc
\aftersectlof
\aftersectlot

Section level mini-tables have no "after" feature (by default):

```
\let\aftersecttoc\empty
\let\aftersectlof\empty
\let\aftersectlot\empty
```

\thispagestyle
\thispageparttocstyle
\thispagepartlofstyle
\thispagepartlotstyle
\thispageminitocstyle
\thispageminilofstyle
\thispageminilotstyle
\thispagesecttocstyle
\thispagesectlofstyle
\thispagesectlotstyle

By default, all the "thispagestyle" features (at part level) use the `empty` page style. It affects only the first page of the mini-table. If `\chapter` is not defined, there is no default "thispagestyle" features at the part level.

```
\@ifundefined{chapter}{%
 \def\thispageparttocstyle{\empty}
 \def\thispagepartlofstyle{\empty}
 \def\thispagepartlotstyle{\empty}}%
{\def\thispageparttocstyle{\thispagestyle{empty}}
 \def\thispagepartlofstyle{\thispagestyle{empty}}
 \def\thispagepartlotstyle{\thispagestyle{empty}}}
\def\thispageminitocstyle{\empty}
\def\thispageminilofstyle{\empty}
\def\thispageminilotstyle{\empty}
\def\thispagesecttocstyle{\empty}
\def\thispagesectlofstyle{\empty}
\def\thispagesectlotstyle{\empty}
```

\mtcsetfeature

In section 9.66.8 on page 363, we will define the `\mtcsetfeature` macro which is much easier user interface to set the mini-tables "features".

9.28 Fake tables of contents

\faketableofcontents
\fakelistoffigures
\fakelistoftables
\fake@starttoc

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 analog commands `\fakelistoffigures` and `\fakelistoftables`, using in fact just 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[5].

```
\def\faketableofcontents{\fake@starttoc{toc}%
    \@ifundefined{c@ptc}{}{\setcounter{ptc}{0}}%
    \@ifundefined{c@mtc}{}{\setcounter{mtc}{0}}%
    \@ifundefined{c@stc}{}{\setcounter{stc}{0}}%
    }
\def\fakelistoffigures{\fake@starttoc{lof}}
\def\fakelistoftables{\fake@starttoc{lot}}
\def\fake@starttoc#1{\begingroup
  \makeatletter
  \if@filesw \expandafter\newwrite\csname tf@#1\endcsname
             \immediate\openout \csname tf@#1\endcsname
             \jobname.#1\relax
  \fi
  \global\@nobreakfalse \endgroup}
```

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

[5] Remember the infamous "`stc0`" bug.

9.29 Depth counters for minilofs and minilots

\AtBeginDocument
\c@lofdepth
\c@lotdepth

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

```
\AtBeginDocument{%
 \@ifundefined{c@lofdepth}{}%
   {\newcounter{minilofdepth}\setcounter{minilofdepth}{2}}%
 \@ifundefined{c@lotdepth}{}%
   {\newcounter{minilotdepth}\setcounter{minilotdepth}{2}}%
}
```

9.30 Chapter level commands

From here, we define the chapter-level commands.

\mtc@markboth
\@mkboth

First, we memorize the marks (not used today, but...):

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

9.31 Starred parts, chapters or sections

\addst@rred
\addcontentsline
\stepcounter
\c@ptc
\c@mtc
\c@stc

We define commands to manage the starred sectionning commands: \part*, \chapter* and \section*. The section-level is different depending if \chapter is defined or not. Eventually, a counter is incremented. A contents line is added in the .toc file, with the right depth to print it (see \l@star... later, in section 9.64 on page 344).

```
\def\addst@rred#1#2{%
  \addcontentsline{toc}{star#1}{#2}%
  \@ifundefined{c@ptc}{}{%
    \expandafter\ifx\csname #1\endcsname\part\relax
      \stepcounter{ptc}%
    \fi
  }%
  \@ifundefined{c@mtc}{}{%
    \expandafter\ifx\csname #1\endcsname\chapter\relax
      \stepcounter{mtc}%
    \fi
  }%
  \@ifundefined{c@stc}{}{%
    \expandafter\ifx\csname #1\endcsname\section\relax
```

```
%%      \@ifundefined{chapter}{\stepcounter{stc}}{}%
      \stepcounter{stc}%
    \fi
  }%
}%
```

\addstarredsection
\chapter
\addst@rred

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

```
\@ifundefined{chapter}{%
\gdef\addstarredsection#1{\addst@rred{section}{#1}}
}%
```

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

```
{%
```

\The@mtc
\firstchapteris
\if@firstchapteris@used@
\adjustmtc
\decrementmtc
\incrementmtc
\themtc
\columnwidth
\mtc@rule
\mlf@rule
\mlt@rule
\plf@rule
\plt@rule
\slf@rule
\slt@rule
\mtcindent

We define now: the internal format of the mtc counter (\The@mtc), the obsolete command \firstchapteris (it just emits a harmless warning), the mtc counter (initialized to 0), the \adjustmtc command (increments the mtc counter, by 1 by default), the \decrementmtc command (decrements the mtc counter by 1), the \incrementmtc command (increments the mtc counter by 1), the format of the mtc counter (\themtc), the counter minitocdepth, initialized to 2, for the depth of a minitoc (analog to the standard tocdepth counter).

We define the horizontal rules to draw before and after minitocs (\mtc@rule), and we copy that definition into analog macros for other kinds of mini-tables. We also set the default value (24pt) of \mtcindent, the indentation for minitocs (both sides). The rules are 0.4pt thick. They are defined via \hrule to stay in vertical mode for the final \kern.

```
\def\The@mtc{\arabic{mtc}}
\def\firstchapteris#1%
  {\mtcPackageWarning[<W0003>]{minitoc}%
     {\string\firstchapteris \space is an obsolete
      \MessageBreak
      command}%
     \@firstchapteris@used@true}
\newcounter{mtc}
\setcounter{mtc}{0}
\newcommand{\adjustmtc}[1][1]{\addtocounter{mtc}{#1}}
\def\decrementmtc{\addtocounter{mtc}{-1}}
\def\incrementmtc{\addtocounter{mtc}{+1}}
\gdef\themtc{\arabic{mtc}}
\newcounter{minitocdepth}
\setcounter{minitocdepth}{2}
\def\mtc@rule{\kern-3\p@ \hrule \@width\columnwidth \kern2.6\p@}
\let\mlf@rule\mtc@rule
\let\mlt@rule\mtc@rule
```

```
\let\plf@rule\mtc@rule
\let\plt@rule\mtc@rule
\let\slf@rule\mtc@rule
\let\slt@rule\mtc@rule
\mtcindent=24\p@
```

9.32 Font commands for the mini-tables

\mtcfont
\mtcSfont
\mtcSSfont
\mtcSSSfont
\mtcPfont
\mtcSPfont
\mlffont
\mlfSfont
\mltfont
\mltSfont
\mtifont

We define these commands with full NFSS [165] descriptions. These definitions are effective if \chapter is defined. The fonts for titles are also defined here. See also the \mtcsetfont macro (section 9.66.2 on page 347) and the \mtcsettitlefont macro later (section 9.66.3 on page 352).

```
\def\mtcfont{\small\rmfamily\upshape\mdseries}
\def\mtcSfont{\small\rmfamily\upshape\bfseries}
\let\mtcSSfont\mtcfont
\let\mtcSSSfont\mtcfont
\let\mtcPfont\mtcfont
\let\mtcSPfont\mtcfont
\let\mlffont\mtcfont
\let\mlfSfont\mtcfont
\let\mltfont\mtcfont
\let\mltSfont\mtcfont
\def\mtifont{\large\rmfamily\upshape\bfseries}
```

\coffeefont

And \coffeefont is used for "coffee breaks ☕" in the minutes package [169].

```
\def\coffeefont{\small\rmfamily\slshape\mdseries}
```

9.33 Internal commands to position the mini-table titles

\df@mtitc
\df@mtilf
\df@mtilt
\do@mtitc
\do@mtilf
\do@mtilt

The commands \miniXXX and \dominiXXX accept an optional argument to left justify, center, right justify or omit the title of the chapter-level mini-tables. By default, these titles are left justified. The choice made in a \dominiXXX command is global and memorized in \df@mtitc, \df@mtilf or \df@mtilt; the choice made in a \miniXXX command is local and stored in \do@mtitc, \do@mtilf or \do@mtilt. See the \minitoc@ macro later (section 9.35.1 on page 272). An empty title needs a vertical correction (Frank Mittelbach).

\c@mti
\l@mti
\r@mti
\e@mti
\n@mti

Centering, flushleft, flushright or empty titles:

```
\def\c@mti#1{\null\hfill #1\hfill\null}
\def\l@mti#1{\null #1\hfill\null}
\def\r@mti#1{\null\hfill #1\null}
\def\e@mti#1{\vspace{-\baselineskip}}
\def\n@mti#1{\vspace{-\baselineskip}}
```

\l@mti
\do@mtitc
\df@mtitc
\do@mtilf
\df@mtilf
\do@mtilt
\df@mtilt

Default: titles on left:

```
\let\do@mtitc\l@mti
\let\df@mtitc\l@mti
\let\do@mtilf\l@mti
\let\df@mtilf\l@mti
\let\do@mtilt\l@mti
\let\df@mtilt\l@mti
```

9.34 The mtc@verse environment

mtc@verse
\iftightmtc
\ifktightmtc
\\
\list
\itemsep
\itemindent
\listparindent
\topsep
\parsep
\mtcindent

Each minitoc is placed inside a mtc@verse environment. This environment is analog to the standard verse environment and hence defined via two commands: \mtc@verse and \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, \topsep. \parsep is set to zero if the tight option is active (to reduce the spacing of the lines). \parskip is set to zero if the k-tight option is active (to reduce the spacing of the lines). Both margins are set to \mtcindent. \endmtc@verse terminates the list and discourages a page break.

```
\def\mtc@verse{\let\\=\@centercr
  \list{}{\itemsep\z@
  \itemindent \z@
  \listparindent \itemindent
  \partopsep\z@
  \iftightmtc \parsep\z@ \fi
  \ifktightmtc \parskip\z@ \fi
  \topsep=1ex
  \leftmargin\mtcindent
  \rightmargin\leftmargin}\item[]}
\def\endmtc@verse{\nopagebreak[4]\endlist}
```

9.35 The \minitoc, \minilof, and \minilot commands

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

9.35.1 The \minitoc command

\minitoc
\chapter

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

\dominitoc
\minitoc

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 "l". It seems tortuous, but it is simple to use: we have a default behaviour (l) which can be altered globally via the optional argument of \dominitoc, or locally via the optional argument of \minitoc.

\minitoc
\minitoc@
\@ifnextchar

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

```
\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 9.78.2.2 on page 391).

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

\@tocfile
\if@longextensions

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.

```
\if@longextensions@
\def\@tocfile{mtc\The@mtc}%
\else
\def\@tocfile{M\The@mtc}%
\fi
```

\mtc@CkFile
\if@mtc@FE
\if@mtc@empty@minitoc@

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

```
    \mtc@CkFile{\jobname.\@tocfile}
    \if@mtc@FE
```

```
\mtcPackageInfo[<I0006>]{minitoc}%
   {\jobname.\@tocfile\space is empty}
\@mtc@empty@minitoc@true
\else
```

\thispageminitocstyle

We call \thispageminitocstyle 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.

```
\thispageminitocstyle
%%   \mtc@markboth{\MakeUppercase{\mtctitle}}{\MakeUppercase{\mtctitle}}%
```

\beforeminitoc
samepage
\do@mtitc
\e@mti
\n@mti
\c@mti
\l@mti
\r@mti
\df@mtic
\mtc@CkStr
\mtctitle
\if@mtc@FE
\mtcfont
\mtifont
\mtc@rule
\columnwidth
tabular

We call \beforeminitoc, then begin a samepage environment (to try to discourage page breaks in a minitoc) 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 minitoc (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 \mtcfont.

```
\beforeminitoc
\relax\begin{samepage}%
\if #1e\let\do@mtitc\e@mti
 \else\if #1n\let\do@mtitc\n@mti
 \else\if #1c\let\do@mtitc\c@mti
 \else\if #1l\let\do@mtitc\l@mti
 \else\if #1r\let\do@mtitc\r@mti
 \else\if #1d\let\do@mtitc\df@mtitc
\fi\fi\fi\fi\fi\fi
\mtc@CkStr{\mtctitle}\if@mtc@FE \let\do@mtitc\e@mti\relax\fi
\raggedright
\parskip=\z@%
\reset@font\mtcfont%
\parindent=\z@%
\nopagebreak[4]%
\kern-0.8\baselineskip\nopagebreak[4]%
\par\noindent %%
\ifx\mtc@rule\relax
 \begin{tabular}{@{}p{\columnwidth}@{}}
 \reset@font\mtifont\do@mtitc{\mtc@v\mtctitle}\\
 \end{tabular}%
\else
 \begin{tabular}{@{}p{\columnwidth}@{}}
 \reset@font\mtifont\do@mtitc{\mtc@v\mtctitle}\\\hline
 \end{tabular}%
\fi
```

\mtc@zrule
\mtc@BBR
\mtcindent
mtc@verse

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

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

\c@tocdepth
\c@minitocdepth
\\
\mtc@BBR

We force the effective depth of the mini-table (\c@tocdepth) to the required depth (\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 enough, then inhibit a page break. The blank line is necessary to avoid a parasite negative indentation.

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

\mtc@pgno
\@dottedtocline
\@undottedtocline
\mtc@hook@beforeinputfile
\mtc@setform
\mtcsetformat
\mtc@strut

We test the presence of leaders and page numbers, then print the minitoc by inputing the minitoc file. But before reading the minitoc file, we must call the hook macro (asked for by Donald Arseneau for his notoccite package [9]) \mtc@hook@beforeinputfile and the macro \mtc@setform which adjusts some layout parameters (defined by the user via some \mtcsetformat commands). We work in a group to keep local some macro redefinitions.

```
\begingroup
 \makeatletter
 \@ifundefined{mtc@pgno}%
 {\let\@dottedtocline\@undottedtocline}{}
 \@fileswfalse\mtc@hook@beforeinputfile
 \mtc@setform%
 \@input{\jobname.\@tocfile}
 \vspace{-1ex} \vspace{-\baselineskip}
 \leavevmode\mtc@strut
 \global\@nobreakfalse\endgroup
```

mtc@verse
\mtc@bottom@rule
\\
samepage
\afterminitoc

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

```
\end{mtc@verse}%
\kernafterminitoc
\nopagebreak[4]\mtc@bottom@rule\null\leavevmode\\%
\vskip-1.0\baselineskip\mtc@zrule
\end{samepage}%
\par\pagebreak[1]\vspace*{-1ex}\afterminitoc\fi}
```

\mtc@bottom@rule
\mtc@rule
\columnwidth

And we define the bottom rule for a minitoc, with some space under the minitoc:

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

9.35.2 The \minilof command

\minilof

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

\minilof
\chapter

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

\dominilof
\minilof

This command accepts an optional argument, whose default value has eventually been 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 "l". It seems tortuous, but it is simple to use: we have a default behaviour (l) which can be altered globally via the optional argument of \dominilof, or locally via the optional argument of \minilof.

\minilof
\minilof@
\@ifnextchar

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

```
\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 9.78.2.2 on page 391).

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

\@tocfile
\if@longextensions@

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

```
\if@longextensions@%
\def\@tocfile{mlf\The@mtc}%
\else
\def\@tocfile{F\The@mtc}%
\fi
```

\mtc@CkFile
\if@mtc@FE
\if@mtc@empty@minilof@

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

```
    \mtc@CkFile{\jobname.\@tocfile}
    \if@mtc@FE
    \mtcPackageInfo[<I0006>]{minitoc}%
       {\jobname.\@tocfile\space is empty}
    \@mtc@empty@minilof@true
    \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.

```
    \thispageminilofstyle
%%    \mtc@markboth{\MakeUppercase{\mlftitle}}{\MakeUppercase{\mlftitle}}%
```

\beforeminilof
samepage
\do@mtitc
\e@mti
\n@mti
\c@mti
\l@mti
\r@mti
\df@mtic
\mtc@CkStr
\mtctitle
\if@mtc@FE
\mlffont
\mtifont
\mlf@rule
\columnwidth
tabular

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.

```
    \beforeminilof
    \relax\begin{samepage}%
    \if #1e\let\do@mtilf\e@mti
    \else\if #1n\let\do@mtilf\n@mti
    \else\if #1c\let\do@mtilf\c@mti
    \else\if #1l\let\do@mtilf\l@mti
    \else\if #1r\let\do@mtilf\r@mti
    \else\if #1d\let\do@mtilf\df@mtilf
    \fi\fi\fi\fi\fi\fi
    \mtc@CkStr{\mlftitle}\if@mtc@FE \let\do@mtilf\e@mti\relax\fi
    \raggedright
    \parskip=\z@
    \reset@font\mlffont
    \parindent=\z@
    \nopagebreak[4]%
    \kern-0.8\baselineskip\nopagebreak[4]%
    \par\noindent
    \ifx\mlf@rule\relax
    \begin{tabular}{@{}p{\columnwidth}@{}}
```

```
      \reset@font\mtifont\do@mtilf{\mtc@v\mlftitle}\\
      \end{tabular}%
      \else
      \begin{tabular}{@{}p{\columnwidth}@{}}
      \reset@font\mtifont\do@mtilf{\mtc@v\mlftitle}\\\hline
      \end{tabular}%
      \fi
```

\mtc@zrule
\mtc@BBR
\mtcindent
mtc@verse

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

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

\c@lofdepth
\c@minilofdepth
\\
\mtc@BBR

We force the effective depth of the mini-table (\c@tocdepth) to the required depth (\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 enough, then inhibit a page break. The blank line is necessary to avoid a parasite negative indentation.

```
      \@ifundefined{c@lofdepth}%
        {}%
        {\c@lofdepth=\c@minilofdepth
       \ifnum\c@lofdepth<1\relax\c@lofdepth=1\fi}
       \leavevmode\\\mtc@BBR\vskip -.5\baselineskip
```

\mtc@pgno
\@dottedtocline
\@undottedtocline
\mtc@hook@beforeinputfile
\mlf@setform
\mtcsetformat
\mtc@strut

We test the presence of leaders and page numbers, then print the minilof by inputing the minilof file. But before reading the minilof file, we must call the hook macro (asked for by Donald Arseneau for his notoccite package [9]) \mtc@hook@beforeinputfile and the macro \mlf@setform which adjusts some layout parameters (defined by the user via some \mtcsetformat commands). We work in a group to keep local some macro redefinitions.

```
\begingroup
  \makeatletter
  \@ifundefined{mlf@pgno}%
  {\let\@dottedtocline\@undottedtocline}{}
  \@fileswfalse\mtc@hook@beforeinputfile
  \mlf@setform
  \@input{\jobname.\@tocfile}
  \vspace{-1ex} \vspace{-\baselineskip}
  \leavevmode\mtc@strut
  \global\@nobreakfalse\endgroup
```

mtc@verse
\mtc@bottom@rule
\\
samepage
\afterminilof

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

```
\end{mtc@verse}%
\kernafterminilof
\nopagebreak[4]\mlf@rule\null\leavevmode\\%
\vskip-1.0\baselineskip\mtc@zrule\end{samepage}%
\par\pagebreak[1]\vspace*{-1ex}\afterminilof\fi}%
```

9.35.3 The \minilot command

\minilot

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

\minilot
\chapter

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

\dominilot
\minilot

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 "l". It seems tortuous, but it is simple to use: we have a default behaviour (l) which can be altered globally via the optional argument of \dominilot, or locally via the optional argument of \minilot.

\minilot
\minilot@
\@ifnextchar

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

```
\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 9.78.2.2 on page 391).

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

\@tocfile
\if@longextensions@

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.

```
\if@longextensions@%
\def\@tocfile{mlt\The@mtc}%
\else
\def\@tocfile{T\The@mtc}%
\fi
```

\mtc@CkFile
\if@mtc@FE
\if@mtc@empty@minilot@

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

```
        \mtc@CkFile{\jobname.\@tocfile}
        \if@mtc@FE
        \mtcPackageInfo[<I0006>]{minitoc}%
           {\jobname.\@tocfile\space is empty}
        \@mtc@empty@minilot@true
        \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.

```
        \thispageminilotstyle
%%        \mtc@markboth{\MakeUppercase{\mlttitle}}{\MakeUppercase{\mlttitle}}%
```

\beforeminilot
samepage
\do@mtitc
\e@mti
\n@mti
\c@mti
\l@mti
\r@mti
\df@mtic
\mtc@CkStr
\mtctitle
\if@mtc@FE
\mltfont
\mtifont
\mlt@rule
\columnwidth
tabular

We call \beforeminilot, then begin a samepage environment (to try to discourage page breaks in a minilot) 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 minilot (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 \mltfont.

```
        \beforeminilot
        \relax\begin{samepage}%
        \if #1e\let\do@mtilt\e@mti
        \else\if #1n\let\do@mtilt\n@mti
        \else\if #1c\let\do@mtilt\c@mti
        \else\if #1l\let\do@mtilt\l@mti
        \else\if #1r\let\do@mtilt\r@mti
        \else\if #1d\let\do@mtilt\df@mtilt
        \fi\fi\fi\fi\fi\fi
        \mtc@CkStr{\mlttitle}\if@mtc@FE \let\do@mtilt\e@mti\relax\fi
        \raggedright
        \parskip=\z@%
        \reset@font\mltfont%
        \parindent=\z@%
```

```
      \nopagebreak[4]%
      \kern-0.8\baselineskip\nopagebreak[4]%
      \par\noindent
      \ifx\mlt@rule\relax
      \begin{tabular}{@{}p{\columnwidth}@{}}
      \reset@font\mtifont\do@mtilt{\mtc@v\mlttitle}\\
      \end{tabular}%
    \else
      \begin{tabular}{@{}p{\columnwidth}@{}}
      \reset@font\mtifont\do@mtilt{\mtc@v\mlttitle}\\\hline
      \end{tabular}%
    \fi
```

\mtc@zrule
\mtc@BBR
\mtcindent
mtc@verse

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

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

\c@lotdepth
\c@minilotdepth
\\
\mtc@BBR

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

```
      \@ifundefined{c@lotdepth}%
      {}%
      {\c@lotdepth=\c@minilotdepth
       \ifnum\c@lotdepth<1\relax\c@lotdepth=1\fi}
       \leavevmode\\\mtc@BBR\vskip -.5\baselineskip
```

\mtc@pgno
\@dottedtocline
\@undottedtocline
\mtc@hook@beforeinputfile
\mlt@setform
\mtcsetformat
\mtc@strut

We test the presence of leaders and page numbers, then print the minilot by inputing the minilot file. But before reading the minilot file, we must call the hook macro (asked for by Donald Arseneau for his notoccite package [9]) \mtc@hook@beforeinputfile and the macro \mlt@setform which adjusts some layout parameters (defined by the user via some \mtcsetformat commands). We work in a group to keep local some macro redefinitions.

```
\begingroup
  \makeatletter
  \@ifundefined{mlt@pgno}%
  {\let\@dottedtocline\@undottedtocline}{}
  \@fileswfalse\mtc@hook@beforeinputfile
  \mlt@setform
```

```
\@input{\jobname.\@tocfile}
\vspace{-1ex} \vspace{-\baselineskip}
\leavevmode\mtc@strut
\global\@nobreakfalse\endgroup
```

mtc@verse
\mtc@bottom@rule
\\
samepage
\afterminilot

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

```
        \end{mtc@verse}%
        \kernafterminilot
        \nopagebreak[4]\mlt@rule\null\leavevmode\\%
        \vskip-1.0\baselineskip\mtc@zrule\end{samepage}%
        \par\pagebreak[1]\vspace*{-1ex}\afterminilot\fi}%
```

9.36 Patching the \chapter command, continued

\l@xchapter
\@dottedtocline
\l@chapter
\xchapter

First, we define \l@xchapter which is like \l@chapter, but with a huge depth, to inhibit its printing (except if you cheat):

```
\def\l@xchapter{\@dottedtocline{\@M}{1em}{2.3em}}
\def\xchapter{xchapter}
```

\@chapter
\sv@chapter
\addcontentsline
\ignorespaces

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).

```
\let\sv@chapter\@chapter
\def\@chapter[#1]#2{\sv@chapter[{#1}]{#2}\relax%
   \addcontentsline{lof}{xchapter}{#1}%
   \addcontentsline{lot}{xchapter}{#1}%
   \ignorespaces}
```

\mtc@schapter
\@schapter
\addtocontents
\chapterbegin
\chapterend

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.

```
\let\mtc@schapter\@schapter
\def\@schapter{\addtocontents{toc}{\protect\chapterend}\mtc@schapter}
\def\@schapter{\addtocontents{@@@}{\protect\chapterbegin}\mtc@schapter}
\let\chapterbegin\relax
\let\chapterend\relax
```

9.37 The \addstarred... commands

\addstarredsection
\addstarredchapter
\addstarredpart
\chapter
\section
\part
\addst@rred

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 command \addstarredchapter. If the command \part is defined, we define the command \addstarredpart. We use the utility command \addst@rred defined in section 9.31 on page 268.

```
\@ifundefined{chapter}%
   {\@ifundefined{section}%
      {}{\def\addstarredsection#1{\addst@rred{section}{#1}}}}%
   {\def\addstarredchapter#1{\addst@rred{chapter}{#1}}}
\@ifundefined{part}%
   {}{\def\addstarredpart#1{\addst@rred{part}{#1}}}
```

9.38 TOC entries without leaders

\@Undottedtocline
\coffeefont

We define two internal macros to format TOC entries without leaders. The macro \@Undottedtocline prints no page number, but \@Undottedtoclinep prints it.

```
\def\@Undottedtocline#1#2#3#4#5{%
  \ifnum #1>\c@tocdepth\relax \else
    \vskip \z@ \@plus.2\p@
    {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
     \parindent #2\relax\@afterindenttrue
     \interlinepenalty\@M
     \leavevmode
     \@tempdima #3\relax
     \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
     {\coffeefont #4}\nobreak
     \nobreak\null
     \par}%
  \fi}
```

\@Undottedtoclinep
\coffeefont

The same but with the page number:

```
\def\@Undottedtoclinep#1#2#3#4#5{%
  \ifnum #1>\c@tocdepth\relax \else
    \vskip \z@ \@plus.2\p@
    {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
     \parindent #2\relax\@afterindenttrue
     \interlinepenalty\@M
     \leavevmode
     \@tempdima #3\relax
     \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
     {#4}\nobreak
```

```
        \hfill
    \nobreak\null
    \hb@xt@\@pnumwidth{\hfil\normalfont \normalcolor #5}%
    \par}%
\fi}
```

9.39 Mini-tables with or without leaders

\minitoc@
\minilof@
\minilot@
\@dottedtocline
\@Undottedtoclinep
\sv@minitoc@
\sv@minilof@
\sv@minilot@

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

```
\@ifundefined{chapter}{}{%
  \let\sv@minitoc@\minitoc@
  \def\minitoc@[#1]{{\ifundottedmtc\let\@dottedtocline\@Undottedtoclinep\fi
     \sv@minitoc@[#1]}}%
  \let\sv@minilof@\minilof@
  \def\minilof@[#1]{{\ifundottedmtc\let\@dottedtocline\@Undottedtoclinep\fi
     \sv@minilof@[#1]}}%
  \let\sv@minilot@\minilot@
  \def\minilot@[#1]{{\ifundottedmtc\let\@dottedtocline\@Undottedtoclinep\fi
     \sv@minilot@[#1]}}}
```

\sv@parttoc@
\sv@partlof@
\sv@partlot@
\ifundottedmtc
\parttoc@
\partlof@
\partlot@

For the part level:

```
\@ifundefined{part}{}{%
  \let\sv@parttoc@\parttoc@
  \def\parttoc@[#1]{{\ifundottedmtc\let\@dottedtocline\@Undottedtoclinep\fi
     \sv@parttoc@[#1]}}%
  \let\sv@partlof@\partlof@
  \def\partlof@[#1]{{\ifundottedmtc\let\@dottedtocline\@Undottedtoclinep\fi
     \sv@partlof@[#1]}}%
  \let\sv@partlot@\partlot@
  \def\partlot@[#1]{{\ifundottedmtc\let\@dottedtocline\@Undottedtoclinep\fi
     \sv@partlot@[#1]}}}
```

\sv@secttoc@
\sv@sectlof@
\sv@sectlot@
\ifundottedmtc
\secttoc@
\sectlof@
\sectlot@

For the section level:

```
\@ifundefined{chapter}{%
  \@ifundefined{section}{}{%
     \let\sv@secttoc@\secttoc@
     \def\secttoc@[#1]{{\ifundottedmtc\let\@dottedtocline\@Undottedtoclinep\fi
        \sv@secttoc@[#1]}}%
     \let\sv@sectlof@\sectlof@
```

```
    \def\sectlof@[#1]{{\ifundottedmtc\let\@dottedtocline\@Undottedtoclinep\fi
       \sv@sectlof@[#1]}}%
    \let\sv@sectlot@\sectlot@
    \def\sectlot@[#1]{{\ifundottedmtc\let\@dottedtocline\@Undottedtoclinep\fi
       \sv@sectlot@[#1]}}}}{}
```

9.40 The \dominitoc command and its siblings

\dominitoc
\dominilof
\dominilot
\contentsline
\chapbegin
\starchapter

The three commands \dominitoc, \dominilof and \dominilot are, of course, very similar. They take the \jobname.toc file (resp. the \jobname.lof and \jobname.lot files) produced by the previous LaTeX run and cut it in slices (one slice per chapter or starred chapter) into the \jobname.mtc⟨*N*⟩ files (resp. the \jobname.mlf⟨*N*⟩ and \jobname.mlt⟨*N*⟩ files), using specific lines in the \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
\dominitoc@
\@@dominitoc
\if@dominitoc@used@

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

The macros are \dominitoc (user interface), which calls \dominitoc@[l] (or with the optional argument of \dominitoc). Then \dominitoc@[l] processes its argument and calls \@@dominitoc. \@@dominitoc calls \@dominitoc (passing \jobname as argument) then close the minitoc file written. \@dominitoc reset to zero the counter of mini-tables, calls \MTC@next#1.toc (where #1 is the value of \jobname), then reset again to zero the counter of mini-tables. Each call to \dominitoc@ (i.e., to dominitoc) sets the flag \@dominitoc@used@true. This will be used later for a hint (which detects that you have correctly called \minitoc *after* \dominitoc and that both or neither have been called). See section 9.78.2.2 on page 391. The code is similar for \dominilof and \dominilot.

\dominitoc
\@dominitoc
\MTC@next

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

```
\def\@dominitoc#1{{%
  \makeatletter
  \setcounter{mtc}{0}
  \MTC@next#1.toc\relax\\}\setcounter{mtc}{0}}
\def\dominitoc{\@ifnextchar[{\dominitoc@}{\dominitoc@[l]}}
```

\dominilof
\@dominilof
\MLF@next

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

```
\def\@dominilof#1{{%
  \makeatletter
  \setcounter{mtc}{0}
  \MLF@next#1.lof\relax\\}\setcounter{mtc}{0}}
\def\dominilof{\@ifnextchar[{\dominilof@}{\dominilof@[l]}}
```

\dominilot
\@dominilot
\MLT@next

The \dominilot command extracts information from the .lot file and create the minilots files, with the adequate extension.

```
\def\@dominilot#1{{%
  \makeatletter
  \setcounter{mtc}{0}
  \MLT@next#1.lot\relax\\}\setcounter{mtc}{0}}
\def\dominilot{\@ifnextchar[{\dominilot@}{\dominilot@[l]}}
```

\if@dominitoc@used@
\if@mtc@hints@
\@mtc@hints@given@true
\df@mtitc
\e@mti
\n@mti
\c@mti
\l@mti
\r@mti
\@@dominitoc

Some code to flag the use of the command and manage the position of the minitoc title; a hint detects any spurious invocation.

```
\def\dominitoc@[#1]{%
\if@mtc@hints@
   \if@dominitoc@used@
   \mtcPackageInfo[<I0045>]{minitoc(hints)}%
         {The \string\dominitoc \space command
          \MessageBreak
          has been invoked more than once
          \MessageBreak}
   \global\@mtc@hints@given@true
   \fi
\fi
\global\@dominitoc@used@true
\if #1e\let\df@mtitc\e@mti%
\else\if #1n\let\df@mtitc\n@mti%
\else\if #1c\let\df@mtitc\c@mti%
\else\if #1l\let\df@mtitc\l@mti%
\else\if #1r\let\df@mtitc\r@mti%
\fi\fi\fi\fi\fi%
\@@dominitoc}
```

\if@dominilof@used@
\if@mtc@hints@
\@mtc@hints@given@true
\df@mtilf
\e@mti
\n@mti
\c@mti
\l@mti
\r@mti
\@@dominilof

Some code to flag the use of the command and manage the position of the minilof title; a hint detects any spurious invocation.

```
\def\dominilof@[#1]{%
\if@mtc@hints@
   \if@dominilof@used@
```

```
   \mtcPackageInfo[<I0045>]{minitoc(hints)}%
        {The \string\dominilof \space command
         \MessageBreak
         has been invoked more than once
         \MessageBreak}
   \global\@mtc@hints@given@true
   \fi
\fi
\global\@dominilof@used@true
\if #1e\let\df@mtilf\e@mti%
\else\if #1n\let\df@mtilf\n@mti%
\else\if #1c\let\df@mtilf\c@mti%
\else\if #1l\let\df@mtilf\l@mti%
\else\if #1r\let\df@mtilf\r@mti%
\fi\fi\fi\fi\fi%
\@@dominilof}
```

\if@dominilot@used@
\if@mtc@hints@
\@mtc@hints@given@true
\df@mtilt
\e@mti
\n@mti
\c@mti
\l@mti
\r@mti
\@@dominilot

Some code to flag the use of the command and manage the position of the minilot title; a hint detects any spurious invocation.

```
\def\dominilot@[#1]{%
\if@mtc@hints@
   \if@dominilot@used@
   \mtcPackageInfo[<I0045>]{minitoc(hints)}%
        {The \string\dominilot \space command
         \MessageBreak
         has been invoked more than once
         \MessageBreak}
   \global\@mtc@hints@given@true
   \fi
\fi
\global\@dominilot@used@true
\if #1e\let\df@mtilt\e@mti%
\else\if #1n\let\df@mtilt\n@mti%
\else\if #1c\let\df@mtilt\c@mti%
\else\if #1l\let\df@mtilt\l@mti%
\else\if #1r\let\df@mtilt\r@mti%
\fi\fi\fi\fi\fi%
\@@dominilot}
```

\@@dominitoc
\@@dominilof
\@@dominilot
\tf@mtc

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

```
\def\@@dominitoc{\@dominitoc{\jobname}\immediate\closeout\tf@mtc}
\def\@@dominilof{\@dominilof{\jobname}\immediate\closeout\tf@mtc}
\def\@@dominilot{\@dominilot{\jobname}\immediate\closeout\tf@mtc}
```

9.40.1 Analysis and splitting of the TOC file

This is done via a loop managed by the following macros [6]:

\MTC@next
\MTC@list
\MTC@loop

Processes the next entry in the list and removes it from the head of the list:

```
\def\MTC@next#1\relax#2\\{%
  \edef\MTC@list{#2}%
  \MTC@loop{#1}%
}
```

\MTC@toc
\MTC@list
\MTC@explist

Check if the list is empty:

```
\def\MTC@toc{%
  \ifx\MTC@list\@empty\else\expandafter\MTC@explist\fi
}
```

\MTC@contentsline
\arabic
\chapter
\themtc
\tf@mtc

The macro \MTC@contentsline analyses the lines read from the TOC file and detects interesting keywords. If \chapter is found, the mtc counter (which simulates the chapter counter, but is absolute) is incremented and a new minitoc file is created.

```
\def\MTC@contentsline#1#2#3#4{% %%HO/BJ: 4 instead of 3 parameters
  \gdef\themtc{\arabic{mtc}}% %%HO: space removed
  \expandafter\ifx\csname #1\endcsname\chapter
    \stepcounter{mtc}%
```

\if@longextensions@
\themtc
\mtcname
\tf@mtc
\closeout
\openout

We test if long or short extensions are used, to build the name of the mini-table file, then open it (after closing the file descriptor):

```
    \if@longextensions@%
      \mtcPackageInfo[<I0033>]{minitoc}%
         {Writing\space\jobname.mtc\themtc\@gobble}%
      \def\mtcname{\jobname.mtc\themtc}%
    \else
      \mtcPackageInfo[<I0033>]{minitoc}%
         {Writing\space\jobname.M\themtc\@gobble}%
      \def\mtcname{\jobname.M\themtc}%
    \fi
    \immediate\closeout\tf@mtc
    \immediate\openout\tf@mtc=\mtcname
  \fi
```

[6] This code is derived from the xr package [89], 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.

\mtc@toks

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

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

\MTC@WriteContentsline
\section
\subsection
\subsubsection
\paragraph
\subparagraph

Each interesting contents line is copied, with a font command added before it. We begin with the standard sectionning commands, below \chapter:

```
\expandafter\ifx\csname #1\endcsname\section
  \MTC@WriteContentsline{#1}{mtcS}{#3}{#4}%
\fi
\expandafter\ifx\csname #1\endcsname\subsection
  \MTC@WriteContentsline{#1}{mtcSS}{#3}{#4}%
\fi
\expandafter\ifx\csname #1\endcsname\subsubsection
  \MTC@WriteContentsline{#1}{mtcSSS}{#3}{#4}%
\fi
\expandafter\ifx\csname #1\endcsname\paragraph
  \MTC@WriteContentsline{#1}{mtcP}{#3}{#4}%
\fi
\expandafter\ifx\csname #1\endcsname\subparagraph
  \MTC@WriteContentsline{#1}{mtcSP}{#3}{#4}%
\fi
```

\coffee
\MTC@WriteCoffeeline

A coffee break contents line ☕ is written for \coffee:

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

\starchapter
\stepcounter
\if@longextensions@
\mtcname
\themtc
\tf@mtc
\closeout
\openout

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

```
\expandafter\ifx\csname #1\endcsname\starchapter
  \stepcounter{mtc}%
  \if@longextensions@
    \mtcPackageInfo[<I0033>]{minitoc}%
       {Writing\space\jobname.mtc\themtc\@gobble}%
    \def\mtcname{\jobname.mtc\themtc}%
  \else
    \mtcPackageInfo[<I0033>]{minitoc}%
       {Writing\space\jobname.M\themtc\@gobble}%
    \def\mtcname{\jobname.M\themtc}%
  \fi
  \immediate\closeout\tf@mtc
  \immediate\openout\tf@mtc=\mtcname
\fi
```

\starsection
\MTC@WriteContentsline
\starsubsection
\starsubsubsection
\starparagraph
\starsubparagraph

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

```
  \expandafter\ifx\csname #1\endcsname\starsection
    \MTC@WriteContentsline{#1}{mtcS}{#3}{#4}%
  \fi
  \expandafter\ifx\csname #1\endcsname\starsubsection
    \MTC@WriteContentsline{#1}{mtcSS}{#3}{#4}%
  \fi
  \expandafter\ifx\csname #1\endcsname\starsubsubsection
    \MTC@WriteContentsline{#1}{mtcSSS}{#3}{#4}%
  \fi
  \expandafter\ifx\csname #1\endcsname\starparagraph
    \MTC@WriteContentsline{#1}{mtcP}{#3}{#4}%
  \fi
  \expandafter\ifx\csname #1\endcsname\starsubparagraph
    \MTC@WriteContentsline{#1}{mtcSP}{#3}{#4}%
  \fi
}
```

\MTC@explist
\MTC@next
\MTC@list

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:

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

\MTC@loop
\openin
\@inputcheck
\MTC@toc
\MTC@read

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

```
\def\MTC@loop#1{\openin\@inputcheck#1\relax
  \ifeof\@inputcheck
    \mtcPackageWarning[<W0010>]{minitoc}%
       {No file #1.
        \MessageBreak
        MINITOCS NOT PREPARED}%
    \expandafter\MTC@toc
  \else
    \mtcPackageInfo[<I0024>]{minitoc}{PREPARING MINITOCS FROM #1}%
    \expandafter\MTC@read
  \fi
}
```

\MTC@read
\read
\@inputcheck
\MTC@line

Read the next entry of the .toc file.

```
\def\MTC@read{%
  \read\@inputcheck to\MTC@line
```

\MTC@test
\MTC@line
\MTC@

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

```
  \expandafter\MTC@test\MTC@line.....\MTC@% %%HO: . added
}%
```

\MTC@test
\contentsline
\MTC@contentsline
\mtc@string
\@input
\MTC@list
\chapterend
\closeout
\tf@mtc
\openout
\chapterbegin
\addtocounter
\MTC@toc
\MTC@read

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

```
%%HO/BJ: now patch \MTC@test,
%%HO/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. Thanks to A.J. "Tony" Roberts.

```
\long\def\MTC@test#1#2#3#4#5#6\MTC@{% %%HO: #6 added
  \ifx#1\contentsline
    \let\mtc@string\string
    \MTC@contentsline{#2}{#3}{#4}{#5}%
    %%HO/BJ: 4. parameter added by Tony Roberts
    \let\mtc@string\relax
  \else\ifx#1\@input
    \edef\MTC@list{\MTC@list#2\relax}%
  \else\ifx#1\chapterend
    \immediate\closeout\tf@mtc
    \immediate\openout\tf@mtc=\jobname.mtc
  \else\ifx#1\chapterbegin
    \addtocounter{mtc}{-1}%
  \fi\fi\fi\fi
  \ifeof\@inputcheck
    \expandafter\MTC@toc
  \else
    \expandafter\MTC@read
  \fi
}%
```

9.41 Mini-lists of figures

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

9.41.1 Analysis and splitting of the list of figures file

\MLF@next
\MLF@list
\MLF@loop

This is done via a loop managed by the following macros:

Processes the next entry in the list and removes it from the head of the list:

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

\MLF@lof
\MLF@list
\MLF@explist

Checks if the list is empty:

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

\MLF@contentsline
\arabic
\xchapter

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

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

\if@longextensions@
\themtc
\mlfname
\closeout
\tf@mtc
\openout

The name of the minilof file is built from \jobname and a long or short extension:

```
    \if@longextensions@%
      \mtcPackageInfo[<I0033>]{minitoc}%
         {Writing\space\jobname.mlf\themtc\@gobble}
      \def\mlfname{\jobname.mlf\themtc}%
    \else
      \mtcPackageInfo[<I0033>]{minitoc}%
         {Writing\space\jobname.F\themtc\@gobble}
      \def\mlfname{\jobname.F\themtc}%
    \fi
    \immediate\closeout\tf@mtc
    \immediate\openout\tf@mtc=\mlfname
  \fi
```

\figure
\subfigure
\mtc@toks
\MTC@WriteContentsline

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

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

```
  \expandafter\ifx\csname #1\endcsname\subfigure
    \mtc@toks{\noexpand\leavevmode#2}%
    \MTC@WriteContentsline{#1}{mlfS}{#3}{#4}%
  \fi
}
```

\MLF@explist
\MLF@next
\MLF@list

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

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

\MLF@loop
\openin
\@inputcheck
\MLF@lof
\MLF@read

And now, we scan the .lof file:

```
\def\MLF@loop#1{\openin\@inputcheck#1\relax
  \ifeof\@inputcheck
    \mtcPackageWarning[<W0008>]{minitoc}%
       {No file #1.
        \MessageBreak
        MINILOFS NOT PREPARED}%
    \expandafter\MLF@lof
  \else
    \mtcPackageInfo[<I0034>]{minitoc}%
       {PREPARING MINILOFS FROM #1}%
    \expandafter\MLF@read\fi}
```

\MLF@read
\read
\@inputcheck

Read the next entry in the .lof file:

```
\def\MLF@read{%
  \read\@inputcheck to\MLF@line
```

\MLF@line
\MLF@test
\MLF@

The ..... make sure that \MLF@test has enough arguments:

```
  \expandafter\MLF@test\MLF@line.....\MLF@% %%HO: . added
  }%
```

\MLF@test

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

\contentsline
\mtc@string
\MLF@contentsline
\@input
\MLF@list
\chapterend
\closeout
\tf@mtc
\openout
\chapterbegin
\addtocounter
\MLF@lof
\MLF@read

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.

```
\long\def\MLF@test#1#2#3#4#5#6\MLF@{% %%HO: #6 added
  \ifx#1\contentsline
    \let\mtc@string\string
    \MLF@contentsline{#2}{#3}{#4}{#5}% %%HO: #4 added
    \let\mtc@string\relax
  \else\ifx#1\@input
      \edef\MLF@list{\MLF@list#2\relax}%
  \else\ifx#1\chapterend
      \immediate\closeout\tf@mtc
      \immediate\openout\tf@mtc=\jobname.mtc
  \else\ifx#1\chapterbegin
    \addtocounter{mtc}{-1}%
  \fi\fi\fi\fi
  \ifeof\@inputcheck\expandafter\MLF@lof
  \else\expandafter\MLF@read\fi}%
```

9.42 Mini-lists of tables

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

9.42.1 Analysis and splitting of the list of tables file

\MLT@next
\MLT@list
\MLT@loop

This is done via a loop managed by the following macros:

Processes the next entry in the list and removes it from the head of the list:

```
\def\MLT@next#1\relax#2\\{%
  \edef\MLT@list{#2}%
  \MLT@loop{#1}}
```

\MLT@lot
\MLT@list
\MLT@explist

Checks if the list is empty:

```
\def\MLT@lot{%
  \ifx\MLT@list\@empty\else\expandafter\MLT@explist\fi}
```

\MLT@contentsline
\arabic
\xchapter

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

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

\if@longextensions@
\themtc
\mltname
\closeout
\tf@mtc
\openout

The name of the minilot file it build from \jobname and a long or short extension:

```
    \if@longextensions@%
      \mtcPackageInfo[<I0033>]{minitoc}%
         {Writing\space\jobname.mlt\themtc\@gobble}%
      \def\mltname{\jobname.mlt\themtc}%
    \else
      \mtcPackageInfo[<I0033>]{minitoc}%
         {Writing\space\jobname.T\themtc\@gobble}%
      \def\mltname{\jobname.T\themtc}%
    \fi
    \immediate\closeout\tf@mtc
    \immediate\openout\tf@mtc=\mltname
  \fi
```

\table
\subtable
\mtc@toks
\MTC@WriteContentsline

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

```
  \expandafter\ifx\csname #1\endcsname\table
    \mtc@toks{\noexpand\leavevmode#2}%
    \MTC@WriteContentsline{#1}{mlt}{#3}{#4}%
  \fi
  \expandafter\ifx\csname #1\endcsname\subtable
    \mtc@toks{\noexpand\leavevmode#2}%
    \MTC@WriteContentsline{#1}{mltS}{#3}{#4}%
  \fi
}
```

\MLT@explist
\MLT@next
\MLT@list

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

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

\MLT@loop
\openin
\@inputcheck
\MLT@lot
\MLT@read

And now, we scan the .lot file:

```
\def\MLT@loop#1{\openin\@inputcheck#1\relax
  \ifeof\@inputcheck
    \mtcPackageWarning[<W0009>]{minitoc}%
```

```
      {No file #1.
       \MessageBreak
       MINILOTS NOT PREPARED}%
    \expandafter\MLT@lot
  \else
    \mtcPackageInfo[<I0037>]{minitoc}%
       {PREPARING MINILOTS FROM #1}%
    \expandafter\MLT@read\fi}
```

\MLT@read
\read
\@inputcheck

Read the next entry in the `.lot` file:

```
\def\MLT@read{%
  \read\@inputcheck to\MLT@line
```

\MLT@line
\MLT@test
\MLT@

The `.....` make sure that `\MLT@test` has enough arguments:

```
  \expandafter\MLT@test\MLT@line.....\MLT@% %%HO: . added
  }%
```

\MLT@test

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

\contentsline
\mtc@string
\MLT@contentsline
\@input
\MLT@list
\chapterend
\closeout
\tf@mtc
\openout
\chapterbegin
\addtocounter
\MLT@lot
\MLT@read

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.

```
\long\def\MLT@test#1#2#3#4#5#6\MLT@{% %%HO: #6 added
  \ifx#1\contentsline
    \let\mtc@string\string
    \MLT@contentsline{#2}{#3}{#4}{#5}% %%HO: #4 added
    \let\mtc@string\relax
  \else\ifx#1\@input
     \edef\MLT@list{\MLT@list#2\relax}%
  \else\ifx#1\chapterend
     \immediate\closeout\tf@mtc
     \immediate\openout\tf@mtc=\jobname.mtc
  \else\ifx#1\chapterbegin
    \addtocounter{mtc}{-1}%
  \fi\fi\fi\fi
  \ifeof\@inputcheck\expandafter\MLT@lot
  \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).

```
}%
```

9.43 Macro to write a contents line

\mtc@dot
\MTC@WriteContentsline
\mtc@param
\write
\tf@mtc
\@resetfont
\mtc@string
\contentsline
\mtc@toks

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

The arguments of \MTC@WriteContentsline are:

#1: the #1 argument of \MTC@contentsline;

#2: font shorthand = : \csname #2font\endcsname;

#3: the #3 argument of \MTC@contentsline;

#4: the #4 argument of \MTC@contentsline (hyperlink).

```
\def\mtc@dot{.}
\def\MTC@WriteContentsline#1#2#3#4{%
  \def\mtc@param{#4}%
   \immediate\write\tf@mtc{%
    {\string\reset@font
     \expandafter\string\csname #2font\endcsname
     \string\mtc@string
     \string\contentsline{#1}%
     {\the\mtc@toks}%
     {\string\reset@font
      \expandafter\string\csname #2font\endcsname
      \space #3%
     }%
     \ifx\mtc@dot\mtc@param
     \else
       {#4}% %%HO/BJ: #4 is hyperlink
     \fi
    }%
  }%
}
```

\MTC@WriteCoffeeline
\write
\tf@mtc
\@resetfont
\coffeefont
\mtc@string
\mtc@toks

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

#1: the #1 argument of \MTC@contentsline;

#2: the #3 argument of \MTC@contentsline.

```
\def\MTC@WriteCoffeeline#1#2#3{%
  \immediate\write\tf@mtc{%
    {\string\reset@font \string\coffeefont \string\mtc@string
     {\the\mtc@toks}%
     {\string\reset@font \string\coffeefont \space #3%
     }%
```

```
    }%
  }%
}
```

9.44 Depth counters for partlofs and partlots

\AtBeginDocument
\newcounter
\setcounter
\c@lofdepth
\c@lotdepth

If the counters lofdepth and lotdepth are defined, we create the corresponding new counters: partlofdepth and partlotdepth. These counters are initialised to 2. This is done after the loading of the packages, in an \AtBeginDocument block:

```
\AtBeginDocument{%
 \@ifundefined{c@lofdepth}{}%
   {\newcounter{partlofdepth}\setcounter{partlofdepth}{2}}%
 \@ifundefined{c@lotdepth}{}%
   {\newcounter{partlotdepth}\setcounter{partlotdepth}{2}}%
}
```

9.45 Part level commands

\xpart
\theptc
\Thepart
\adjustptc
\decrementptc
\incrementptc
\firstpartis
\firstpartis
\if@firstpartis@used@
\newcounter
\setcounter

If \part is defined, we define some utility commands, a counter (ptc) for the parttocs and related commands (\theptc, \Thepart, \adjustptc, \decrementptc, \incrementptc), the obsolete command \firstpartis, and the depth counter parttocdepth.

```
\@ifundefined{part}{}%
{%
\def\xpart{xpart}
\def\Thepart{\arabic{ptc}}
\def\firstpartis#1%
  {\mtcPackageWarning[<W0004>]{minitoc}%
     {\string\firstpartis \space is an obsolete
      \MessageBreak
      command}%
     \@firstpartis@used@true}
\newcounter{ptc}
\setcounter{ptc}{0}
\newcommand{\adjustptc}[1][1]{\addtocounter{ptc}{#1}}
\def\decrementptc{\addtocounter{ptc}{-1}}
\def\incrementptc{\addtocounter{ptc}{+1}}
\def\theptc{\arabic{ptc}}
\newcounter{parttocdepth}
\setcounter{parttocdepth}{2}
```

\ptc@rule
\columnwidth

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.

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

\ptcindent

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

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

\ptcfont
\ptcSfont
\ptcSSfont
\ptcSSSfont
\ptcPfont
\ptcSPfont
\plffont
\plfSfont
\pltfont
\pltSfont
\ptifont

9.46 Fonts for the parttocs

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):

```
\@ifundefined{chapter}{%
   \def\ptcfont{\small\rmfamily\upshape\mdseries} % the parttoc
   \def\ptcSfont{\small\rmfamily\upshape\bfseries}% (sections)
   \let\ptcSSfont\ptcfont     % (subsections)
   \let\ptcSSSfont\ptcfont    % (subsubsections)
   \let\ptcPfont\ptcfont      % (paragraphs)
   \let\ptcSPfont\ptcfont     % (subparagraphs)
   \let\plffont\ptcfont       % (figures)
   \let\plfSfont\ptcfont      % (subfigures)
   \let\pltfont\ptcfont       % (tables)
   \let\pltSfont\ptcfont      % (subtables)
   \def\ptifont{\Large\rmfamily\upshape\bfseries}% titles
}%
```

\ptcfont
\ptcCfont
\ptcSfont
\ptcSSfont
\ptcSSSfont
\ptcPfont
\ptcSPfont
\plffont
\plfSfont
\pltfont
\pltSfont
\ptifont

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

```
{%
   \def\ptcfont{\normalsize\rmfamily\upshape\mdseries} % the parttoc
   \def\ptcCfont{\normalsize\rmfamily\upshape\bfseries}% (chapters)
   \def\ptcSfont{\normalsize\rmfamily\upshape\mdseries}% (sections)
   \let\ptcSSfont\ptcfont     % (subsections)
   \let\ptcSSSfont\ptcfont    % (subsubsections)
   \let\ptcPfont\ptcfont      % (paragraphs)
   \let\ptcSPfont\ptcfont     % (subparagraphs)
   \let\plffont\ptcfont       % (figures)
```

```
  \let\plfSfont\ptcfont      % (subfigures)
  \let\pltfont\ptcfont       % (tables)
  \let\pltSfont\ptcfont      % (subtables)
  \def\ptifont{\LARGE\rmfamily\upshape\bfseries}% titles
}
```

9.47 Default titles for part-level mini-tables

\parttoc
\partlof
\partlot

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

\c@pti
\l@pti
\r@pti
\e@pti
\n@pti

If \chapter is undefined, the definitions are very simple, for centered, flushleft, flushright or empty titles. Here, empty titles need a vertical correction (Frank Mittelbach).

```
\@ifundefined{chapter}{%
\def\c@pti#1{\null\hfill #1\hfill\null}
\def\l@pti#1{\null #1\hfill\null}
\def\r@pti#1{\null\hfill #1\null}
\def\e@pti#1{\vspace{-\baselineskip}}
\def\n@pti#1{\vspace{-\baselineskip}}
}%
```

\e@pti
\n@pti

But, if \chapter is defined, we must simulate the formatting of a chapter head, which is more complex. Here, empty titles need a vertical correction (Frank Mittelbach).

```
{%
\def\e@pti#1{\vspace{-\baselineskip}}
\def\n@pti#1{\vspace{-\baselineskip}}
```

\l@pti
\if@twocolumn
\@topnewpage
\@afterheading
\ptifont
\@makephead@l
\mtcgapbeforeheads
\mtcgapafterheads

For a title on the left, we must test if the main text is on two columns:

```
\def\l@pti#1{\if@twocolumn
    \@topnewpage[\@makephead@l{#1}]%
    \else
    \@makephead@l{#1}%
    \@afterheading
    \fi}
\def\@makephead@l#1{%
    \vspace*{\mtcgapbeforeheads}%
    {\parindent \z@ \raggedright
     \ptifont
     #1\par
     \nobreak
```

```
     \vskip \mtcgapafterheads\hbox{}
    }}
```

\r@pti
\if@twocolumn
\@topnewpage
\@makephead@r
\@afterheading
\ptifont
\mtcgapbeforeheads
\mtcgapafterheads

For a title on the right, we must also test if the main text is on two columns:

```
\def\r@pti#1{\if@twocolumn
    \@topnewpage[\@makephead@r{#1}]%
    \else
    \@makephead@r{#1}%
    \@afterheading
    \fi}
\def\@makephead@r#1{%
    \vspace*{\mtcgapbeforeheads}%
    {\parindent \z@ \raggedleft
     \ptifont
     #1\par
     \nobreak
     \vskip \mtcgapafterheads\hbox{}
    }}
```

\c@pti
\if@twocolumn
\@topnewpage
\@makephead@c
\@afterheading
\ptifont
\mtcgapbeforeheads
\mtcgapafterheads

For a centered title, we must also test if the main text is on two columns:

```
\def\c@pti#1{\if@twocolumn
    \@topnewpage[\@makephead@c{#1}]%
    \else
    \@makephead@c{#1}%
    \@afterheading
    \fi}
\def\@makephead@c#1{%
    \vspace*{\mtcgapbeforeheads}%
    {\parindent \z@ \centering
     \ptifont
     #1\par
     \nobreak
     \vskip \mtcgapafterheads\hbox{}
    }}%
}
```

\l@pti
\do@ptitc
\df@ptitc
\do@ptilf
\df@ptilf
\do@ptilt
\df@ptilt

By default, titles are on left:

```
\let\do@ptitc\l@pti
\let\df@ptitc\l@pti
\let\do@ptilf\l@pti
\let\df@ptilf\l@pti
\let\do@ptilt\l@pti
\let\df@ptilt\l@pti
```

9.48 The ptc@verse environment

ptc@verse
\ptc@verse
\endptc@verse
\\
\@centercr
\list
\itemsep
\itemindent
\listparindent
\topsep
\parsep
\parskip
\partopsep
\ptcindent
\iftightmtc
\ifktightmtc

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) \\, then call \list{} and set some dimensions like \itemsep, \itemindent, \listparindent, \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.

```
\def\ptc@verse{\let\\=\@centercr
  \list{}{\itemsep\z@
          \itemindent \z@
          \listparindent \itemindent
          \partopsep\z@
          \iftightmtc \parsep\z@ \fi
          \ifktightmtc \parskip\z@ \fi
          \topsep=1ex
          \leftmargin\ptcindent
          \rightmargin\leftmargin}\item[]}
\def\endptc@verse{\nopagebreak[4]\endlist}
```

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

\parttoc
\partlof
\partlot

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.

9.49.1 The \parttoc command

\parttoc
\parttoc@

This command must be used after \part if you need a parttoc (no automatic 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:

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

\parttoc@
\if@parttoc@used@
\if@longextensions@
\@tocfile
\Thepart

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

```
\def\parttoc@[#1]{%
\global\@parttoc@used@true
\if@longextensions@%
\def\@tocfile{ptc\Thepart}%
\else
\def\@tocfile{P\Thepart}%
\fi
```

\mtc@CkFile
\if@mtc@FE
\@tocfile

Then, we check the presence of the parttoc file and give a warning if it is not here:

```
    \mtc@CkFile{\jobname.\@tocfile}
    \if@mtc@FE
    \mtcPackageInfo[<I0006>]{minitoc}%
       {\jobname.\@tocfile\space is empty}
    \@mtc@empty@parttoc@true
    \else
```

\beforeparttoc

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

```
    \beforeparttoc
```

\mtc@markboth
\@mkboth
\thispageparttocstyle
\MakeUppercase
\ptctitle

If \chapter is defined, we just set the page marks with the parttoc title and set the page style:

```
    \@ifundefined{chapter}{}{%
    \global\let\mtc@markboth\markboth
    \global\let\@mkboth\markboth
    \thispageparttocstyle
    \mtc@markboth{\MakeUppercase{\ptctitle}}{\MakeUppercase{\ptctitle}}%
    }%
```

\do@ptitc
\e@pti
\n@pti
\c@pti
\l@pti
\r@pti
\df@pti
\mtc@CkStr
\ptctitle
\if@mtc@FE
samepage

A samepage environment is begun, then the argument is treated to set the position of the parttoc title. If the title string is empty, this forces the positionning.

```
    \relax\begin{samepage}%
    \if #1e\let\do@ptitc\e@pti
    \else\if #1n\let\do@ptitc\n@pti
    \else\if #1c\let\do@ptitc\c@pti
    \else\if #1l\let\do@ptitc\l@pti
    \else\if #1r\let\do@ptitc\r@pti
    \else\if #1d\let\do@ptitc\df@ptitc
    \fi\fi\fi\fi\fi\fi
    \mtc@CkStr{\ptctitle}\if@mtc@FE \let\do@ptitc\e@pti\relax\fi
```

\raggedright
\parskip
\ptcfont

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

```
\raggedright
\parskip=\z@%
\reset@font\ptcfont%
\parindent=\z@%
\nopagebreak[4]%
\kern-0.8\baselineskip\nopagebreak[4]%
\par\noindent
\nopagebreak[4]%
```

\ptc@rule
tabular
\columnwidth
\ptifont
\do@ptitc
\mtc@v
\ptctitle
\hline

The parttoc 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 parttoc.

```
\ifx\ptc@rule\relax
\begin{tabular}{@{}p{\columnwidth}@{}}
\reset@font\ptifont\do@ptitc{\mtc@v\ptctitle}\\
\end{tabular}%
\else
\begin{tabular}{@{}p{\columnwidth}@{}}
\reset@font\ptifont\do@ptitc{\mtc@v\ptctitle}\\\hline
\end{tabular}%
\fi
```

\mtc@zrule
\mtc@BBR
\ptcindent

Then, we adjust the position under the top rule and set the indentation and some formatting parameters:

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

ptc@verse
\c@tocdepth
\c@parttocdepth
\mtc@BBR

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

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

\ptc@pgno
\@dottedtocline
\@undottedtocline
\mtc@hook@beforeinputfile
\ptc@setform
\@tocfile
\mtc@strut
ptc@verse

If the contents lines must have no numbers, we replace the macro \@dottedtocline with its undotted version. For chapter-level entries, we must invoke \l@chapter ignoring the page number argument. A hook (redefinissable command) is added, and the formatting settings coming from \mtcsetformat are activated via \ptc@setform. Then the parttoc file is inserted, followed by a strut, and the ptc@verse environment is terminated.

```
\begingroup
  \makeatletter
  \@ifundefined{ptc@pgno}%
  {\let\@dottedtocline\@undottedtocline}{}
  \@ifundefined{ptc@pgno}%
  {\let\l@chapter@SVPN\l@chapter%
    \def\l@chapter##1##2{\l@chapter@SVPN{##1}{\hbox{}}}}{}
  \@fileswfalse\mtc@hook@beforeinputfile
  \ptc@setform
  \@input{\jobname.\@tocfile}
  \vspace{-1ex} \vspace{-1\baselineskip}
  \leavevmode\mtc@strut
  \global\@nobreakfalse\endgroup
        \end{ptc@verse}%
```

\ptc@rule
\mtc@zrule
samepage
\afterparttoc

The final part is just to add the bottom rule, if necessary, a possible page break (if \chapter is not defined), and \afterparttoc.

```
        \kernafterparttoc
        \nopagebreak[4]\ptc@rule\null\leavevmode\\%
        \vskip-1.0\baselineskip\mtc@zrule\end{samepage}%
        \par\@ifundefined{chapter}{\pagebreak[1]\vspace*{-1ex}}%
        \afterparttoc\fi}%
```

9.49.2 The \partlof command

\partlof
\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:

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

\partlof@
\if@partlof@used@
\if@longextensions@
\@tocfile
\Thepart

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

```
\def\partlof@[#1]{%
\global\@partlof@used@true
\if@longextensions@%
\def\@tocfile{plf\Thepart}%
```

```
\else
\def\@tocfile{G\Thepart}%
\fi
```

\mtc@CkFile
\if@mtc@FE
\@tocfile

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

```
    \mtc@CkFile{\jobname.\@tocfile}
    \if@mtc@FE
    \mtcPackageInfo[<I0006>]{minitoc}%
       {\jobname.\@tocfile\space is empty}
    \@mtc@empty@partlof@true
    \else
```

\beforepartlof

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

```
    \beforepartlof
```

\mtc@markboth
\@mkboth
\thispagepartlofstyle
\MakeUppercase
\plftitle

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

```
    \@ifundefined{chapter}{}{%
    \global\let\mtc@markboth\markboth
    \global\let\@mkboth\markboth
    \thispagepartlofstyle
    \mtc@markboth{\MakeUppercase{\plftitle}}{\MakeUppercase{\plftitle}}%
    }%
```

\do@ptilf
\e@pti
\n@pti
\c@pti
\l@pti
\r@pti
\df@pti
\mtc@CkStr
\plftitle
\if@mtc@FE
samepage

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.

```
    \relax\begin{samepage}%
    \if #1e\let\do@ptilf\e@pti
    \else\if #1n\let\do@ptilf\n@pti
    \else\if #1c\let\do@ptilf\c@pti
    \else\if #1l\let\do@ptilf\l@pti
    \else\if #1r\let\do@ptilf\r@pti
    \else\if #1d\let\do@ptilf\df@ptilf
    \fi\fi\fi\fi\fi\fi
    \mtc@CkStr{\plftitle}\if@mtc@FE \let\do@ptilf\e@pti\relax\fi
```

\raggedright
\parskip
\plffont

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

```
    \raggedright
    \parskip=\z@%
    \reset@font\plffont%
```

\plf@rule
tabular
\columnwidth
\ptifont
\do@ptilf
\mtc@v
\plftitle
\hline

The parttoc 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 partlof.

```
\parindent=\z@%
\nopagebreak[4]%
\kern-0.8\baselineskip\nopagebreak[4]%
\par\noindent
\ifx\plf@rule\relax
\begin{tabular}{@{}p{\columnwidth}@{}}
\reset@font\ptifont\do@ptilf{\mtc@v\plftitle}\\
\end{tabular}%
\else
\begin{tabular}{@{}p{\columnwidth}@{}}
\reset@font\ptifont\do@ptilf{\mtc@v\plftitle}\\\hline
\mtc@hstrut\\
\end{tabular}%
\fi
```

\mtc@zrule
\mtc@BBR
\ptcindent

Then, we adjust the position under the top rule and set the indentation and some formatting parameters:

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

ptc@verse
\mtc@BBR

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

```
\begin{ptc@verse}%
\@ifundefined{c@lofdepth}%
{}%
{\c@lofdepth=\c@partlofdepth
 \ifnum\c@lofdepth<1\relax\c@lofdepth=1\fi}
 \leavevmode\\\mtc@BBR\vskip -.5\baselineskip
```

\plf@pgno
\@dottedtocline
\@undottedtocline
\mtc@hook@beforeinputfile
\plf@setform
\@tocfile
\mtc@strut
ptc@verse

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 \mtcsetformat are activated via \plf@setform. Then the partlof file is inserted, followed by a strut, and the ptc@verse environment is terminated.

```
\begingroup
  \makeatletter
  \@ifundefined{plf@pgno}%
```

```
{\let\@dottedtocline\@undottedtocline}{}
\@fileswfalse\mtc@hook@beforeinputfile
\plf@setform
\@input{\jobname.\@tocfile}
\vspace{-1ex} \vspace{-1\baselineskip}
\leavevmode\mtc@strut
\global\@nobreakfalse\endgroup
      \end{ptc@verse}%
```

\plf@rule
\mtc@zrule
samepage
\afterpartlof

The final part is just to add the bottom rule, if necessary, a possible page break (if \chapter is not defined), and \afterpartlof. The blank line (\\) is essential.

```
      \kernafterpartlof
      \nopagebreak[4]\plf@rule\null\leavevmode\\%
      \vskip-1.0\baselineskip\mtc@zrule\end{samepage}%
      \par\@ifundefined{chapter}{\pagebreak[1]\vspace*{-1ex}}%
      \afterpartlof\fi}
```

9.49.3 The \partlot command

\partlot
\partlot@

This command must be used after \part if you need a partlot (no automatic partlot). First, \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:

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

\partlot@
\if@partlot@used@
\if@longextensions@
\@tocfile
\Thepart

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

```
%
\def\partlot@[#1]{%
\global\@partlot@used@true
\if@longextensions@%
\def\@tocfile{plt\Thepart}%
\else
\def\@tocfile{U\Thepart}%
\fi
```

\mtc@CkFile
\if@mtc@FE
\@tocfile

Then, we check the presence of the partlot file and give a warning if it is not here:

```
      \mtc@CkFile{\jobname.\@tocfile}
      \if@mtc@FE
      \mtcPackageInfo[<I0006>]{minitoc}%
```

```
          {\jobname.\@tocfile\space is empty}
      \@mtc@empty@partlof@true
      \else
```

\beforepartlot

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

```
      \beforepartlot
```

\mtc@markboth
\@mkboth
\thispagepartlotstyle
\MakeUppercase
\plttitle

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

```
      \@ifundefined{chapter}{}{%
      \global\let\mtc@markboth\markboth
      \global\let\@mkboth\markboth
      \thispagepartlotstyle
      \mtc@markboth{\MakeUppercase{\plttitle}}{\MakeUppercase{\plttitle}}%
      }%
```

\do@ptilt
\e@pti
\n@pti
\c@pti
\l@pti
\r@pti
\df@pti
\mtc@CkStr
\plttitle
\if@mtc@FE
samepage

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.

```
      \relax\begin{samepage}%
      \if #1e\let\do@ptilt\e@pti
      \else\if #1n\let\do@ptilt\n@pti
      \else\if #1c\let\do@ptilt\c@pti
      \else\if #1l\let\do@ptilt\l@pti
      \else\if #1r\let\do@ptilt\r@pti
      \else\if #1d\let\do@ptilt\df@ptilt
      \fi\fi\fi\fi\fi\fi
      \mtc@CkStr{\plttitle}\if@mtc@FE \let\do@ptilt\e@pti\relax\fi
```

\raggedright
\parskip
\pltfont

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

```
      \raggedright
      \parskip=\z@%
      \reset@font\pltfont%
      \parindent=\z@%
      \nopagebreak[4]%
      \kern-0.8\baselineskip\nopagebreak[4]%
      \par\noindent
```

\plt@rule
tabular
\columnwidth
\ptifont
\do@ptilt
\mtc@v
\plttitle
\hline

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.

```
\ifx\plt@rule\relax
\begin{tabular}{@{}p{\columnwidth}@{}}
\reset@font\ptifont\do@ptilt{\mtc@v\plttitle}\\
\end{tabular}%
\else
\begin{tabular}{@{}p{\columnwidth}@{}}
\reset@font\ptifont\do@ptilt{\mtc@v\plttitle}\\\hline
\mtc@hstrut\\
\end{tabular}%
\fi
```

\mtc@zrule
\mtc@BBR
\ptcindent

Then, we adjust the position under the top rule and set the indentation and some formatting parameters:

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

ptc@verse
\mtc@BBR

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.

```
\begin{ptc@verse}%
\@ifundefined{c@lotdepth}%
{}%
{\c@lotdepth=\c@partlotdepth
 \ifnum\c@lotdepth<1\relax\c@lotdepth=1\fi}
 \leavevmode\\\mtc@BBR\vskip -.5\baselineskip
```

\plt@pgno
\@dottedtocline
\@undottedtocline
\mtc@hook@beforeinputfile
\plt@setform
\@tocfile
\mtc@strut
ptc@verse

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 \mtcsetformat are activated via \plt@setform. Then the partlot file is inserted, followed by a strut, and the ptc@verse environment is terminated.

```
\begingroup
  \makeatletter
  \@ifundefined{plt@pgno}%
  {\let\@dottedtocline\@undottedtocline}{}
  \@fileswfalse\mtc@hook@beforeinputfile
  \plt@setform
  \@input{\jobname.\@tocfile}
  \vspace{-1ex} \vspace{-1\baselineskip}
```

```
\leavevmode\mtc@strut
\global\@nobreakfalse\endgroup
      \end{ptc@verse}%
```

\ptc@rule
\mtc@zrule
samepage
\afterpartlot

The final part is just to add the bottom rule, if necessary, a possible page break (if \chapter is not defined), and \afterpartlot. The blank line (\\) is essential.

```
      \kernafterpartlot
      \nopagebreak[4]\plt@rule\null\leavevmode\\%
      \vskip-1.0\baselineskip\mtc@zrule\end{samepage}%
      \par\@ifundefined{chapter}{\pagebreak[1]\vspace*{-1ex}}%
      \afterpartlot\fi}
```

9.50 Auxiliary commands for printing parttocs

\@dottedtocline
\l@xpart
\l@pchapter
\l@psect
\pchapter
\psect

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 commands are similar to \l@subsection, only the arguments have been altered:

```
\def\l@xpart{\@dottedtocline{\@M}{1.0em}{2.3em}}
\def\l@pchapter{\@dottedtocline{1}{1.0em}{2.3em}}
\def\l@psect{\@dottedtocline{2}{1.0em}{2.3em}}
\def\pchapter{pchapter}
\def\psect{psect}
```

9.51 Patching the \part command, continued

\sv@part
\mtc@svpart
\@part
\addtocontents
\sv@spart
\ptc@spart
\@spart
\stepcounter
\partbegin
\partend

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).

```
\let\sv@part\mtc@svpart
\def\@part[#1]#2{\sv@part[{#1}]{#2}\relax
\addcontentsline{lof}{xpart}{#1}%
\addcontentsline{lot}{xpart}{#1}%
\addcontentsline{toc}{xpart}{#1}%
\stepcounter{ptc}}
\let\sv@spart\@spart
\def\@spart{\stepcounter{ptc}\sv@spart}
\let\ptc@spart\@spart
\def\@spart{\addtocontents{toc}{\protect\partend}\ptc@spart}
```

```
\def\@spart{\addtocontents{toc}{\protect\partbegin}\ptc@spart}
\let\partend\relax
\let\partbegin\relax
```

9.52 The \doparttoc command and its siblings

\doparttoc
\dopartlof
\dopartlot

The \doparttoc command works like the \dominitoc command, \dopartlof like \dominilof and \dopartlot like \dominilot.

\@doparttoc
\PTC@next
\setcounter

The \doparttoc command extracts information from the .toc file and creates the .ptc⟨*N*⟩ files (.ptc becomes .P on MS-DOS).

```
\def\@doparttoc#1{{%
  \makeatletter
  \setcounter{ptc}{0}%
  \PTC@next#1.toc\relax\\}\setcounter{ptc}{0}}%
```

\@dopartlof
\PLF@next
\setcounter

The \dopartlof command extracts information from the .lof file and creates the .plf⟨*N*⟩ files (.plf becomes .G on MS-DOS).

```
\def\@dopartlof#1{{%
  \makeatletter
  \setcounter{ptc}{0}%
  \PLF@next#1.lof\relax\\}\setcounter{ptc}{0}}%
```

\@dopartlot
\PLT@next
\setcounter

The \dopartlot command extracts information from the .lot file and creates the .plt⟨*N*⟩ files (.plt becomes .U on MS-DOS).

```
\def\@dopartlot#1{{%
  \makeatletter
  \setcounter{ptc}{0}%
  \PLT@next#1.lot\relax\\}\setcounter{ptc}{0}}%
```

\doparttoc
\dopartlof
\dopartlot

We define the user macros, who detect the optional argument:

```
\def\doparttoc{\@ifnextchar[{\doparttoc@}{\doparttoc@[l]}}
\def\dopartlof{\@ifnextchar[{\dopartlof@}{\dopartlof@[l]}}
\def\dopartlot{\@ifnextchar[{\dopartlot@}{\dopartlot@[l]}}
```

\doparttoc@
\if@mtc@hints@
\if@doparttoc@used@
\@mtc@hints@given@true
\df@ptitc
\e@pti
\n@pti
\c@pti
\l@pti
\r@pti

We treat the optional argument of \doparttoc (it becomes the default position for titles of parttocs) and flag this macro as used; a hint detects any spurious invocation.

```
\def\doparttoc@[#1]{%
\if@mtc@hints@
   \if@doparttoc@used@
   \mtcPackageInfo[<I0045>]{minitoc(hints)}%
        {The \string\doparttoc \space command
         \MessageBreak
         has been invoked more than once
         \MessageBreak}
   \global\@mtc@hints@given@true
   \fi
\fi
\global\@doparttoc@used@true
\if #1e\let\df@ptitc\e@pti%
\else\if #1n\let\df@ptitc\n@pti%
\else\if #1c\let\df@ptitc\c@pti%
\else\if #1l\let\df@ptitc\l@pti%
\else\if #1r\let\df@ptitc\r@pti%
\fi\fi\fi\fi\fi%
\@@doparttoc}
```

\dopartlof@
\if@dopartlof@used@
\if@mtc@hints@
\@mtc@hints@given@true
\df@ptilf
\e@pti
\n@pti
\c@pti
\l@pti
\r@pti

We treat the optional argument of \dopartlof (it becomes the default position for titles of partlofs) and flag this macro as used, a hint detects any spurious invocation.

```
\def\dopartlof@[#1]{%
\if@mtc@hints@
   \if@dopartlof@used@
   \mtcPackageInfo[<I0045>]{minitoc(hints)}%
        {The \string\dopartlof \space command
         \MessageBreak
         has been invoked more than once
         \MessageBreak}
   \global\@mtc@hints@given@true
   \fi
\fi
\global\@dopartlof@used@true
\if #1e\let\df@ptilf\e@pti%
\else\if #1n\let\df@ptilf\n@pti%
\else\if #1c\let\df@ptilf\c@pti%
\else\if #1l\let\df@ptilf\l@pti%
\else\if #1r\let\df@ptilf\r@pti%
\fi\fi\fi\fi\fi%
\@@dopartlof}
```

\dopartlot@
\if@mtc@hints@
\@mtc@hints@given@true
\if@dopartlot@used@
\df@ptilt
\e@pti
\n@pti
\c@pti
\l@pti
\r@pti

We treat the optional argument of \dopartlot (it becomes the default position for titles of partlofs) and flag this macro as used; a hint detects any spurious invocation.

```
\def\dopartlot@[#1]{%
\if@mtc@hints@
   \if@dopartlot@used@
   \mtcPackageInfo[<I0045>]{minitoc(hints)}%
        {The \string\dopartlot \space command
         \MessageBreak
         has been invoked more than once
         \MessageBreak}
   \global\@mtc@hints@given@true
   \fi
\fi
\global\@dopartlot@used@true
\if #1e\let\df@ptilt\e@pti%
\else\if #1n\let\df@ptilt\n@pti%
\else\if #1c\let\df@ptilt\c@pti%
\else\if #1l\let\df@ptilt\l@pti%
\else\if #1r\let\df@ptilt\r@pti%
\fi\fi\fi\fi\fi%
\@@dopartlot}
```

\@@doparttoc
\@@dopartlof
\@@dopartlot
\tf@mtc

These macros invoke the \@dopart... commands to create the mini-table file, then close the file descriptor.

```
\def\@@doparttoc{\@doparttoc{\jobname}\immediate\closeout\tf@mtc}
\def\@@dopartlof{\@dopartlof{\jobname}\immediate\closeout\tf@mtc}
\def\@@dopartlot{\@dopartlot{\jobname}\immediate\closeout\tf@mtc}
```

9.52.1 Processing macros for the parttocs

\PTC@next
\PTC@list
\PTC@loop

Processing the next entry in the list and remove it from the head of the list:

```
\def\PTC@next#1\relax#2\\{%
  \edef\PTC@list{#2}%
  \PTC@loop{#1}}
```

\PTC@toc
\PTC@list
\PTC@explist

Check if the list is empty:

```
\def\PTC@toc{%
  \ifx\PTC@list\@empty\else\expandafter\PTC@explist\fi}
```

\PTC@contentsline
\part
\theptc
\tf@mtc
\ptcname
\MTC@WriteContentsLine

The macro \PTC@contentsline analyses the lines read from the TOC file and detects interesting keywords. If \part is found, the ptc counter is incremented and a new partlof file is created.

```
\def\PTC@contentsline#1#2#3#4{% %%HO/DV: 4 instead of 3 parameters
  \expandafter\ifx\csname #1\endcsname\part
    \stepcounter{ptc}%
    \if@longextensions@%
      \mtcPackageInfo[<I0033>]{minitoc}%
         {Writing\space\jobname.ptc\theptc\@gobble}%
      \def\ptcname{\jobname.ptc\theptc}%
    \else
      \mtcPackageInfo[<I0033>]{minitoc}%
         {Writing\space\jobname.P\theptc\@gobble}%
      \def\ptcname{\jobname.P\theptc}%
    \fi
    \immediate\closeout\tf@mtc
    \immediate\openout\tf@mtc=\ptcname
  \fi
  \expandafter\ifx\csname #1\endcsname\starpart\relax
    \stepcounter{ptc}%
```

\if@longextensions@
\ptcname

We test if long or short extensions are used, to build the name of the mini-table file, then open it:

```
    \if@longextensions@%
      \mtcPackageInfo[<I0033>]{minitoc}%
         {Writing\space\jobname.ptc\theptc}%
      \def\ptcname{\jobname.ptc\theptc}%
    \else
      \mtcPackageInfo[<I0033>]{minitoc}%
         {Writing\space\jobname.P\theptc}%
      \def\ptcname{\jobname.P\theptc}%
    \fi
    \immediate\closeout\tf@mtc
    \immediate\openout\tf@mtc=\ptcname
  \fi
```

\mtc@toks

The token register \mtc@toks is used to pass the entry to \MTC@WriteContentsline:

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

\MTC@WriteContentsline
\chapter
\pchapter
\section
\coffee
\subsection
\subsubsection
\paragraph
\subparagraph

Now, we filter the relevant contents lines; this code extracts and writes info for chapters, sections, etc.:

```
  \expandafter\ifx\csname #1\endcsname\chapter
    \MTC@WriteContentsline{#1}{ptcC}{#3}{#4}%
  \fi
```

```
\expandafter\ifx\csname #1\endcsname\pchapter
  \MTC@WriteContentsline{#1}{ptcC}{#3}{#4}%
\fi
\expandafter\ifx\csname #1\endcsname\section
  \MTC@WriteContentsline{#1}{ptcS}{#3}{#4}%
\fi
\expandafter\ifx\csname #1\endcsname\coffee
  \MTC@WriteCoffeeline{#1}{#3}%
\fi
\expandafter\ifx\csname #1\endcsname\subsection
  \MTC@WriteContentsline{#1}{ptcSS}{#3}{#4}%
\fi
\expandafter\ifx\csname #1\endcsname\subsubsection
  \MTC@WriteContentsline{#1}{ptcSSS}{#3}{#4}%
\fi
\expandafter\ifx\csname #1\endcsname\paragraph
  \MTC@WriteContentsline{#1}{ptcP}{#3}{#4}%
\fi
\expandafter\ifx\csname #1\endcsname\subparagraph
  \MTC@WriteContentsline{#1}{ptcSP}{#3}{#4}%
\fi
```

\MTC@WriteContentsline
\starchapter
\starsection
\starsubsection
\starsubsubsection
\starparagraph
\starsubparagraph

And for the starred sectionning commands:

```
\expandafter\ifx\csname #1\endcsname\starchapter
%%HO: the following line should be disabled: \stepcounter{ptc}%
  \MTC@WriteContentsline{#1}{ptcC}{#3}{#4}%
\fi
\expandafter\ifx\csname #1\endcsname\starsection
  \MTC@WriteContentsline{#1}{ptcS}{#3}{#4}%
\fi
\expandafter\ifx\csname #1\endcsname\starsubsection
  \MTC@WriteContentsline{#1}{ptcSS}{#3}{#4}%
\fi
\expandafter\ifx\csname #1\endcsname\starsubsubsection
  \MTC@WriteContentsline{#1}{ptcSSS}{#3}{#4}%
\fi
\expandafter\ifx\csname #1\endcsname\starparagraph
  \MTC@WriteContentsline{#1}{ptcP}{#3}{#4}%
\fi
\expandafter\ifx\csname #1\endcsname\starsubparagraph
  \MTC@WriteContentsline{#1}{ptcSP}{#3}{#4}%
\fi
}
```

\PTC@explist
\PTC@next
\PTC@list

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

```
\def\PTC@explist{\expandafter\PTC@next\PTC@list\\}
```

\PTC@loop
\PTC@toc
\PTC@read

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

```
\def\PTC@loop#1{\openin\@inputcheck#1\relax
  \ifeof\@inputcheck
    \mtcPackageWarning[<W0013>]{minitoc}%
       {No file #1
        \MessageBreak
        PARTTOCS NOT PREPARED}%
    \expandafter\PTC@toc
  \else
    \mtcPackageInfo[<I0025>]{minitoc}%
       {PREPARING PARTTOCS FROM #1}%
    \expandafter\PTC@read\fi}
```

\PTC@read
\PTC@line

Read the next entry of the .toc file.

```
\def\PTC@read{%
  \read\@inputcheck to\PTC@line
```

\PTC@test
\PTC@line

The ..... make sure that \PTC@test has enough arguments:

```
  \expandafter\PTC@test\PTC@line.....\PTC@% %%HO: . added
  }%
```

\PTC@test

The \PTC@test macro finds the "interesting" commands in the TOC file, mainly to delimit parts:

```
%%HO/BJ: now patch \PTC@test,
%%HO/BJ: call \PTC@contentsline with 4 instead of 3 parameters
```

\PTC@contentsline
\mtc@string
\PTC@list
\PTC@toc
\PTC@read
\partend

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.

```
\long\def\PTC@test#1#2#3#4#5#6\PTC@{% %%HO: #6 added
  \ifx#1\contentsline
    \let\mtc@string\string
    \PTC@contentsline{#2}{#3}{#4}{#5}%
    %%HO/DV: 4 instead of 3 parameters
    \let\mtc@string\relax
  \else\ifx#1\@input
     \edef\PTC@list{\PTC@list#2\relax}%
  \else\ifx#1\partend
     \immediate\closeout\tf@mtc
     \immediate\openout\tf@mtc=\jobname.mtc
```

```
\else\ifx#1\partbegin
   \addtocounter{ptc}{-1}%
\fi\fi\fi\fi
\ifeof\@inputcheck\expandafter\PTC@toc
\else\expandafter\PTC@read\fi}%
```

9.52.2 Processing macros for the partlofs

\PLF@next
\PLF@list
\PLF@loop

Processing the next entry in the list and remove it from the head of the list:

```
\def\PLF@next#1\relax#2\\{%
  \edef\PLF@list{#2}%
  \PLF@loop{#1}}
```

\PLF@lof
\PLF@list
\PLF@explist

Check if the list is empty:

```
\def\PLF@lof{%
  \ifx\PLF@list\@empty\else\expandafter\PLF@explist\fi}
```

\PLF@contentsline
\part
\theptc
\tf@mtc
\plfname
\PLF@WriteContentsLine

The macro \PLF@contentsline analyses the lines read from the LOF file and detects interesting keywords. If \part is found, the ptc counter is incremented and a new partlof file is created.

```
\def\PLF@contentsline#1#2#3#4{% %%HO: #4 added
  \expandafter\ifx\csname #1\endcsname\xpart
    \stepcounter{ptc}%
```

\if@longextensions@
\plfname

We test if long or short extensions are used, to build the name of the mini-table file, then open it:

```
    \if@longextensions@%
      \mtcPackageInfo[<I0033>]{minitoc}%
         {Writing\space\jobname.plf\theptc}%
      \def\plfname{\jobname.plf\theptc}%
    \else
      \mtcPackageInfo[<I0033>]{minitoc}%
         {Writing\space\jobname.G\theptc}%
      \def\plfname{\jobname.G\theptc}%
    \fi
    \immediate\closeout\tf@mtc
    \immediate\openout\tf@mtc=\plfname
  \fi
```

\mtc@toks

The token register \mtc@toks is used to pass the entry to \MTC@WriteContentsline:

\figure
\subfigure
\mtc@toks
\MTC@WriteContentsline

Now, we filter the relevant contents lines:

```
\expandafter\ifx\csname #1\endcsname\figure
  \mtc@toks{\noexpand\leavevmode#2}%
  \MTC@WriteContentsline{#1}{plf}{#3}{#4}%
\fi
\expandafter\ifx\csname #1\endcsname\subfigure
  \mtc@toks{\noexpand\leavevmode#2}%
  \MTC@WriteContentsline{#1}{plfS}{#3}{#4}%
\fi
}
```

\PLF@explist
\PLF@next
\PLF@list

The loop to read the lines of the LOF file; expands the list of entries and call \PLF@next to process the first one:

```
\def\PLF@explist{\expandafter\PLF@next\PLF@list\\}
```

\PLF@loop
\PLF@lof
\PLF@read

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

```
\def\PLF@loop#1{\openin\@inputcheck#1\relax
  \ifeof\@inputcheck
    \mtcPackageWarning[<W0011>]{minitoc}%
       {No file #1
        \MessageBreak
        PARTLOFS NOT PREPARED}%
    \expandafter\PLF@lof
  \else
    \mtcPackageInfo[<I0035>]{minitoc}%
       {PREPARING PARTLOFS FROM #1}%
    \expandafter\PLF@read\fi}
```

\PLF@read
\PLF@line

Read the next entry of the .lof file.

```
\def\PLF@read{%
  \read\@inputcheck to\PLF@line
```

\PLF@test
\PLF@line

The ..... make sure that \PLF@test has enough arguments:

```
  \expandafter\PLF@test\PLF@line.....\PLF@% %%HO: . added
  }%
```

\PLF@test

The \PLF@test macro finds the "interesting" commands in the LOF file, mainly to delimit parts:

```
%%HO/BJ: now patch \PLF@test,
%%HO/BJ: call \PLF@contentsline with 4 instead of 3 parameters
```

\PLF@contentsline
\mtc@string
\PLF@list
\PLF@lof
\PLF@read
\partend

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.

```
\long\def\PLF@test#1#2#3#4#5#6\PLF@{% %%HO: #6 added
  \ifx#1\contentsline
    \let\mtc@string\string
    \PLF@contentsline{#2}{#3}{#4}{#5}% %%HO: #4 added
    \let\mtc@string\relax
  \else\ifx#1\@input
      \edef\PLF@list{\PLF@list#2\relax}%
  \else\ifx#1\partend
      \immediate\closeout\tf@mtc
      \immediate\openout\tf@mtc=\jobname.mtc
  \else\ifx#1\partbegin
      \addtocounter{ptc}{-1}%
  \fi\fi\fi\fi
  \ifeof\@inputcheck\expandafter\PLF@lof
  \else\expandafter\PLF@read\fi}%
```

9.52.3 Processing macros for the partlots

\PLT@next
\PLT@list
\PLT@loop

Processing the next entry in the list and remove it from the head of the list:

```
\def\PLT@next#1\relax#2\\{%
  \edef\PLT@list{#2}%
  \PLT@loop{#1}}
```

\PLT@lot
\PLT@list
\PLT@explist

Check if the list is empty:

```
\def\PLT@lot{%
  \ifx\PLT@list\@empty\else\expandafter\PLT@explist\fi}
```

\PLT@contentsline
\part
\theptc
\tf@mtc
\pltname
\PLT@WriteContentsLine

The macro \PLT@contentsline analyses the lines read from the LOT file and detects interesting keywords. If \part is found, the ptc counter is incremented and a new partlot file is created.

```
\def\PLT@contentsline#1#2#3#4{% %%HO: #4 added
```

```
\expandafter\ifx\csname #1\endcsname\xpart
  \stepcounter{ptc}%
```

\if@longextensions@
\pltname

We test if long or short extensions are used, to build the name of the mini-table file, then open it:

```
  \if@longextensions@%
    \mtcPackageInfo[<I0033>]{minitoc}%
       {Writing\space\jobname.plt\theptc}%
    \def\pltname{\jobname.plt\theptc}%
  \else
    \mtcPackageInfo[<I0033>]{minitoc}%
       {Writing\space\jobname.U\theptc}%
    \def\pltname{\jobname.U\theptc}%
  \fi
  \immediate\closeout\tf@mtc
  \immediate\openout\tf@mtc=\pltname
\fi
```

\mtc@toks

The token register \mtc@toks is used to pass the entry to \MTC@WriteContentsline:

\table
\subtable
\mtc@toks
\MTC@WriteContentsline

Now, we filter the relevant contents lines:

```
\expandafter\ifx\csname #1\endcsname\table
  \mtc@toks{\noexpand\leavevmode#2}%
  \MTC@WriteContentsline{#1}{plt}{#3}{#4}%
\fi
\expandafter\ifx\csname #1\endcsname\subtable
  \mtc@toks{\noexpand\leavevmode#2}%
  \MTC@WriteContentsline{#1}{pltS}{#3}{#4}%
\fi
}
```

\PLT@explist
\PLT@next
\PLT@list

The loop to read the lines of the LOT file; expands the list of entries and call \PLT@next to process the first one:

```
\def\PLT@explist{\expandafter\PLT@next\PLT@list\\}
```

\PLT@loop
\PLT@lot
\PLT@read

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

```
\def\PLT@loop#1{\openin\@inputcheck#1\relax
  \ifeof\@inputcheck
    \mtcPackageWarning[<W0012>]{minitoc}%
```

```
      {No file #1
       \MessageBreak
       PARTLOTS NOT PREPARED}%
    \expandafter\PLT@lot
  \else
    \mtcPackageInfo[<I0038>]{minitoc}%
       {PREPARING PARTLOTS FROM #1}%
    \expandafter\PLT@read\fi}
```

\PLT@read
\PLT@line

Read the next entry of the `.lot` file.

```
\def\PLT@read{%
  \read\@inputcheck to\PLT@line
```

\PLT@test
\PLT@line

The `.....` make sure that `\PLT@test` has enough arguments:

```
  \expandafter\PLT@test\PLT@line.....\PLT@% %%HO: . added
  }%
```

\PLT@test

The `\PLT@test` macro finds the "interesting" commands in the LOT file, mainly to delimit parts:

```
%%HO/BJ: now patch \PLT@test,
%%HO/BJ: call \PLT@contentsline with 4 instead of 3 parameters
```

\PLT@contentsline
\mtc@string
\PLT@list
\PLT@lot
\PLT@read
\partend

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.

```
\long\def\PLT@test#1#2#3#4#5#6\PLT@{% %%HO: #6 added
  \ifx#1\contentsline
    \let\mtc@string\string
    \PLT@contentsline{#2}{#3}{#4}{#5}% %%HO: #4 added
    \let\mtc@string\relax
  \else\ifx#1\@input
      \edef\PLT@list{\PLT@list#2\relax}%
  \else\ifx#1\partend
      \immediate\closeout\tf@mtc
      \immediate\openout\tf@mtc=\jobname.mtc
  \else\ifx#1\partbegin
      \addtocounter{ptc}{-1}%
  \fi\fi\fi\fi
  \ifeof\@inputcheck\expandafter\PLT@lot
  \else\expandafter\PLT@read\fi}%
```

End of the part level stuff (begun in section 9.45 on page 297):

```
}%
```

9.53 Depth counters for sectlofs and sectlots

\AtBeginDocument
\c@lofdepth
\c@lotdepth
\newcounter
\setcounter

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:

```
\AtBeginDocument{%
 \@ifundefined{c@lofdepth}{}%
   {\newcounter{sectlofdepth}\setcounter{sectlofdepth}{2}}%
 \@ifundefined{c@lotdepth}{}%
   {\newcounter{sectlotdepth}\setcounter{sectlotdepth}{2}}%
}
```

9.54 Section-level commands

\if@mtc@chapter@undef@
\if@mtc@section@def@

The section-level commands are defined only if \chapter is *not* defined, hence in article-like document classes, and only if \section is defined:

```
\if@mtc@chapter@undef@
\if@mtc@section@def@
```

\firstsectionis
\adjuststc
\decrementstc
\incrementstc
\stc@rule
\stcindent
\columnwidth

We define the obsolete command \firstsectionis (with its harmless warning), the counter stc of secttocs, the \adjuststc, \decrementstc and \incrementstc commands, the depth counter secttocdepth 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).

```
\def\firstsectionis#1%
  {\mtcPackageWarning[<W0005>]{minitoc}%
     {\string\firstsectionis \space is an obsolete
      \MessageBreak
      command}%
     \@firstsectionis@used@true}
\newcounter{stc}\setcounter{stc}{0}%
\newcommand{\adjuststc}[1][1]{\addtocounter{stc}{#1}}%
\def\decrementstc{\addtocounter{stc}{-1}}%
\def\incrementstc{\addtocounter{stc}{+1}}%
\newcounter{secttocdepth}\setcounter{secttocdepth}{2}%
```

```
\def\stc@rule{\rule[3\p@]{\columnwidth}{.4\p@}\vspace*{2.6\p@}}%
\newlength\stcindent \stcindent=24\p@
```

9.55 Fonts commands for secttocs and co.

\stcfont
\stcSSfont
\stcSSSfont
\stcPfont
\stcSPfont
\slffont
\slfSfont
\sltfont
\sltSfont
\stifont

We define the fonts commands for the secttocs, sectlofs and sectlots and their titles:

```
\def\stcfont{\small\rmfamily\upshape\mdseries}    % secttoc
\def\stcSSfont{\small\rmfamily\upshape\bfseries}  % (subsections)
\let\stcSSSfont\stcfont    % (subsubsections)
\let\stcPfont\stcfont      % (paragraphs)
\let\stcSPfont\stcfont     % (subparagraphs)
\let\slffont\stcfont       % sectlof (figures)
\let\slfSfont\stcfont      % sectlof (subfigures)
\let\sltfont\stcfont       % sectlot (tables)
\let\sltSfont\stcfont      % sectlot (subtables)
\def\stifont{\large\rmfamily\upshape\bfseries}    % titles
```

9.56 Internal macros for title positionning

Some internal macros for title positionning, from the optional arguments of \dosecttoc and \secttoc commands (and siblings):

\l@sti
\c@sti
\r@sti
\e@sti
\n@sti

Centering, flushleft, flushright or empty titles (with a vertical correction for empty titles, from Frank Mittelbach):

```
\def\c@sti#1{\null\hfill #1\hfill\null}
\def\l@sti#1{\null #1\hfill\null}
\def\r@sti#1{\null\hfill #1\null}
\def\e@sti#1{\vspace{-\baselineskip}}
\def\n@sti#1{\vspace{-\baselineskip}}
```

\do@stitc
\df@stitc
\do@stilf
\df@stilf
\do@stilt
\df@stilt
\l@sti

By default, titles are flushleft.

```
\let\do@stitc\l@sti
\let\df@stitc\l@sti
\let\do@stilf\l@sti
\let\df@stilf\l@sti
\let\do@stilt\l@sti
\let\df@stilt\l@sti
```

9.57 The stc@verse environment

stc@verse
\iftightmtc
\ifktightmtc

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.

```
\def\stc@verse{\let\\=\@centercr
  \list{}{\itemsep\z@
          \itemindent \z@
          \listparindent \itemindent
          \partopsep \z@
          \iftightmtc \parsep \z@ \fi
          \ifktightmtc \parskip \z@ \fi
          \topsep 1ex %% 20060731 was \z@
          \leftmargin\stcindent
          \rightmargin\leftmargin}\item[]}
\def\endstc@verse{\nopagebreak[4]\endlist}
```

9.58 The \secttoc, \sectlof, and \sectlot commands

These three commands are very similar.

9.58.1 The \secttoc command

\secttoc
\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:

```
\def\secttoc{\@ifnextchar[{\secttoc@}{\secttoc@[d]}}
```

\secttoc@
\if@secttoc@used@
\if@longextensions@
\@tocfile
\thestc

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

```
\def\secttoc@[#1]{%
\global\@secttoc@used@true
\if@longextensions@%
\def\@tocfile{stc\thestc}%
\else
\def\@tocfile{S\thestc}%
\fi
```

\mtc@CkFile
\if@mtc@FE
\@tocfile

Then, we check the presence and the emptiness of the secttoc file and give a warning if it is not here or is empty:

```
\mtc@CkFile{\jobname.\@tocfile}
\if@mtc@FE
\mtcPackageInfo[<I0006>]{minitoc}%
   {\jobname.\@tocfile\space is empty}
\@mtc@empty@secttoc@true
\else
```

\beforesecttoc
\thispagesecttocstyle

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:

```
\beforesecttoc
\thispagesecttocstyle
```

\do@stitc
\e@sti
\n@sti
\c@sti
\l@sti
\r@sti
\df@sti
\mtc@CkStr
\stctitle
\if@mtc@FE
samepage

We begin a samepage environment, then treat the positionning argument. If the title is empty, we simulate the "e" positionning.

```
%%      \mtc@markboth{\MakeUppercase{\stctitle}}{\MakeUppercase{\stctitle}}%
        \relax\begin{samepage}%
        \if #1e\let\do@stitc\e@sti
        \else\if #1n\let\do@stitc\n@sti
        \else\if #1c\let\do@stitc\c@sti
        \else\if #1l\let\do@stitc\l@sti
        \else\if #1r\let\do@stitc\r@sti
        \else\if #1d\let\do@stitc\df@stitc
        \fi\fi\fi\fi\fi\fi
        \mtc@CkStr{\stctitle}\if@mtc@FE \let\do@stitc\e@sti\relax\fi
```

\raggedright
\parskip
\stcfont

We adjust some formatting parameters and avoid a page break between the title and the secttoc, then we set the font:

```
\raggedright
\parskip=\z@%
\reset@font\stcfont%
\parindent=\z@%
\nopagebreak[4]%
```

\stc@rule
tabular
\columnwidth
\stifont
\do@stitc
\mtc@v
\stctitle
\hline

The secttoc 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 secttoc.

```
\kern-0.8\baselineskip\nopagebreak[4]%
\par\noindent
```

```
        \nopagebreak[4]%
        \ifx\stc@rule\relax
        \begin{tabular}{@{}p{\columnwidth}@{}}
        \reset@font\stifont\do@stitc{\mtc@v\stctitle}\\
        \end{tabular}%
        \else
        \begin{tabular}{@{}p{\columnwidth}@{}}
        \reset@font\stifont\do@stitc{\mtc@v\stctitle}\\\hline
        \end{tabular}%
        \fi
```

\mtc@zrule
\mtc@BBR
\stcindent

Then, we adjust the position under the top rule and set the indentation and some formatting parameters:

```
        \nopagebreak[4]\null\leavevmode\mtc@zrule\\\mtc@BBR
        \leftmargin\stcindent
        \rightmargin\stcindent
        \itemindent=\z@\labelwidth=\z@%
        \labelsep=\z@\listparindent=\z@%
```

stc@verse
\c@tocdepth
\c@secttocdepth
\mtc@BBR

We enter in a stc@verse environment to format the secttoc. The toc depth is forced (locally) to secttocdepth. A little trick is necessary to adjust the position.

```
        \begin{stc@verse}\c@tocdepth=\c@secttocdepth%
        \leavevmode\\\mtc@BBR\vskip -.5\baselineskip
```

\stc@pgno
\@dottedtocline
\@undottedtocline
\mtc@hook@beforeinputfile
\stc@setform
\@tocfile
\mtc@strut
stc@verse

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 \mtcsetformat are activated via \stc@setform. Then the secttoc file is inserted, followed by a strut, and the stc@verse environment is terminated.

```
\begingroup
  \makeatletter
  \stc@setform%
  \@ifundefined{stc@pgno}%
  {\let\@dottedtocline\@undottedtocline}{}
  \@fileswfalse\mtc@hook@beforeinputfile
  \stc@setform%
  \@input{\jobname.\@tocfile}
  \vspace{-1ex} \vspace{-\baselineskip}
  \leavevmode\mtc@strut
  \global\@nobreakfalse\endgroup
        \end{stc@verse}%
```

\stc@rule
\mtc@zrule
samepage
\afterseccttoc

The final part is just to add the bottom rule, if necessary, a possible page break and \afterseccttoc.

```
\kernafterseccttoc
\nopagebreak[4]\stc@rule\null\leavevmode\\%
\vskip-1.0\baselineskip\mtc@zrule\end{samepage}%
\par\pagebreak[1]\vspace*{-1ex}\afterseccttoc\fi}%
```

9.58.2 The \sectlof command

\sectlof
\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:

```
\def\sectlof{\@ifnextchar[{\sectlof@}{\sectlof@[d]}}
```

\sectlof@
\if@sectlof@used@
\if@longextensions@
\@tocfile
\thestc

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

```
\def\sectlof@[#1]{%
\global\@sectlof@used@true
\if@longextensions@%
\def\@tocfile{slf\thestc}%
\else
\def\@tocfile{H\thestc}%
\fi
```

\mtc@CkFile
\if@mtc@FE
\@tocfile

Then, we check the presence and the emptiness of the sectlof file and give a warning if it is not here or is empty:

```
\mtc@CkFile{\jobname.\@tocfile}
\if@mtc@FE
\mtcPackageInfo[<I0006>]{minitoc}%
   {\jobname.\@tocfile\space is empty}
\@mtc@empty@sectlof@true
\else
```

\beforesectlof
\thispagesectlofstyle

If the sectlof file is present and not empty, we can insert it, but we must add some presentation code: first, \beforesectlof, of course, and the page style feature:

```
\thispagesectlofstyle
%%   \mtc@markboth{\MakeUppercase{\slftitle}}{\MakeUppercase{\slftitle}}%
\beforesectlof
```

\do@stilf
\e@sti
\n@sti
\c@sti
\l@sti
\r@sti
\df@sti
\mtc@CkStr
\slftitle
\if@mtc@FE
samepage

We begin a samepage environment, then treat the positionning argument. If the title is empty, we simulate the "e" positionning.

```
\relax\begin{samepage}%
\if #1e\let\do@stilf\e@sti
\else\if #1n\let\do@stilf\n@sti
\else\if #1c\let\do@stilf\c@sti
\else\if #1l\let\do@stilf\l@sti
\else\if #1r\let\do@stilf\r@sti
\else\if #1d\let\do@stilf\df@stilf
\fi\fi\fi\fi\fi\fi
\mtc@CkStr{\slftitle}\if@mtc@FE \let\do@stilf\e@sti\relax\fi
```

\raggedright
\parskip
\slffont

We adjust some formatting parameters and avoid a page break between the title and the sectlof, then we set the font:

```
\raggedright
\parskip=\z@%
\reset@font\slffont%
\parindent=\z@%
\nopagebreak[4]%
```

\slf@rule
tabular
\stifont
\columnwidth
\do@stilf
\mtc@v
\slftitle
\hline

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.

```
\kern-0.8\baselineskip\nopagebreak[4]%
\par\noindent
\ifx\slf@rule\relax
\begin{tabular}{@{}p{\columnwidth}@{}}
\reset@font\stifont\do@stilf{\mtc@v\slftitle}\\
\end{tabular}%
\else
\begin{tabular}{@{}p{\columnwidth}@{}}
\mtc@hstrut
\reset@font\stifont\do@stilf{\mtc@v\slftitle}\\\hline
\end{tabular}%
\fi
```

\mtc@zrule
\mtc@BBR
\stcindent

Then, we adjust the position under the top rule and set the indentation and some formatting parameters:

```
\nopagebreak[4]\null\leavevmode\mtc@zrule\\\mtc@BBR
\leftmargin\stcindent
\rightmargin\stcindent
\itemindent=\z@\labelwidth=\z@%
\labelsep=\z@\listparindent=\z@%
```

stc@verse
\c@tocdepth
\c@sectlofdepth
\mtc@BBR

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.

```
        \begin{stc@verse}%
        \@ifundefined{c@lofdepth}%
          {}%
          {\c@lofdepth=\c@sectlofdepth
           \ifnum\c@lofdepth<1\relax\c@lofdepth=1\fi}
        \leavevmode\\\mtc@BBR\vskip -.5\baselineskip
```

\slf@pgno
\@dottedtocline
\@undottedtocline
\mtc@hook@beforeinputfile
\slf@setform
\@tocfile
\mtc@strut
stc@verse

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 \mtcsetformat are activated via \slf@setform. Then the sectlof file is inserted, followed by a strut, and the stc@verse environment is terminated.

```
\begingroup
  \makeatletter
  \@ifundefined{slf@pgno}%
  {\let\@dottedtocline\@undottedtocline}{}
  \@fileswfalse\mtc@hook@beforeinputfile
  \slf@setform%
  \@input{\jobname.\@tocfile}
  \global\@nobreakfalse\endgroup
        \end{stc@verse}%
```

\stc@rule
\mtc@zrule
samepage
\aftersectlof

The final part is just to add the bottom rule, if necessary, a possible page break and \aftersectlof. The blank line (\\) is essential.

```
        \kernaftersectlof
        \nopagebreak[4]\slf@rule\null\leavevmode\\%
        \vskip-1.0\baselineskip\mtc@zrule\end{samepage}%
        \par\pagebreak[1]\vspace*{-1ex}\aftersectlof\fi}%
```

9.58.3 The \sectlot command

\sectlot
\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:

```
\def\sectlot{\@ifnextchar[{\sectlot@}{\sectlot@[d]}}
```

\sectlot@
\if@sectlot@used@
\if@longextensions@
\@tocfile
\thestc

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

```
\def\sectlot@[#1]{%
\global\@sectlot@used@true
\if@longextensions@%
\def\@tocfile{slt\thestc}%
\else
\def\@tocfile{I\thestc}%
\fi
```

\mtc@CkFile
\if@mtc@FE
\@tocfile

Then, we check the presence and the emptiness of the sectlot file and give a warning if it is not here or is empty:

```
        \mtc@CkFile{\jobname.\@tocfile}
        \if@mtc@FE
        \mtcPackageInfo[<I0006>]{minitoc}%
           {\jobname.\@tocfile\space is empty}
        \@mtc@empty@sectlot@true
        \else
```

\beforesectlot
\thispagesectlotstyle

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:

```
        \thispagesectlotstyle
%%      \mtc@markboth{\MakeUppercase{\slttitle}}{\MakeUppercase{\slttitle}}%
        \beforesectlot
```

\do@stilt
\e@sti
\n@sti
\c@sti
\l@sti
\r@sti
\df@sti
\mtc@CkStr
\slttitle
\if@mtc@FE
samepage

We begin a samepage environment, then treat the positionning argument. If the title is empty, we simulate the "e" positionning.

```
        \relax\begin{samepage}%
        \if #1e\let\do@stilt\e@sti
        \else\if #1n\let\do@stilt\n@sti
        \else\if #1c\let\do@stilt\c@sti
        \else\if #1l\let\do@stilt\l@sti
        \else\if #1r\let\do@stilt\r@sti
        \else\if #1d\let\do@stilt\df@stilt
        \fi\fi\fi\fi\fi\fi
        \mtc@CkStr{\slttitle}\if@mtc@FE \let\do@stilt\e@sti\relax\fi
```

\raggedright
\parskip
\sltfont

We adjust some formatting parameters and avoid a page break between the title and the sectlot, then we set the font:

```
\raggedright
\parskip=\z@%
\reset@font\sltfont%
\parindent=\z@%
\nopagebreak[4]%
```

\stc@rule
tabular
\stifont
\columnrule
\do@stilt
\mtc@v
\slttitle
\hline

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.

```
\kern-0.8\baselineskip\nopagebreak[4]%
\par\noindent
\ifx\slt@rule\relax
\begin{tabular}{@{}p{\columnwidth}@{}}
\reset@font\stifont\do@stilt{\mtc@v\slttitle}\\
\end{tabular}%
\else
\begin{tabular}{@{}p{\columnwidth}@{}}
\mtc@hstrut
\reset@font\stifont\do@stilt{\mtc@v\slttitle}\\\hline
\end{tabular}%
\fi
```

\mtc@zrule
\mtc@BBR
\stcindent

Then, we adjust the position under the top rule and set the indentation and some formatting parameters:

```
\nopagebreak[4]\null\leavevmode\mtc@zrule\\\mtc@BBR
\leftmargin\stcindent
\rightmargin\stcindent
\itemindent=\z@\labelwidth=\z@%
\labelsep=\z@\listparindent=\z@%
```

stc@verse
\c@tocdepth
\c@sectlotdepth
\mtc@BBR

We enter in a `stc@verse` environment to format the sectlot. The toc depth is forced (locally) to `sectlotdepth`. A little trick is necessary to adjust the position.

```
\begin{stc@verse}%
\@ifundefined{c@lotdepth}%
  {}%
  {\c@lotdepth=\c@sectlotdepth
\ifnum\c@lotdepth<1\relax\c@lotdepth=1\fi}
\leavevmode\\\mtc@BBR\vskip -.5\baselineskip
```

\slt@pgno
\@dottedtocline
\@undottedtocline
\mtc@hook@beforeinputfile
\slt@setform
\@tocfile
\mtc@strut
stc@verse

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 `\mtcsetformat` are activated via `\slt@setform`. Then the sectlot file is inserted, followed by a strut, and the `stc@verse` environment is terminated.

```
\begingroup
 \makeatletter
 \@ifundefined{slt@pgno}%
 {\let\@dottedtocline\@undottedtocline}{}
  \gdef\thestc{\arabic{stc}}
 \@fileswfalse\mtc@hook@beforeinputfile
 \slt@setform%
 \@input{\jobname.\@tocfile}
 \global\@nobreakfalse\endgroup
       \end{stc@verse}%
```

\stc@rule
\mtc@zrule
samepage
\aftersectlot

The final part is just to add the bottom rule, if necessary, a possible page break and \aftersectlot.

```
       \kernaftersectlof
       \nopagebreak[4]\slt@rule\null\leavevmode\\%
       \vskip-1.0\baselineskip\mtc@zrule\end{samepage}%
       \par\pagebreak[1]\vspace*{-1ex}\aftersectlot\fi}%
```

9.59 Auxiliary internal commands, section level

\l@xsect
\l@xsection
\@dottedtocline
\l@schapter
\xsect
\xsection
\schapter

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).

```
\def\l@xsect{\@dottedtocline{\@M}{1.0em}{2.3em}}
\def\l@xsection{\@dottedtocline{\@M}{1.0em}{2.3em}}
\def\l@schapter{\@dottedtocline{1}{1.0em}{2.3em}}
\def\xsect{xsect}
\def\xsection{xsection}
\def\schapter{schapter}
```

9.60 Patching the \section command (continued)

\@sect
\addcontentsline

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).

```
\let\sv@sect\@sect
\gdef\@sect#1#2#3#4#5#6[#7]#8{%
\ifnum #2=1\relax
```

```
   \addcontentsline{lof}{xsect}{#7}%
   \addcontentsline{lot}{xsect}{#7}%
\fi
\sv@sect{#1}{#2}{#3}{#4}{#4}{#5}{#6}[{#7}]{#8}}
```

\section
\starsection
\addcontentsline

If it is a section (unstarred or starred via \starsection), we add a xsect entry in the LOF and in the LOT.

```
\def\@sect#1#2#3#4#5#6[#7]#8{
\expandafter
\ifx\csname #1\endcsname\section\relax
   \addcontentsline{lof}{xsect}{#7}%
   \addcontentsline{lot}{xsect}{#7}%
\fi
\ifx\csname #1\endcsname\starsection\relax
   \addcontentsline{lof}{xsect}{#7}%
   \addcontentsline{lot}{xsect}{#7}%
\fi
```

\@svsec
\refstepcounter
\@tempskipa
\@hangfrom
\addcontentsline
\numberline
\@svsechd
\@xsect

And the remainder of the section header formatting:

```
\ifnum #2>\c@secnumdepth\relax
   \let\@svsec\@empty
\else
   \refstepcounter{#1}%
   \edef\@svsec{\csname the#1\endcsname\hskip 1em}%
\fi
\@tempskipa #5\relax
\ifdim \@tempskipa>\z@
   \begingroup #6\relax
      \@hangfrom{\hskip #3\relax\@svsec}%
         {\interlinepenalty \@M #8\par}%
   \endgroup
   \csname #1mark\endcsname{#7}\addcontentsline
   {toc}{#1}{\ifnum #2>\c@secnumdepth\relax
             \else
             \protect\numberline{\csname the#1\endcsname}%
             \fi
             #7}%
\else
   \def\@svsechd{#6\hskip #3\relax
   \@svsec #8\csname #1mark\endcsname
   {#7}\addcontentsline
   {toc}{#1}{\ifnum #2>\c@secnumdepth\relax
             \else
                \protect\numberline{\csname the#1\endcsname}
             \fi
   #7}}%
\fi
\@xsect{#5}}
```

\@sect
\sectbegin
\sectend
\stc@sect
\addtocontents

Then we patch the unstarred branch (\@sect). We define also the delimiting commands \sectbegin and \sectend commands. We do not add \sectbegin if it is a subsection or deeper.

```
\let\stc@sect\@sect
\def\@sect#1#2#3#4#5#6[#7]#8{%
  \ifnum #2<1 \relax
  \addtocontents{toc}{\protect\sectbegin}
  \fi
  \stc@sect{#1}{#2}{#3}{#4}{#5}{#6}[{#7}]{#8}}
\let\sectend\relax
\let\sectbegin\relax
```

9.61 The \dosecttoc command and siblings

The \dosecttoc command is very similar to \dominitoc.

\dosecttoc
\@dosecttoc
\STC@next

The \dosecttoc command extracts information from the .toc file and creates the .stc⟨*N*⟩ files (.stc becomes .S on MS-DOS).

```
\def\@dosecttoc#1{{%
  \makeatletter
  \setcounter{stc}{0}
  \STC@next#1.toc\relax\\}\setcounter{stc}{0}}
```

\dosectlof
\@dosectlof
\SLF@next

The \dosectlof command extracts information from the .lof file and creates the .slf⟨*N*⟩ files (.slf becomes .H on MS-DOS).

```
\def\@dosectlof#1{{%
  \makeatletter
  \setcounter{stc}{0}
  \SLF@next#1.lof\relax\\}\setcounter{stc}{0}}
```

\dosectlot
\@dosectlot
\PLT@next

The \dosectlot command extracts information from the .lot file and creates the .slt⟨*N*⟩ files (.slt becomes .V on MS-DOS).

```
\def\@dosectlot#1{{%
  \makeatletter
  \setcounter{stc}{0}
  \SLT@next#1.lot\relax\\}\setcounter{stc}{0}}
```

\dosecttoc
\dosectlof
\dosectlot

We define the user-level macros, who detect the optional argument:

```
\def\dosecttoc{\@ifnextchar[{\dosecttoc@}{\dosecttoc@[l]}}
\def\dosectlof{\@ifnextchar[{\dosectlof@}{\dosectlof@[l]}}
\def\dosectlot{\@ifnextchar[{\dosectlot@}{\dosectlot@[l]}}
```

\dosecttoc@
\if@mtc@hints@
\@mtc@hints@given@true
\if@dosecttoc@used@
\df@stitc
\e@sti
\n@sti
\c@sti
\l@sti
\r@sti

We treat the optional argument of \dosecttoc (it becomes the default position for titles of secttocs) and flag this macro as used; a hint detects any spurious invocation.

```
\def\dosecttoc@[#1]{%
\if@mtc@hints@
   \if@dosecttoc@used@
   \mtcPackageInfo[<I0045>]{minitoc(hints)}%
        {The \string\dosecttoc \space command
         \MessageBreak
         has been invoked more than once
         \MessageBreak}
   \global\@mtc@hints@given@true
   \fi
\fi
\global\@dosecttoc@used@true
\if #1e\let\df@stitc\e@sti%
\else\if #1n\let\df@stitc\n@sti%
\else\if #1c\let\df@stitc\c@sti%
\else\if #1l\let\df@stitc\l@sti%
\else\if #1r\let\df@stitc\r@sti%
\fi\fi\fi\fi\fi%
\@@dosecttoc}
```

\dosectlof@
\if@mtc@hints@
\@mtc@hints@given@true
\if@dosectlof@used@
\df@stilf
\e@sti
\n@sti
\c@sti
\l@sti
\r@sti

We treat the optional argument of \dosectlof (it becomes the default position for titles of sectlofs) and flag this macro as used; a hint detects any spurious invocation.

```
\def\dosectlof@[#1]{%
\if@mtc@hints@
   \if@dosectlof@used@
   \mtcPackageInfo[<I0045>]{minitoc(hints)}%
        {The \string\dosectlof \space command
         \MessageBreak
         has been invoked more than once
         \MessageBreak}
   \global\@mtc@hints@given@true
   \fi
\fi
\global\@dosectlof@used@true
\if #1e\let\df@stilf\e@sti%
\else\if #1n\let\df@stilf\n@sti%
\else\if #1c\let\df@stilf\c@sti%
\else\if #1l\let\df@stilf\l@sti%
\else\if #1r\let\df@stilf\r@sti%
\fi\fi\fi\fi\fi%
```

```
\@@dosectlof}
```

\dosectlot@
\if@mtc@hints@
\@mtc@hints@given@true
\if@dosectlot@used@
\df@stilt
\e@sti
\n@sti
\c@sti
\l@sti
\r@sti

We treat the optional argument of `\dosectlot` (it becomes the default position for titles of sectlofs) and flag this macro as used; a hint detects any spurious invocation.

```
\def\dosectlot@[#1]{%
\if@mtc@hints@
   \if@dosectlot@used@
   \mtcPackageInfo[<I0045>]{minitoc(hints)}%
        {The \string\dosectlot \space command
         \MessageBreak
         has been invoked more than once
         \MessageBreak}
   \global\@mtc@hints@given@true
   \fi
\fi
\global\@dosectlot@used@true
\if #1e\let\df@stilt\e@sti%
\else\if #1n\let\df@stilt\n@sti%
\else\if #1c\let\df@stilt\c@sti%
\else\if #1l\let\df@stilt\l@sti%
\else\if #1r\let\df@stilt\r@sti%
\fi\fi\fi\fi\fi%
\@@dosectlot}
```

\@@dosecttoc
\@@dosectlof
\@@dosectlot
\tf@mtc

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

```
\def\@@dosecttoc{\@dosecttoc{\jobname}\immediate\closeout\tf@mtc}
\def\@@dosectlof{\@dosectlof{\jobname}\immediate\closeout\tf@mtc}
\def\@@dosectlot{\@dosectlot{\jobname}\immediate\closeout\tf@mtc}
```

\STC@next
\STC@list
\STC@loop

Processing the next entry in the list and remove it from the head of the list:

```
\def\STC@next#1\relax#2\\{%
   \edef\STC@list{#2}%
   \STC@loop{#1}}
```

\STC@toc
\STC@list
\STC@explist

Check if the list is empty:

```
\def\STC@toc{%
   \ifx\STC@list\@empty\else\expandafter\STC@explist\fi}
```

\STC@contentsline
\section
\thestc
\tf@mtc
\stcname
\MTC@WriteContentsLine

The macro \STC@contentsline analyses the lines read from the TOC file. If \section is found, the stc counter is incremented and a new secttoc file is created.

```
\def\STC@contentsline#1#2#3#4{% %%HO: #4 added
  \gdef\thestc{\arabic{stc}}% %%HO: space removed
  \expandafter\ifx\csname #1\endcsname\section
    \stepcounter{stc}%
%%    \def\thestc{\arabic{stc}} %% HO: removed
```

\if@longextensions@
\stcname

We test if long or short extensions are used, to build the name of the mini-table file, then open it:

```
    \if@longextensions@%
      \mtcPackageInfo[<I0033>]{minitoc}%
         {Writing\space\jobname.stc\thestc}%
      \def\stcname{\jobname.stc\thestc}%
    \else
      \mtcPackageInfo[<I0033>]{minitoc}%
         {Writing\space\jobname.S\thestc}%
      \def\stcname{\jobname.S\thestc}%
    \fi
    \immediate\closeout\tf@mtc
    \immediate\openout\tf@mtc=\stcname
  \fi
```

\mtc@toks
\MTC@WriteCoffeeline
\MTC@WriteContentsline
\coffee
\subsection
\subsubsection
\paragraph
\subparagraph

Now, we filter the relevant contents lines:

```
  \mtc@toks{\noexpand\leavevmode #2}%
  \expandafter\ifx\csname #1\endcsname\coffee
    \MTC@WriteCoffeeline{#1}{#3}%
  \fi
  \expandafter\ifx\csname #1\endcsname\subsection
    \MTC@WriteContentsline{#1}{stcSS}{#3}{#4}%
  \fi
  \expandafter\ifx\csname #1\endcsname\subsubsection
    \MTC@WriteContentsline{#1}{stcSSS}{#3}{#4}%
  \fi
  \expandafter\ifx\csname #1\endcsname\paragraph
    \MTC@WriteContentsline{#1}{stcP}{#3}{#4}%
  \fi
  \expandafter\ifx\csname #1\endcsname\subparagraph
    \MTC@WriteContentsline{#1}{stcSP}{#3}{#4}%
  \fi
```

\starsection
\stepcounter
\thestc
\arabic
\if@longextensions@
\stcname
\closeout
\openout

A starred section terminates the current section and creates a new secttoc file:

```
  \ifx\csname #1\endcsname\starsection
    \stepcounter{stc}%
```

```
    \gdef\thestc{\arabic{stc}}
    \if@longextensions@%
      \mtcPackageInfo[<I0033>]{minitoc}%
         {Writing\space\jobname.stc\thestc}%
      \def\stcname{\jobname.stc\thestc}%
    \else
      \mtcPackageInfo[<I0033>]{minitoc}%
         {Writing\space\jobname.S\thestc}%
      \def\stcname{\jobname.S\thestc}%
    \fi
    \immediate\closeout\tf@mtc
    \immediate\openout\tf@mtc=\stcname
  \fi
```

\MTC@WriteContentsline
\starsubsection
\starsubsubsection
\starparagraph
\starsubparagraph

We process the entries for starred sectionning commands:

```
  \expandafter\ifx\csname #1\endcsname\starsubsection
    \MTC@WriteContentsline{#1}{stcSS}{#3}{#4}%
  \fi
  \expandafter\ifx\csname #1\endcsname\starsubsubsection
    \MTC@WriteContentsline{#1}{stcSSS}{#3}{#4}%
  \fi
  \expandafter\ifx\csname #1\endcsname\starparagraph
    \MTC@WriteContentsline{#1}{stcP}{#3}{#4}%
  \fi
  \expandafter\ifx\csname #1\endcsname\starsubparagraph
    \MTC@WriteContentsline{#1}{stcSP}{#3}{#4}%
  \fi
}
```

\STC@explist
\STC@next
\STC@list

The loop to read the lines of the TOC file; expands the list of entries and call \STC@next to process the first one.

```
\def\STC@explist{\expandafter\STC@next\STC@list\\}
```

\STC@loop
\STC@toc
\STC@read

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

```
\def\STC@loop#1{\openin\@inputcheck#1\relax
  \ifeof\@inputcheck
    \mtcPackageWarning[<W0016>]{minitoc}%
       {No file #1
        \MessageBreak
        SECTTOCS NOT PREPARED}%
    \expandafter\STC@toc
  \else
    \mtcPackageInfo[<I0026>]{minitoc}%
       {PREPARING SECTTOCS FROM #1}%
    \expandafter\STC@read\fi}
```

\STC@read
\STC@line

Read the next entry of the .toc file.

```
\def\STC@read{%
  \read\@inputcheck to\STC@line
```

\STC@test
\STC@line

The ..... make sure that \STC@test has enough arguments:

```
  \expandafter\STC@test\STC@line.....\STC@% %%HO: . added
  }%
```

\STC@test
\STC@contentsline
\mtc@string
\STC@list
\STC@toc
\STC@read
\sectend
\sectbegin

The \STC@test macro finds the "interesting" commands in the TOC file, mainly to delimit sections;

```
\long\def\STC@test#1#2#3#4#5#6\STC@{% %%HO: #6 added
  \ifx#1\contentsline
    \let\mtc@string\string
    \STC@contentsline{#2}{#3}{#4}{#5}% %%HO: #4 added
    \let\mtc@string\relax
  \else\ifx#1\@input
      \edef\STC@list{\STC@list#2\relax}%
  \else\ifx#1\sectend
      \immediate\closeout\tf@mtc
      \immediate\openout\tf@mtc=\jobname.mtc
  \else\ifx#1\sectbegin
      \addtocounter{stc}{-1}%
  \fi\fi\fi\fi
  \ifeof\@inputcheck\expandafter\STC@toc
  \else\expandafter\STC@read\fi}%
```

\SLF@next
\SLF@list
\SLF@loop

Processing the next entry in the list and remove it from the head of the list:

```
\def\SLF@next#1\relax#2\\{%
  \edef\SLF@list{#2}%
  \SLF@loop{#1}}
```

\SLF@lof
\SLF@list
\SLF@explist

Check if the list is empty:

```
\def\SLF@lof{%
  \ifx\SLF@list\@empty\else\expandafter\SLF@explist\fi}
```

\SLF@contentsline
\section
\thestc
\tf@mtc
\slfname
\MTC@WriteContentsLine

The macro \SLF@contentsline analyses the lines read from the LOF file. If \section is found, the stc counter is incremented and a new sectlof file is created.

```
\def\SLF@contentsline#1#2#3#4{% %%HO: #4 added
```

```
  \gdef\thestc{\arabic{stc}}% %%HO: space removed
  \expandafter\ifx\csname #1\endcsname\xsect
    \stepcounter{stc}%
%%    \gdef\thestc{\arabic{stc}} %%HO: removed
```

\if@longextensions@
\slfname

We test if long or short extensions are used, to build the name of the mini-table file, then open it:

```
    \if@longextensions@%
      \mtcPackageInfo[<I0033>]{minitoc}%
         {Writing\space\jobname.slf\thestc}%
      \def\slfname{\jobname.slf\thestc}%
    \else
      \mtcPackageInfo[<I0033>]{minitoc}%
         {Writing\space\jobname.H\thestc}%
      \def\slfname{\jobname.H\thestc}%
    \fi
    \immediate\closeout\tf@mtc
    \immediate\openout\tf@mtc=\slfname
  \fi
```

\mtc@toks
\MTC@WriteContentsline
\figure
\subfigure

Now, we filter the relevant contents lines:

```
  \mtc@toks{\noexpand\leavevmode #2}%
  \expandafter\ifx\csname #1\endcsname\figure
    \MTC@WriteContentsline{#1}{slf}{#3}{#4}%
  \fi
  \expandafter\ifx\csname #1\endcsname\subfigure
    \MTC@WriteContentsline{#1}{slfS}{#3}{#4}%
  \fi
}
```

\SLF@explist
\SLF@next
\SLF@list

The loop to read the lines of the LOF file; expands the list of entries and call \SLF@next to process the first one.

```
\def\SLF@explist{\expandafter\SLF@next\SLF@list\\}
```

\SLF@loop
\SLF@lof
\SLF@read

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

```
\def\SLF@loop#1{\openin\@inputcheck#1\relax
  \ifeof\@inputcheck
    \mtcPackageWarning[<W0014>]{minitoc}%
       {No file #1
        \MessageBreak
        SECTLOFS NOT PREPARED}%
```

```
    \expandafter\SLF@lof
  \else
    \mtcPackageInfo[<I0036>]{minitoc}%
       {PREPARING SECTLOFS FROM #1}%
    \expandafter\SLF@read\fi}
```

Read the next entry of the `.lof` file.

\SLF@read
\SLF@test
\SLF@line

The ..... make sure that \SLF@test has enough arguments:

```
\def\SLF@read{%
  \read\@inputcheck to\SLF@line
  \expandafter\SLF@test\SLF@line.....\SLF@% %%HO: . added
  }%
```

\SLF@test
\SLF@contentsline
\mtc@string
\SLF@list
\SLF@lof
\SLF@read
\sectend
\sectbegin

The \SLF@test macro finds the "interesting" commands in the LOF file, mainly to delimit sections;

```
\long\def\SLF@test#1#2#3#4#5#6\SLF@{% %%HO: #6 added
  \ifx#1\contentsline
    \let\mtc@string\string
    \SLF@contentsline{#2}{#3}{#4}{#5}% %%HO: #4 added
    \let\mtc@string\relax
  \else\ifx#1\@input
      \edef\SLF@list{\SLF@list#2\relax}%
  \else\ifx#1\sectend
      \immediate\closeout\tf@mtc
      \immediate\openout\tf@mtc=\jobname.mtc
  \else\ifx#1\sectbegin
      \addtocounter{stc}{-1}%
  \fi\fi\fi\fi
  \ifeof\@inputcheck\expandafter\SLF@lof
  \else\expandafter\SLF@read\fi}%
```

\SLT@next
\SLT@list
\SLT@loop

Processing the next entry in the list and remove it from the head of the list:

```
\def\SLT@next#1\relax#2\\{%
  \edef\SLT@list{#2}%
  \SLT@loop{#1}}
```

\SLT@lot
\SLT@list
\SLT@explist

Check if the list is empty:

```
\def\SLT@lot{%
  \ifx\SLT@list\@empty\else\expandafter\SLT@explist\fi}
```

\SLT@contentsline
\section
\thestc
\tf@mtc
\sltname
\MTC@WriteContentsLine

The macro \SLT@contentsline analyses the lines read from the LOT file. If \section is found, the stc counter is incremented and a new sectlot file is created.

```
\def\SLT@contentsline#1#2#3#4{% %%HO: #4 added
  \gdef\thestc{\arabic{stc}}% %%HO: space removed
  \expandafter\ifx\csname #1\endcsname\xsect
    \stepcounter{stc}%
%%    \gdef\thestc{\arabic{stc}} %%HO: removed
```

\if@longextensions@
\sltname

We test if long or short extensions are used, to build the name of the mini-table file, then open it:

```
    \if@longextensions@%
      \mtcPackageInfo[<I0033>]{minitoc}%
         {Writing\space\jobname.slt\thestc}%
      \def\sltname{\jobname.slt\thestc}%
    \else
      \mtcPackageInfo[<I0033>]{minitoc}%
         {Writing\space\jobname.V\thestc}%
      \def\sltname{\jobname.V\thestc}%
    \fi
    \immediate\closeout\tf@mtc
    \immediate\openout\tf@mtc=\sltname
  \fi
```

\mtc@toks
\MTC@WriteContentsline
\table
\subtable

Now, we filter the relevant contents lines:

```
  \mtc@toks{\noexpand\leavevmode #2}%
  \expandafter\ifx\csname #1\endcsname\table
    \MTC@WriteContentsline{#1}{slt}{#3}{#4}%
  \fi
  \expandafter\ifx\csname #1\endcsname\subtable
    \MTC@WriteContentsline{#1}{sltS}{#3}{#4}%
  \fi
}
```

\SLT@explist
\SLT@next
\SLT@list

The loop to read the lines of the LOT file; expands the list of entries and call \SLT@next to process the first one.

```
\def\SLT@explist{\expandafter\SLT@next\SLT@list\\}
```

\SLT@loop
\SLT@lot
\SLT@read

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

```
\def\SLT@loop#1{\openin\@inputcheck#1\relax
  \ifeof\@inputcheck
```

```
    \mtcPackageWarning[<W0015>]{minitoc}%
       {No file #1
        \MessageBreak
        SECTLOTS NOT PREPARED}%
    \expandafter\SLT@lot
  \else
    \mtcPackageInfo[<I0039>]{minitoc}%
       {PREPARING SECTLOTS FROM #1}%
    \expandafter\SLT@read\fi}
```

Read the next entry of the `.lot` file.

\SLT@read
\SLT@test
\SLT@line

The `.....` make sure that `\SLT@test` has enough arguments:

```
\def\SLT@read{%
  \read\@inputcheck to\SLT@line
  \expandafter\SLT@test\SLT@line.....\SLT@% %%HO: . added
  }%
```

\SLT@test
\SLT@contentsline
\mtc@string
\SLT@list
\SLT@lot
\SLT@read
\sectend
\sectbegin

The `\SLT@test` macro finds the "interesting" commands in the LOT file, mainly to delimit sections;

```
\long\def\SLT@test#1#2#3#4#5#6\SLT@{% %%HO: #6 added
  \ifx#1\contentsline
    \let\mtc@string\string
    \SLT@contentsline{#2}{#3}{#4}{#5}% %%HO: #4 added
    \let\mtc@string\relax
  \else\ifx#1\@input
      \edef\SLT@list{\SLT@list#2\relax}%
  \else\ifx#1\sectend
      \immediate\closeout\tf@mtc
      \immediate\openout\tf@mtc=\jobname.mtc
  \else\ifx#1\sectbegin
      \addtocounter{stc}{-1}%
  \fi\fi\fi\fi
  \ifeof\@inputcheck\expandafter\SLT@lot
  \else\expandafter\SLT@read\fi}%
```

9.62 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 that test:

```
\fi% end of \if@mtc@section@def@
\fi% end of \if@mtc@chapter@undef@
```

9.63 The \mtcprepare command

\mtcprepare
\@ifnextchar
\mtcprepare@
\@ifundefined
\IfFileExists
\jobname
\doparttoc
\dopartlof
\dopartlot
\dominitoc
\dominilof
\dominilot
\dosecttoc
\dosectlof
\dosectlot
\mtcPackageInfo

This command tests the availability of the \do... minitoc preparation commands and of the contents files, then calls as much as possible of these preparation commands. A hint is given.

```
\def\mtcprepare{\@ifnextchar[{\mtcprepare@}{\mtcprepare@[l]}}%
\def\mtcprepare@[#1]{%
\@ifundefined{part}{}{%
\IfFileExists{\jobname.toc}{\doparttoc[#1]}{}%
\IfFileExists{\jobname.lof}{\dopartlof[#1]}{}%
\IfFileExists{\jobname.lot}{\dopartlot[#1]}{}%
}%
\@ifundefined{chapter}{%
  \@ifundefined{section}{}%
  {%
\IfFileExists{\jobname.toc}{\dosecttoc[#1]}{}%
\IfFileExists{\jobname.lof}{\dosectlof[#1]}{}%
\IfFileExists{\jobname.lot}{\dosectlot[#1]}{}%
  }%
}{%
\IfFileExists{\jobname.toc}{\dominitoc[#1]}{}%
\IfFileExists{\jobname.lof}{\dominilof[#1]}{}%
\IfFileExists{\jobname.lot}{\dominilot[#1]}{}%
}%
\if@mtc@hints@
  \@mtc@hints@given@true
  \mtcPackageInfo[<I0048>]{minitoc(hints)}%
     {Using \string\mtcprepare\space may induce some
      \MessageBreak
      hints about the preparation commands,
      \MessageBreak
      because it invokes ALL the preparation
      \MessageBreak
      commands allowed by the document class,
      \MessageBreak
      without any previous check\@gobble}%
\fi
}
```

9.64 Necessary \l@... commands

\l@listof
\l@starpart
\l@starchapter
\l@starsection
\l@starsubsection
\l@starsubsubsection
\l@starparagraph
\l@starsubparagraph

Some \l@... commands (analog to \l@section or \l@paragraph) are required to format some entries in the mini-tables, for starred sectionning commands essentially:

```
\@ifundefined{section}{}{\let\l@listof\l@section}
\@ifundefined{chapter}{}{\let\l@listof\l@chapter}
\@ifundefined{part}{}{\let\l@starpart\l@part}
\@ifundefined{chapter}{}{\let\l@starchapter\l@chapter}
```

```
\@ifundefined{section}{}{\let\l@starsection\l@section}
\@ifundefined{subsection}{}{\let\l@starsubsection\l@subsection}
\@ifundefined{subsubsection}{}{\let\l@starsubsubsection\l@subsubsection}
\@ifundefined{paragraph}{}{\let\l@starparagraph\l@paragraph}
\@ifundefined{subparagraph}{}{\let\l@starsubparagraph\l@subparagraph}
```

9.65 The horizontal rules and their default values

\noptcrule
\nomtcrule
\nostcrule
\ptcrule
\mtcrule
\stcrule
\noplfrule
\nomlfrule
\noslfrule
\plfrule
\mlfrule
\slfrule
\nopltrule
\nomltrule
\nosltrule
\pltrule
\mltrule
\sltrule
\columnwidth

We define here the various commands to activate ou inhibit the horizontal rules in the various kinds of mini-tables. Each such command is an indirect definition of the corresponding horizontal rule. The rules are .4 pt high horizontal rules.

```
\def\noptcrule{\let\ptc@rule\relax}
\def\nomtcrule{\let\mtc@rule\relax}
\def\nostcrule{\let\stc@rule\relax}
\def\ptcrule{\def\ptc@rule{\kern-3\p@ \hrule width \columnwidth \kern2.6\p@}}
\def\mtcrule{\def\mtc@rule{\kern-3\p@ \hrule width \columnwidth \kern2.6\p@}}
\def\stcrule{\def\stc@rule{\kern-3\p@ \hrule width \columnwidth \kern2.6\p@}}
\def\ptc@rule{\rule[3\p@]{\columnwidth}{.4\p@}\vspace*{2.6\p@}}
\def\mtc@rule{\rule[3\p@]{\columnwidth}{.4\p@}\vspace*{2.6\p@}}
\def\stc@rule{\rule[3\p@]{\columnwidth}{.4\p@}\vspace*{2.6\p@}}

\def\noplfrule{\let\plf@rule\relax}
\def\nomlfrule{\let\mlf@rule\relax}
\def\noslfrule{\let\slf@rule\relax}
\def\plfrule{\def\plf@rule{\kern-3\p@ \hrule width \columnwidth \kern2.6\p@}}
\def\mlfrule{\def\mlf@rule{\kern-3\p@ \hrule width \columnwidth \kern2.6\p@}}
\def\slfrule{\def\slf@rule{\kern-3\p@ \hrule width \columnwidth \kern2.6\p@}}
\def\plf@rule{\rule[3\p@]{\columnwidth}{.4\p@}\vspace*{2.6\p@}}
\def\mlf@rule{\rule[3\p@]{\columnwidth}{.4\p@}\vspace*{2.6\p@}}
\def\slf@rule{\rule[3\p@]{\columnwidth}{.4\p@}\vspace*{2.6\p@}}

\def\nopltrule{\let\plt@rule\relax}
\def\nomltrule{\let\mlt@rule\relax}
\def\nosltrule{\let\slt@rule\relax}
\def\pltrule{\def\plt@rule{\kern-3\p@ \hrule width \columnwidth \kern2.6\p@}}
\def\mltrule{\def\mlt@rule{\kern-3\p@ \hrule width \columnwidth \kern2.6\p@}}
\def\sltrule{\def\slt@rule{\kern-3\p@ \hrule width \columnwidth \kern2.6\p@}}
\def\plt@rule{\rule[3\p@]{\columnwidth}{.4\p@}\vspace*{2.6\p@}}
\def\mlt@rule{\rule[3\p@]{\columnwidth}{.4\p@}\vspace*{2.6\p@}}
\def\slt@rule{\rule[3\p@]{\columnwidth}{.4\p@}\vspace*{2.6\p@}}
```

9.66 The \mtcset... commands

These commands [7] have been introduced to build a nicer user interface, and each of them replaces numerous user commands, offering a rather unified and logical syntax.

9.66.1 Keywords for the \mtcset... commands

\@namedef
\@nameuse

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:

```
\@namedef{mtc@typetable@parttoc}{ptc}\def\mtc@typetable@parttoc{ptc}
\@namedef{mtc@typetable@partlof}{plf}\def\mtc@typetable@partlof{plf}
\@namedef{mtc@typetable@partlot}{plt}\def\mtc@typetable@partlot{plt}
\@namedef{mtc@typetable@minitoc}{mtc}\def\mtc@typetable@minitoc{mtc}
\@namedef{mtc@typetable@minilof}{mlf}\def\mtc@typetable@minilof{mlf}
\@namedef{mtc@typetable@minilot}{mlt}\def\mtc@typetable@minilot{mlt}
\@namedef{mtc@typetable@secttoc}{stc}\def\mtc@typetable@secttoc{stc}
\@namedef{mtc@typetable@sectlof}{slf}\def\mtc@typetable@sectlof{slf}
\@namedef{mtc@typetable@sectlot}{slt}\def\mtc@typetable@sectlot{slt}
```

\@namedef

Then another family (typetitle) for the titles of the mini-tables:

```
\@namedef{mtc@typetitle@parttoc}{pti}\def\mtc@typetitle@parttoc{pti}
\@namedef{mtc@typetitle@partlof}{pti}\def\mtc@typetitle@partlof{pti}
\@namedef{mtc@typetitle@partlot}{pti}\def\mtc@typetitle@partlot{pti}
\@namedef{mtc@typetitle@minitoc}{mti}\def\mtc@typetitle@minitoc{mti}
\@namedef{mtc@typetitle@minilof}{mti}\def\mtc@typetitle@minilof{mti}
\@namedef{mtc@typetitle@minilot}{mti}\def\mtc@typetitle@minilot{mti}
\@namedef{mtc@typetitle@secttoc}{sti}\def\mtc@typetitle@secttoc{sti}
\@namedef{mtc@typetitle@sectlof}{sti}\def\mtc@typetitle@sectlof{sti}
\@namedef{mtc@typetitle@sectlot}{sti}\def\mtc@typetitle@sectlot{sti}
```

[7] The general concept of the \mtcset... commands was proposed by Benjamin Bayart.

\@namedef We define a family (YN) of keywords to recognize the keywords "off" and "on", with their many synonyms [8] and meaning false or true[9] :

```
\@namedef{mtc@YN@off}{no}\def\mtc@YN@off{no}
\@namedef{mtc@YN@OFF}{no}\def\mtc@YN@OFF{no}
\@namedef{mtc@YN@no}{no}\def\mtc@YN@no{no}
\@namedef{mtc@YN@NO}{no}\def\mtc@YN@NO{no}
\@namedef{mtc@YN@n}{no}\def\mtc@YN@n{no}
\@namedef{mtc@YN@N}{no}\def\mtc@YN@N{no}
\@namedef{mtc@YN@false}{no}\def\mtc@YN@false{no}
\@namedef{mtc@YN@FALSE}{no}\def\mtc@YN@FALSE{no}
\@namedef{mtc@YN@faux}{no}\def\mtc@YN@faux{no}
\@namedef{mtc@YN@FAUX}{no}\def\mtc@YN@FAUX{no}
\@namedef{mtc@YN@f}{no}\def\mtc@YN@f{no}
\@namedef{mtc@YN@F}{no}\def\mtc@YN@F{no}
\@namedef{mtc@YN@NON}{no}\def\mtc@YN@NON{no}
\@namedef{mtc@YN@non}{no}\def\mtc@YN@non{no}
\@namedef{mtc@YN@0}{no}\expandafter\def\csname mtc@YN@0\endcsname{no}
\@namedef{mtc@YN@-}{no}\expandafter\def\csname mtc@YN@-\endcsname{no}
%
\@namedef{mtc@YN@on}{}\def\mtc@YN@on{}
\@namedef{mtc@YN@ON}{}\def\mtc@YN@ON{}
\@namedef{mtc@YN@yes}{}\def\mtc@YN@yes{}
\@namedef{mtc@YN@YES}{}\def\mtc@YN@YES{}
\@namedef{mtc@YN@y}{}\def\mtc@YN@y{}
\@namedef{mtc@YN@Y}{}\def\mtc@YN@Y{}
\@namedef{mtc@YN@true}{}\def\mtc@YN@true{}
\@namedef{mtc@YN@TRUE}{}\def\mtc@YN@TRUE{}
\@namedef{mtc@YN@t}{}\def\mtc@YN@t{}
\@namedef{mtc@YN@T}{}\def\mtc@YN@T{}
\@namedef{mtc@YN@vrai}{}\def\mtc@YN@vrai{}
\@namedef{mtc@YN@VRAI}{}\def\mtc@YN@VRAI{}
\@namedef{mtc@YN@v}{}\def\mtc@YN@v{}
\@namedef{mtc@YN@V}{}\def\mtc@YN@V{}
\@namedef{mtc@YN@OUI}{}\def\mtc@YN@OUI{}
\@namedef{mtc@YN@oui}{}\def\mtc@YN@oui{}
\@namedef{mtc@YN@O}{}\def\mtc@YN@O{}
\@namedef{mtc@YN@o}{}\def\mtc@YN@o{}
\@namedef{mtc@YN@1}{}\expandafter\def\csname mtc@YN@1\endcsname{}
\@namedef{mtc@YN@+}{}\expandafter\def\csname mtc@YN@+\endcsname{}
```

9.66.2 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

[8] This (deliberately 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.

[9] O and o are the letter O, 0 is the zero digit.

highest sectionning level); "*" represents "any level", and is used to set the global default font for a given kind of mini-table.

```
\@namedef{mtc@sectlevel@chapter}{C}\def\mtc@sectlevel@chapter{C}
\@namedef{mtc@sectlevel@section}{S}\def\mtc@sectlevel@section{S}
\@namedef{mtc@sectlevel@subsection}{SS}\def\mtc@sectlevel@subsection{SS}
\@namedef{mtc@sectlevel@subsubsection}{SSS}\def\mtc@sectlevel@subsubsection{SSS}
\@namedef{mtc@sectlevel@paragraph}{P}\def\mtc@sectlevel@paragraph{P}
\@namedef{mtc@sectlevel@subparagraph}{SP}\def\mtc@sectlevel@subparagraph{SP}
\@namedef{mtc@sectlevel@*}{}\expandafter\def\csname mtc@sectlevel@*\endcsname{}
\@namedef{mtc@sectlevel@figure}{}\def\mtc@sectlevel@figure{}
\@namedef{mtc@sectlevel@table}{}\def\mtc@sectlevel@table{}
\@namedef{mtc@sectlevel@subfigure}{S}\def\mtc@sectlevel@subfigure{S}
\@namedef{mtc@sectlevel@subtable}{S}\def\mtc@sectlevel@subtable{S}
```

`\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 [165] basic commands usually.

`\if@mtc@setfont@` First, we declare a flag, set true:

```
\newif\if@mtc@setfont@\@mtc@setfont@true
```

`\mtcsetfont` Then, we begin the command, which has three arguments:

```
\newcommand{\mtcsetfont}[3]{%
```

`\mtc@mta@abbrev` `\if@mtc@setfont@` `\@nameuse` The two first arguments of this command are keywords. They must be translated into the effective strings. We process the first argument, a keyword from the `typetable` family. The result is stored in `\mtc@mta@abbrev`. Example: if `#1` is `minitoc`, we get `mtc`.

```
\def\mtc@mta@abbrev{X}
\@mtc@setfont@true
\expandafter\ifx\csname mtc@typetable@#1\endcsname\relax
   \@mtc@setfont@false
   \def\mtc@mta@abbrev{X}
   \mtcPackageError[<E0013>]{minitoc}%
      {\string\mtcsetfont \space has a wrong first argument
       \MessageBreak
       (#1).
       \MessageBreak
```

```
      It should be a mini-table type
      \MessageBreak
      (parttoc...sectlot)}%
     {Correct the source code.
      \MessageBreak
      Type <return> and rerun LaTeX}
\else
   \edef\mtc@mta@abbrev{\@nameuse{mtc@typetable@#1}}
\fi
```

\mtc@level@abbrev
\if@mtc@setfont@
\@nameuse

The second argument, a keyword from the family `sectlevel`, is processed the same way and the result is stored into a macro `\mtc@level@abbrev`. Example: if `#2` is the `subparagraph` keyword, we get SP.

```
\def\mtc@level@abbrev{X}
\expandafter\ifx\csname mtc@sectlevel@#2\endcsname\relax
   \@mtc@setfont@false
   \def\mtc@level@abbrev{X}
   \mtcPackageError[<E0014>]{minitoc}%
      {\string\mtcsetfont \space has a wrong second argument
       \MessageBreak
       (#2).
       \MessageBreak
       It should be a sectionning level
       \MessageBreak
       (part...subparagraph) or * }%
      {Correct the source code.
       \MessageBreak
       Type <return> and rerun LaTeX}
\else
   \edef\mtc@level@abbrev{\@nameuse{mtc@sectlevel@#2}}
\fi
```

\mtc@tmp@name
\mtc@mta@abbrev
\mtc@level@abbrev

Then, we construct the effective macro to be applied:

```
\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. Special care must be taken for testing via internal defined commands (*quarks*) with `@mtcqk` at the end of their names).

```
\def\parttoc@mtcqk{parttoc@mtcqk}
\def\minitoc@mtcqk{minitoc@mtcqk}
\def\secttoc@mtcqk{secttoc@mtcqk}
```

```
\def\partlof@mtcqk{partlof@mtcqk}
\def\minilof@mtcqk{minilof@mtcqk}
\def\sectlof@mtcqk{sectlof@mtcqk}
\def\partlot@mtcqk{partlot@mtcqk}
\def\minilot@mtcqk{minilot@mtcqk}
\def\sectlot@mtcqk{sectlot@mtcqk}
\def\part@mtcqk{part@mtcqk}
\def\chapter@mtcqk{chapter@mtcqk}
\def\section@mtcqk{section@mtcqk}
\def\subsection@mtcqk{subsection@mtcqk}
\def\subsubsection@mtcqk{subsubsection@mtcqk}
\def\paragraph@mtcqk{paragraph@mtcqk}
\def\subparagraph@mtcqk{subparagraph@mtcqk}
\def\figure@mtcqk{figure@mtcqk}
\def\table@mtcqk{table@mtcqk}
\def\subfigure@mtcqk{subfigure@mtcqk}
\def\subtable@mtcqk{subtable@mtcqk}
\@mtc@setfont@true
\expandafter\ifx\csname #1@mtcqk\endcsname\parttoc@mtcqk\relax
   \expandafter\ifx\csname #2@mtcqk\endcsname\figure@mtcqk\relax\@mtc@setfont@false\fi
   \expandafter\ifx\csname #2@mtcqk\endcsname\subfigure@mtcqk\relax\@mtc@setfont@false\fi
   \expandafter\ifx\csname #2@mtcqk\endcsname\table@mtcqk\relax\@mtc@setfont@false\fi
   \expandafter\ifx\csname #2@mtcqk\endcsname\subtable@mtcqk\relax\@mtc@setfont@false\fi
\fi
\expandafter\ifx\csname #1@mtcqk\endcsname\partlof@mtcqk\relax
   \expandafter\ifx\csname #2@mtcqk\endcsname\table@mtcqk\relax\@mtc@setfont@false\fi
   \expandafter\ifx\csname #2@mtcqk\endcsname\subtable@mtcqk\relax\@mtc@setfont@false\fi
\fi
\expandafter\ifx\csname #1@mtcqk\endcsname\partlot@mtcqk\relax
   \expandafter\ifx\csname #2@mtcqk\endcsname\figure@mtcqk\relax\@mtc@setfont@false\fi
   \expandafter\ifx\csname #2@mtcqk\endcsname\subfigure@mtcqk\relax\@mtc@setfont@false\fi
\fi
\expandafter\ifx\csname #1@mtcqk\endcsname\minitoc@mtcqk\relax
   \expandafter\ifx\csname #2@mtcqk\endcsname\part@mtcqk\relax\@mtc@setfont@false\fi
   \expandafter\ifx\csname #2@mtcqk\endcsname\chapter@mtcqk\relax\@mtc@setfont@false\fi
   \expandafter\ifx\csname #2@mtcqk\endcsname\figure@mtcqk\relax\@mtc@setfont@false\fi
   \expandafter\ifx\csname #2@mtcqk\endcsname\subfigure@mtcqk\relax\@mtc@setfont@false\fi
   \expandafter\ifx\csname #2@mtcqk\endcsname\table@mtcqk\relax\@mtc@setfont@false\fi
   \expandafter\ifx\csname #2@mtcqk\endcsname\subtable@mtcqk\relax\@mtc@setfont@false\fi
\fi
\expandafter\ifx\csname #1@mtcqk\endcsname\minilof@mtcqk\relax
   \expandafter\ifx\csname #2@mtcqk\endcsname\part@mtcqk\relax\@mtc@setfont@false\fi
   \expandafter\ifx\csname #2@mtcqk\endcsname\chapter@mtcqk\relax\@mtc@setfont@false\fi
   \expandafter\ifx\csname #2@mtcqk\endcsname\table@mtcqk\relax\@mtc@setfont@false\fi
   \expandafter\ifx\csname #2@mtcqk\endcsname\subtable@mtcqk\relax\@mtc@setfont@false\fi
\fi
\expandafter\ifx\csname #1@mtcqk\endcsname\minilot@mtcqk\relax
   \expandafter\ifx\csname #2@mtcqk\endcsname\part@mtcqk\relax\@mtc@setfont@false\fi
   \expandafter\ifx\csname #2@mtcqk\endcsname\chapter@mtcqk\relax\@mtc@setfont@false\fi
   \expandafter\ifx\csname #2@mtcqk\endcsname\figure@mtcqk\relax\@mtc@setfont@false\fi
   \expandafter\ifx\csname #2@mtcqk\endcsname\subfigure@mtcqk\relax\@mtc@setfont@false\fi
\fi
\expandafter\ifx\csname #1@mtcqk\endcsname\secttoc@mtcqk\relax
   \expandafter\ifx\csname #2@mtcqk\endcsname\part@mtcqk\relax\@mtc@setfont@false\fi
```

```
   \expandafter\ifx\csname #2@mtcqk\endcsname\chapter@mtcqk\relax\@mtc@setfont@false\fi
   \expandafter\ifx\csname #2@mtcqk\endcsname\section@mtcqk\relax\@mtc@setfont@false\fi
   \expandafter\ifx\csname #2@mtcqk\endcsname\figure@mtcqk\relax\@mtc@setfont@false\fi
   \expandafter\ifx\csname #2@mtcqk\endcsname\subfigure@mtcqk\relax\@mtc@setfont@false\fi
   \expandafter\ifx\csname #2@mtcqk\endcsname\table@mtcqk\relax\@mtc@setfont@false\fi
   \expandafter\ifx\csname #2@mtcqk\endcsname\subtable@mtcqk\relax\@mtc@setfont@false\fi
\fi
\expandafter\ifx\csname #1@mtcqk\endcsname\sectlof@mtcqk\relax
   \expandafter\ifx\csname #2@mtcqk\endcsname\part@mtcqk\relax\@mtc@setfont@false\fi
   \expandafter\ifx\csname #2@mtcqk\endcsname\chapter@mtcqk\relax\@mtc@setfont@false\fi
   \expandafter\ifx\csname #2@mtcqk\endcsname\section@mtcqk\relax\@mtc@setfont@false\fi
   \expandafter\ifx\csname #2@mtcqk\endcsname\table@mtcqk\relax\@mtc@setfont@false\fi
   \expandafter\ifx\csname #2@mtcqk\endcsname\subtable@mtcqk\relax\@mtc@setfont@false\fi
\fi
\expandafter\ifx\csname #1@mtcqk\endcsname\sectlot@mtcqk\relax
   \expandafter\ifx\csname #2@mtcqk\endcsname\part@mtcqk\relax\@mtc@setfont@false\fi
   \expandafter\ifx\csname #2@mtcqk\endcsname\chapter@mtcqk\relax\@mtc@setfont@false\fi
   \expandafter\ifx\csname #2@mtcqk\endcsname\section@mtcqk\relax\@mtc@setfont@false\fi
   \expandafter\ifx\csname #2@mtcqk\endcsname\figure@mtcqk\relax\@mtc@setfont@false\fi
   \expandafter\ifx\csname #2@mtcqk\endcsname\subfigure@mtcqk\relax\@mtc@setfont@false\fi
\fi
```

`\if@mtc@setfont@`
`\mtc@tmp@name`
`\mtc@mta@abbrev`
`\mtc@level@abbrev`
`\mtc@toks`

If the combinaison is legal, we apply it, i.e., we redefine the meaning of the constructed macro with the sequence of commands given as third argument of `\mtcsetfont` and we log that event (we store the third argument in a token register to can print it *verbatim*).

If the combinaison is not legal, an error message is displayed.

```
\if@mtc@setfont@
   \def\mtc@tmp@name{\mtc@mta@abbrev\mtc@level@abbrev font}
   \mtc@toks{#3}
   \mtcPackageInfo[<I0015>]{minitoc}%
      {\string\mtcsetfont\space redefines the macro
       \MessageBreak
       "\mtc@tmp@name" as "\the\mtc@toks"}
   \expandafter\def\csname\mtc@tmp@name\endcsname{#3}
\else
   \mtcPackageError[<E0024>]{minitoc}%
      {The macro \string\mtcsetfont\space has incompatible
       \MessageBreak
       first (#1) and second (#2) arguments}%
      {Correct the source code.
       \MessageBreak
       Type <return> and rerun LaTeX}
\fi
}
```

9.66.3 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 [165] 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:

```
\newif\if@mtc@settitlefont@\@mtc@settitlefont@true
```

\mtcsettitlefont And we begin the definition of the \mtcsettitlefont command, which has two arguments:

```
\newcommand{\mtcsettitlefont}[2]{%
```

\mtc@mtatf@abbrev \if@mtc@settitlefont@ \@nameuse We process the first argument, a keyword of the typetitle family, then the result is stored into \mtc@mtatf@abbrev:

```
\def\mtc@mtatf@abbrev{X}
\@mtc@settitlefont@true
\expandafter\ifx\csname mtc@typetitle@#1\endcsname\relax
   \@mtc@settitlefont@false
   \def\mtc@mtatf@abbrev{X}
   \mtcPackageError[<E0022>]{minitoc}%
      {\string\mtcsettitlefont \space has a wrong first argument
       \MessageBreak
       (#1).
       \MessageBreak
       It should be a mini-table type
       \MessageBreak
       (parttoc...sectlot)}%
      {Correct the source code.
       \MessageBreak
       Type <return> and rerun LaTeX}
\else
   \edef\mtc@mtatf@abbrev{\@nameuse{mtc@typetitle@#1}}
\fi
```

\if@mtc@settitlefont@
\mtc@tmptf@name
\mtc@mtatf@abbrev
\mtc@toks

Then we build the name of the effective command and apply this command:

```
\if@mtc@settitlefont@
   \def\mtc@tmptf@name{\mtc@mtatf@abbrev font}
   \mtc@toks{#2}
   \mtcPackageInfo[<I0018>]{minitoc}%
      {\string\mtcsettitlefont\space redefines the macro
       \MessageBreak
       "\mtc@tmptf@name" as
       \MessageBreak
       "\the\mtc@toks"}
   \expandafter\def\csname\mtc@tmptf@name\endcsname{#2}
\else
   \mtcPackageError[<E0034>]{minitoc}%
      {The macro \string\\mtcsettitlefont\space uses
       \MessageBreak
       an illegal type of table (#1)}%
      {Correct the source code.
       \MessageBreak
       Type <return> and rerun LaTeX}{\relax}
\fi
}
```

9.66.4 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:

```
\newif\if@mtc@settitle@\@mtc@settitle@true
```

\mtcsettitle

Then we define the \mtcsettitle command, which has two arguments:

```
\newcommand{\mtcsettitle}[2]{%
```

\mtc@mtati@abbrev
\if@mtc@settitle@
\@nameuse

We process the first argument, a keyword of the typetable family. The result is stored in \mtc@mtati@abbrev:

```
\def\mtc@mtati@abbrev{X}
```

```
\@mtc@settitle@true
\expandafter\ifx\csname mtc@typetable@#1\endcsname\relax
   \@mtc@settitle@false
   \def\mtc@mtati@abbrev{X}
   \mtcPackageError[<E0021>]{minitoc}%
      {\string\mtcsettitle \space has a wrong first argument
       \MessageBreak
       (#1).
       \MessageBreak
       It should be a mini-table type
       \MessageBreak
       (parttoc...sectlot)}%
      {Correct the source code.
       \MessageBreak
       Type <return> and rerun LaTeX}
\else
   \edef\mtc@mtati@abbrev{\@nameuse{mtc@typetable@#1}}
\fi
```

\if@mtc@settitle@
\mtc@tmpti@name
\mtc@mtati@abbrev
\mtc@toks

And we construct the name of the effective macro and apply it:

```
\if@mtc@settitle@
   \def\mtc@tmpti@name{\mtc@mtati@abbrev title}
   \mtc@toks{#2}
   \mtcPackageInfo[<I0017>]{minitoc}%
      {\string\mtcsettitle\space redefines the macro
       \MessageBreak
       "\mtc@tmpti@name" as
       \MessageBreak
       "\the\mtc@toks"}
   \expandafter\def\csname\mtc@tmpti@name\endcsname{#2}
\else
   \mtcPackageError[<E0033>]{minitoc}%
      {The macro \string\mtcsettitle\space uses
       \MessageBreak
       an illegal type of table (#1)}%
      {Correct the source code.
       \MessageBreak
       Type <return> and rerun LaTeX}{\relax}
\fi
}
```

9.66.5 The \mtcsetformat command

\@namedef

We define first the keywords (family formatparam) for the three formatting parameters that this command can alter:

```
\@namedef{mtc@formatparam@dotinterval}{dotsep}%
\def\mtc@arg@dotinterval{dotsep}
```

```
\@namedef{mtc@formatparam@tocrightmargin}{tocrmarg}%
\def\mtc@arg@tocrightmargin{tocrightmargin}
\@namedef{mtc@formatparam@pagenumwidth}{pnumwidth}%
\def\mtc@arg@pagenumwidth{\mtc@arg@pagenumwidth}
%% \@namedef{mtc@arg@numwidth}{numwidth} %not yet available
%% \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:

```
\AtBeginDocument{%
```

\@pnumwidth
\ptcpnumwidth
\mtcpnumwidth
\stcpnumwidth
\plfpnumwidth
\mlfpnumwidth
\slfpnumwidth
\pltpnumwidth
\mltpnumwidth
\sltpnumwidth

We take, if possible, the default value of \@pnumwith for each type of mini-tables:

```
\@ifundefined{ptcpnumwidth}{\let\ptcpnumwidth\@pnumwidth}{}%
\@ifundefined{stcpnumwidth}{\let\stcpnumwidth\@pnumwidth}{}%
\@ifundefined{mtcpnumwidth}{\let\mtcpnumwidth\@pnumwidth}{}%
\@ifundefined{plfpnumwidth}{\let\plfpnumwidth\@pnumwidth}{}%
\@ifundefined{mlfpnumwidth}{\let\mlfpnumwidth\@pnumwidth}{}%
\@ifundefined{slfpnumwidth}{\let\slfpnumwidth\@pnumwidth}{}%
\@ifundefined{pltpnumwidth}{\let\pltpnumwidth\@pnumwidth}{}%
\@ifundefined{mltpnumwidth}{\let\mltpnumwidth\@pnumwidth}{}%
\@ifundefined{sltpnumwidth}{\let\sltpnumwidth\@pnumwidth}{}%
```

\@tocrmarg
\ptctocrmarg
\mtclofrmarg
\stclotrmarg
\plftocrmarg
\mlflofrmarg
\slflotrmarg
\plttocrmarg
\mltlofrmarg
\sltlotrmarg

We take, if possible, the default value of \@tocrmarg for each type of mini-tables:

```
\@ifundefined{ptctocrmarg}{\let\ptctocrmarg\@tocrmarg}{}%
\@ifundefined{mtctocrmarg}{\let\mtctocrmarg\@tocrmarg}{}%
\@ifundefined{stctocrmarg}{\let\stctocrmarg\@tocrmarg}{}%
\@ifundefined{plftocrmarg}{\let\plftocrmarg\@tocrmarg}{}%
\@ifundefined{mlftocrmarg}{\let\mlftocrmarg\@tocrmarg}{}%
\@ifundefined{slftocrmarg}{\let\slftocrmarg\@tocrmarg}{}%
\@ifundefined{plttocrmarg}{\let\plttocrmarg\@tocrmarg}{}%
\@ifundefined{mlttocrmarg}{\let\mlttocrmarg\@tocrmarg}{}%
\@ifundefined{slttocrmarg}{\let\slttocrmarg\@tocrmarg}{}%
```

\@dotsep
\ptcdotsep
\mtcdotsep
\stcdotsep
\plfdotsep
\mlfdotsep
\slfdotsep
\pltdotsep
\mltdotsep
\sltdotsep

We take, if possible, the default value of \@dotsep for each type of mini-tables:

```
\@ifundefined{ptcdotsep}{\let\ptcdotsep\@dotsep}{}%
\@ifundefined{mtcdotsep}{\let\mtcdotsep\@dotsep}{}%
\@ifundefined{stcdotsep}{\let\stcdotsep\@dotsep}{}%
\@ifundefined{plfdotsep}{\let\plfdotsep\@dotsep}{}%
\@ifundefined{mlfdotsep}{\let\mlfdotsep\@dotsep}{}%
\@ifundefined{slfdotsep}{\let\slfdotsep\@dotsep}{}%
\@ifundefined{pltdotsep}{\let\pltdotsep\@dotsep}{}%
\@ifundefined{mltdotsep}{\let\mltdotsep\@dotsep}{}%
\@ifundefined{sltdotsep}{\let\sltdotsep\@dotsep}{}%
```

And we terminate the \AtBeginDocument block:

```
}
```

\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 ptc@verse The \ptc@setform macro is invoked in ptc@verse to set format parameters:

```
\def\ptc@setform{%
\let\@pnumwidth\ptcpnumwidth\relax
\let\@tocrmarg\ptctocrmarg\relax
\let\@dotsep\ptcdotsep\relax
}
```

\mtc@setform mtc@verse The \mtc@setform macro is invoked in mtc@verse to set format parameters:

```
\def\mtc@setform{%
\let\@pnumwidth\mtcpnumwidth\relax
\let\@tocrmarg\mtctocrmarg\relax
\let\@dotsep\mtcdotsep\relax
}
```

\stc@setform stc@verse The \stc@setform macro is invoked in stc@verse to set format parameters:

```
\def\stc@setform{%
\let\@pnumwidth\stcpnumwidth\relax
\let\@tocrmarg\stctocrmarg\relax
\let\@dotsep\stcdotsep\relax
}
```

\plf@setform ptc@verse The \plf@setform macro is invoked in ptc@verse to set format parameters:

```
\def\plf@setform{%
\let\@pnumwidth\plfpnumwidth\relax
\let\@tocrmarg\plftocrmarg\relax
\let\@dotsep\plfdotsep\relax
}
```

\mlf@setform
mtc@verse

The \mlf@setform macro is invoked in mtc@verse to set format parameters:

```
\def\mlf@setform{%
\let\@pnumwidth\mlfpnumwidth\relax
\let\@tocrmarg\mlftocrmarg\relax
\let\@dotsep\mlfdotsep\relax
}
```

\slf@setform
stc@verse

The \slf@setform macro is invoked in stc@verse to set format parameters:

```
\def\slf@setform{%
\let\@pnumwidth\slfpnumwidth\relax
\let\@tocrmarg\slftocrmarg\relax
\let\@dotsep\slfdotsep\relax
}
```

\plt@setform
ptc@verse

The \plt@setform macro is invoked in ptc@verse to set format parameters:

```
\def\plt@setform{%
\let\@pnumwidth\pltpnumwidth\relax
\let\@tocrmarg\plttocrmarg\relax
\let\@dotsep\pltdotsep\relax
}
```

\mlt@setform
mtc@verse

The \mlt@setform macro is invoked in mtc@verse to set format parameters:

```
\def\mlt@setform{%
\let\@pnumwidth\plfpnumwidth\relax
\let\@tocrmarg\plftocrmarg\relax
\let\@dotsep\plfdotsep\relax
}
```

\slt@setform
stc@verse

The \slt@setform macro is invoked in stc@verse to set format parameters:

```
\def\slt@setform{%
\let\@pnumwidth\plfpnumwidth\relax
\let\@tocrmarg\plftocrmarg\relax
\let\@dotsep\plfdotsep\relax
}
```

\if@mtc@setformat@
\mtcsetformat

We now define a flag and the \mtcsetformat command, which has the following syntax:

\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.

```
\newif\if@mtc@setformat@\@mtc@setformat@true
\newcommand{\mtcsetformat}[3]{%
```

`\mtc@mtf@abbrev` `\mtc@fparam@abbrev`

We now process the first argument and store the result in `\mtc@mtf@abbrev`:

```
\def\mtc@mtf@abbrev{X}
\def\mtc@fparam@abbrev{X}
\@mtc@setformat@true
\expandafter\ifx\csname mtc@typetable@#1\endcsname\relax
  \@mtc@setformat@false
  \def\mtc@mtf@abbrev{X}
  \mtcPackageError[<E0015>]{minitoc}%
     {\string\mtcsetformat \space has a wrong first argument
      \MessageBreak
      (#1).
      \MessageBreak
      It should be a mini-table type
      \MessageBreak
      (parttoc...sectlot)}%
     {Correct the source code.
      \MessageBreak
      Type <return> and rerun LaTeX}
\else
  \edef\mtc@mtf@abbrev{\@nameuse{mtc@typetable@#1}}
\fi
```

`\mtc@fparam@abbrev` `\@nameuse`

Then we process the second argument and store the result into a macro `\mtc@fparam@abbrev`:

```
\expandafter\ifx\csname mtc@formatparam@#2\endcsname\relax
 \@mtc@setformat@false
  \def\mtc@fparam@abbrev{X}
  \mtcPackageError[<E0016>]{minitoc}%
     {\string\mtcsetformat \space has a wrong second argument
      \MessageBreak
      (#2).
      \MessageBreak
      It should be a formatting param choosen from:
      \MessageBreak
      pagenumwidth, tocrightmargin, dotinterval}%
     {Correct the source code.
      \MessageBreak
      Type <return> and rerun LaTeX}
\else
  \edef\mtc@fparam@abbrev{\@nameuse{mtc@formatparam@#2}}
\fi
```

\if@mtc@setformat@
\mtc@tmpfm@name
\mtc@mtf@abbrev
\mtc@fparam@abbrev

The name of the storage macro is built and it receives the third parameter as value:

```
\if@mtc@setformat@
  \def\mtc@tmpfm@name{\mtc@mtf@abbrev\mtc@fparam@abbrev}
  \mtcPackageInfo[<I0016>]{minitoc}%
     {\string\mtcsetformat\space redefines the macro
      \MessageBreak
      "\mtc@tmpfm@name" as "\string#3"}
  \expandafter\def\csname\mtc@tmpfm@name\endcsname{#3}
\else
  \mtcPackageError[<E0025>]{minitoc}%
     {The macro \string\mtcsetformat\space has incompatible
      \MessageBreak
      first (#1) and second (#2) arguments}%
      {Correct the source code.
       \MessageBreak
       Type <return> and rerun LaTeX}
\fi
}
```

9.66.6 The \mtcsetpagenumbers command

This command activates of 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@
\if@mtc@spn@ok@

We define some flags:

```
\newif\if@mtc@setpagenumbers@ \@mtc@setpagenumbers@false
\newif\if@mtc@spn@ok@
```

\mtcsetpagenumbers
\@ifundefined
\mtcsetpagenumbers@

We define the user-level macro. If the first argument is a star, we call the internal macro \mtcsetpagenumbers@ for each type of mini-table available; else, we call this internal macro only once, for the specified type of mini-table.

```
\newcommand{\mtcsetpagenumbers}[2]{%
  \expandafter\ifx\csname #1\endcsname\*\relax
    \@ifundefined{part}{}%
    {\mtcsetpagenumbers@{parttoc}{#2}
     \mtcsetpagenumbers@{partlof}{#2}
```

```
      \mtcsetpagenumbers@{partlot}{#2}}
     \@ifundefined{chapter}{}%
     {\mtcsetpagenumbers@{minitoc}{#2}
      \mtcsetpagenumbers@{minilof}{#2}
      \mtcsetpagenumbers@{minilot}{#2}}
     \@ifundefined{section}{}%
     {\mtcsetpagenumbers@{secttoc}{#2}
      \mtcsetpagenumbers@{sectlof}{#2}
      \mtcsetpagenumbers@{sectlot}{#2}}
   \else
   \mtcsetpagenumbers@{#1}{#2}%
  \fi
}
```

\mtcsetpagenumber@

Then the \mtcsetpagenumbers@ internal macro, with two arguments:

```
\newcommand{\mtcsetpagenumbers@}[2]{%
```

\mtc@mttpn@abbrev
\mtc@pnsw@abbrev

We process the first argument, a keyword of the typetable family, and store the result in \mtc@mttpn@abbrev:

```
\def\mtc@mttpn@abbrev{X}
\@mtc@setpagenumbers@true
\def\mtc@pnsw@abbrev{}
\expandafter\ifx\csname mtc@typetable@#1\endcsname\relax
   \@mtc@setpagenumbers@false
   \def\mtc@pnsw@abbrev{X}
   \def\mtc@mttpn@abbrev{X}
   \mtcPackageError[<E0017>]{minitoc}%
      {\string\mtcsetpagenumbers \space has a wrong first
       \MessageBreak
       argument (#1)}%
      {It should be a mini-table type
       \MessageBreak
       (parttoc...sectlot)
       \MessageBreak
       Correct the source code.
       \MessageBreak
       Type <return> and rerun LaTeX}
\else
   \edef\mtc@mttpn@abbrev{\@nameuse{mtc@typetable@#1}}
\fi
```

\if@mtc@spn@ok@
\mtc@pnsw@abbrev
\mtc@mttpn@abbrev
\if@mtc@setpagenumbers@
\mtc@tmppn@name

Then the second argument, a keyword of the YN family, and store the result into a macro \mtc@pnsw@abbrev. The name of the effective macro is built and the macro executed.

```
\@mtc@spn@ok@true
\expandafter\ifx\csname mtc@YN@#2\endcsname\relax
   \@mtc@spn@ok@false
```

```
   \def\mtc@pnsw@abbrev{X}
   \def\mtc@mttpn@abbrev{X}
   \@mtc@setpagenumbers@false
   \def\mtc@mttpn@abbrev{X}
   \mtcPackageError[<E0018>]{minitoc}%
      {\string\mtcsetpagenumbers \space has a wrong second
       \MessageBreak
       argument (#2)}%
      {It should be a boolean value (0/1, yes/no, on/off, ...)
       \MessageBreak
       Correct the source code.
       \MessageBreak
       Type <return> and rerun LaTeX}
\else
   \edef\mtc@pnsw@abbrev{\@nameuse{mtc@YN@#2}}
   \def\mtc@pnsw@abbrevX{X}
   \def\mtc@noX{mtc@noX}
   \def\mtc@tmppn@name{\mtc@pnsw@abbrev\mtc@mttpn@abbrev pagenumbers}
   \expandafter\ifx\csname mtc@\mtc@pnsw@abbrev X\endcsname\mtc@noX
      \mtcPackageInfo[<I0022>]{minitoc}%
         {Page numbers are inhibited
          \MessageBreak
          for the #1s}
   \else
      \mtcPackageInfo[<I0021>]{minitoc}%
         {Page numbers are activated
          \MessageBreak for the #1s}
   \fi
   \csname\mtc@tmppn@name\endcsname{}
\fi
}
```

9.66.7 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@
\if@mtc@sru@ok@

We define some flags:

```
\newif\if@mtc@setrules@ \@mtc@setrules@false
\newif\if@mtc@sru@ok@
```

\mtcsetrules
\@ifundefined
\mtcsetrules@

We define the user-level macro. If the first argument is a star, we call the internal macro \mtcsetrules@ for each type of mini-table available; else, we call this internal macro only once, for the specified type of mini-table.

```
\newcommand{\mtcsetrules}[2]{%
  \expandafter\ifx\csname #1\endcsname\*\relax
    \@ifundefined{part}{}%
    {\mtcsetrules@{parttoc}{#2}
     \mtcsetrules@{partlof}{#2}
     \mtcsetrules@{partlot}{#2}}
    \@ifundefined{chapter}{}%
    {\mtcsetrules@{minitoc}{#2}
     \mtcsetrules@{minilof}{#2}
     \mtcsetrules@{minilot}{#2}}
    \@ifundefined{section}{}%
    {\mtcsetrules@{secttoc}{#2}
     \mtcsetrules@{sectlof}{#2}
     \mtcsetrules@{sectlot}{#2}}
  \else
  \mtcsetrules@{#1}{#2}%
 \fi
}
```

\mtcsetrules@

Then the \mtcsetrules@ internal macro, which has two arguments:

```
\newcommand{\mtcsetrules@}[2]{%
```

\mtc@mttru@abbrev
\if@mtc@setrules@
\mtc@rusw@abbrev
\@nameuse

We process the first argument, a keyword of the typetable family and store the result in a macro \mtc@mttru@abbrev:

```
\def\mtc@mttru@abbrev{X}
\@mtc@setrules@true
\def\mtc@rusw@abbrev{}
\expandafter\ifx\csname mtc@typetable@#1\endcsname\relax
  \@mtc@setrules@false
  \def\mtc@rusw@abbrev{X}
  \def\mtc@mttru@abbrev{X}
  \mtcPackageError[<E0019>]{minitoc}%
     {\string\mtcsetrules \space has a wrong first argument
      \MessageBreak
      (#1)}%
     {It should be a mini-table type
      \MessageBreak
      (parttoc...sectlot)
      \MessageBreak
      Correct the source code.
      \MessageBreak
      Type <return> and rerun LaTeX}
\else
  \edef\mtc@mttru@abbrev{\@nameuse{mtc@typetable@#1}}
```

```
\fi
```

\if@mtc@sru@ok@
\mtc@rusw@abbrev
\mtc@mttru@abbrev
if@setrules@false
\mtc@noX
\mtc@tmppn@name

Then the second argument, a keyword of the YN family, and store the result in a macro \mtc@rusw@abbrev. The name of the effective macro is built and the macro executed.

```
\@mtc@sru@ok@true
\expandafter\ifx\csname mtc@YN@#2\endcsname\relax
   \@mtc@sru@ok@false
   \def\mtc@rusw@abbrev{X}
   \def\mtc@mttru@abbrev{X}
   \@mtc@setrules@false
   \mtcPackageError[<E0020>]{minitoc}%
      {\string\mtcsetrules \space has a wrong second argument
       \MessageBreak
       (#2)}%
      {It should be a boolean value (0/1, yes/no, on/off, ...)
       \MessageBreak
       Correct the source code.
       \MessageBreak
       Type <return> and rerun LaTeX}
\else
   \edef\mtc@rusw@abbrev{\@nameuse{mtc@YN@#2}}
   \def\mtc@rusw@abbrevX{X}
   \def\mtc@noX{mtc@noX}
   \def\mtc@tmppn@name{\mtc@rusw@abbrev\mtc@mttru@abbrev rule}
   \expandafter\ifx\csname mtc@\mtc@rusw@abbrev X\endcsname\mtc@noX
      \mtcPackageInfo[<I0008>]{minitoc}%
         {Horizontal rules are inhibited
          \MessageBreak
          for the #1s}
   \else
      \mtcPackageInfo[<I0007>]{minitoc}%
         {Horizontal rules are activated
          \MessageBreak
          for the #1s}
   \fi
   \csname\mtc@tmppn@name\endcsname{}
\fi
}
```

9.66.8 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 (ltypetable) for the long names of the types of mini-tables:

```
\@namedef{mtc@ltypetable@parttoc}{parttoc}\def\mtc@ltypetable@parttoc{parttoc}
```

```
\@namedef{mtc@ltypetable@partlof}{partlof}\def\mtc@ltypetable@partlof{partlof}
\@namedef{mtc@ltypetable@partlot}{partlot}\def\mtc@ltypetable@partlot{partlot}
\@namedef{mtc@ltypetable@minitoc}{minitoc}\def\mtc@ltypetable@minitoc{minitoc}
\@namedef{mtc@ltypetable@minilof}{minilof}\def\mtc@ltypetable@minilof{minilof}
\@namedef{mtc@ltypetable@minilot}{minilot}\def\mtc@ltypetable@minilot{minilot}
\@namedef{mtc@ltypetable@secttoc}{secttoc}\def\mtc@ltypetable@secttoc{secttoc}
\@namedef{mtc@ltypetable@sectlof}{sectlof}\def\mtc@ltypetable@sectlof{sectlof}
\@namedef{mtc@ltypetable@sectlot}{sectlot}\def\mtc@ltypetable@sectlot{sectlot}
```

A family (`featureparam`) for the type of feature:

```
\@namedef{mtc@featureparam@before}{before}%
   \def\mtc@featureparam@before{before}
\@namedef{mtc@featureparam@after}{after}%
   \def\mtc@featureparam@after{after}
\@namedef{mtc@featureparam@pagestyle}{thispage}%
   \def\mtc@featureparam@pagestyle{thispage}
```

And a family (`ft3`) to add "style" if it is a "pagestyle" feature:

```
\@namedef{mtc@ft3@before}{}\expandafter\def\csname mtc@ft3@before\endcsname{}
\@namedef{mtc@ft3@after}{}\expandafter\def\csname mtc@ft3@after\endcsname{}
\@namedef{mtc@ft3@pagestyle}{style}%
    \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@`
`\mtcsetfeature`

We define a flag and the `\mtcsetfeature` command, with three arguments:

```
\newif\if@mtc@setfeature@\@mtc@setfeature@true
\newcommand{\mtcsetfeature}[3]{%
```

`\mtc@mtfeat@abbrev`
`\mtc@featparam@abbrev`

We process the first argument, a keyword of the `ltypetable` family, and store the result in `\mtc@mtfeat@abbrev`:

```
\def\mtc@mtfeat@abbrev{X}
\def\mtc@featparam@abbrev{X}
\@mtc@setfeature@true
\expandafter\ifx\csname mtc@ltypetable@#1\endcsname\relax
  \@mtc@setfeature@false
  \def\mtc@mtfeat@abbrev{X}
```

```
  \mtcPackageError[<E0011>]{minitoc}%
     {\string\mtcsetfeature \space has a wrong first argument
      \MessageBreak
      (#1).
      \MessageBreak
      It should be a mini-table type
      \MessageBreak
      (parttoc...sectlot)}%
     {Correct the source code.
      \MessageBreak
      Type <return> and rerun LaTeX}
\else
  \edef\mtc@mtfeat@abbrev{\@nameuse{mtc@ltypetable@#1}}
\fi
```

\if@mtc@setfeature@
\mtc@featparam@abbrev
\mtc@featparam@third
\@nameuse

The second argument is a keyword of the `featureparam` family, the result is stored in `\mtc@featparam@`; and the complement is computed from the first argument, interpreted as a keyword of the `ft3` family and whose result is stored in `\mtc@featparam@third`.

```
\expandafter\ifx\csname mtc@featureparam@#2\endcsname\relax
 \@mtc@setfeature@false
  \def\mtc@featparam@abbrev{X}
  \def\mtc@featparam@third{X}
  \mtcPackageError[<E0012>]{minitoc}%
     {\string\mtcsetfeature \space has a wrong second argument
      \MessageBreak
      (#2).
      \MessageBreak
      It should be a feature param
      \MessageBreak
      (before, after, pagestyle)}%
     {Correct the source code.
      \MessageBreak
      Type <return> and rerun LaTeX}
\else
  \edef\mtc@featparam@abbrev{\@nameuse{mtc@featureparam@#2}}
  \edef\mtc@featparam@third{\@nameuse{mtc@ft3@#2}}
\fi
```

\if@mtc@setfeature@
\mtc@tmpfeat@name
\mtc@featparam@abbrev
\mtc@mtfeat@abbrev
\mtc@featparam@third

The name of the effective macro is built by concatenating these three pieces (named `\mtc@featparam@abbrev`, `\mtc@mtfeat@abbrev`, and `\mtc@featparam@third` respectively), then this macro is executed:

```
\if@mtc@setfeature@
  \def\mtc@tmpfeat@name%
     {\mtc@featparam@abbrev\mtc@mtfeat@abbrev\mtc@featparam@third}
  \mtcPackageInfo[<I0014>]{minitoc}%
     {\string\mtcsetfeature\space redefines the macro
      \MessageBreak
      "\csname mtc@tmpfeat@name\endcsname" as
```

```
      \MessageBreak
      "\string#3"}
  \expandafter\def\csname\mtc@tmpfeat@name\endcsname{#3}
\else
  \mtcPackageError[<E0023>]{minitoc}%
     {The macro \string\mtcsetfeature\space has incompatible
      \MessageBreak
      first (#1) and second (#2) arguments}%
     {Correct the source code.
      \MessageBreak
      Type <return> and rerun LaTeX}
\fi}
```

9.66.9 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 [96]).

\if@mtc@setdepth@ First, we declare a flag, set true:

```
\newif\if@mtc@setdepth@\@mtc@setdepth@true
```

\mtcsetdepth Then we define the \mtcsetdepth command, with two arguments:

```
\newcommand{\mtcsetdepth}[2]{%
```

\mtc@mtade@abbrev \if@mtc@setdepth@ \@nameuse We process the first argument, a keyword of the ltypetable family. The result is stored in \mtc@mtade@abbrev:

```
\def\mtc@mtade@abbrev{X}
\@mtc@setdepth@true
\expandafter\ifx\csname mtc@ltypetable@#1\endcsname\relax
   \@mtc@setdepth@false
   \def\mtc@mtade@abbrev{X}
   \mtcPackageError[<E0009>]{minitoc}%
      {\string\mtcsetdepth \space has a wrong first argument
       \MessageBreak
       (#1).
       \MessageBreak
       It should be a mini-table type
       \MessageBreak
```

```
       (parttoc...sectlot)}%
      {Correct the source code.
       \MessageBreak
       Type <return> and rerun LaTeX}
\else
   \edef\mtc@mtade@abbrev{\@nameuse{mtc@ltypetable@#1}}
\fi
```

\if@mtc@setdepth@
\mtc@tmpde@name
\mtc@mtade@abbrev
\mtc@toks
\setcounter

And we construct the name of the effective counter and gave it the value:

```
\if@mtc@setdepth@
   \def\mtc@tmpde@name{\mtc@mtade@abbrev depth}
   \expandafter\noexpand\@ifundefined{c@\mtc@mtade@abbrev depth}%
      {\mtcPackageError[<E0008>]{minitoc}%
       {\string\mtcsetdepth \space attempts to use
        \MessageBreak
             an undefined counter (#1depth).}%
      {Correct the source code.
       \MessageBreak
       Type <return> and rerun LaTeX}}{%
   \mtc@toks{#2} % trick for explicit message using \the.
   \mtcPackageInfo[<I0013>]{minitoc}%
      {\string\mtcsetdepth\space redefines the counter
       \MessageBreak
       "\mtc@tmpde@name" as "\the\mtc@toks"}%
    \expandafter\csname c@\mtc@tmpde@name\endcsname=#2}%
\else
   \mtcPackageError[<E0010>]{minitoc}%
      {\string\mtcsetdepth:\space Illegal type of table (#1)}%
      {Correct the source code.
       \MessageBreak
       Type <return> and rerun LaTeX}{\relax}
\fi}% end of \mtcsetdepth
```

9.67 The mtchideinmaintoc environment and siblings

\if@mtc@Himtoc@

The flag \if@mtc@Himtoc@ is used to detect an incorrect imbrication of this environment:

```
\newif\if@mtc@Himtoc@ \@mtc@Himtoc@false
```

\mtc@savetocdepth
\mtc@restoretocdepth
mtchideinmaintoc
\if@mtc@Himtoc@
\mtc@sv@tocdepth
\arabic
\addtocontents
\setcounter

We define a macro \mtc@savetocdepth to save the current value of the counter tocdepth. Then we define this environment, which inserts into the TOC file this command and commands of the form \setcounter{tocdepth}{...}. Note that \xdef is necessary! It also save and restore the value of the counter tocdepth, as the optional argument is the hiding depth of the entries in the main TOC.

```
\newcommand{\mtc@savetocdepth}{\xdef\mtc@sv@tocdepth{\arabic{tocdepth}}}%
\newcommand{\mtc@restoretocdepth}{\setcounter{tocdepth}{\mtc@sv@tocdepth}}%
\newenvironment{mtchideinmaintoc}[1][-1]%
{\if@mtc@Himtoc@\mtcPackageError[<E0005>]{minitoc}%
    {Imbrication of mtchideinmaintoc environments}%
    {The hiding in main ToC could be incorrect}\fi
 \global\@mtc@Himtoc@true
 \addtocontents{toc}{\protect\mtc@savetocdepth}%
 \addtocontents{toc}{\protect\setcounter{tocdepth}{#1}}}%
{\if@mtc@Himtoc@\else\mtcPackageError[<E0031>]{minitoc}%
    {Unbalanced mtchideinmaintoc environment}%
    {The hiding in main ToC could be incorrect}\fi
 \global\@mtc@Himtoc@false
 \addtocontents{toc}{\protect\mtc@restoretocdepth}}%
```

\AtBeginDocument
\if@mtc@Himlof@
mtchideinmainlof
\mtc@sv@lofdepth
\mtc@sv@tocdepth
\mtc@svf@tocdepth
\arabic
\addtocontents
\setcounter

The mtchideinmainlof and mtchideinmainlot environments are similar, but we must verify the presence of the associated depth counter, so we have two versions of each of these environments. This must be done *after* the loading of the packages.

First, for the list of figures:

```
\newif\if@mtc@Himlof@ \@mtc@Himlof@false
\AtBeginDocument{%
\@ifundefined{c@lofdepth}{%
   \newenvironment{mtchideinmainlof}[1][-1]%
   {\if@mtc@Himlof@\mtcPackageError[<E0003>]{minitoc}%
       {Imbrication of mtchideinmainlof environments}%
       {The hiding in main LoF could be incorrect}\fi
    \global\@mtc@Himlof@true
    \def\mtc@sv@tocdepth{\arabic{tocdepth}}%
    \def\mtc@sv@lofdepth{\arabic{tocdepth}}%
    \addtocontents{lof}{\protect\mtc@savetocdepth}%
    \addtocontents{lof}{\protect\setcounter{tocdepth}{#1}}}%
   {\if@mtc@Himlof@\else\mtcPackageError[<E0029>]{minitoc}%
       {Unbalanced mtchideinmainlof environment}%
       {The hiding in main LoF could be incorrect}\fi
    \global\@mtc@Himtoc@false
    \addtocontents{lof}{\protect\mtc@restoretocdepth}%
}}%
{%
   \newcommand{\mtc@savelofdepth}{\xdef{\mtc@sv@lofdepth{\arabic{lofdepth}}}}%
   \newcommand{\mtc@restorelofdepth}{\setcounter{lofdepth}{\mtc@sv@lofdepth}}%
   \newenvironment{mtchideinmainlof}[1][-1]%
   {\if@mtc@Himlof@\mtcPackageError[<E0003>]{minitoc}%
       {Imbrication of mtchideinmainlof environments}%
       {The hiding in main LoF could be incorrect}\fi
    \global\@mtc@Himlof@true
    \addtocontents{lof}{\protect\mtc@savelofdepth}%
    \addtocontents{lof}{\protect\setcounter{tocdepth}{#1}}}%
   {\if@mtc@Himlof@\else\mtcPackageError[<E0029>]{minitoc}%
       {Unbalanced mtchideinmainlof environment}%
       {The hiding in main LoF could be incorrect}\fi
```

```
    \global\@mtc@Himlof@false
    \addtocontents{lof}{\protect\mtc@restoretocdepth}}}}
```

\AtBeginDocument
\if@mtc@Himlot@
mtchideinmainlot
\mtc@sv@lotdepth
\mtc@sv@tocdepth
\mtc@svt@tocdepth
\arabic
\addtocontents
\setcounter

Then for the list of tables:

```
\newif\if@mtc@Himlot@ \@mtc@Himlot@false
\AtBeginDocument{%
\@ifundefined{c@lotdepth}{%
   \newenvironment{mtchideinmainlot}[1][-1]%
   {\if@mtc@Himlot@\mtcPackageError[<E0004>]{minitoc}%
       {Imbrication of mtchideinmainlot environments}%
       {The hiding in main LoT could be incorrect}\fi
    \global\@mtc@Himlot@true
    \def\mtc@sv@tocdepth{\arabic{tocdepth}}%
    \def\mtc@sv@lotdepth{\arabic{tocdepth}}%
    \addtocontents{lot}{\protect\mtc@savetocdepth}%
    \addtocontents{lot}{\protect\setcounter{tocdepth}{#1}}}%
   {\if@mtc@Himlot@\else\mtcPackageError[<E0030>]{minitoc}%
       {Unbalanced mtchideinmainlot environment}%
       {The hiding in main LoT could be incorrect}\fi
    \global\@mtc@Himlot@false
    \addtocontents{lot}{\protect\mtc@restoretocdepth}%
}}%
{%
   \newcommand{\mtc@savelotdepth}{\xdef{\mtc@sv@lotdepth{\arabic{lotdepth}}}}%
   \newcommand{\mtc@restorelotdepth}{\setcounter{lotdepth}{\mtc@sv@lotdepth}}%
   \newenvironment{mtchideinmainlot}[1][-1]%
   {\if@mtc@Himlot@\mtcPackageError[<E0004>]{minitoc}%
       {Imbrication of mtchideinmainlot environments}%
       {The hiding in main LoT could be incorrect}\fi
    \global\@mtc@Himlot@true
    \addtocontents{lot}{\protect\mtc@savelotdepth}%
    \addtocontents{lot}{\protect\setcounter{lotdepth}{#1}}}%
   {\if@mtc@Himlot@\else\mtcPackageError[<E0030>]{minitoc}%
       {Unbalanced mtchideinmainlot environment}%
       {The hiding in main LoT could be incorrect}\fi
    \global\@mtc@Himlot@false
    \addtocontents{lot}{\protect\mtc@restorelotdepth}}}}
```

9.68 Fixing the “Glossary” entry in the TOC

This macro is complex. Its syntax is:

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

\@ifundefined
\mtc@glofix@level

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:

```
\@ifundefined{chapter}{%
   \@ifundefined{section}%
      {\mtcPackageWarningNoLine[<W0001>]{minitoc}%
       {\string\chapter\space and \string\section\space are undefined.%
        \MessageBreak
        Cannot use \string\mtcfixglossary \space without
        \MessageBreak
        optional argument [part]}%
       \@ifundefined{part}%
         {\mtcPackageError[<E0001>]{minitoc}%
            {But \string\part\space is undefined}%
            {\string\mtcfixglossary\space not usable}}%
       {\mtcPackageWarningNoLine[<W0006>]{minitoc}%
          {\string\mtcfixglossary\space can only be used
           \MessageBreak
           with the [part] optional argument,
           \MessageBreak
           which becomes the default}%
        \def\mtc@glofix@level{part}%
       }}%
       {\def\mtc@glofix@level{section}}}%
{\def\mtc@glofix@level{chapter}}
```

\if@mtcfixglossary@
\mtcfixglossary
\addcontentsline

Then we define a flag (\if@mtcfixglossary@) and the command \mtcfixglossary, which adds the necessary lines in the TOC, the LOF and the LOT.

```
\newif\if@mtcfixglossary@ \@mtcfixglossary@false
\newcommand{\mtcfixglossary}[1][\mtc@glofix@level]{%
   \@mtcfixglossary@false
   \expandafter%
    \ifx\csname #1\endcsname\part\relax\@mtcfixglossary@true\fi
   \expandafter%
    \ifx\csname #1\endcsname\chapter\relax\@mtcfixglossary@true\fi
   \expandafter%
    \ifx\csname #1\endcsname\section\relax\@mtcfixglossary@true\fi
   \if@mtcfixglossary@
   \addcontentsline{lof}{x\mtc@glofix@level}{}%
   \addcontentsline{lot}{x\mtc@glofix@level}{}%
   \csname mtcadd\mtc@glofix@level\endcsname\relax
   \else
   \mtcPackageError[<E0026>]{minitoc}%
      {The optional argument of \string\mtcfixglossary
       \MessageBreak
       is wrong}%
      {It must be omitted (\mtc@glofix@level), or be part, chapter or section}%
   \fi
}%
```

9.69 Fixing the "Index" entry in the TOC

This macro is complex. Its syntax is:

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

\@ifundefined
\mtc@ixfix@level

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:

```
\@ifundefined{chapter}{%
   \@ifundefined{section}%
      {\mtcPackageWarningNoLine[<W0002>]{minitoc}%
       {\string\chapter\space and \string\section\space are undefined.%
        \MessageBreak
        Cannot use \string\mtcfixindex \space without
        \MessageBreak
        optional argument [part]}%
       \@ifundefined{part}%
         {\mtcPackageError[<E0002>]{minitoc}%
            {But \string\part\space is undefined}%
            {\string\mtcfixindex\space not usable}}%
       {\mtcPackageWarningNoLine[<W0007>]{minitoc}%
          {\string\mtcfixindex\space can only be used with
           \MessageBreak
           the [part] optional argument,
           \MessageBreak
           which becomes the default}%
        \def\mtc@ixfix@level{part}%
       }}%
       {\def\mtc@ixfix@level{section}}}%
{\def\mtc@ixfix@level{chapter}}
```

\if@mtcfixindex@
\mtcfixindex
\addcontentsline

Then we define a flag and the command \mtcfixindex, which adds the necessary lines in the TOC, the LOF and the LOT.

```
\newif\if@mtcfixindex@ \@mtcfixindex@false
\newcommand{\mtcfixindex}[1][\mtc@ixfix@level]{%
   \@mtcfixindex@false
   \expandafter%
    \ifx\csname #1\endcsname\part\relax\@mtcfixindex@true\fi
   \expandafter%
    \ifx\csname #1\endcsname\chapter\relax\@mtcfixindex@true\fi
   \expandafter%
    \ifx\csname #1\endcsname\section\relax\@mtcfixindex@true\fi
   \if@mtcfixindex@
   \addcontentsline{lof}{x\mtc@ixfix@level}{}%
```

```
   \addcontentsline{lot}{x\mtc@ixfix@level}{}%
   \csname mtcadd\mtc@ixfix@level\endcsname\relax
   \else
   \mtcPackageError[<E0027>]{minitoc}%
      {The optional argument of \string\mtcfixindex
       \MessageBreak
       is wrong}%
      {It must be omitted (\mtc@ixfix@level), or be part, chapter or section}%
   \fi
}%
```

9.70 The \mtcselectlanguage command

\mtcselectlanguage
\if@mtc@insellang@
\IfFileExists
\@input

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@ is true while we are in this macro.

```
\newif\if@mtc@insellang@ \@mtc@insellang@false
\def\mtcselectlanguage#1{%
     \@mtc@insellang@true
     \InputIfFileExists{#1.mld}%
       {\mtcPackageInfo[<I0010>]{minitoc}{The #1 language is selected.%
         \MessageBreak
         }}%
       {\mtcPackageError[<E0006>]{minitoc}%
          {#1 is not a known language,
           \MessageBreak
           #1.mld not found.
           \MessageBreak
           Command ignored}%
          {See the minitoc documentation.
           \MessageBreak
           Correct the source using a valid language name.
           \MessageBreak
           Press RETURN}}%
     \@mtc@insellang@false
}
```

9.71 The \mtcloadmlo internal command

\mtcloadmlo
\if@mtc@insellang@
\IfFileExists
\@input

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 invoked directly by the user. This is verified via the flag `\if@mtc@insellang@`.

```
\def\mtcloadmlo#1{%
    \if@mtc@insellang@
    \InputIfFileExists{#1.mlo}%
      {\mtcPackageInfo[<I0011>]{minitoc}%
         {#1 minitoc language object selected.
          \MessageBreak}}%
      {\mtcPackageError[<E0007>]{minitoc}%
         {#1 is not a known minitoc
          \MessageBreak
          language object file (.mlo),
          \MessageBreak
          #1.mlo not found.
          \MessageBreak
          Command ignored}%
         {See the minitoc documentation.
          \MessageBreak
          Correct the source using a valid language name.
          \MessageBreak
          Press RETURN}}%
    \else
      \mtcPackageError[<E0032>]{minitoc}%
          {You are using the \string\mtcloadmlo\space command
           \MessageBreak
           outside of a .mld file}%
          {It will be ignored}
      \@mtc@insellang@false
    \fi
}
```

9.72 The "coffee breaks"

`\addcoffeeline` `\addtocontents` `\coffeeline` `\l@coffee` `\@Undottedtocline`

For the minutes package [169] (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.

```
\def\addcoffeeline#1#2#3{%
  \addtocontents{#1}{\protect\coffeeline{#2}{#3}{\null}}}
\def\coffeeline#1{\csname l@#1\endcsname}
\newcommand*\l@coffee{\@Undottedtocline{1}{1.5em}{2.3em}}
```

9.73 Initialization of counters

\AtBeginDocument
\@ifundefined
\setcounter

At the beginning of the document, we initialize the absolute counters for parts, chapters and sections, if they are defined.

```
\AtBeginDocument{%
\@ifundefined{c@ptc}{}{\setcounter{ptc}{0}}
\@ifundefined{c@mtc}{}{\setcounter{mtc}{0}}
\@ifundefined{c@stc}{}{\setcounter{stc}{0}}}
```

9.74 Declarations for simple options

These options are just setting a flag.

9.74.1 Options tight and loose, k-tight and k-loose

\DeclareOption
\iftightmtc
\ifktightmtc

These options influence the interline separation in the mini-tables.

```
\DeclareOption{tight}{\tightmtctrue}
\DeclareOption{loose}{\tightmtcfalse} % default
\DeclareOption{k-tight}{\ktightmtctrue}
\DeclareOption{k-loose}{\ktightmtcfalse} % default
```

9.74.2 Options checkfiles and nocheckfiles

\DeclareOption
\if@mtc@checkfiles

These options activate or inhibit the checking for empty mini-table files.

```
\DeclareOption{checkfiles}{\@mtc@checkfilestrue} % default
\DeclareOption{nocheckfiles}{\@mtc@checkfilesfalse}
```

9.74.3 Options dotted and undotted

\DeclareOption
\ifundottedmtc

These options activate or inhibit the leaders (lines of dots) in the mini-tables.

```
\DeclareOption{undotted}{\undottedmtctrue}
\DeclareOption{dotted}{\undottedmtcfalse} % default
```

9.74.4 Option notoccite

\DeclareOption
\if@mtc@notoccite@

This option will later load the notoccite package [9].

```
\DeclareOption{notoccite}{\@mtc@notoccite@true}
```

9.74.5 Option shortext

\DeclareOption
\if@longextensions@

This option forces the use of short extensions.

```
\DeclareOption{shortext}{\@longextensions@false
\mtcPackageWarningNoLine[<W0020>]{minitoc}%
   {You have forced the use of short extensions}}
```

9.75 The insection option

\if@mtc@ss@insection@
\@ifundefined
\DeclareOption

This option is available only if \chapter is not defined and \section defined. It is to be revised when chapter/section level commands will ever be allowed together, sometime in the far away future, with a lot of luck (and work)[10].

```
\newif\if@mtc@ss@insection@ \@mtc@ss@insection@false
\@ifundefined{chapter}{%
   \@ifundefined{section}{\DeclareOption{insection}{%
   \mtcPackageError[<E0035>]{minitoc}%
      {You have used the ‘insection’ option in
       \MessageBreak
       a document where chapters are defined.
       \MessageBreak
       This is not compatible: option ignored.}%
      {Remove this option.
       \MessageBreak
       Type <return> and rerun LaTeX}
                          }}%
                         {%
                          \DeclareOption{insection}%
                           {\@mtc@ss@insection@true}%
                         }%
}{}
```

[10] Please, do not dream too much!

9.76 The listfiles and nolistfiles options

\if@mtc@listfiles@
\DeclareOption

The listfiles option creates a file containing a list of the auxiliary files created by the minitoc package. This is the default. This file is named *document*.maf. The nolistfiles option inhibits this listing.

```
\newif\if@mtc@listfiles@ \@mtc@listfiles@true
\DeclareOption{listfiles}{\@mtc@listfiles@true}
\DeclareOption{nolistfiles}{\@mtc@listfiles@false}
```

9.77 Language options

\@gobblethree
\mtc@listmisslanguages
\mtc@addmisslanguage
\mtc@LML
\MessageBreak

First, we define an utility macro (\@gobblethree), a list of the missing files (accumulated in the \mtc@listmisslanguages macro), and a macro (\mtc@addmisslanguage) to add a file name to the list:

```
\def\@gobblethree#1#2#3{\empty}
\def\mtc@listmisslanguages{}
\def\mtc@addmisslanguage#1{%
    \let\mtc@LML\mtc@listmisslanguages
    \edef\mtc@listmisslanguages{\mtc@LML \MessageBreak #1}}
```

\if@mtc@misslang
\mtc@setlangopt
\mtc@setlangopto
\IfFileExists
\DeclareOption
\mtc@addmisslanguage
\mtcPackageWarningNoLine

Before defining a language option, we must verify that the corresponding .mld file exists, and, if necessary, that the corresponding .mlo file exists. Hence, we must first define a flag \if@mtc@misslang and two macros to test the presence of these files; if the files are available, we define the language option.

```
\newif\if@mtc@misslang\@mtc@misslangfalse
\newcommand{\mtc@setlangopt}[1]{%
   \IfFileExists{#1.mld}%
      {\DeclareOption{#1}{\mtcselectlanguage{#1}}}%
      {\@mtc@misslangtrue
       \mtc@addmisslanguage{#1.mld}
       \mtcPackageInfo[<I0050>]{minitoc}%
          {The required "#1.mld" file is missing.
           \MessageBreak
           The "#1" language option
           \MessageBreak
           will not be available.
           \MessageBreak
           Please install it from a recent distribution
           \MessageBreak
           or from the CTAN archives\@gobble}}%
}%
\newcommand{\mtc@setlangopto}[1]{%
   \IfFileExists{#1.mlo}%
```

```
    {\mtc@setlangopt{#1}}%
    {\@mtc@misslangtrue
     \mtc@addmisslanguage{#1.mlo}
      \mtcPackageInfo[<I0051>]{minitoc}%
         {The required "#1.mlo" file is missing.
          \MessageBreak
          The "#1" language option
          \MessageBreak
          will not be available.
          \MessageBreak
          Please install it from a recent distribution
          \MessageBreak
          or from the CTAN archives\@gobble}%
  \IfFileExists{#1.mld}{}%
      {\@mtc@misslangtrue
       \mtc@addmisslanguage{#1.mld}
      \mtcPackageInfo[<I0050>]{minitoc}%
         {The required "#1.mld" file is missing.
          \MessageBreak
          The "#1" language option will not be available.
          \MessageBreak
          Please install it from a recent distribution
          \MessageBreak
          or from the CTAN archives\@gobble}%
      }%
    }%
}%
```

\if@mtc@misslang
\mtc@setlangoptm
\IfFileExists
\DeclareOption
\mtcPackageError
\mtc@addmisslanguage

Some .mld files are mandatory (english.mld because english is the default language), so their absence is a serious error:

```
\newcommand{\mtc@setlangoptm}[1]{%
  \IfFileExists{#1.mld}%
     {\DeclareOption{#1}{\mtcselectlanguage{#1}}}%
     {\@mtc@misslangtrue
      \mtc@addmisslanguage{#1.mld}
      \mtcPackageError[<E0038>]{minitoc}%
         {Your minitoc installation is incomplete.
          \MessageBreak
          A mandatory minitoc language object file,
          \MessageBreak
          #1.mld, is not found.
          \MessageBreak
          We will try to continue with
          \MessageBreak
          current/default values}%
         {See the minitoc documentation.
          \MessageBreak
          Please fix your minitoc installation.
          \MessageBreak
          Press <return> to continue}%
```

\providecommand
\ptctitle
\plftitle
\plttitle
\mtctitle
\mlftitle
\mlttitle
\stctitle
\slftitle
\slttitle

We must define the default titles (english):

```
          \providecommand{\ptctitle}{Table of Contents}%
          \providecommand{\plftitle}{List of Figures}%
          \providecommand{\plttitle}{List of Tables}%
          \providecommand{\mtctitle}{Contents}%
          \providecommand{\mlftitle}{Figures}%
          \providecommand{\mlttitle}{Tables}%
          \providecommand{\stctitle}{Contents}%
          \providecommand{\slftitle}{Figures}%
          \providecommand{\slttitle}{Tables}%
}}%
```

\AtEndDocument
\if@mtc@misslang
\mtcPackageWarningNoLine
\MessageBreak
\mtc@listmisslanguages
\@gobblethre

If a .mld or .mlo file is missing, we signal that at the end of the document:

```
\AtEndDocument{%
  \if@mtc@misslang
     \mtcPackageWarningNoLine[<W0093>]{minitoc}%
       {Some "*.mld" or "*.mlo" files are missing
        \MessageBreak
        in your installation.
        \MessageBreak
        Search for the I0050 and I0051 info messages
        \MessageBreak
        in the \string\jobname.log file.
        \MessageBreak
        Please install the missing files from
        \MessageBreak
        a recent distribution
        \MessageBreak
        or from the CTAN archives}%
     \mtcPackageWarningNoLine[<W0094>]{minitoc}%
       {Missing minitoc language file(s)\string:
        \MessageBreak
        \mtc@listmisslanguages\@gobblethree}%
  \fi
}%
```

\DeclareOption
\mtc@setlangopt
\mtc@setlangopto
\mtc@setlangoptm

Each language option reads the corresponding *language*.mld file via the specialized macro \mtcselectlanguage, after verification by \mtc@setlangopt or \mtc@setlangopto (when a .mlo file is required), by \mtc@setlangoptm when the language is mandatory. If the file does not exist, a standard error message is displayed. The language options are (should be) in alphabetical order (to make maintenance easier). Several options could load the same file, but, by convention, there should be a *language*.mld file for each language option, given that this file may load another one (as american.mld loads english.mld).

```
\mtc@setlangopt{acadian}%
\mtc@setlangopt{acadien}%
\mtc@setlangopt{afrikaan}%
\mtc@setlangopt{afrikaans}%
```

```
\mtc@setlangopt{albanian}%
\mtc@setlangopt{american}%
\mtc@setlangopt{arab}%
\mtc@setlangopt{arab2}%
\mtc@setlangopt{arabi}%
\mtc@setlangopt{arabic}%
\mtc@setlangopt{armenian}%
\mtc@setlangopt{australian}%
\mtc@setlangopt{austrian}%
\mtc@setlangopt{bahasa}%
\mtc@setlangopt{bahasai}%
\mtc@setlangopt{bahasam}%
\mtc@setlangopt{bangla}%
\mtc@setlangopt{basque}%
\mtc@setlangopt{bicig}%
\mtc@setlangopt{bicig2}%
\mtc@setlangopt{bicig3}%
\mtc@setlangopt{bithe}%
\mtc@setlangopt{brazil}%
\mtc@setlangopt{brazilian}%
\mtc@setlangopt{breton}%
\mtc@setlangopt{british}%
\mtc@setlangopt{bulgarian}%
\mtc@setlangopt{bulgarianb}%
\mtc@setlangopt{buryat}%
\mtc@setlangopt{buryat2}%
\mtc@setlangopt{canadian}%
\mtc@setlangopt{canadien}%
\mtc@setlangopt{castillan}%
\mtc@setlangopt{castillian}%
\mtc@setlangopt{catalan}%
\mtc@setlangopto{chinese1}%            % .mlo
\mtc@setlangopto{chinese2}%            % .mlo
\mtc@setlangopt{croatian}%
\mtc@setlangopt{czech}%
\mtc@setlangopt{danish}%
\mtc@setlangopt{devanagari}%
\mtc@setlangopt{dutch}%
\mtc@setlangoptm{english}%             % mandatory
\mtc@setlangopt{english1}%
\mtc@setlangopt{english2}%
\mtc@setlangopt{esperant}%
\mtc@setlangopt{esperanto}%
\mtc@setlangopt{estonian}%
\mtc@setlangopt{ethiopia}%
\mtc@setlangopt{ethiopian}%
\mtc@setlangopt{ethiopian2}%
\mtc@setlangopto{farsi1}%              % .mlo
\mtc@setlangopto{farsi2}%              % .mlo
\mtc@setlangopt{farsi3}%
\mtc@setlangopt{finnish}%
\mtc@setlangopt{finnish2}%
\mtc@setlangopt{francais}%
\mtc@setlangopt{french}%
```

```
\mtc@setlangopt{french1}%
\mtc@setlangopt{french2}%
\mtc@setlangopt{frenchb}%
\mtc@setlangopt{frenchle}%
\mtc@setlangopt{frenchpro}%
\mtc@setlangopt{galician}%
\mtc@setlangopt{german}%
\mtc@setlangopt{germanb}%
\mtc@setlangopt{germanb2}%
\mtc@setlangopt{greek}%
\mtc@setlangopt{greek-mono}%
\mtc@setlangopt{greek-polydemo}%
\mtc@setlangopt{greek-polykatha}%
\mtc@setlangopt{guarani}%
\mtc@setlangopto{hangul1}%              % .mlo
\mtc@setlangopto{hangul2}%              % .mlo
\mtc@setlangopto{hangul3}%              % .mlo
\mtc@setlangopto{hangul4}%              % .mlo
\mtc@setlangopto{hangul-u8}%            % .mlo
\mtc@setlangopto{hanja1}%               % .mlo
\mtc@setlangopto{hanja2}%               % .mlo
\mtc@setlangopto{hanja-u8}%             % .mlo
\mtc@setlangopt{hebrew}%
\mtc@setlangopt{hebrew2}%
\mtc@setlangopt{hindi}%
\mtc@setlangopt{hindi-modern}%
\mtc@setlangopt{hungarian}%
\mtc@setlangopt{icelandic}%
\mtc@setlangopt{indon}%
\mtc@setlangopt{indonesian}%
\mtc@setlangopt{interlingua}%
\mtc@setlangopt{irish}%
\mtc@setlangopt{italian}%
\mtc@setlangopt{italian2}%
\mtc@setlangopto{japanese}%             % .mlo
\mtc@setlangopto{japanese2}%            % .mlo
\mtc@setlangopto{japanese3}%            % .mlo
\mtc@setlangopto{japanese4}%            % .mlo
\mtc@setlangopto{japanese5}%            % .mlo
\mtc@setlangopto{japanese6}%            % .mlo
\mtc@setlangopt{kannada}%
\mtc@setlangopt{khalkha}%
\mtc@setlangopt{latin}%
\mtc@setlangopt{latin2}%
\mtc@setlangopt{latvian}%
\mtc@setlangopt{letton}%
\mtc@setlangopt{lithuanian}%
\mtc@setlangopt{lowersorbian}%
\mtc@setlangopt{lsorbian}%
\mtc@setlangopt{magyar}%
\mtc@setlangopt{magyar2}%
\mtc@setlangopt{magyar3}%
\mtc@setlangopt{malay}%
\mtc@setlangopt{malayalam-keli}%
```

```
\mtc@setlangopto{malayalam-omega}%     % .mlo
\mtc@setlangopt{malayalam-rachana}%
\mtc@setlangopt{malayalam-rachana2}%
\mtc@setlangopt{manju}%
\mtc@setlangopt{meyalu}%
\mtc@setlangopt{mongol}%
\mtc@setlangopt{naustrian}%
\mtc@setlangopt{ngerman}%
\mtc@setlangopt{newzealand}%
\mtc@setlangopt{ngermanb}%
\mtc@setlangopt{ngermanb2}%
\mtc@setlangopt{norsk}%
\mtc@setlangopt{norsk2}%
\mtc@setlangopt{nynorsk}%
\mtc@setlangopt{nynorsk2}%
\mtc@setlangopt{polish}%
\mtc@setlangopt{polish2}%
\mtc@setlangopt{polski}%
\mtc@setlangopt{portuges}%
\mtc@setlangopt{portuguese}%
\mtc@setlangopt{romanian}%
\mtc@setlangopt{romanian2}%
\mtc@setlangopt{romanian3}%
\mtc@setlangopt{russian}%
\mtc@setlangopt{russianb}%
\mtc@setlangopt{russianc}%
\mtc@setlangopt{russian2m}%
\mtc@setlangopt{russian2o}%
\mtc@setlangopto{russian-cca}%         % .mlo
\mtc@setlangopto{russian-cca1}%        % .mlo
\mtc@setlangopto{russian-lh}%          % .mlo
\mtc@setlangopto{russian-lhcyralt}%    % .mlo
\mtc@setlangopto{russian-lhcyrkoi}%    % .mlo
\mtc@setlangopto{russian-lhcyrwin}%    % .mlo
\mtc@setlangopt{samin}%
\mtc@setlangopt{scottish}%
\mtc@setlangopt{serbian}%
\mtc@setlangopt{serbianc}%
\mtc@setlangopt{slovak}%
\mtc@setlangopt{slovene}%
\mtc@setlangopt{spanish}%
\mtc@setlangopt{spanish2}%
\mtc@setlangopt{spanish3}%
\mtc@setlangopt{spanish4}%
\mtc@setlangopt{swedish}%
\mtc@setlangopt{swedish2}%
\mtc@setlangopto{thai}%                % .mlo
\mtc@setlangopt{turkish}%
\mtc@setlangopt{uighur}%
\mtc@setlangopt{uighur2}%
\mtc@setlangopt{uighur3}%
\mtc@setlangopt{UKenglish}%
\mtc@setlangopt{ukraineb}%
\mtc@setlangopt{ukrainian}%
```

```
\mtc@setlangopt{uppersorbian}%
\mtc@setlangopt{USenglish}%
\mtc@setlangopt{usorbian}%
\mtc@setlangopt{vietnam}%
\mtc@setlangopt{vietnamese}%
\mtc@setlangopt{welsh}%
\mtc@setlangopt{xalx}%
\mtc@setlangopt{xalx2}%
\mtc@setlangopt{xalx3}%
```

9.78 The hints option

\DeclareOption
\if@mtc@hints@

We declare the hints (default) and nohints options:

```
\DeclareOption{hints}{\@mtc@hints@true}
\DeclareOption{nohints}{\@mtc@hints@false}
```

\mtc@hints@begindoc
\AtBeginDocument

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, then 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
\AtEndDocument

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 *document*.log file and/or warnings on the screen and in the *document*.log file. Note that the hints option does not signal errors, only infos and warnings, so it does not stop the LaTeX run.

9.78.1 First part: \mtc@hints@begindoc

\if@mtc@abstract@loaded@
\mtc@hints@begindoc
\if@mtc@toc@used@
\if@mtc@lof@used@
\if@mtc@lot@used@

We declare some flags and the first part of the hints option (for a \AtBeginDocument block):

```
\newif\if@mtc@abstract@loaded@ \@mtc@abstract@loaded@false
\newif\if@mtc@toc@used@ \global\@mtc@toc@used@false
\newif\if@mtc@lof@used@ \global\@mtc@lof@used@false
\newif\if@mtc@lot@used@ \global\@mtc@lot@used@false
\def\mtc@hints@begindoc{%
\mtcPackageInfo[<I0049>]{minitoc(hints)}%
   {==> You requested the hints option
    \MessageBreak
    Some hints are eventually given below\@gobble}%
```

9.78.1.1 Hint about the alphanum package

\@ifpackageloaded
\if@mtc@hints@given@

We test the presence of the alphanum package (part of the jura class [85]), and emit a warning, because this package is incompatible with minitoc:

```
\@ifpackageloaded{alphanum}%
   {\@mtc@hints@given@true
    \mtcPackageWarningNoLine[<W0025>]{minitoc(hints)}%
       {--- The alphanum package is loaded.
        \MessageBreak
        It is incompatible
        \MessageBreak
        with the minitoc package}}{}%
```

9.78.1.2 Hint about the appendix package

\@ifpackageloaded
\if@mtc@hints@given@

We test the presence of the appendix package [252]:

```
\@ifpackageloaded{appendix}{%
\@mtc@hints@given@true
\mtcPackageInfo[<I0042>]{minitoc(hints)}%
   {--- The appendix package is loaded.
    \MessageBreak
    See the minitoc package documentation
    \MessageBreak
    for specific precautions\@gobble}}{}%
```

9.78.1.3 Hint about the tocbibind package

\@ifpackageloaded
\if@mtc@hints@given@

We test the presence of the tocbibind package [253]:

```
\@ifpackageloaded{tocbibind}%
   {\@mtc@hints@given@true
   \mtcPackageInfo[<I0046>]{minitoc(hints)}%
      {--- The tocbibind package is loaded.
       \MessageBreak
       See the minitoc package documentation
       \MessageBreak
       for specific precautions\@gobble}}{}%
```

9.78.1.4 Hint about the KOMA-Script classes

\@ifclassloaded
\if@mtc@hints@given@

We test the presence of each minitoc-compatible KOMA-Script class [147, 195]:

```
\@ifclassloaded{scrbook}%
   {\@mtc@hints@given@true
   \mtcPackageInfo[<I0043>]{minitoc(hints)}%
      {--- The KOMAScript scrbook class is loaded.
       \MessageBreak
       See the minitoc package documentation
       \MessageBreak
       for specific precautions\@gobble}}{}%
\@ifclassloaded{scrreprt}%
   {\@mtc@hints@given@true
   \mtcPackageInfo[<I0043>]{minitoc(hints)}%
      {--- The KOMAScript scrreprt class is loaded.
       \MessageBreak
       See the minitoc package documentation
       \MessageBreak
       for specific precautions\@gobble}}{}%
\@ifclassloaded{scrartcl}%
   {\@mtc@hints@given@true
   \mtcPackageInfo[<I0043>]{minitoc(hints)}%
      {--- The KOMAScript scrartcl class is loaded.
       \MessageBreak
       See the minitoc package documentation
       \MessageBreak
       for specific precautions\@gobble}}{}%
```

9.78.1.5 Hint about the tocloft package

\@ifpackageloaded
\if@mtc@hints@given@

We test the presence of the tocloft package [250]:

```
\@ifpackageloaded{tocloft}%
   {\@mtc@hints@given@true
    \mtcPackageInfo[<I0047>]{minitoc(hints)}%
       {--- The tocloft package is loaded.
        \MessageBreak
        See the minitoc package documentation
        \MessageBreak
        for specific precautions\@gobble}}{}%
```

9.78.1.6 Hint about the titletoc package

\@ifpackageloaded
\if@mtc@hints@given@

We test the presence of the titletoc package [33], and emit a warning, because this package is incompatible with minitoc:

```
\@ifpackageloaded{titletoc}%
   {\@mtc@hints@given@true
    \mtcPackageWarningNoLine[<W0040>]{minitoc(hints)}%
       {--- The titletoc package is loaded.
        \MessageBreak
```

```
        It is incompatible
        \MessageBreak
        with the minitoc package}}{}%
```

9.78.1.7 Hint about the placeins package

\@ifpackageloaded
\if@mtc@ss@insection@
\@ifpackagewith
\if@mtc@hints@given@

We test if the placeins package [10] is loaded and, if yes, we check that the right options are selected:

```
\@ifpackageloaded{placeins}%
  {\if@mtc@ss@insection@
   \@ifpackagewith{placeins}{section}{}%
      {\@mtc@hints@given@true
       \mtcPackageWarningNoLine[<W0031>]{minitoc(hints)}%
          {--- The placeins package is loaded
           \MessageBreak
           without the section option,
           \MessageBreak
           but minitoc used the insection option
           \MessageBreak
           which implies it. Try to inverse the
           \MessageBreak
           loading order and use consistent options.
           \MessageBreak
           You may have got a message
           \MessageBreak
   ! LaTeX Error: Option clash for package placeins}%
      }%
   \@ifpackagewith{placeins}{above}%
      {\@mtc@hints@given@true
       \mtcPackageWarningNoLine[<W0084>]{minitoc(hints)}%
          {--- The placeins package is loaded
           \MessageBreak
           with the above option,
           \MessageBreak
           but minitoc used the insection option
           \MessageBreak
           which is incompatible with it.
           \MessageBreak
           Try to remove the above option
           \MessageBreak
           and use consistent options}%
      }{}%
   \@ifpackagewith{placeins}{below}%
      {\@mtc@hints@given@true
       \mtcPackageWarningNoLine[<W0085>]{minitoc(hints)}%
          {--- The placeins package is loaded
           \MessageBreak
           with the below option,
           \MessageBreak
           but minitoc used the insection option
```

```
            \MessageBreak
            which is incompatible with it.
            \MessageBreak
            Try to remove the below option
            \MessageBreak
            and use consistent options}%
        }{}%
    \fi
   \@ifpackagelater{placeins}{2005/04/18}{}{%
      \@mtc@hints@given@true
      \mtcPackageWarningNoLine[<W0032>]{minitoc(hints)}%
         {--- The placeins package loaded is
          \MessageBreak
          too old. You should use a version
          \MessageBreak
          dated of 2005/04/18 at least}%
   }%
   }{}%
```

9.78.1.8 Hint about the memoir class

\@ifclassloaded
\if@mtc@hints@given@

We test if the memoir class [257, 258] is loaded:

```
\@ifclassloaded{memoir}%
   {\@mtc@hints@given@true
    \mtcPackageInfo[<I0044>]{minitoc(hints)}%
       {--- The memoir class is loaded.
        \MessageBreak
        See the minitoc package documentation
        \MessageBreak
        for specific precautions\@gobble}{}%
   }{}%
```

9.78.1.9 Hint about the amsart and amsproc classes

\@ifclassloaded
\if@mtc@hints@given@

We test if the amsart or amsproc class is loaded and emit a warning, because these classes are incompatible with minitoc:

```
\@ifclassloaded{amsart}%
   {\@mtc@hints@given@true
    \mtcPackageWarningNoLine[<W0026>]{minitoc(hints)}%
       {--- The amsart class is loaded.
        \MessageBreak
        It is incompatible
        \MessageBreak
        with the minitoc package}}{}%
\@ifclassloaded{amsproc}%
```

```
   {\@mtc@hints@given@true
    \mtcPackageWarningNoLine[<W0027>]{minitoc(hints)}%
       {--- The amsproc class is loaded.
        \MessageBreak
        It is incompatible
        \MessageBreak with the minitoc package}}{}%
```

9.78.1.10 Hint about the amsbook class

\@ifclassloaded
\if@mtc@hints@given@

We test if the amsbook class is loaded:

```
\@ifclassloaded{amsbook}%
   {\@mtc@hints@given@true
    \mtcPackageInfo[<I0041>]{minitoc(hints)}%
       {--- The amsbook class is loaded.
        \MessageBreak
        See the minitoc package documentation
        \MessageBreak
        for specific precautions\@gobble}{}%
   }{}%
```

9.78.1.11 Hint about the abstract package

\@ifpackageloaded
\@ifpackagewith
\if@mtc@hints@given@

We test the presence of the abstract package [251], then its options:

```
\@ifpackageloaded{abstract}%
   {\@mtc@abstract@loaded@true%
   \@ifpackagewith{abstract}{addtotoc}%
      {\@mtc@hints@given@true
       \mtcPackageInfo[<I0040>]{minitoc(hints)}%
          {The ‘‘abstract’’ package has been
           \MessageBreak
           loaded with the ‘‘addtotoc’’ option.
           \MessageBreak
           You need to look at the
           \MessageBreak
           documentation to adjust.
           \MessageBreak}
      }}{}
```

9.78.1.12 Hint about the jura class

\@ifclassloaded
\if@mtc@hints@given@

We test if the jura class is loaded and emit a warning, because this class is incompatible with minitoc:

```
\@ifclassloaded{jura}%
  {\@mtc@hints@given@true
   \mtcPackageWarningNoLine[<W0029>]{minitoc(hints)}%
      {--- The jura class is loaded.
       \MessageBreak
       It is incompatible
       \MessageBreak with the minitoc package}}{}%
```

9.78.1.13 Hint about the alteration of the sectionning commands

\part
\chapter
\section
\AtBeginDocument
\mtc@hints@part
\mtc@hints@chapter
\mtc@hints@section

To check if the sectionning commands \part, \chapter or \section have been altered by some package or in the preamble, we compare them (when executing a \AtBeginDocument block) with their saved versions (saved by the minitoc package when it is loaded) \mtc@hints@part, \mtc@hints@chapter and \mtc@hints@section. For each sectionning command, we must perform the comparaison for the command itself, its unstarred branch and its starred branch. But the hyperref package [214] may interfere, hence the formal precautions in the messages.

9.78.1.13.1 Alteration of \part

\if@mtc@hints@w@
\@ifundefined
\part
\mtc@hints@part
\@part
\mtc@hints@@part
\@spart
\mtc@hints@@spart
\if@mtc@hints@given@
\if@mtc@hyper@used@

We check the alteration of \part, \@part and \@spart:

```
\@mtc@hints@w@false
\@ifundefined{part}{}{\ifx\part\mtc@hints@part\relax
   \else\@mtc@hints@w@true\fi}
\@ifundefined{part}{}{\ifx\@part\mtc@hints@@part\relax
   \else\@mtc@hints@w@true\fi}
\@ifundefined{part}{}{\ifx\@spart\mtc@hints@@spart\relax
   \else\@mtc@hints@w@true\fi}
\if@mtc@hints@w@\@mtc@hints@given@true%
  \mtcPackageWarningNoLine[<W0030>]{minitoc(hints)}%
     {--- The \string\part\space command is altered
          \MessageBreak
          after minitoc}
  \if@mtc@hyper@used@
     \mtcPackageWarningNoLine[<W0023>]{minitoc(hints)}%
        {--- It may be the consequence
             \MessageBreak
             of loading the ``hyperref'' package}
  \fi
\fi
```

9.78.1.13.2 Alteration of \chapter

\if@mtc@hints@w@
\@ifundefined
\chapter
\mtc@hints@chapter
\@chapter
\mtc@hints@@chapter
\@schapter
\mtc@hints@@schapter
\if@mtc@hints@given@
\if@mtc@hyper@used@

We check the alteration of \chapter, \@chapter and \@schapter:

```
\@mtc@hints@w@false
\@ifundefined{chapter}{}{\ifx\chapter\mtc@hints@chapter\relax
   \else\@mtc@hints@w@true\fi}%
\@ifundefined{chapter}{}{\ifx\@chapter\mtc@hints@@chapter\relax
   \else\@mtc@hints@w@true\fi}%
\@ifundefined{chapter}{}{\ifx\@schapter\mtc@hints@@schapter\relax
   \else\@mtc@hints@w@true\fi}%
\if@mtc@hints@w@\@mtc@hints@given@true%
  \mtcPackageWarningNoLine[<W0028>]{minitoc(hints)}%
     {--- The \string\chapter\space command is altered
          \MessageBreak
          after minitoc}
  \if@mtc@hyper@used@
     \mtcPackageWarningNoLine[<W0023>]{minitoc(hints)}%
        {--- It may be the consequence
             \MessageBreak
             of loading the ``hyperref'' package}
  \fi
\fi
```

9.78.1.13.3 Alteration of \section

\if@mtc@hints@w@
\@ifundefined
\section
\mtc@hints@section
\@sect
\mtc@hints@@sect
\@ssect
\mtc@hints@@ssect
\if@mtc@hints@given@
\if@mtc@hyper@used@

We check the alteration of \section, \@sect and \@ssect:

```
\@mtc@hints@w@false
\@ifundefined{chapter}%
   {\@ifundefined{section}{}{\ifx\section\mtc@hints@section\relax\else
      \@mtc@hints@w@true\fi}
   \@ifundefined{section}{}{\ifx\@sect\mtc@hints@@sect\relax\else
      \@mtc@hints@w@true\fi}
   \@ifundefined{section}{}{\ifx\@ssect\mtc@hints@@ssect\relax\else
      \@mtc@hints@w@true\fi}
\if@mtc@hints@w@\@mtc@hints@given@true%
  \mtcPackageWarningNoLine[<W0039>]{minitoc(hints)}%
     {--- The \string\section\space command is altered
          \MessageBreak
          after minitoc}
  \if@mtc@hyper@used@
     \mtcPackageWarningNoLine[<W0023>]{minitoc(hints)}%
        {--- It may be the consequence
             \MessageBreak
             of loading the ``hyperref'' package}
  \fi
  \relax\else\fi}{}%
```

9.78.1.14 Hint about the consistency of the calling sequences of some commands

And finally, we prepare the consistency 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@
\if@mtc@toc@used@
\if@mtc@lof@used@
\if@mtc@lot@used@

Hence we initialize some flags:

```
\if@mtc@hints@
\global\@mtc@toc@used@false
\global\@mtc@lof@used@false
\global\@mtc@lot@used@false
```

\mtc@sv@tableofcontents
\tableofcontents
\if@mtc@toc@used@
\mtc@sv@listoffigures
\listoffigures
\if@mtc@lof@used@
\mtc@sv@listoftables
\listoftables
\if@mtc@lot@used@

Then we patch the involved commands to set the corresponding flag when they are used. First, the commands for the main summaries:

```
\let\mtc@sv@tableofcontents\tableofcontents
\def\tableofcontents%
    {\global\@mtc@toc@used@true\mtc@sv@tableofcontents}
\let\mtc@sv@listoffigures\listoffigures
\def\listoffigures%
    {\global\@mtc@lof@used@true\mtc@sv@listoffigures}
\let\mtc@sv@listoftables\listoftables
\def\listoftables%
    {\global\@mtc@lot@used@true\mtc@sv@listoftables}
```

\mtc@sv@fktableofcontents
\faketableofcontents
\if@mtc@toc@used@
\mtc@sv@fklistoffigures
\fakelistoffigures
\if@mtc@lof@used@
\mtc@sv@fklistoftables
\fakelistoftables
\if@mtc@lot@used@

Then, their "fake" siblings:

```
\let\mtc@sv@fktableofcontents\faketableofcontents
\def\faketableofcontents%
    {\global\@mtc@toc@used@true\mtc@sv@fktableofcontents}
\let\mtc@sv@fklistoffigures\fakelistoffigures
\def\fakelistoffigures%
    {\global\@mtc@lof@used@true\mtc@sv@fklistoffigures}
\let\mtc@sv@fklistoftables\fakelistoftables
\def\fakelistoftables%
    {\global\@mtc@lot@used@true\mtc@sv@fklistoftables}
\fi
```

\mtc@hints@begindoc

And the \mtc@hints@begindoc definition is finished (it begins in section 9.78.1 on page 382):

```
}
```

9.78.2 Final part: \mtc@hints@enddoc

\mtc@hints@enddoc
\AtEndDocument

The final part of the hints option is executed via \AtEndDocument. Its code is in the \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:

```
\def\mtc@hints@enddoc{%
```

9.78.2.1 Hint about \sect-lof|lot and the insection option

\if@mtc@sect@floats@
\if@dosectlof@used@
\if@dosectlot@used@
\if@sectlof@used@
\if@sectlot@used@
\if@mtc@section@def@

We look if some section-level lists of figures or tables have been requested.

```
\if@dosectlof@used@\@mtc@sect@floats@true\fi
\if@dosectlot@used@\@mtc@sect@floats@true\fi
\if@sectlof@used@\@mtc@sect@floats@true\fi
\if@sectlot@used@\@mtc@sect@floats@true\fi
\if@mtc@section@def@
```

\if@mtc@placeinsLoaded@
\if@mtc@sect@floats@
\if@mtc@hints@w@
\if@mtc@hints@given@

If yes, we verify that the placeins package [10] 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@placeinsLoaded@ \else
  \if@mtc@sect@floats@%
   \mtcPackageWarningNoLine[<W0056>]{minitoc(hints)}%
      {You are using \string\dosectlof\space and/or
       \MessageBreak
       \string\dosectlot, \string\sectlof\space and/or \string\sectlot,
       \MessageBreak
       hence the ``insection'' package
       \MessageBreak
       option is recommended}%
   \@mtc@hints@w@true \@mtc@hints@given@true
  \fi
\fi
```

9.78.2.2 Final part of the consistency tests

We test if \parttoc has been used without \doparttoc, etc., for each pair of preparation/insertion commands.

\if@mtc@part@def@
\if@parttoc@used@
\if@doparttoc@used@
\if@mtc@hints@given@
\if@partlof@used@
\if@dopartlof@used@
\if@partlot@used@
\if@dopartlot@used@

For the part level commands:

```
\if@mtc@part@def@
\if@parttoc@used@
  \if@doparttoc@used@\else
  \mtcPackageWarningNoLine[<W0062>]{minitoc(hints)}%
     {You have used \string\parttoc,
      \MessageBreak
      but not \string\doparttoc}
  \@mtc@hints@given@true
  \fi
\fi
\if@partlof@used@
  \if@dopartlof@used@\else
  \mtcPackageWarningNoLine[<W0060>]{minitoc(hints)}%
     {You have used \string\partlof,
      \MessageBreak
      but not \string\dopartlof}
  \@mtc@hints@given@true
  \fi
\fi
\if@partlot@used@
  \if@dopartlot@used@\else
  \mtcPackageWarningNoLine[<W0061>]{minitoc(hints)}%
     {You have used \string\partlot,
      \MessageBreak
      but not \string\dopartlot}
  \@mtc@hints@given@true
  \fi
\fi
\fi
```

\if@mtc@chapter@def@
\if@minitoc@used@
\if@dominitoc@used@
\if@mtc@hints@given@
\if@minilof@used@
\if@dominilof@used@
\if@minilot@used@
\if@dominilot@used@

For the chapter level commands:

```
\if@mtc@chapter@def@
\if@minitoc@used@
  \if@dominitoc@used@\else
  \mtcPackageWarningNoLine[<W0059>]{minitoc(hints)}%
     {You have used \string\minitoc,
      \MessageBreak
      but not \string\dominitoc}
  \@mtc@hints@given@true
  \fi
\fi
\if@minilof@used@
  \if@dominilof@used@\else
  \mtcPackageWarningNoLine[<W0057>]{minitoc(hints)}%
     {You have used \string\minilof,
      \MessageBreak
      but not \string\dominilof}
  \@mtc@hints@given@true
  \fi
```

```
\fi
\if@minilot@used@
  \if@dominilot@used@\else
  \mtcPackageWarningNoLine[<W0058>]{minitoc(hints)}%
     {You have used \string\minilot,
      \MessageBreak
      but not \string\dominilot}
  \@mtc@hints@given@true
  \fi
\fi
\fi
```

\if@mtc@section@def@
\if@secttoc@used@
\if@dosecttoc@used@
\if@mtc@hints@given@
\if@sectlof@used@
\if@dosectlof@used@
\if@sectlot@used@
\if@dosectlot@used@

For the section level commands:

```
\if@mtc@section@def@
\if@secttoc@used@
  \if@dosecttoc@used@\else
  \mtcPackageWarningNoLine[<W0065>]{minitoc(hints)}%
     {You have used \string\secttoc,
      \MessageBreak
      but not \string\dosecttoc}
  \@mtc@hints@given@true
  \fi
\fi
\if@sectlof@used@
  \if@dosectlof@used@\else
  \mtcPackageWarningNoLine[<W0063>]{minitoc(hints)}%
     {You have used \string\sectlof,
      \MessageBreak
      but not \string\dosectlof}
  \@mtc@hints@given@true
  \fi
\fi
\if@sectlot@used@
  \if@dosectlot@used@\else
  \mtcPackageWarningNoLine[<W0064>]{minitoc(hints)}%
     {You have used \string\sectlot,
      \MessageBreak
      but not \string\dosectlot}
  \fi
\fi
\fi
```

9.78.2.3 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 it is a vain call. We do the same for each analog command.

\if@mtc@part@def@
\if@doparttoc@used@
\if@parttoc@used@
\if@mtc@hints@given@
\if@dopartlof@used@
\if@partlof@used@
\if@dopartlot@used@
\if@partlot@used@

Part level commands:

```
\if@mtc@part@def@
 \if@doparttoc@used@
   \if@parttoc@used@\else
   \mtcPackageWarningNoLine[<W0075>]{minitoc(hints)}%
      {You have used \string\doparttoc,
       \MessageBreak
       but not \string\parttoc}
   \@mtc@hints@given@true
   \fi
 \fi
 \if@dopartlof@used@
   \if@partlof@used@\else
   \mtcPackageWarningNoLine[<W0076>]{minitoc(hints)}%
      {You have used \string\dopartlof,
       \MessageBreak
       but not \string\partlof}
   \@mtc@hints@given@true
   \fi
 \fi
 \if@dopartlot@used@
   \if@partlot@used@\else
   \mtcPackageWarningNoLine[<W0077>]{minitoc(hints)}%
      {You have used \string\dopartlot,
       \MessageBreak
       but not \string\partlot}
   \@mtc@hints@given@true
   \fi
 \fi
\fi
```

\if@mtc@chapter@def@
\if@dominitoc@used@
\if@minitoc@used@
\if@mtc@hints@given@
\if@dominilof@used@
\if@minilof@used@
\if@dominilot@used@
\if@minilot@used@

Chapter level commands:

```
\if@mtc@chapter@def@
 \if@dominitoc@used@
   \if@minitoc@used@\else
   \mtcPackageWarningNoLine[<W0078>]{minitoc(hints)}%
      {You have used \string\dominitoc,
       \MessageBreak
       but not \string\minitoc}
   \@mtc@hints@given@true
   \fi
 \fi
 \if@dominilof@used@
   \if@minilof@used@\else
   \mtcPackageWarningNoLine[<W0079>]{minitoc(hints)}%
      {You have used \string\dominilof,
       \MessageBreak
       but not \string\minilof}
   \@mtc@hints@given@true
   \fi
```

```
\fi
\if@dominilot@used@
  \if@minilot@used@\else
  \mtcPackageWarningNoLine[<W0080>]{minitoc(hints)}%
     {You have used \string\dominilot,
      \MessageBreak
      but not \string\minilot}
  \@mtc@hints@given@true
  \fi
\fi
\fi
```

\if@mtc@section@def@
\if@dosecttoc@used@
\if@secttoc@used@
\if@mtc@hints@given@
\if@dosectlof@used@
\if@sectlof@used@
\if@dosectlot@used@
\if@sectlot@used@

Section level commands:

```
\if@mtc@section@def@
\if@dosecttoc@used@
  \if@secttoc@used@\else
  \mtcPackageWarningNoLine[<W0081>]{minitoc(hints)}%
     {You have used \string\dosecttoc,
      \MessageBreak
      but not \string\secttoc}
  \@mtc@hints@given@true
  \fi
\fi
\if@dosectlof@used@
  \if@sectlof@used@\else
  \mtcPackageWarningNoLine[<W0082>]{minitoc(hints)}%
     {You have used \string\dosectlof,
      \MessageBreak
      but not \string\sectlof}
  \@mtc@hints@given@true
  \fi
\fi
\if@dosectlot@used@
  \if@sectlot@used@\else
  \mtcPackageWarningNoLine[<W0083>]{minitoc(hints)}%
     {You have used \string\dosectlot,
      \MessageBreak
      but not \string\sectlot}
  \fi
\fi
\fi
\fi
```

9.78.2.4 Check if the main tables have been prepared (second part)

Another consistency test verifies that if the macro `\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@
\if@mtc@hints@given@
\if@parttoc@used@
\ifmtc@toc@used@
\if@partlof@used@
\ifmtc@lof@used@
\if@partlot@used@
\ifmtc@lot@used@

Part level commands:

```
\if@mtc@part@def@
\if@parttoc@used@
  \if@mtc@toc@used@\else
  \mtcPackageWarningNoLine[<W0071>]{minitoc(hints)}%
     {You have used \string\parttoc\space but not
      \MessageBreak
      \string\tableofcontents
      \MessageBreak
      nor \string\faketableofcontents}
  \@mtc@hints@given@true
  \fi
\fi
\if@partlof@used@
  \if@mtc@lof@used@\else
  \mtcPackageWarningNoLine[<W0069>]{minitoc(hints)}%
     {You have used \string\partlof\space but not
      \MessageBreak
      \string\listoffigures
      \MessageBreak
      nor \string\fakelistoffigures}
  \@mtc@hints@given@true
  \fi
\fi
\if@partlot@used@
  \if@mtc@lot@used@\else
  \mtcPackageWarningNoLine[<W0070>]{minitoc(hints)}%
     {You have used \string\partlot\space but not
      \MessageBreak
      \string\listoftables
      \MessageBreak
      nor \string\fakelistoftables}
  \@mtc@hints@given@true
  \fi
\fi
\fi
```

\if@mtc@chapter@def@
\if@mtc@hints@given@
\if@minitoc@used@
\ifmtc@toc@used@
\if@minilof@used@
\ifmtc@lof@used@
\if@minilot@used@
\ifmtc@lot@used@

Chapter level commands:

```
\if@mtc@chapter@def@
\if@minitoc@used@
  \if@mtc@toc@used@\else
  \mtcPackageWarningNoLine[<W0068>]{minitoc(hints)}%
     {You have used \string\minitoc\space but not
      \MessageBreak
      \string\tableofcontents
      \MessageBreak
      nor \string\faketableofcontents}
  \@mtc@hints@given@true
  \fi
\fi
```

```
\if@minilof@used@
  \if@mtc@lof@used@\else
  \mtcPackageWarningNoLine[<W0066>]{minitoc(hints)}%
     {You have used \string\minilof\space but not
      \MessageBreak
      \string\listoffigures
      \MessageBreak
      nor \string\fakelistoffigures}
  \@mtc@hints@given@true
  \fi
\fi
\if@minilot@used@
  \if@mtc@lot@used@\else
  \mtcPackageWarningNoLine[<W0067>]{minitoc(hints)}%
     {You have used \string\minilot\space but not
      \MessageBreak
      \string\listoftables
      \MessageBreak
      nor \string\fakelistoftables}
  \@mtc@hints@given@true
  \fi
\fi
\fi
```

\if@mtc@section@def@
\if@mtc@hints@given@
\if@secttoc@used@
\ifmtc@toc@used@
\if@sectlof@used@
\ifmtc@lof@used@
\if@sectlot@used@
\ifmtc@lot@used@

Section level commands:

```
\if@mtc@section@def@
\if@secttoc@used@
  \if@mtc@toc@used@\else
  \mtcPackageWarningNoLine[<W0074>]{minitoc(hints)}%
     {You have used \string\secttoc\space but not
      \MessageBreak
      \string\tableofcontents
      \MessageBreak
      nor \string\faketableofcontents}
  \@mtc@hints@given@true
  \fi
\fi
\if@sectlof@used@
  \if@mtc@lof@used@\else
  \mtcPackageWarningNoLine[<W0072>]{minitoc(hints)}%
     {You have used \string\sectlof\space but not
      \MessageBreak
      \string\listoffigures
      \MessageBreak
      nor \string\fakelistoffigures}
  \@mtc@hints@given@true
  \fi
\fi
\if@sectlot@used@
  \if@mtc@lot@used@\else
  \mtcPackageWarningNoLine[<W0073>]{minitoc(hints)}%
```

```
      {You have used \string\sectlot\space but not
       \MessageBreak
       \string\listoftables
       \MessageBreak
       nor \string\fakelistoftables}
   \@mtc@hints@given@true
   \fi
 \fi
\fi
```

9.78.2.5 Check the number of mini-tables, in case of short extensions

\mtc@hints@checklongext
\if@longextensions@
\if@mtc@part@def@
\value
\if@mtc@hints@given@true
\if@mtc@chapter@def@
\if@mtc@section@def@

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 [11], either redesign the document.

```
\def\mtc@hints@checklongext{%
\if@longextensions@
\else
  \if@mtc@part@def@
    \ifnum 99 < \value{ptc}\relax
       \@mtc@hints@given@true
       \mtcPackageWarningNoLine[<W0054>]{minitoc(hints)}%
          {You have used short extensions
           \MessageBreak
           and more than 99 parts (\arabic{ptc})}
    \fi
  \fi
  \if@mtc@chapter@def@
    \ifnum 99 < \value{mtc}\relax
       \@mtc@hints@given@true
       \mtcPackageWarningNoLine[<W0053>]{minitoc(hints)}%
          {You have used short extensions
           \MessageBreak
           and more than 99 chapters (\arabic{mtc})}
    \fi
  \else
    \if@mtc@section@def@
      \ifnum 99 < \value{stc}\relax
         \@mtc@hints@given@true
         \mtcPackageWarningNoLine[<W0055>]{minitoc(hints)}%
            {You have used short extensions
             \MessageBreak
             and more than 99 sections (\arabic{stc})}
      \fi
    \fi
  \fi
```

[11] On the long term, a good investment.

```
\fi}
\mtc@hints@checklongext
```

9.78.2.6 Final part of the hint about the sectsty package

\if@mtc@sectstyLoaded@
\if@mtc@sectstyLoaded@a@
\if@mtc@hints@given@

We test if sectsty has been loaded before (correct) or after (incorrect) minitoc. See section 9.9.1 on page 250.

```
\if@mtc@sectstyLoaded@\else
  \if@mtc@sectstyLoaded@a@
     \mtcPackageWarningNoLine[<W0037>]{minitoc(hints)}%
     {The sectsty package should be
      \MessageBreak
      loaded BEFORE the minitoc package}
     \@mtc@hints@given@true
  \fi
\fi
```

9.78.2.7 Final part of the hint about the varsects package

\if@mtc@varsectsLoaded@
\if@mtc@varsectsLoaded@a@
\if@mtc@hints@given@

We test if varsects has been loaded before (correct) or after (incorrect) minitoc. See section 9.9.2 on page 250.

```
\if@mtc@varsectsLoaded@\else
  \if@mtc@varsectsLoaded@a@
     \mtcPackageWarningNoLine[<W0038>]{minitoc(hints)}%
     {The varsects package should be
      \MessageBreak
      loaded BEFORE the minitoc package}
     \@mtc@hints@given@true
  \fi
\fi
```

9.78.2.8 Final part of the hint about the fncychap package

\if@mtc@fncychapLoaded@
\if@mtc@fncychapLoaded@a@
\if@mtc@hints@given@

We test if fncychap has been loaded before (correct) or after (incorrect) minitoc. See section 9.9.3 on page 250.

```
\if@mtc@fncychapLoaded@\else
  \if@mtc@fncychapLoaded@a@
     \mtcPackageWarningNoLine[<W0086>]{minitoc(hints)}%
     {The fncychap package should be
      \MessageBreak
      loaded BEFORE the minitoc package}
```

```
    \@mtc@hints@given@true
  \fi
\fi
```

9.78.2.9 Final part of the hint about the hangcaption package

\if@mtc@HgcLoaded@
\if@mtc@HgcLoaded@a@
\if@mtc@hints@given@

We test if hangcaption has been loaded before (correct) or after (incorrect) minitoc. See section 9.9.4 on page 251.

```
\if@mtc@HgcLoaded@\else
  \if@mtc@HgcLoaded@a@
    \mtcPackageWarningNoLine[<W0092>]{minitoc(hints)}%
    {The hangcaption package should be
     \MessageBreak
     loaded BEFORE the minitoc package}
    \@mtc@hints@given@true
  \fi
\fi
```

9.78.2.10 Final part of the hint about the quotchap package

\if@mtc@quotchapLoaded@
\if@mtc@quotchapLoaded@a@
\if@mtc@hints@given@

We test if quotchap has been loaded before (correct) or after (incorrect) minitoc. See section 9.9.5 on page 251.

```
\if@mtc@quotchapLoaded@\else
  \if@mtc@quotchapLoaded@a@
    \mtcPackageWarningNoLine[<W0087>]{minitoc(hints)}%
    {The quotchap package should be
     \MessageBreak
     loaded BEFORE the minitoc package}
    \@mtc@hints@given@true
  \fi
\fi
```

9.78.2.11 Final part of the hint about the romannum package

\if@mtc@romannumLoaded@
\if@mtc@romannumLoaded@a@
\if@mtc@hints@given@

We test if romannum has been loaded before (correct) or after (incorrect) minitoc. See section 9.9.6 on page 251.

```
\if@mtc@romannumLoaded@\else
  \if@mtc@romannumLoaded@a@
    \mtcPackageWarningNoLine[<W0088>]{minitoc(hints)}%
    {The romannum package should be
     \MessageBreak
```

```
      loaded BEFORE the minitoc package}
     \@mtc@hints@given@true
  \fi
\fi
```

9.78.2.12 Final part of the hint about the sfheaders package

\if@mtc@sfheadersLoaded@
\if@mtc@sfheadersLoaded@a@
\if@mtc@hints@given@

We test if sfheaders has been loaded before (correct) or after (incorrect) minitoc. See section 9.9.7 on page 251.

```
\if@mtc@sfheadersLoaded@\else
  \if@mtc@sfheadersLoaded@a@
     \mtcPackageWarningNoLine[<W0089>]{minitoc(hints)}%
     {The sfheaders package should be
      \MessageBreak
      loaded BEFORE the minitoc package}
     \@mtc@hints@given@true
  \fi
\fi
```

9.78.2.13 Final part of the hint about the alnumsec package

\if@mtc@alnumsecLoaded@
\if@mtc@alnumsecLoaded@a@
\if@mtc@hints@given@

We test if alnumsec has been loaded before (correct) or after (incorrect) minitoc. See section 9.9.8 on page 252.

```
\if@mtc@alnumsecLoaded@\else
  \if@mtc@alnumsecLoaded@a@
     \mtcPackageWarningNoLine[<W0090>]{minitoc(hints)}%
     {The alnumsec package should be
      \MessageBreak
      loaded BEFORE the minitoc package}
     \@mtc@hints@given@true
  \fi
\fi
```

9.78.2.14 Final part of the hint about the captcont package

\if@mtc@captcontLoaded@
\if@mtc@captcontLoaded@a@
\if@mtc@hints@given@

We test if captcont has been loaded before (correct) or after (incorrect) minitoc. See section 9.9.9 on page 252.

```
\if@mtc@captcontLoaded@\else
  \if@mtc@captcontLoaded@a@
     \mtcPackageWarningNoLine[<W0091>]{minitoc(hints)}%
     {The captcont package should be
```

```
      \MessageBreak
      loaded BEFORE the minitoc package}
     \@mtc@hints@given@true
  \fi
\fi
```

9.78.2.15 Final part of the hint about the caption package

\if@mtc@captionLoaded@
\if@mtc@captionLoaded@a@
\if@mtc@hints@given@

We test if caption has been loaded before (correct) or after (incorrect) minitoc. See section 9.9.10 on page 252.

```
\if@mtc@captionLoaded@\else
  \if@mtc@captionLoaded@a@
     \mtcPackageWarningNoLine[<W0033>]{minitoc(hints)}%
     {The caption package should be
      \MessageBreak
      loaded BEFORE the minitoc package}
     \@mtc@hints@given@true
  \fi
\fi
```

9.78.2.16 Final part of the hint about the caption2 package

\if@mtc@captionIILoaded@
\if@mtc@captionIILoaded@a@
\if@mtc@hints@given@

We test if caption2 has been loaded before (correct) or after (incorrect) minitoc. See section 9.9.11 on page 252.

```
\if@mtc@captionIILoaded@\else
  \if@mtc@captionIILoaded@a@
     \mtcPackageWarningNoLine[<W0034>]{minitoc(hints)}%
     {The caption2 package should be
      \MessageBreak
      loaded BEFORE the minitoc package}
     \@mtc@hints@given@true
  \fi
\fi
```

9.78.2.17 Final part of the hint about the ccaption package

\if@mtc@ccaptionLoaded@
\if@mtc@ccaptionLoaded@a@
\if@mtc@hints@given@

We test if ccaption has been loaded before (correct) or after (incorrect) minitoc. See section 9.9.12 on page 253.

```
\if@mtc@ccaptionLoaded@\else
  \if@mtc@ccaptionLoaded@a@
     \mtcPackageWarningNoLine[<W0035>]{minitoc(hints)}%
```

```
    {The ccaption package should be
     \MessageBreak
     loaded BEFORE the minitoc package}
    \@mtc@hints@given@true
  \fi
\fi
```

9.78.2.18 Final part of the hint about the mcaption package

\if@mtc@mcaptionLoaded@
\if@mtc@mcaptionLoaded@a@
\if@mtc@hints@given@

We test if mcaption has been loaded before (correct) or after (incorrect) minitoc. See section 9.9.13 on page 253.

```
\if@mtc@mcaptionLoaded@\else
  \if@mtc@mcaptionLoaded@a@
    \mtcPackageWarningNoLine[<W0036>]{minitoc(hints)}%
    {The mcaption package should be
     \MessageBreak
     loaded BEFORE the minitoc package}
    \@mtc@hints@given@true
  \fi
\fi
```

9.78.2.19 Check if empty mini-tables have been detected

We test for each kind of mini-tables.

\if@mtc@empty@parttoc@

For parttocs:

```
\if@mtc@empty@parttoc@
  \mtcPackageWarningNoLine[<W0046>]{minitoc(hints)}%
  {You have attempted to insert
   \MessageBreak
   empty parttocs}
  \@mtc@hints@given@true
\fi
```

\if@mtc@empty@partlof@

For partlofs:

```
\if@mtc@empty@partlof@
  \mtcPackageWarningNoLine[<W0044>]{minitoc(hints)}%
  {You have attempted to insert
   \MessageBreak
   empty partlofs}
  \@mtc@hints@given@true
\fi
```

`\if@mtc@empty@partlot@` For partlots:

```
\if@mtc@empty@partlot@
   \mtcPackageWarningNoLine[<W0045>]{minitoc(hints)}%
   {You have attempted to insert
    \MessageBreak
    empty partlots}
   \@mtc@hints@given@true
\fi
```

`\if@mtc@empty@minitoc@` For minitocs:

```
\if@mtc@empty@minitoc@
   \mtcPackageWarningNoLine[<W0043>]{minitoc(hints)}%
   {You have attempted to insert
    \MessageBreak
    empty minitocs}
   \@mtc@hints@given@true
\fi
```

`\if@mtc@empty@minilof@` For minilofs:

```
\if@mtc@empty@minilof@
   \mtcPackageWarningNoLine[<W0041>]{minitoc(hints)}%
   {You have attempted to insert
    \MessageBreak
    empty minilofs}
   \@mtc@hints@given@true
\fi
```

`\if@mtc@empty@minilot@` For minilots:

```
\if@mtc@empty@minilot@
   \mtcPackageWarningNoLine[<W0042>]{minitoc(hints)}%
   {You have attempted to insert
    \MessageBreak
    empty minilots}
   \@mtc@hints@given@true
\fi
```

`\if@mtc@empty@secttoc@` For secttocs:

```
\if@mtc@empty@secttoc@
   \mtcPackageWarningNoLine[<W0049>]{minitoc(hints)}%
   {You have attempted to insert
    \MessageBreak
```

```
    empty secttocs}
   \@mtc@hints@given@true
\fi
```

\if@mtc@empty@sectlof@ For sectlofs:

```
\if@mtc@empty@sectlof@
   \mtcPackageWarningNoLine[<W0047>]{minitoc(hints)}%
   {You have attempted to insert
    \MessageBreak
    empty sectlofs}
   \@mtc@hints@given@true
\fi
```

\if@mtc@empty@sectlot@ For sectlots:

```
\if@mtc@empty@sectlot@
   \mtcPackageWarningNoLine[<W0042>]{minitoc(hints)}%
   {You have attempted to insert
    \MessageBreak
    empty sectlots}
   \@mtc@hints@given@true
\fi
```

9.78.2.20 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 *document*.log file.

\if@firstpartis@used@ Obsolete macro \firstpartis:

```
\if@firstpartis@used@
   \mtcPackageWarningNoLine[<W0051>]{minitoc(hints)}%
   {You have invoked an obsolete
    \MessageBreak
    command: \string\firstpartis}
   \@mtc@hints@given@true
\fi
```

\if@firstchapteris@used@ Obsolete macro \firstchapteris:

```
\if@firstchapteris@used@
   \mtcPackageWarningNoLine[<W0050>]{minitoc(hints)}%
   {You have invoked an obsolete
```

```
    \MessageBreak
    command: \string\firstchapteris}
   \@mtc@hints@given@true
\fi
```

\if@firstsectionis@used@ Obsolete macro \firstsectionis:

```
\if@firstsectionis@used@
   \mtcPackageWarningNoLine[<W0052>]{minitoc(hints)}%
   {You have invoked an obsolete
    \MessageBreak
    command: \string\firstsectionis}
   \@mtc@hints@given@true
\fi
```

9.78.2.21 Check if some hints have been written

\if@mtc@hints@given@ \mtc@hints@enddoc 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 *document*.log file if no problems have been detected. And we terminate the \mtc@hints@enddoc macro by a closing brace.

```
\if@mtc@hints@given@
   \mtcPackageWarningNoLine[<W0024>]{minitoc(hints)}%
      {Some hints have been written
       \MessageBreak
       in the \jobname.log file}
\else
   \mtcPackageInfo[<I0019>]{minitoc(hints)}%
      {No hints have been written
       \MessageBreak
       in the \jobname.log file.\@gobble}
\fi
}
```

9.79 Processing of options

\InputIfFileExists \ExecuteOptions First, if possible, we apply the default language option, english:

```
    \InputIfFileExists{english.mld}%
      {\ExecuteOptions{english}}%
```

\mtcPackageError
\providecommand
\ptctitle
\plftitle
\plttitle
\mtctitle
\mlftitle
\mlttitle
\stctitle
\slftitle
\slttitle

Else, we signal a severe error and provide the missing default titles:

```
      {\mtcPackageError[<E0036>]{minitoc}%
         {Your minitoc installation is incomplete.
          \MessageBreak
          The minitoc language object file (.mld),
          \MessageBreak
          english.mld is not found.
          \MessageBreak
          We will try to continue with default values}%
         {See the minitoc documentation.
          \MessageBreak
          Please fix your minitoc installation.
          \MessageBreak
          Press <return> to continue}%
       \providecommand{\ptctitle}{Table of Contents}%
       \providecommand{\plftitle}{List of Figures}%
       \providecommand{\plttitle}{List of Tables}%
       \providecommand{\mtctitle}{Contents}%
       \providecommand{\mlftitle}{Figures}%
       \providecommand{\mlttitle}{Tables}%
       \providecommand{\stctitle}{Contents}%
       \providecommand{\slftitle}{Figures}%
       \providecommand{\slttitle}{Tables}%
     }%
```

\ProcessOptions*

Then, we execute all requested options: for most options, it is just setting a flag, or loading a file for the language options.

```
\ProcessOptions*
```

We now examine the flags for some options and execute the necessary actions.

9.79.1 Processing the insection option

\if@mtc@ss@insection@
\RequirePackage
\@ifpackageloaded
\if@mtc@placeinsLoaded@

For the insection option, we load the placeins package [10] with its options verbose and section, after the flafter package (described in [162] and [189, page 286]); the correct loading is verified:

```
\if@mtc@ss@insection@
   \RequirePackage{flafter}[2000/07/23]%
   \RequirePackage[section,verbose]{placeins}[2005/04/18]%
   \@ifpackageloaded{placeins}%
      {\@mtc@placeinsLoaded@true}{\@mtc@placeinsLoaded@false}%
\fi
```

9.79.2 Processing the notoccite option

\if@mtc@notoccite@
\RequirePackage

For the notoccite option, we just load the notoccite package [9]:

```
\if@mtc@notoccite@
   \RequirePackage{notoccite}%
\fi
```

9.79.3 Processing the listfiles option

\mtc@maf
\tf@mtc
\if@longextensions@
\mtc@maf@long
\mtc@maf@short
\mtc@addtomaf
\IfFileExists
\jobname

We define the \mtc@maf macro which closes \tf@mtc and reopens it to write into the file *document*.maf. It calls \mtc@maf@long or \mtc@maf@short (long or short extensions), then closes \tf@mtc. \mtc@maf@long or \mtc@maf@short writes the names of the existing auxiliary files using decrementing loops on the associated counters, and includes *document*.mtc in the list (but *not* the *document*.maf [12] file). We must also check the existence of \jobname.mtc1 if long extensions are used.

```
\def\mtc@maf{%
       \mtcPackageInfo[<I0009>]{minitoc}%
                      {Listing minitoc auxiliary files.
                       \MessageBreak
                       Creating the \jobname.maf file\@gobble}
       \immediate\closeout\tf@mtc
       \immediate\openout\tf@mtc \jobname.maf
       \if@longextensions@\mtc@maf@long\else\mtc@maf@short\fi
       \immediate\closeout\tf@mtc}
\def\mtc@addtomaf#1{%
  \IfFileExists{#1}{\immediate\write\tf@mtc{#1}}{}}
\def\mtc@maf@long{%
       \mtc@addtomaf{\jobname.mtc}
\@ifundefined{c@ptc}{}{\loop\ifnum\c@ptc>\z@\relax
       \mtc@addtomaf{\jobname.ptc\arabic{ptc}}
       \mtc@addtomaf{\jobname.plf\arabic{ptc}}
       \mtc@addtomaf{\jobname.plt\arabic{ptc}}
       \advance\c@ptc\m@ne\repeat}
\@ifundefined{c@mtc}{\mtc@addtomaf{\jobname.mtc1}}{%
       \loop\ifnum\c@mtc>\z@\relax
       \mtc@addtomaf{\jobname.mtc\arabic{mtc}}
       \mtc@addtomaf{\jobname.mlf\arabic{mtc}}
       \mtc@addtomaf{\jobname.mlt\arabic{mtc}}
       \advance\c@mtc\m@ne\repeat}
\@ifundefined{c@stc}{}{\loop\ifnum\c@stc>\z@\relax
       \mtc@addtomaf{\jobname.stc\arabic{stc}}
       \mtc@addtomaf{\jobname.slf\arabic{stc}}
       \mtc@addtomaf{\jobname.slt\arabic{stc}}
       \advance\c@stc\m@ne\repeat}}
\def\mtc@maf@short{%
```

[12] Some users could made a cleanup using this file as a list of files to delete, so it should not be in the list.

```
        \mtc@addtomaf{\jobname.mtc}
\@ifundefined{c@ptc}{}{\loop\ifnum\c@ptc>\z@\relax
        \mtc@addtomaf{\jobname.P\arabic{ptc}}
        \mtc@addtomaf{\jobname.G\arabic{ptc}}
        \mtc@addtomaf{\jobname.U\arabic{ptc}}
        \advance\c@ptc\m@ne\repeat}
\@ifundefined{c@mtc}{}{\loop\ifnum\c@mtc>\z@\relax
        \mtc@addtomaf{\jobname.M\arabic{mtc}}
        \mtc@addtomaf{\jobname.F\arabic{mtc}}
        \mtc@addtomaf{\jobname.T\arabic{mtc}}
        \advance\c@mtc\m@ne\repeat}
\@ifundefined{c@stc}{}{\loop\ifnum\c@stc>\z@\relax
        \mtc@addtomaf{\jobname.S\arabic{stc}}
        \mtc@addtomaf{\jobname.H\arabic{stc}}
        \mtc@addtomaf{\jobname.V\arabic{stc}}
        \advance\c@stc\m@ne\repeat}}
```

\if@mtc@listfiles@ \AtEndDocument \mtc@maf

If this option is active, we call \mtc@maf in a \AtEndDocument block.

```
\if@mtc@listfiles@{\AtEndDocument{\mtc@maf}}\else\fi
```

9.79.4 Processing the hints option

\if@mtc@hints@ \AtBeginDocument \mtc@hints@begindoc \AtEndDocument \mtc@hints@enddoc

For the hints option, we set its first part in a \AtBeginDocument block and its third (last) part in a \AtEndDocument block:

```
\if@mtc@hints@
   \AtBeginDocument{\mtc@hints@begindoc}%
   \AtEndDocument{\mtc@hints@enddoc}%
\fi
```

9.79.5 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 \mtc@hints@part \part \mtc@hints@@part \@part \mtc@hints@@spart \@spart

For the \part command:

```
\@ifundefined{part}{}{\let\mtc@hints@part\part
                        \let\mtc@hints@@part\@part
                        \let\mtc@hints@@spart\@spart}
```

\@ifundefined
\mtc@hints@chapter
\chapter
\mtc@hints@@chapter
\@chapter
\mtc@hints@@schapter
\@schapter

For the \chapter command:

```
\@ifundefined{chapter}{}{\let\mtc@hints@chapter\chapter
                         \let\mtc@hints@@chapter\@chapter
                         \let\mtc@hints@@schapter\@schapter}
```

\@ifundefined
\mtc@hints@section
\section
\mtc@hints@@sect
\@sect
\mtc@hints@@ssect
\@ssect

For the \section command:

```
\@ifundefined{section}{}{\let\mtc@hints@section\section
                         \let\mtc@hints@@sect\@sect
                         \let\mtc@hints@@ssect\@ssect}
```

9.80 Trapping the undefined preparation and insertion commands

\mtc@classck
\mtcPackageError

It may happen that you use a preparation command (like \dominitoc) or an insertion command (like \dominitoc) in a document using a class where that command is not available (like article). To get a better diagnostic for such errors, we intercept such commands by providing a default definition which just emits an error message. These default definitions are made in a \AtBeginDocument block.

```
\def\mtc@classck#1{%
  \mtcPackageError[<E0037>]{minitoc}%
     {The \csname #1\endcsname\space command is incompatible
      \MessageBreak
      with the document class}%
     {Correct the source code.
      \MessageBreak
      Type <return> and rerun LaTeX}%
}
```

A \AtBeginDocument bloc:

\AtBeginDocument

```
\AtBeginDocument{%
```

\providecommand
\doparttoc
\dopartlof
\dopartlot

Part-level preparation commands:

```
 \providecommand{\doparttoc}[1][l]{\mtc@classck{doparttoc}}%
 \providecommand{\dopartlof}[1][l]{\mtc@classck{dopartlof}}%
 \providecommand{\dopartlot}[1][l]{\mtc@classck{dopartlot}}%
```

\dominitoc
\dominilof
\dominilot

Chapter-level preparation commands:

```
\providecommand{\dominitoc}[1][l]{\mtc@classck{dominitoc}}%
\providecommand{\dominilof}[1][l]{\mtc@classck{dominilof}}%
\providecommand{\dominilot}[1][l]{\mtc@classck{dominilot}}%
```

\dosecttoc
\dosectlof
\dosectlot

Section-level preparation commands:

```
\providecommand{\dosecttoc}[1][l]{\mtc@classck{dosecttoc}}%
\providecommand{\dosectlof}[1][l]{\mtc@classck{dosectlof}}%
\providecommand{\dosectlot}[1][l]{\mtc@classck{dosectlot}}%
```

\parttoc
\partlof
\partlot

```
% \ifcase\LANG\relax
% Part-level insertion commands:
% \or\relax
% Commandes d'insertion au niveau partie:
% \fi
\providecommand{\parttoc}[1][l]{\mtc@classck{parttoc}}%
\providecommand{\partlof}[1][l]{\mtc@classck{partlof}}%
\providecommand{\partlot}[1][l]{\mtc@classck{partlot}}%
```

\minitoc
\minilof
\minilot

Chapter-level insertion commands:

```
\providecommand{\minitoc}[1][l]{\mtc@classck{minitoc}}%
\providecommand{\minilof}[1][l]{\mtc@classck{minilof}}%
\providecommand{\minilot}[1][l]{\mtc@classck{minilot}}%
```

\secttoc
\sectlof
\sectlot

Section-level insertion commands:

```
\providecommand{\secttoc}[1][l]{\mtc@classck{secttoc}}%
\providecommand{\sectlof}[1][l]{\mtc@classck{sectlof}}%
\providecommand{\sectlot}[1][l]{\mtc@classck{sectlot}}%
}
```

And the package is terminated.

⟨/minitoc⟩

9.81 The fminitoc.dtx file

\jobname
\input

This short file is necessary to create the french documentation. Its rôle is to set \jobname to fminitoc in place of minitoc. As minitoc.ins generates the minitoc.lan and fminitoc.lan files which set a language number \LANG, and minitoc.dtx reads then the \jobname.lan file, the documentation can be in several languages (english and french here) in minitoc.dtx, the language being selected by \ifcase\LANG\relax ... \or\relax ... \fi constructs. The \relax primitives are necessary to avoid bad surprises.

```
⟨*fminitoc⟩
\ProvidesFile{fminitoc.dtx}%
             [2007/01/09 minitoc french documentation start file]
\input{minitoc.dtx}
⟨/fminitoc⟩
```

Chapter 10

Commented code of mtcoff

Contents

10.1 Why mtcoff?

The minitoc package [106, 107] 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 desactivated 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.

10.2 Identification of the package

\NeedsTeXFormat
\ProvidesPackage

First, we identify the package and check the version of LaTeX [1]; we need the mtcmess package to write messages with unique identifiers.

```
⟨*mtcoff⟩
\NeedsTeXFormat{LaTeX2e}%
\ProvidesPackage{mtcoff}[2006/06/27 v30 The mtcoff package]
\RequirePackage{mtcmess}[2006/03/14]
```

10.3 Faking counters and dimensions

\count@
\dimen@
\c@minitocdepth
\mtcindent
\mtcskipamount
\bigskipamount
\c@parttocdepth
\mtcindent
\c@secttocdepth
\stcindent

As minitoc declares some counters and dimensions registers, we fake them using \count@ or \dimen@. For \mtcskipamount, we must use its default definition, \bigskipamount.

```
\let\c@minitocdepth\count@
\let\mtcindent\dimen@
\let\mtcskipamount\bigskipamount
\let\c@parttocdepth\count@
\let\ptcindent\dimen@
```

[1] This checking is not really useful for the mtcoff package itself, but it is good to check that your version of LaTeX is not too old to support minitoc.

```
\let\c@secttocdepth\count@
\let\stcindent\dimen@
```

\c@mtc
\c@ptc
\c@stc

The basic mini-table counters are provided by \count@:

```
\let\c@mtc\count@
\let\c@ptc\count@
\let\c@stc\count@
```

\mtcgapbeforeheads
\mtcgapafterheads

The gaps before and after parttoc heads receive their default values:

```
\def\mtcgapbeforeheads{50\p@}
\def\mtcgapafterheads{40\p@}
```

\@ifundefined
\kernafterparttoc
\kernafterpartlof
\kernafterpartlot
\kernaftersecttoc
\kernaftersectlof
\kernaftersectlot
\kernafterminitoc
\kernafterminilof
\kernafterminilot

We must define the macros for the vertical kernings between the minitables and their before the bottom rule. The default values are used. We must issue a warning if one of these macros is used.

```
   \@ifundefined{part}{}{%
      \def\kernafterparttoc{%
       \mtcoffwarn@true
       \mtcPackageWarning[<F0008>]{mtcoff}%
         {The macro \string\kernafterparttoc
          \MessageBreak
          should not be used out of context
          \MessageBreak}%
        \kern-1.\baselineskip\kern.5ex}%
      \def\kernafterpartlof{%
       \mtcoffwarn@true
       \mtcPackageWarning[<F0008>]{mtcoff}%
         {The macro \string\kernafterpartlof
          \MessageBreak
          should not be used out of context
          \MessageBreak}%
        \kern-1.\baselineskip\kern.5ex}%
      \def\kernafterpartlot{%
       \mtcoffwarn@true
       \mtcPackageWarning[<F0008>]{mtcoff}%
         {The macro \string\kernafterpartlot
          \MessageBreak
          should not be used out of context
          \MessageBreak}%
        \kern-1.\baselineskip\kern.5ex}%
                        }%
   \@ifundefined{chapter}{%
      \@ifundefined{section}{}%
                            {%
          \def\kernaftersecttoc{%
```

```
                \mtcoffwarn@true
                \mtcPackageWarning[<F0008>]{mtcoff}%
                  {The macro \string\kernaftersecttoc
                   \MessageBreak
                   should not be used out of context
                   \MessageBreak}%
            \kern-1.\baselineskip\kern.5ex}%
        \def\kernaftersectlof{%
                \mtcoffwarn@true
                \mtcPackageWarning[<F0008>]{mtcoff}%
                  {The macro \string\kernaftersectlof
                   \MessageBreak
                   should not be used out of context
                   \MessageBreak}%
            \kern-1.\baselineskip\kern.5ex}%
        \def\kernaftersectlot{%
                \mtcoffwarn@true
                \mtcPackageWarning[<F0008>]{mtcoff}%
                  {The macro \string\kernaftersectlot
                   \MessageBreak
                   should not be used out of context
                   \MessageBreak}%
            \kern-1.\baselineskip\kern.5ex}%
                        }%
                    }%
                    {%
        \def\kernafterminitoc{%
                \mtcoffwarn@true
                \mtcPackageWarning[<F0008>]{mtcoff}%
                  {The macro \string\kernafterminitoc
                   \MessageBreak
                   should not be used out of context
                   \MessageBreak}%
             \kern-.5\baselineskip\kern.5ex}%
        \def\kernafterminilof{%
                \mtcoffwarn@true
                \mtcPackageWarning[<F0008>]{mtcoff}%
                  {The macro \string\kernafterminilof
                   \MessageBreak
                   should not be used out of context
                   \MessageBreak}%
             \kern-1.\baselineskip\kern0.ex}%
        \def\kernafterminilot{%
                \mtcoffwarn@true
                \mtcPackageWarning[<F0008>]{mtcoff}%
                  {The macro \string\kernafterminilot
                   \MessageBreak
                   should not be used out of context
                   \MessageBreak}%
             \kern-1.\baselineskip\kern0.ex}%
                    }%
```

10.4 Faking simple commands

\mtcskip
\faketableofcontents
\fakelistoffigures
\fakelistoftables
\adjustptc
\adjustmtc
\adjuststc
\decrementptc
\decrementmtc
\decrementstc
\incrementptc
\incrementmtc
\incrementstc

Some user commands are easy to fake:

```
\let\mtcskip\relax
\let\faketableofcontents\relax
\let\fakelistoffigures\relax
\let\fakelistoftables\relax
\newcommand{\adjustptc}[1][1]{\relax}
\newcommand{\adjustmtc}[1][1]{\relax}
\newcommand{\adjuststc}[1][1]{\relax}
\let\decrementptc\relax
\let\decrementmtc\relax
\let\decrementstc\relax
\let\incrementptc\relax
\let\incrementmtc\relax
\let\incrementstc\relax
```

\partend
\partbegin
\chapterend
\chapterbegin
\sectend
\sectbegin

The following commands are not directly called by the user, in normal circumstances, but must be faked:

```
\let\partend\relax
\let\partbegin\relax
\let\chapterend\relax
\let\chapterbegin\relax
\let\sectend\relax
\let\sectbegin\relax
```

10.5 Faking 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@.

```
\def\gobbleopt@[#1]{\relax}
```

\@ifnextchar
\doparttoc
\dopartlof
\dopartlot
\parttoc
\partlof
\partlot

Commands for part level mini-tables:

```
\def\doparttoc{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[l]}}
\def\dopartlof{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[l]}}
\def\dopartlot{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[l]}}
\def\parttoc{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[l]}}
\def\partlof{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[l]}}
\def\partlot{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[l]}}
```

\@ifnextchar
\dominitoc
\dominilof
\dominilot
\minitoc
\minilof
\minilot

Commands for chapter level mini-tables:

```
\def\dominitoc{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[l]}}
\def\dominilof{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[l]}}
\def\dominilot{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[l]}}
\def\minitoc{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[l]}}
\def\minilof{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[l]}}
\def\minilot{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[l]}}
```

\@ifnextchar
\dosecttoc
\dosectlof
\dosectlot
\secttoc
\sectlof
\sectlot

Commands for section level mini-tables:

```
\def\dosecttoc{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[l]}}
\def\dosectlof{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[l]}}
\def\dosectlot{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[l]}}
\def\secttoc{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[l]}}
\def\sectlof{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[l]}}
\def\sectlot{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[l]}}
```

\@ifnextchar
\mtcprepare

Command \mtcprepare:

```
\def\mtcprepare{\@ifnextchar[{\gobbleopt@}{\gobbleopt@[l]}}
```

10.6 Disabling the internal commands

\@gobbletwo
\tf@mtc
\mtc@string
\appendixmtc
\l@xchapter
\xchapter
\pchapter
\psect
\l@xpart
\xpart
\l@xsect
\xsect

We need also to disable some minitoc commands, with \relax (macros with no argument) or \@gobbletwo (macros with two arguments):

```
\let\tf@mtc\count@
\let\mtc@string\relax
\let\appendixmtc\relax
\let\l@xchapter\@gobbletwo
\let\xchapter\relax
\let\pchapter\relax
\let\psect\relax
\let\l@xpart\@gobbletwo
\let\xpart\relax
\let\l@xsect\@gobbletwo
\let\xsect\relax
```

10.7 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
\ptcCfont
\ptcSfont
\ptcSSfont
\ptcSSSfont
\ptcPfont
\ptcSPfont
\plffont
\plfSfont
\pltfont
\pltSfont
\ptifont

Fonts for part level mini-tables:

```
\let\ptcfont\empty
\let\ptcCfont\empty
\let\ptcSfont\empty
\let\ptcSSfont\empty
\let\ptcSSSfont\empty
\let\ptcPfont\empty
\let\ptcSPfont\empty
\let\plffont\empty
\let\plfSfont\empty
\let\pltfont\empty
\let\pltSfont\empty
\let\ptifont\empty
```

\mtcfont
\mtcSfont
\mtcSSfont
\mtcSSSfont
\mtcPfont
\mtcSPfont
\mlffont
\mlfSfont
\mltfont
\mltSfont
\mtifont

Fonts for chapter level mini-tables:

```
\let\mtcfont\empty
\let\mtcSfont\empty
\let\mtcSSfont\empty
\let\mtcSSSfont\empty
\let\mtcPfont\empty
\let\mtcSPfont\empty
\let\mlffont\empty
\let\mlfSfont\empty
\let\mltfont\empty
\let\mltSfont\empty
\let\mtifont\empty
```

\stcfont
\stcSSfont
\stcSSSfont
\stcPfont
\stcSPfont
\slffont
\slfSfont
\sltfont
\sltSfont
\stifont

Fonts for section level mini-tables:

```
\let\stcfont\empty
\let\stcSSfont\empty
\let\stcSSSfont\empty
\let\stcPfont\empty
\let\stcSPfont\empty
\let\slffont\empty
\let\slfSfont\empty
\let\sltfont\empty
\let\sltSfont\empty
\let\stifont\empty
```

\coffeefont Font for "coffee" ☕ lines:

```
\let\coffeefont\empty
```

10.8 Disabling the \mtcset... commands

\@gobbletwo
\mtcsetdepth
\mtcsetfont
\mtcsettitlefont
\mtcsettitle
\mtcsetformat
\mtcsetfeature
\mtcsetpagenumbers
\mtcsetrules

These commands use two or three mandatory arguments:

```
\let\mtcsetdepth\@gobbletwo
\def\mtcsetfont#1#2#3{\empty}
\let\mtcsettitlefont\@gobbletwo
\let\mtcsettitle\@gobbletwo
\def\mtcsetformat#1#2#3{\empty}
\def\mtcsetfeature#1#2#3{\empty}
\let\mtcsetpagenumbers\@gobbletwo
\let\mtcsetrules\@gobbletwo
```

10.9 Disabling the new \l@... commands

\l@starpart
\l@starchapter
\l@starsection
\l@starsubsection
\l@starsubsubsection
\l@starparagraph
\l@starsubparagraph

The minitoc package defines the \l@starXXX commands to format TOC entries for starred sectionning commands. We reset to the unstarred version, when necessary:

```
\@ifundefined{part}{}{\let\l@starpart\l@part}
\@ifundefined{chapter}{}{\let\l@starchapter\l@chapter}
\@ifundefined{section}{}{\let\l@starsection\l@section}
\@ifundefined{subsection}{}{\let\l@starsubsection\l@subsection}
\@ifundefined{subsubsection}{}{\let\l@starsubsubsection\l@subsubsection}
\@ifundefined{paragraph}{}{\let\l@starparagraph\l@paragraph}
\@ifundefined{subparagraph}{}{\let\l@starsubparagraph\l@subparagraph}
```

10.10 Ignore the obsolete commands

\@gobble
\firstpartis
\firstchapteris
\firstsectionis

We just ignore the obsolete commands (with one mandatory argument):

```
\let\firstpartis\@gobble
\let\firstchapteris\@gobble
\let\firstsectionis\@gobble
```

10.11 Disabling the \mtcselectlanguage command

\@gobble
\mtcselectlanguage

This command has one mandatory argument:

```
\let\mtcselectlanguage\@gobble
```

10.12 Disabling the \mtcloadmlo command

\@gobble
\mtcloadmlo

This command has one mandatory argument:

```
\let\mtcloadmlo\@gobble
```

10.13 Disabling the commands for the horizontal rules

\ptcrule
\noptcrule
\mtcrule
\nomtcrule
\stcrule
\nostcrule
\plfrule
\noplfrule
\mlfrule
\nomlfrule
\slfrule
\noslfrule
\pltrule
\nopltrule
\mltrule
\nomltrule
\sltrule
\nosltrule

These commands have no argument:

```
\let\ptcrule\relax
\let\noptcrule\relax
\let\mtcrule\relax
\let\nomtcrule\relax
\let\stcrule\relax
\let\nostcrule\relax
\let\plfrule\relax
\let\noplfrule\relax
\let\mlfrule\relax
\let\nomlfrule\relax
\let\slfrule\relax
\let\noslfrule\relax
\let\pltrule\relax
\let\nopltrule\relax
\let\mltrule\relax
\let\nomltrule\relax
\let\sltrule\relax
\let\nosltrule\relax
```

10.14 Disabling the commands for the page numbers

\ptcpagenumbers
\noptcpagenumbers
\mtcpagenumbers
\nomtcpagenumbers
\stcpagenumbers
\nostcpagenumbers
\plfpagenumbers
\noplfpagenumbers
\mlfpagenumbers
\nomlfpagenumbers
\slfpagenumbers
\noslfpagenumbers
\pltpagenumbers
\nopltpagenumbers
\mltpagenumbers
\nomltpagenumbers
\sltpagenumbers
\nosltpagenumbers

These commands have no argument:

```
\let\mtcpagenumbers\relax
\let\nomtcpagenumbers\relax
\let\stcpagenumbers\relax
\let\nostcpagenumbers\relax
\let\ptcpagenumbers\relax
\let\noptcpagenumbers\relax
\let\mlfpagenumbers\relax
\let\nomlfpagenumbers\relax
\let\slfpagenumbers\relax
\let\noslfpagenumbers\relax
\let\plfpagenumbers\relax
\let\noplfpagenumbers\relax
\let\mltpagenumbers\relax
\let\nomltpagenumbers\relax
\let\sltpagenumbers\relax
\let\nosltpagenumbers\relax
\let\pltpagenumbers\relax
\let\nopltpagenumbers\relax
```

10.15 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
\beforepartlof
\beforepartlot
\afterparttoc
\afterpartlof
\afterpartlot
\thispageparttocstyle
\thispagepartlofstyle
\thispagepartlotstyle

Commands for part level mini-tables:

```
\let\beforeparttoc\empty
\let\beforepartlof\empty
\let\beforepartlot\empty
\let\afterparttoc\empty
\let\afterpartlof\empty
\let\afterpartlot\empty
\let\thispageparttocstyle\empty
\let\thispagepartlofstyle\empty
\let\thispagepartlotstyle\empty
```

\beforeminitoc
\beforeminilof
\beforeminilot
\afterminitoc
\afterminilof
\afterminilot
\thispageminitocstyle
\thispageminilofstyle
\thispageminilotstyle

Commands for chapter level mini-tables:

```
\let\beforeminitoc\empty
\let\beforeminilof\empty
\let\beforeminilot\empty
\let\afterminitoc\empty
\let\afterminilof\empty
\let\afterminilot\empty
\let\thispageminitocstyle\empty
\let\thispageminilofstyle\empty
\let\thispageminilotstyle\empty
```

\beforesecttoc
\beforesectlof
\beforesectlot
\aftersecttoc
\aftersectlof
\aftersectlot
\thispagesecttocstyle
\thispagesectlofstyle
\thispagesectlotstyle

Commands for section level mini-tables:

```
\let\beforesecttoc\empty
\let\beforesectlof\empty
\let\beforesectlot\empty
\let\aftersecttoc\empty
\let\aftersectlof\empty
\let\aftersectlot\empty
\let\thispagesecttocstyle\empty
\let\thispagesectlofstyle\empty
\let\thispagesectlotstyle\empty
```

10.16 Disabling miscellaneous flags and commands

\if@longextensions@
\iftightmtc
\ifktightmtc
\ifundottedmtc
\l@listof
\ifmtcsecondpart
\chapter

There are some flags and commands that it is wise to declare:

```
\newif\if@longextensions@ \@longextensions@true
\newif\iftightmtc \tightmtcfalse
\newif\ifktightmtc \ktightmtcfalse
\newif\ifundottedmtc \undottedmtcfalse
\newif\ifmtcsecondpart \mtcsecondpartfalse
\let\l@listof\chapter
```

10.17 Caution for some commands

\AtBeginDocument
\ifmtcoffwarn@

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:

```
\newif\ifmtcoffwarn@ \mtcoffwarn@false
\AtEndDocument{\ifmtcoffwarn@
```

```
   \mtcPackageWarningNoLine[<F0007>]{mtcoff}%
      {You should scan (backwards) your .log
       \MessageBreak
       file to find some commands needing
       \MessageBreak
       to be replaced if you decide to
       \MessageBreak
       DEFINITELY stop using minitoc for this
       \MessageBreak
       document. It is more wise to keep the
       \MessageBreak
       \string\usepackage\space lines for minitoc and mtcoff
       \MessageBreak
       and to comment out only one of them}
\fi}
```

\mtcaddchapter
\mtcaddsection
\mtcaddpart
\ifmtcoffwarn@
\mtc@ck
\addcontentsline

Then these commands are disabled and they set the flag and give a warning (useful to get the line number):

```
\newcommand{\mtcaddchapter}[1][]{\mtcoffwarn@true
   \mtcPackageWarning[<F0004>]{mtcoff}%
   {\protect\mtcaddchapter{...} should be replaced
    \MessageBreak
    by \protect\addcontentsline{toc}{chapter}{...}
    \MessageBreak}
\def\mtc@ck{#1}
\ifx\mtc@ck\empty
\else
   \addcontentsline{toc}{chapter}{#1}%
\fi}
\newcommand{\mtcaddsection}[1][]{\mtcoffwarn@true
   \mtcPackageWarning[<F0006>]{mtcoff}%
      {\protect\mtcaddsection{...} should be replaced
       \MessageBreak
       by \protect\addcontentsline{toc}{section}{...}
       \MessageBreak}
\def\mtc@ck{#1}
\ifx\mtc@ck\empty
\else
     \addcontentsline{toc}{part}{#1}%
\fi}
\newcommand{\mtcaddpart}[1][]{\mtcoffwarn@true
   \mtcPackageWarning[<F0005>]{mtcoff}%
      {\protect\mtcaddpart{...} should be replaced
       \MessageBreak
       by \protect\addcontentsline{toc}{part}{...}
       \MessageBreak}
\def\mtc@ck{#1}
\ifx\mtc@ck\empty
\else
   \addcontentsline{toc}{part}{#1}%
\fi}
```

10.18 Disabling commands for "coffee"

\addcoffeeline
\coffeeline
\@gobble
\@Undottedtocline
\@Undottedtoclinep

We disable the commands relative to "coffee" lines, and the specific version of contents lines without leaders of dots:

```
\def\addcoffeeline#1#2#3{\relax}
\let\coffeeline\@gobble
\let\l@coffee\relax
\def\@Undottedtocline#1#2#3#4#5{\relax}
\def\@Undottedtoclinep#1#2#3#4#5{\relax}
```

10.19 Disabling the mtchideinmain... environments

mtchideinmaintoc
mtchideinmainlof
mtchideinmainlot

These environments accept one optional argument:

```
\newenvironment{mtchideinmaintoc}[1][-1]%
    {\empty}{\empty}
\newenvironment{mtchideinmainlof}[1][-1]%
    {\empty}{\empty}
\newenvironment{mtchideinmainlot}[1][-1]%
    {\empty}{\empty}
```

10.20 Inhibition of the \mtc@saveXXXdepth and \mtc@restoreXXXdepth internal commands

\mtc@savetocdepth
\mtc@savelofdepth
\mtc@savelotdepth
\mtc@restoretocdepth
\mtc@restorelofdepth
\mtc@restorelotdepth

We must inhibit these commands, inserted in the .toc, .lof and .lot files by the hiding commands. So we will not have to delete these files when switching from the minitoc package to the mtcoff package.

```
\let\mtc@savetocdepth\empty
\let\mtc@savelofdepth\empty
\let\mtc@savelotdepth\empty
\let\mtc@restoretocdepth\empty
\let\mtc@restorelofdepth\empty
\let\mtc@restorelotdepth\empty
```

10.21 Disabling the \mtcfixglossary command

\mtcfixglossary

This command accepts one optional argument:

```
\newcommand{\mtcfixglossary}[1][]{\relax}
```

10.22 Disabling the \mtcfixindex command

\mtcfixindex

This command accepts one optional argument:

```
\newcommand{\mtcfixindex}[1][]{\relax}
```

10.23 Disabling the \addstarred... commands

\ifmtcoffwarn@
\addstarredpart
\addstarredchapter
\addstarredsection
\addcontentsline

These commands should be replaced by standard commands, but mtcoff simulates and gives a warning, which will be reminded at the end of document:

```
\def\addstarredpart#1{\mtcoffwarn@true
   \mtcPackageWarning[<F0002>]{mtcoff}%
   {\protect\addstarredpart{...} should be replaced by
    \MessageBreak
    \protect\addcontentsline{toc}{part}{...}
    \MessageBreak}
   \addcontentsline{toc}{part}{#1}}
\def\addstarredchapter#1{\mtcoffwarn@true
   \mtcPackageWarning[<F0001>]{mtcoff}%
   {\protect\addstarredchapter{...} should be replaced by
    \MessageBreak
    \protect\addcontentsline{toc}{chapter}{...}
    \MessageBreak}
   \addcontentsline{toc}{chapter}{#1}}
\def\addstarredsection#1{\mtcoffwarn@true
   \mtcPackageWarning[<F0003>]{mtcoff}%
   {\protect\addstarredsection{...} should be replaced by
    \MessageBreak
    \protect\addcontentsline{toc}{section}{...}
    \MessageBreak}
   \addcontentsline{toc}{section}{#1}}
```

And the mtcoff package is terminated.

⟨/mtcoff⟩

Chapter 11

Commented code of mtcmess

\mtcPackageInfo
\PackageInfo
\MessageBreak
\mtcPackageWarning
\PackageWarning
\mtcPackageWarningNoLine
\PackageWarningNoLine
\mtcPackageError
\PackageError

To make easier the search of a message in the documentation[1], we will assign an unique identifier to each message of the minitoc and mtcoff packages. As the standard commands for such messages do not include this feature, we make extended versions, with the same syntax, plus a first optional argument:

```
⟨*mtcmess⟩
\ProvidesPackage{mtcmess}[2006/03/14]%
\NeedsTeXFormat{LaTeX2e}[1996/06/01]%
\newcommand{\mtcPackageInfo}[3][]%
    {\PackageInfo{#2}{#1\MessageBreak #3}}%
\newcommand{\mtcPackageWarning}[3][]%
    {\PackageWarning{#2}{#1\MessageBreak #3}}%
\newcommand{\mtcPackageWarningNoLine}[3][]%
    {\PackageWarningNoLine{#2}{#1\MessageBreak #3}}%
\newcommand{\mtcPackageError}[4][]%
    {\PackageError{#2}{#1\MessageBreak #3}{#4}}%
⟨/mtcmess⟩
```

Hence the first line of the message will contain the package name and the unique identifier of the message.

These macros are defined in a separate package because they are used by at least two packages (minitoc and mtcoff) and because they could be useful for other packages.

[1] By example, using the search facility of some PDF reader utility.

Chapter 12

Patch for the memoir class

This code must be loaded to fix an incompatibility of the minitoc package with some recent versions of the memoir class. This correction is no more necessary after the 2005/09/25 version of memoir.

```
⟨*mtcpatchmem⟩
\NeedsTeXFormat{LaTeX2e}[1996/06/01]%
\ProvidesPackage{mtcpatchmem}%
   [2007/01/09 v51 Package mtcpatchmem]
\RequirePackage{mtcmess}[2006/03/14]
\mtcPackageInfo[<M0001>]{mtcpatchmem}%
   {mtcpatchmem package to patch the memoir class\@gobble}
\renewcommand{\@m@mchapter}[1][]{%
  \def\ch@pt@c{#1}% capture first optional arg
   \@ifnextchar[{\@chapter}{\@chapter[]}%
}
\def\@chapter[#1]#2{%
% if |\ch@pt@c| is empty, no [ was found at all. Use |#2| as
% entry for all fields.
   \ifx\ch@pt@c\@empty
     \def\f@rtoc{#2}%
     \def\f@rhdr{#2}%
   \else
% otherwise at least one [ was found. If |#1| is empty then only
% one was found.
     \let\f@rtoc\ch@pt@c
     \ifx\@empty#1\@empty
       \let\f@rhdr\ch@pt@c
     \else
       \def\f@rhdr{#1}%
     \fi
   \fi
   \ifnum \c@secnumdepth >\m@ne\relax
     \if@mainmatter
       \refstepcounter{chapter}%
```

```
    \fi
  \fi
  \chaptermark{\f@rhdr}%
  \ifartopt
    \@makechapterhead{#2}%
    \@afterheading
  \else
    \insertchapterspace
    \if@twocolumn
      \@topnewpage[\@makechapterhead{#2}]%
    \else
      \@makechapterhead{#2}%
    \fi
    \@afterheading
  \fi
  \ifnum \c@secnumdepth >\m@ne\relax
    \if@mainmatter
      \ifanappendix
        \addcontentsline{toc}{appendix}{%
          \protect\chapternumberline{\thechapter}\f@rtoc}%
      \else
        \addcontentsline{toc}{chapter}{%
          \protect\chapternumberline{\thechapter}\f@rtoc}%
      \fi
    \else
      \addcontentsline{toc}{chapter}{\f@rtoc}%
    \fi
  \else
    \addcontentsline{toc}{chapter}{\f@rtoc}%
  \fi
  \ifheadnameref\M@gettitle{\f@rhdr}\else\M@gettitle{\f@rtoc}\fi
}
⟨/mtcpatchmem⟩
```

Chapter 13

The language definition (.mld) and object (.mlo) files

Contents

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 [1] in the \usepackage command for the minitoc package, either via the command:

\mtcselectlanguage \mtcselectlanguage{⟨*language*⟩}

[1] It can also be a global option for the document.

\ptctitle
\plftitle
\plttitle
\mtctitle
\mlftitle
\mlttitle
\stctitle
\slftitle
\slttitle

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):

- \ptctitle
- \plftitle
- \plttitle
- \mtctitle
- \mlftitle
- \mlttitle
- \stctitle
- \slftitle
- \slttitle

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 [37, 38, 39, 56], the CJK system [167, 168], the HLaTeX system [146, in korean], the Antomega system [150], the ArabTeX [154, 155], BangTeX [202], Devanāgarī for TeX [204], ethiop [29], FarsiTeX [109][2], guarani [32], malayalam [3] et omal [4], MonTeX [97, 100], or ArmTeX [101] 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 [3], 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 [4], 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.

Since version #49, the minitoc package checks the presence of the *language*.mld file (and of the *language*.mlo file if necessary) for each language option of the package, before validating the option. If a .mld or .mlo is missing, the corresponding language option is not enabled and a warning message is written in the *document*.log file. But the presence of the english.mld file is mandatory, because english is the default language.

13.1 "Acadian" language: acadian.mld

The acadian language [5] is just french, so we load the french.mld file (see section 13.58 on page 461):

```
⟨*acadian⟩
\ProvidesFile{acadian.mld}[2004/12/14]
\mtcselectlanguage{french}%
⟨/acadian⟩
```

[2] By Mohammad Ghodsi (ghodsi@rose.ipm.ac.ir) and FarsiTeX Project Group. See the FarsiTeX site at http://www.farsitex.org

[3] Mainly for chinese, farsi (iranian), hangûl (korean), hanja (korean), japanese, malayalam-omega, thai, and russian variants.

[4] The extension .mlo means *minitoc language object*.

[5] Spoken is some parts of the south of the USA, like Louisiane.

13.2 "Acadien" language: acadien.mld

The "acadien" language[6] is just french ("acadien" is the french term for "acadian"), so we load the french.mld file (see section 13.58 on page 461):

```
⟨*acadien⟩
\ProvidesFile{acadien.mld}[2004/12/14]
\mtcselectlanguage{french}%
⟨/acadien⟩
```

13.3 "Afrikaan" language: afrikaan.mld

The titles for the "afrikaan" language[7] come from the dutch.dtx file (by Johannes Braams) in the babel package [38, 39, 40]:

```
⟨*afrikaan⟩
\ProvidesFile{afrikaan.mld}[2006/01/13]
%% Afrikaan(s) titles for minitoc.sty
%% from dutch.dtx (babel)
%% Braams, Johannes
\def\ptctitle{Inhoudsopgawe}%
\def\plftitle{Lys van figure}%
\def\plttitle{Lys van tabelle}%
%%
\def\mtctitle{Inhoudsopgawe}%
\def\mlftitle{Lys van figure}%
\def\mlttitle{Lys van tabelle}%
%%
\def\stctitle{Inhoudsopgawe}%
\def\slftitle{Lys van figure}%
\def\slttitle{Lys van tabelle}%
⟨/afrikaan⟩
```

13.4 "Afrikaans" language: afrikaans.mld

The term "afrikaans" is a synonym of "afrikaan", so we just load afrikaan.mld (see section 13.3):

```
⟨*afrikaans⟩
\ProvidesFile{afrikaans.mld}[2004/12/14]
\mtcselectlanguage{afrikaan}%
⟨/afrikaans⟩
```

[6] Spoken is some parts of the south of the USA, like Louisiane.
[7] Spoken in South Africa, it has dutch origins; compare with section 13.42 on page 453.

13.5 "Albanian" language: albanian.mld

The titles for the "albanian" language are taken from the albanian.dtx file (with a contribution of Adi Zaimi) in the babel package [38, 39, 83]:

```
⟨*albanian⟩
\ProvidesFile{albanian.mld}[2006/01/13]
%% Albanian titles for minitoc.sty.
%% from albabian.dtx (babel)
%% Contribution of Adi Zaimi (zami1st at yahoo.com).
\def\ptctitle{P\"ermbajta}%
\def\plftitle{Figurat}%
\def\plttitle{Tabelat}%
%%
\def\mtctitle{P\"ermbajta}%
\def\mlftitle{Figurat}%
\def\mltitle{Tabelat}%
%%
\def\stctitle{P\"ermbajta}%
\def\slftitle{Figurat}%
\def\slttitle{Tabelat}%
⟨/albanian⟩
```

13.6 "American" language: american.mld

The "american" language is just like "english"[8], so we just load english.mld (see section 13.43 on page 454):

```
⟨*american⟩
\ProvidesFile{american.mld}[2004/12/14]
\mtcselectlanguage{english}%
⟨/american⟩
```

13.7 "Arab" language: arab.mld

The titles for the "arab" language are taken from the ArabTEX package [154, 155] (by Klaus Lagally), which should be used, with the associated fonts.

```
⟨*arab⟩
\ProvidesFile{arab.mld}[1999/03/16]
%% Arabic titles for minitoc.sty
```

[8] It should be true for the mini-table titles; the languages themselves have some differences, like the hyphenation rules.

```
%% Needs arabic fonts (cf. documentation of arabtex)
%% (strings taken from arabtex; to be used with arabtex)
\def\ptctitle{al-mu.htawayAtu}%
\def\plftitle{qA'imaTu a.s-.suwari}%
\def\plttitle{qA'imaTu al-^gadAwili}
%%
\def\mtctitle{al-mu.htawayAtu}%
\def\mlftitle{qA'imaTu a.s-.suwari}%
\def\mlttitle{qA'imaTu al-^gadAwili}%
%%
\def\stctitle{al-mu.htawayAtu}%
\def\slftitle{qA'imaTu a.s-.suwari}%
\def\slttitle{qA'imaTu al-^gadAwili}%
⟨/arab⟩
```

13.8 "Arab2" language: arab2.mld

The titles for the "arab2" language are taken from the ArabTEX package [154, 155] (by Klaus Lagally), which should be used, with the associated fonts. It is a variant of the "arab" language.

```
⟨*arab2⟩
\ProvidesFile{arab2.mld}[2006/03/31]
%% Arabic titles for minitoc.sty. Variant.
%% Needs arabic fonts (cf. documentation of arabtex)
%% (strings taken from arabtex; to be used with arabtex)
{\makeatletter\global\let\mtcArabTok\a@tok}%
\def\ptctitle{\mtcArabTok(al-muHtawayAtu)}%
\def\plftitle{\mtcArabTok(qAQAimaTu aS-Suwari)}%
\def\plttitle{\mtcArabTok(qAQAimaTu al-GadAwili)}%
%%
\def\mtctitle{\mtcArabTok(al-muHtawayAtu)}%
\def\mlftitle{\mtcArabTok(qAQAimaTu aS-Suwari)}%
\def\mlttitle{\mtcArabTok(qAQAimaTu al-GadAwili)}%
%%
\def\stctitle{\mtcArabTok(al-muHtawayAtu)}%
\def\slftitle{\mtcArabTok(qAQAimaTu aS-Suwari)}%
\def\slttitle{\mtcArabTok(qAQAimaTu al-GadAwili)}%
⟨/arab2⟩
```

13.9 "Arabi" language: arabi.mld

The titles for the "arabi" language are taken from the Arabi package [135] (by Youssef Jabri), which should be used, with the associated fonts.

```
⟨*arabi⟩
\ProvidesFile{arabi.mld}[2006/07/27]
%% Arabic titles for minitoc.sty.
%% Needs arabic fonts (cf. documentation of the Arabi package,
%% Youssef Jabri)
%% (strings taken from arabi.ldf; to be used with arabi)
\def\ptctitle{\R{\alef\lam\fa\ha\ra\seen}}%
\def\plftitle{\R{\qaf\alef\yahamza\meem\T\space \alef\lam\alefhamza\sheen\kaf\alef\lam}}%
\def\plttitle{\R{\qaf\alef\yahamza\meem\T\space \alef\lam\jeem\dal\alef\waw\lam}}%
%%
\def\mtctitle{\R{\alef\lam\fa\ha\ra\seen}}%
\def\mlftitle{\R{\qaf\alef\yahamza\meem\T\space \alef\lam\alefhamza\sheen\kaf\alef\lam}}%
\def\mlttitle{\R{\qaf\alef\yahamza\meem\T\space \alef\lam\jeem\dal\alef\waw\lam}}%
%%
\def\stctitle{\R{\alef\lam\fa\ha\ra\seen}}%
\def\slftitle{\R{\qaf\alef\yahamza\meem\T\space \alef\lam\alefhamza\sheen\kaf\alef\lam}}%
\def\slttitle{\R{\qaf\alef\yahamza\meem\T\space \alef\lam\jeem\dal\alef\waw\lam}}%
%%
⟨/arabi⟩
```

13.10 “Arabic” language: arabic.mld

The “arabic” language is a synomym for “arab”, so we just load arab.mld (see section 13.7 on page 436):

```
⟨*arabic⟩
\ProvidesFile{arabic.mld}[2005/02/10]
\mtcselectlanguage{arab}%
⟨/arabic⟩
```

13.11 “Armenian” language: armenian.mld

The titles for the “armenian” language are taken from the ArmTEX package [101] (by Serguei D’Achian, Arnak Dalalyan and Vardan Akopian), which should be used, with the associated fonts.

```
⟨*armenian⟩
\ProvidesFile{armenian.mld}[1999/06/28]
%% Armenian titles for minitoc.sty
%% from ArmTeX
%% Sergei D’Achian (Serguei.Dachian@univ-lemans.fr)
\def\ptctitle{Bovandakuthyun}%
\def\plftitle{Patkerneri cank}%
\def\plttitle{Aghyusakneri cank}%
%%
```

```
\def\mtctitle{Bovandakuthyun}%
\def\mlftitle{Patkerneri cank}%
\def\mlttitle{Aghyusakneri cank}%
%%
\def\stctitle{Bovandakuthyun}%
\def\slftitle{Patkerneri cank}%
\def\slttitle{Aghyusakneri cank}%
⟨/armenian⟩
```

13.12 "Australian" language: australian.mld

The "australian" language is just like "english", so we just load english.mld (see section 13.43 on page 454):

```
⟨*australian⟩
\ProvidesFile{australian.mld}[2006/01/11]
\mtcselectlanguage{english}%
⟨/australian⟩
```

13.13 "Austrian" language: austrian.mld

For the mini-table titles, the "austrian" language is like the "german" language, so we load german.mld (see section 13.65 on page 465):

```
⟨*austrian⟩
\ProvidesFile{austrian.mld}[2004/12/14]
\mtcselectlanguage{german}%
⟨/austrian⟩
```

13.14 "Bahasa" language: bahasa.mld

The "bahasa" language is just like "bahasai", so we just load bahasai.mld (see section 13.15 on the next page):

```
⟨*bahasa⟩
\ProvidesFile{bahasa.mld}[2006/01/11]
\mtcselectlanguage{bahasai}%
⟨/bahasa⟩
```

13.15 "Bahasai" language: bahasai.mld

The titles of the mini-tables for the "bahasai" language [9] (bahasa indonesia / bahasa meyalu) are taken from the file bahasa.dtx (by Jörg Knappen and Terry Mart) in the babel package [38, 39, 64]. Specific fonts are needed. See also section 13.16. The word "bahasa" means "language" in bahasa. For other names for this language, see sections 13.14 on the page before, 13.87 on page 478, and 13.88 on page 478.

```
⟨*bahasai⟩
\ProvidesFile{bahasai.mld}[2006/01/13]
%% Bahasa Indonesia titles for minitoc.sty
%% From bahasa.dtx in the babel package
%% Knappen, Jörg and Mart, Terry
\def\ptctitle{Daftar Isi}%
\def\plftitle{Daftar Gambar}%
\def\plttitle{Daftar Tabel}%
%%
\def\mtctitle{Daftar Isi}%
\def\mlftitle{Daftar Gambar}%
\def\mlttitle{Daftar Tabel}%
%%
\def\stctitle{Daftar Isi}%
\def\slftitle{Daftar Gambar}%
\def\slttitle{Daftar Tabel}%
⟨/bahasai⟩
```

13.16 "Bahasam" language: bahasam.mld

The titles of the mini-tables for the "bahasam" language (Bahasa Malaysia) [10] are taken from the file bahasam.dtx (by Jörg Knappen, Terry Mart and Bob Margolis) in the babel package [38, 39, 65]. Specific fonts are needed. See also section 13.14 on the preceding page. For other names for this language, see sections 13.111 on page 489 and 13.117 on page 492.

```
⟨*bahasam⟩
\ProvidesFile{bahasam.mld}[2006/12/19]
%% Bahasa Malaysia titles for minitoc.sty
%% From bahasam.dtx in the babel package
%% Knappen, Jörg and Mart, Terry and Margolis, Bob
\def\ptctitle{Kandungan}%
\def\plftitle{Senarai Gambar}%
\def\plttitle{Senarai Jadual}%
%%
\def\mtctitle{Kandungan}%
```

[9] Bahasa is spoken in Indonesia and Malaysia, with different pronunciations and titles but the same writing. Bahasai is the indonesian variant.

[10] Spoken in Indonesia and Malaysia, with different pronunciations and titles but the same writing. Bahasam is the malaysian variant.

```
\def\mlftitle{Senarai Gambar}%
\def\mlttitle{Senarai Jadual}%
%%
\def\stctitle{Kandungan}%
\def\slftitle{Senarai Gambar}%
\def\slttitle{Senarai Jadual}%
⟨/bahasam⟩
```

13.17 "Bangla" language: bangla.mld

The titles for the "bangla" language [11] are taken from the BangTEX package [202] (by Palash Baran PAL); they need specific fonts.

```
⟨*bangla⟩
\ProvidesFile{bangla.mld}[2006/03/31]
%% Bangla titles for minitoc.sty
%% from BangTeX
%% Needs specific fonts
\def\ptctitle{suu\*c*ipotRo}%
\def\plftitle{cho\*b*ir ta\*l*ika}%
\def\plttitle{cho\*k*er ta\*l*ika}%
%%
\def\mtctitle{suu\*c*i}%
\def\mlftitle{cho\*b*ir ta\*l*ika}%
\def\mlttitle{cho\*k*er ta\*l*ika}%
%%
\def\stctitle{suu\*c*i}%
\def\slftitle{cho\*b*ir ta\*l*ika}%
\def\slttitle{cho\*k*er ta\*l*ika}%
⟨/bangla⟩
```

13.18 "Basque" language: basque.mld

The titles for the "basque" language [12] are taken from the basque.dtx file in the babel package [38, 39, 45], by Juan M. AGUIRREGABIRIA and Julio SÁNCHEZ, with help from Zunbeltz IZAOLA. It seems that 8 bits fonts are preferable.

```
⟨*basque⟩
\ProvidesFile{basque.mld}[2006/01/13]
%% Basque titles for minitoc.sty
%% from basque.dtx (babel)
%% Aguirregabiria, Juan M. and Sanchez, Julio
%% Needs special fonts
```

[11] Spoken in Bangladesh and some parts of India.
[12] Spoken in the Basque country, in the north of Spain and south-west of France.

```
\def\ptctitle{Gaien Aurkibidea}%
\def\plftitle{Irudien Zerrenda}%
\def\plttitle{Taulen Zerrenda}%
%%
\def\mtctitle{Gaien Aurkibidea}%
\def\mlftitle{Irudien Zerrenda}%
\def\mlttitle{Taulen Zerrenda}%
%%
\def\stctitle{Gaien Aurkibidea}%
\def\slftitle{Irudien Zerrenda}%
\def\slttitle{Taulen Zerrenda}%
⟨/basque⟩
```

13.19 "Bicig" language: bicig.mld

The titles for the "bicig" language [13] are taken from the MonTEX package [97, 100]. This language requires specific fonts. See also sections 13.118 on page 492, 13.20, and 13.21 on the following page.

```
⟨*bicig⟩
\ProvidesFile{bicig.mld}[1999/03/16]
%% Mongol (Bicig) titles for minitoc.sty
%% needs mongol fonts
\def\ptctitle{\bcg{GarciG}}%
\def\plftitle{\bcg{zuraG-un zigsaalt}}%
\def\plttitle{\bcg{k"usn"agti"in jagsaalt}}%
%%
\def\mtctitle{\bcg{GarciG}}%
\def\mlftitle{\bcg{zuraG-un zigsaalt}}%
\def\mlttitle{\bcg{k"usn"agti"in jagsaalt}}%
%%
\def\stctitle{\bcg{GarciG}}%
\def\slftitle{\bcg{zuraG-un zigsaalt}}%
\def\slttitle{\bcg{k"usn"agti"in jagsaalt}}%
⟨/bicig⟩
```

13.20 "Bicig2" language: bicig2.mld

The titles for the "bicig2" language [14] are taken from the MonTEX package [97, 100]. This language requires specific fonts. See also sections 13.118 on page 492, 13.19, and 13.21 on the following page.

[13] The bicig is a written form of the mongolian language. It is also known as Uighur. See also section 13.161 on page 515.

[14] The bicig, or uighur, is a written form of the mongolian language, bicig2 is a variant. See also section 13.162 on page 515.

```
⟨*bicig2⟩
\ProvidesFile{bicig2.mld}[2005/11/16]
%% Mongol (Bicig2) titles for minitoc.sty
%% needs mongol fonts
\def\ptctitle{garcag}%
\def\plftitle{zirug-un zigsagalda}%
\def\plttitle{kuisunukdu-yin zigsagalda}%
%%
\def\mtctitle{garcag}%
\def\mlftitle{zirug-un zigsagalda}%
\def\mlttitle{kuisunukdu-yin zigsagalda}%
%%
\def\stctitle{garcag}%
\def\slftitle{zirug-un zigsagalda}%
\def\slttitle{kuisunukdu-yin zigsagalda}%
⟨/bicig2⟩
```

13.21 “Bicig3” language: bicig3.mld

The titles for the “bicig3” language [15] are taken from the MonTEX package [97, 100]. This language requires specific fonts. See also sections 13.118 on page 492, 13.19 on the page before and 13.20 on the preceding page.

```
⟨*bicig3⟩
\ProvidesFile{bicig3.mld}[2006/03/31]
%% Mongol (Bicig3) titles for minitoc.sty
%% needs mongol fonts
\def\ptctitle{aguulag=a}%
\def\plftitle{zirug-un zigsagalda}%
\def\plttitle{kuisunukdu-yin zigsagalda}%
%%
\def\mtctitle{aguulag=a}%
\def\mlftitle{zirug-un zigsagalda}%
\def\mlttitle{kuisunukdu-yin zigsagalda}%
%%
\def\stctitle{aguulag=a}%
\def\slftitle{zirug-un zigsagalda}%
\def\slttitle{kuisunukdu-yin zigsagalda}%
⟨/bicig3⟩
```

[15] The bicig, or uighur, is a written form of the mongolian language, bicig3 is a variant. See also section 13.163 on page 516.

13.22 "Bithe" language: bithe.mld

The titles for the "bithe" language [16] are taken from the MonTEX package [97, 100]. This language requires specific fonts. See also sections 13.118 on page 492 and 13.116 on page 492.

The Manju writing, or *bithe* system is a close relative of the Mongolian system; the basical letter shapes are the same. Yet for Manju, a set of diacritics (*dots and circles*) was designed to the effect that all the ambiguities of Mongolian are eliminated.

```
⟨*bithe⟩
\ProvidesFile{bithe.mld}[2005/11/16]
%% Manju (bithe) titles for minitoc.sty
%% needs mongol fonts
\def\ptctitle{garcag}%
\def\plftitle{zirug-un? afaha}%
\def\plttitle{kuisunukdu-yin? afaha}%
%%
\def\mtctitle{garcag}%
\def\mlftitle{zirug-un? afaha}%
\def\mlttitle{kuisunukdu-yin? afaha}%
%%
\def\stctitle{garcag}%
\def\slftitle{zirug-un? afaha}%
\def\slttitle{kuisunukdu-yin? afaha}%
⟨/bithe⟩
```

13.23 "Brazil" language: brazil.mld

The titles for the "brazil" language [17] are taken from the portuges.dtx file (for portugese titles by Jose Pedro Ramalhete) in the babel package [38, 39, 74]:

```
⟨*brazil⟩
\ProvidesFile{brazil.mld}[2006/01/13]
%% Portugues (brazil) titles for minitoc.sty
%% from portuges.dtx (babel)
%% Ramalhete, Jose Pedro and "de Lima", Arnaldo Viegas
\def\ptctitle{Sum\'ario}%
\def\plftitle{Lista de Figuras}%
\plttitle{Lista de Tabelas}%
%%
\def\mtctitle{Sum\'ario}%
\def\mlftitle{Lista de Figuras}%
\def\mlttitle{Lista de Tabelas}%
%%
```

[16] The bithe is a written form of the manju variant of the mongolian language.

[17] It is the portuguese dialect spoken in Brazil. These titles are *different* in Brazil and in Portugal. Arnaldo Viegas de Lima contributed brasilian translations. See section 13.132 on page 499.

```
\def\stctitle{Sum\'ario}%
\def\slftitle{Lista de Figuras}%
\def\slttitle{Lista de Tabelas}%
⟨/brazil⟩
```

13.24 "Brazilian" language: brazilian.mld

The "brazilian" language is just like "brazil", so we just load brazil.mld (see section 13.23 on the preceding page):

```
⟨*brazilian⟩
\ProvidesFile{brazilian.mld}[2005/07/11]
\mtcselectlanguage{brazil}%
⟨/brazilian⟩
```

13.25 "Breton" language: breton.mld

The titles for the "breton" language[18] are taken from breton.dtx file in the babel package [38, 39, 75]:

```
⟨*breton⟩
\ProvidesFile{breton.mld}[2006/01/13]
%% Breton titles for minitoc.sty
%% from breton.dtx (babel)
%% Rolland, Christian
\def\ptctitle{Taolenn}%
\def\plftitle{Listenn ar Figurenno\`u}%
\def\plttitle{Listenn an taolenno\`u}%
%%
\def\mtctitle{Taolenn}%
\def\mlftitle{Listenn ar Figurenno\`u}%
\def\mlttitle{Listenn an taolenno\`u}%
%%
\def\stctitle{Taolenn}%
\def\slftitle{Listenn ar Figurenno\`u}%
\def\slttitle{Listenn an taolenno\`u}%
⟨/breton⟩
```

[18] Spoken as a local celtic dialect in french Brittany.

13.26 "British" language: `british.mld`

The "`british`" language is just like "`english`", so we just load `english.mld` (see section 13.43 on page 454):

```
⟨*british⟩
\ProvidesFile{british.mld}[2005/07/11]
\mtcselectlanguage{english}%
⟨/british⟩
```

13.27 "Bulgarian" language: `bulgarian.mld`

The titles for the "`bulgarian`" language are taken from the `bulgarian.dtx` (adapted from russian by Georgi Boshnakov) file in the babel package [38, 39, 50]; they require specific cyrillic fonts. See also section 13.28 on the following page.

```
⟨*bulgarian⟩
\ProvidesFile{bulgarian.mld}[2006/01/13]
%% Bulgarian titles for minitoc.sty
%% from bulgarian.dtx (babel)
%% Boshnakov, Georgi
%% Needs special fonts
\def\ptctitle{%
   {\cyr\CYRS\cyrhrdsn\cyrd\cyrhrdsn\cyrr\cyrzh\cyra\cyrn\cyri\cyre}}%
\def\plftitle{%
   {\cyr\CYRS\cyrp\cyri\cyrs\cyrhrdsn\cyrk\ %
    \cyrn\cyra\ \cyrf\cyri\cyrg\cyru\cyrr\cyri\cyrt\cyre}}%
\def\plttitle{%
   {\cyr\CYRS\cyrp\cyri\cyrs\cyrhrdsn\cyrk\ %
    \cyrn\cyra\ \cyrt\cyra\cyrb\cyrl\cyri\cyrc\cyri\cyrt\cyre}}%
%%
\def\mtctitle{%
   {\cyr\CYRS\cyrhrdsn\cyrd\cyrhrdsn\cyrr\cyrzh\cyra\cyrn\cyri\cyre}}%
\def\mlftitle{%
   {\cyr\CYRS\cyrp\cyri\cyrs\cyrhrdsn\cyrk\ %
    \cyrn\cyra\ \cyrf\cyri\cyrg\cyru\cyrr\cyri\cyrt\cyre}}%
\def\mlttitle{%
   {\cyr\CYRS\cyrp\cyri\cyrs\cyrhrdsn\cyrk\ %
    \cyrn\cyra\ \cyrt\cyra\cyrb\cyrl\cyri\cyrc\cyri\cyrt\cyre}}%
%%
\def\stctitle{%
   {\cyr\CYRS\cyrhrdsn\cyrd\cyrhrdsn\cyrr\cyrzh\cyra\cyrn\cyri\cyre}}%
\def\slftitle{%
   {\cyr\CYRS\cyrp\cyri\cyrs\cyrhrdsn\cyrk\ %
    \cyrn\cyra\ \cyrf\cyri\cyrg\cyru\cyrr\cyri\cyrt\cyre}}%
\def\slttitle{%
   {\cyr\CYRS\cyrp\cyri\cyrs\cyrhrdsn\cyrk\ %
    \cyrn\cyra\ \cyrt\cyra\cyrb\cyrl\cyri\cyrc\cyri\cyrt\cyre}}%
```

```
⟨/bulgarian⟩
```

13.28 “Bulgarianb” language: bulgarianb.mld

The titles for the “bulgarianb” (upper bulgarian) language are taken from the russianb.dtx file (by Olga G. Lapko, Vladimir Volovich and Werner Lemberg) of the babel package [38, 39, 66, 160]; they require specific cyrillic fonts. See also section 13.27 on the page before.

```
⟨*bulgarianb⟩
\ProvidesFile{bulgarianb.mld}[2006/03/06]
%% Upper bulgarian titles for minitoc.sty
%% From russianb.dtx
%% Needs cyrillic fonts for upper bulgarian
\def\ptctitle{%
 {\cyr\CYRS\cyrhrdsn\cyrd\cyrhrdsn\cyrr\cyrzh\cyra\cyrn\cyri\cyre}}%
\def\plftitle{% Figuri
  {\cyr \CYRF\cyri\cyrg\cyru\cyrr\cyri}}%
\def\plttitle{% Tablici
  {\cyr \CYRT\cyra\cyrb\cyrl\cyri\cyrc\cyri}}%
%%
\def\mtctitle{% Sydyrzhanie
 {\cyr\CYRS\cyrhrdsn\cyrd\cyrhrdsn\cyrr\cyrzh\cyra\cyrn\cyri\cyre}}%
\def\mlftitle{% Figurite
  {\cyr \CYRF\cyri\cyrg\cyru\cyrr\cyri}}%
\def\mlttitle{% Tablici
  {\cyr \CYRT\cyra\cyrb\cyrl\cyri\cyrc\cyri}}%
%%
\def\stctitle{% Sydyrzhanie
 {\cyr\CYRS\cyrhrdsn\cyrd\cyrhrdsn\cyrr\cyrzh\cyra\cyrn\cyri\cyre}}%
\def\slftitle{% Figuri
  {\cyr \CYRF\cyri\cyrg\cyru\cyrr\cyri}}%
\def\slttitle{% Tablici
  {\cyr \CYRT\cyra\cyrb\cyrl\cyri\cyrc\cyri}}%
⟨/bulgarianb⟩
```

13.29 “Buryat” language: buryat.mld

The titles for the “buryat” language [19] are taken from the MonTEX package [97, 100]. This language requires specific fonts. See also section 13.118 on page 492.

```
⟨*buryat⟩
\ProvidesFile{buryat.mld}[1999/03/16]
%% Buryat titles for minitoc.sty
%% Needs special fonts
```

[19] Spoken in some regions of Mongolia, in the Buryat republic, near Lake Baikal.

```
\def\ptctitle{{\mnr Gar{\sh}ag}}%
\def\plftitle{{\mnr Zuraga"i jagsaalt}}%
\def\plttitle{{\mnr X"usn"ag"at"a"i jagsaalt}}%
%%
\def\mtctitle{{\mnr Gar{\sh}ag}}%
\def\mlftitle{{\mnr Zuraga"i jagsaalt}}%
\def\mlttitle{{\mnr X"usn"ag"at"a"i jagsaalt}}%
%%
\def\stctitle{{\mnr Gar{\sh}ag}}%
\def\slftitle{{\mnr Zuraga"i jagsaalt}}%
\def\slttitle{{\mnr X"usn"ag"at"a"i jagsaalt}}%
⟨/buryat⟩
```

13.30 "Buryat2" language: buryat2.mld

The titles for the "buryat2" language (a variant for the "buryat" language, see section 13.29 on the page before) are taken from the MonTEX package [97, 100]. This language requires specific fonts. See also section 13.118 on page 492.

```
⟨*buryat2⟩
\ProvidesFile{buryat2.mld}[1999/03/16]
%% Buryat2 titles for minitoc.sty
%% Needs special fonts
\def\ptctitle{{\mnr Aguulga}}%
\def\plftitle{{\mnr Zuraga"i jagsaalt}}%
\def\plttitle{{\mnr X"usn"ag"at"a"i jagsaalt}}%
%%
\def\mtctitle{{\mnr Aguulga}}%
\def\mlftitle{{\mnr Zuraga"i jagsaalt}}%
\def\mlttitle{{\mnr X"usn"ag"at"a"i jagsaalt}}%
%%
\def\stctitle{{\mnr Aguulga}}%
\def\slftitle{{\mnr Zuraga"i jagsaalt}}%
\def\slttitle{{\mnr X"usn"ag"at"a"i jagsaalt}}%
⟨/buryat2⟩
```

13.31 "Canadian" language: canadian.mld

The "canadian" language (note the final "i*an*") is just the english language spoken in Canada. We just load the file english.mld (see section 13.43 on page 454):

```
⟨*canadian⟩
\ProvidesFile{canadian.mld}[2004/12/14]
\mtcselectlanguage{english}%
⟨/canadian⟩
```

13.32 "Canadien" language: canadien.mld

The "canadien" language (note the final "i*e*n") is just the french language spoken in Canada. We just load the file french.mld (see section 13.58 on page 461):

```
⟨*canadien⟩
\ProvidesFile{canadien.mld}[2004/12/14]
\mtcselectlanguage{french}%
⟨/canadien⟩
```

13.33 "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 13.153 on page 511):

```
⟨*castillan⟩
\ProvidesFile{castillan.mld}[2004/12/14]
%% Castillan (spanish) titles for minitoc.sty
\mtcselectlanguage{spanish}%
⟨/castillan⟩
```

13.34 "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 13.153 on page 511):

```
⟨*castillian⟩
\ProvidesFile{castillian.mld}[2005/07/01]
%% Castillian (spanish) titles for minitoc.sty
\mtcselectlanguage{spanish}%
⟨/castillian⟩
```

13.35 "Catalan" language: catalan.mld

The titles for the "catalan" language [20] are taken from the catalan.dtx file (adapted from spanish by Gonvcal Badenes and Jörg Knappen) in the babel package [38, 39, 47]:

[20] Spoken in Catalunya, the eastern part of Spain, around Barcelona.

```
⟨*catalan⟩
\ProvidesFile{catalan.mld}[2006/01/13]
%% Catalan titles for minitoc.sty
%% from catalan.dtx (babel)
%% Badenes, Goncal
\def\ptctitle{\'Index}%
\def\plftitle{\'Index de figures}%
\def\plttitle{\'Index de taules}%
%%
\def\mtctitle{\'Index}%
\def\mlftitle{Figures}%
\def\mlttitle{Taules}%
%%
\def\stctitle{\'Index}%
\def\slftitle{Figures}%
\def\slttitle{Taules}%
⟨/catalan⟩
```

13.36 "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 [167, 168] (by Werner Lemberg). Special fonts are needed, of course. See also section 13.37.

The titles for the "chinese1" language contain characters that cannot be easily generated, hence we load chinese1.mlo.

```
⟨*chinese1⟩
\ProvidesFile{chinese1.mld}[2005/01/28]
%% From the file file Bg5.cap of the CJK package
%%   for using Asian logographs (Chinese/Japanese/Korean) with LaTeX2e
%%   created by Werner Lemberg <wl@gnu.org>
%% Version 4.5.2 (28-Mar-2003)
%% Chinese captions: character set: Big 5, encoding: Big 5
%%
\mtcloadmlo{chinese1}%
⟨/chinese1⟩
```

13.37 "Chinese2" language: chinese2.mld and chinese2.mlo

The "chinese2" language uses titles taken from the Bg5.cpx file in the CJK system [167, 168] (by Werner Lemberg). Special fonts are needed, of course. See also section 13.36.

The titles for the "chinese2" language contain characters that cannot be easily generated, hence we load chinese2.mlo.

```
⟨*chinese2⟩
\ProvidesFile{chinese2.mld}[2005/01/28]
%% From the file Bg5.cpx of the CJK package
%% for using Asian logographs (Chinese/Japanese/Korean) with LaTeX2e
%% created by Werner Lemberg <wl@gnu.org>
%%
%% Version 4.5.2 (28-Mar-2003)
%% Chinese captions: character set: Big 5, encoding: Big 5, preprocessed
%%
\mtcloadmlo{chinese2}%
⟨/chinese2⟩
```

13.38 "Croatian" language: croatian.mld

The titles for the "croatian" language are taken from the file croatian.dtx file (by Alan Paić) in the babel package [38, 39, 71]:

```
⟨*croatian⟩
\ProvidesFile{croatian.mld}[2006/01/13]
%% Croatian titles for minitoc.sty
%% from croatian.dtx (babel)
%% Pai\'{c}, Alan
\def\ptctitle{Sadr\v{z}aj}%
\def\plftitle{Slike}%
\def\plttitle{Tablice}%
%%
\def\mtctitle{Sadr\v{z}aj}%
\def\mlftitle{Slike}%
\def\mlttitle{Tablice}%
%%
\def\stctitle{Sadr\v{z}aj}%
\def\slftitle{Slike}%
\def\slttitle{Tablice}%
⟨/croatian⟩
```

13.39 "Czech" language: czech.mld

The titles for the "czech" language are taken from the file czech.dtx (contributions by Milos Lokajicek) in the babel package [38, 39, 69]:

```
⟨*czech⟩
\ProvidesFile{czech.mld}[2006/01/13]
```

```
%% Czech titles for minitoc.sty
%% from czech.dtx (babel)
%% Lokajicek, Milos
%% Warning: defines \w as a ring accent
\def\w#1{\accent'27 #1}
\def\ptctitle{Obsah}%
\def\plftitle{Seznam obr\'azk\w{u}}%
\def\plttitle{Seznam tabulek}%
%%
\def\mtctitle{Obsah}%
\def\mlftitle{Seznam obr\'azk\w{u}}%
\def\mlttitle{Seznam tabulek}%
%%
\def\stctitle{Obsah}%
\def\slftitle{Seznam obr\'azk\w{u}}%
\def\slttitle{Seznam tabulek}%
⟨/czech⟩
```

13.40 "Danish" language: danish.mld

The titles for the "danish" language are taken from the danish.dtx file (by Henning Larsen) in the babel package [38, 39, 67]:

```
⟨*danish⟩
\ProvidesFile{danish.mld}[2006/01/13]
%% Danish titles for minitoc.sty
%% fron danish.dtx (babel)
%% Larsen, Henning
\def\ptctitle{Indholdsfortegnelse}%
\def\plftitle{Figurer}%
\def\plttitle{Tabeller}%
%%
\def\mtctitle{Indholdsfortegnelse}%
\def\mlftitle{Figurer}%
\def\mlttitle{Tabeller}%
%%
\def\stctitle{Indholdsfortegnelse}%
\def\slftitle{Figurer}%
\def\slttitle{Tabeller}%
⟨/danish⟩
```

13.41 "Devanagari" language: devanagari.mld

The titles for the "devanagari" language are taken from the devanagari.sty file (by Anshuman Pandey, CV Radhakrishnan, Zdeněk Wagner, John Smith, Kevin Carmody, Richard Mahoney and Dominik Wujastyk) in the Devanāgarī package [204] (Devanāgarī).

See also section 13.83 on page 476. Specific fonts are required. The home page of the package is http://devnag.ramovar.org.

```
⟨*devanagari⟩
\ProvidesFile{devanagari.mld}[2006/08/25]
%% Devanagari (hindi) titles for minitoc.sty
%% from devanagari.sty by
%% Pandey, Anshuman
%% Radhakrishnan, CV
%% Wagner, Zden\v{e}k
%% Smith, John
%% Carmody, Kevin
%% Mahoney, Richard
%% Wujastyk, Dominik
\def\ptctitle{{\dn Evqy{\rs -\re}\8{s}cF}}%
\def\plftitle{{\dn Ec/o{\qva} kF \8{s}cF}}%
\def\plttitle{{\dn tAElkAao\2 kF \8{s}cF}}%
%%
\def\mtctitle{{\dn Evqy{\rs -\re}\8{s}cF}}%
\def\mlftitle{{\dn Ec/o{\qva} kF \8{s}cF}}%
\def\mlttitle{{\dn tAElkAao\2 kF \8{s}cF}}%
%%
\def\stctitle{{\dn Evqy{\rs -\re}\8{s}cF}}%
\def\slftitle{{\dn Ec/o{\qva} kF \8{s}cF}}%
\def\slttitle{{\dn tAElkAao\2 kF \8{s}cF}}%
⟨/devanagari⟩
```

13.42 “Dutch” language: dutch.mld

The titles for the “dutch” language are taken from the dutch.dtx file (by Johannes Braams) in the babel package [38, 39, 40]:

```
⟨*dutch⟩
\ProvidesFile{dutch.mld}[2006/01/13]
% Dutch titles for minitoc.sty
%% from dutch.dtx (babel)
%% Braams, Johannes
\def\ptctitle{Inhoudsopgave}%
\def\plftitle{Lijst van figuren}%
\def\plttitle{Lijst van tabellen}%
%%
\def\mtctitle{Inhoudsopgave}%
\def\mlftitle{Lijst van figuren}%
\def\mlttitle{Lijst van tabellen}%
%%
\def\stctitle{Inhoudsopgave}%
\def\slftitle{Lijst van figuren}%
\def\slttitle{Lijst van tabellen}%
⟨/dutch⟩
```

13.43 "English" language: english.mld

The titles for the "english" language are taken from the english.dtx file (by Johannes Braams) in the babel package [38, 39, 41]. *The presence of the* `english.mld` *file is mandatory, because english is the default language.*

See also sections 13.6 on page 436, 13.12 on page 439, 13.26 on page 446, 13.31 on page 448, 13.120 on page 493, 13.164 on page 516, and 13.168 on page 518.

```
⟨*english⟩
\ProvidesFile{english.mld}[2006/01/13]
%% English titles for minitoc.sty
%% from english.dtx (babel)
%% Braams, Johannes
\def\ptctitle{Table of Contents}%
\def\plftitle{List of Figures}%
\def\plttitle{List of Tables}%
%%
\def\mtctitle{Contents}%
\def\mlftitle{Figures}%
\def\mlttitle{Tables}%
%%
\def\stctitle{Contents}%
\def\slftitle{Figures}%
\def\slttitle{Tables}%
⟨/english⟩
```

13.44 "English1" language: english1.mld

The titles for the "english1" language come from the english.dtx file (by Johannes Braams) in the babel package [38, 39, 41], with some adaptations for the part-level titles.

```
⟨*english1⟩
\ProvidesFile{english1.mld}[2006/03/30]
%% English titles for minitoc.sty
%% from english.dtx (babel) Braams, Johannes
%% ptctitle, plftitle and plttitle modified (JPFD)
\def\ptctitle{\ifnum\value{part}=1\relax
  Table of Contents of the First Part\relax
  \else Table of Contents of Part~\Roman{part}\fi}%
\def\plftitle{\ifnum\value{part}=1\relax
  List of Figures in the First Part\relax
  \else List of Figures in Part~\Roman{part}\fi}%
\def\plttitle{\ifnum\value{part}=1\relax
  List of Tables in the First Part\relax
  \else List of Tables in Part~\Roman{part}\fi}%
%%
\def\mtctitle{Contents}%
```

```
\def\mlftitle{Figures}%
\def\mlttitle{Tables}%
%%
\def\stctitle{Contents}%
\def\slftitle{Figures}%
\def\slttitle{Tables}%
⟨/english1⟩
```

13.45 "English2" language: `english2.mld`

The titles for the "`english2`" language are again taken from the `english.dtx` file (by Johannes Braams) in the babel package [38, 39, 41], with some adaptations for the titles at the part level.

```
⟨*english2⟩
\ProvidesFile{english2.mld}[2006/03/30]
%% English titles for minitoc.sty
%% from english.dtx (babel) Braams, Johannes
%% ptctitle, plftitle and plttitle modified (JPFD)
\def\mtcEnglishIIpart{\ifcase\value{part}%
\or the First Part\or the Second Part\or
the Third Part\or the Fourth Part\or the Fifth Part\or
the Sixth Part\or the Seventh Part\or the Eighth Part\or
the Ninth Part\or the Tenth Part\or the Eleventh Part\or
the Twelfth Part\or the Thirteenth Part\or the Fourteenth Part\or
the Fifteenth Part\or the Sixteenth Part\or the Seventeenth Part\or
the Eighteenth Part\or the Nineteenth Part\or
the Twentieth Part\else Part~\Roman{part}\fi}
\def\ptctitle{Contents of \mtcEnglishIIpart}
\def\plftitle{List of Figures in \mtcEnglishIIpart}
\def\plttitle{List of Tables in \mtcEnglishIIpart}
%%
\def\mtctitle{Contents}%
\def\mlftitle{Figures}%
\def\mlttitle{Tables}%
%%
\def\stctitle{Contents}%
\def\slftitle{Figures}%
\def\slttitle{Tables}%
⟨/english2⟩
```

13.46 "Esperant" language: `esperant.mld`

The titles for the "`esperant`" (espéranto) language are taken from the `esperanto.dtx` file (by Marti Ruiz-Altaba and Jörg Knappen) in the babel package [38, 39, 76]. The esperanto

(artificial) language was created in the 1877–1885 years by Doctor Lejzer Ludwig ZAMENOF[21] (1859–1917) of Warsaw, Poland.

```
⟨*esperant⟩
\ProvidesFile{esperant.mld}[2006/12/19]
%% Esperanto titles for minitoc.sty
%% from esperanto.dtx (babel)
%% Ruiz-Altaba, Marti and Knappen. Jörg
\def\ptctitle{Enhavo}%
\def\plftitle{Listo de figuroj}%
\def\plttitle{Listo de tabeloj}%
%%
\def\mtctitle{Enhavo}%
\def\mlftitle{Listo de figuroj}%
\def\mlttitle{Listo de tabeloj}%
%%
\def\stctitle{Enhavo}%
\def\slftitle{Listo de figuroj}%
\def\slttitle{Listo de tabeloj}%
⟨/esperant⟩
```

13.47 "Esperanto" language: esperanto.mld

The "esperanto" and "esperant" languages are synonyms, so we just load the esperant.mld file (see section 13.46 on the page before):

```
⟨*esperanto⟩
\ProvidesFile{esperanto.mld}[2004/12/14]
\mtcselectlanguage{esperant}%
⟨/esperanto⟩
```

13.48 "Estonian" language: estonian.mld

The titles for the "estonian" language are taken from the estonian.dtx file (by Enn SAAR) in the babel package [38, 39, 77]:

```
⟨*estonian⟩
\ProvidesFile{estonian.mld}[2006/01/13]
%% Estonian titles for minitoc.sty
%% from estonian.dtx (babel)
%% Saar, Enn
\def\ptctitle{Sisukord}%
\def\plftitle{Joonised}%
```

[21] See http://uea.org/ and http://www.esperanto-france.org/ for more information.

```
\def\plttitle{Tabelid}%
%%
\def\mtctitle{Sisukord}%
\def\mlftitle{Joonised}%
\def\mlttitle{Tabelid}%
%%
\def\stctitle{Sisukord}%
\def\slftitle{Joonised}%
\def\slttitle{Tabelid}%
⟨/estonian⟩
```

13.49 "Ethiopia" language: ethiopia.mld

The titles for the "ethiopia" language are taken from the ethiop package [29]. Specific fonts are needed. See also section 13.51 on the following page.

```
⟨*ethiopia⟩
\ProvidesFile{ethiopia.mld}[1999/03/16]
%% Ethopian titles for minitoc.sty
%% Needs special fonts
\def\ptctitle{yezate}%
\def\plftitle{%
    ya\eth@doaltchar{85}‘elo\eth@doaltchar{109}
    mAwe\eth@doaltchar{187}}%
\def\plttitle{%
    yasane\eth@doaltchar{176}ra\eth@doaltchar{149}
    mAwe\eth@doaltchar{187}}%
%%
\def\mtctitle{yezate}%
\def\mlftitle{%
    ya\eth@doaltchar{85}‘elo\eth@doaltchar{109}
    mAwe\eth@doaltchar{187}}%
\def\mlttitle{%
    yasane\eth@doaltchar{176}ra\eth@doaltchar{149}
    mAwe\eth@doaltchar{187}}%
%%
\def\stctitle{yezate}%
\def\slftitle{%
    ya\eth@doaltchar{85}‘elo\eth@doaltchar{109}
    mAwe\eth@doaltchar{187}}%
\def\slttitle{%
    yasane\eth@doaltchar{176}ra\eth@doaltchar{149}
    mAwe\eth@doaltchar{187}}%
⟨/ethiopia⟩
```

13.50 "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 13.49 on the page before). See also section 13.49 on the preceding page.

```
⟨*ethiopian⟩
\ProvidesFile{ethiopian.mld}[2004/12/14]
%% Needs ethopian special fonts
\mtcselectlanguage{ethiopia}%
⟨/ethiopian⟩
```

13.51 "Ethiopian2" language: ethiopian2.mld

The titles for the "ethiopian2" language (for Omega) are taken from the ethiop package [29]. Specific fonts are needed. See also section 13.49 on the page before.

```
⟨*ethiopian2⟩
\ProvidesFile{ethiopian2.mld}[2006/01/30]
%% Ethopian titles for minitoc.sty with Omega.
%% Needs special fonts
\def\ptctitle{^^^^12ed^^^^12d8^^^^1275}%
\def\plftitle{^^^^12e8^^^^1225^^^^12d5^^^^120e^^^^127d
  ^^^^121b^^^^12cd^^^^132b}%
\def\plttitle{^^^^12e8^^^^1230^^^^1295^^^^1320^^^^1228^^^^12e5
  ^^^^121b^^^^12cd^^^^132b}%
%%
\def\mtctitle{^^^^12ed^^^^12d8^^^^1275}%
\def\mlftitle{^^^^12e8^^^^1225^^^^12d5^^^^120e^^^^127d
  ^^^^121b^^^^12cd^^^^132b}%
\def\mlttitle{^^^^12e8^^^^1230^^^^1295^^^^1320^^^^1228^^^^12e5
  ^^^^121b^^^^12cd^^^^132b}%
%%
\def\stctitle{^^^^12ed^^^^12d8^^^^1275}%
\def\slftitle{^^^^12e8^^^^1225^^^^12d5^^^^120e^^^^127d
  ^^^^121b^^^^12cd^^^^132b}%
\def\slttitle{^^^^12e8^^^^1230^^^^1295^^^^1320^^^^1228^^^^12e5
  ^^^^121b^^^^12cd^^^^132b}%
⟨/ethiopian2⟩
```

13.52 "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 [109] system[22], by Mohammad Ghodsi, Behdad Esfahbod, Roozbeh Pournader, Hassan Abolhassani, and others. Special fonts are needed, of course. See also section 13.53.

The titles for the "farsi1" language contain characters that cannot be easily generated, hence we load farsi1.mlo.

```
⟨*farsi1⟩
\ProvidesFile{farsi1.mld}[2005/09/13]
%% From farsi.sty of the FarsiTeX project
%% by Dr Mohammad Ghodsi,
%% Roozbeh Pournader (roozbeh@sharif.edu),
%% Hassan Abolhassani, and others.
%% http://www.farsitex.org
\mtcloadmlo{farsi1}%
⟨/farsi1⟩
```

13.53 "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 [109][22], by Mohammad Ghodsi, Roozbeh Pournader, Behdad Esfahbod, Hassan Abolhassani, and others. Special fonts are needed, of course. See also section 13.52.

The titles for the "farsi2" language contain characters that cannot be easily generated, hence we load farsi2.mlo.

```
⟨*farsi2⟩
\ProvidesFile{farsi2.mld}[2005/09/13]
%% From farsi.sty of the FarsiTeX project
%% by Dr Mohammad Ghodsi,
%% Roozbeh Pournader (roozbeh@sharif.edu),
%% Hassan Abolhassani, and others.
%% http://www.farsitex.org
\mtcloadmlo{farsi2}%
⟨/farsi2⟩
```

[22] By Mohammad Ghodsi (ghodsi@rose.ipm.ac.ir) and FarsiTEX Project Group. See the FarsiTEX site at http://www.farsitex.org

13.54 "Farsi3" language: farsi3.mld

There are several variants for the farsi language, spoken in Iran. The "farsi3" language uses titles taken from the farsi.ldf file in the Arabi system[135], by Youssef Jabri. Special fonts are needed, of course.

```
⟨*farsi3⟩
\ProvidesFile{farsi3.mld}[2006/07/27]
%% From farsi.ldf of the Arabi system
%% by Youssef Jabri.
\def\ptctitle{\FR{\fa\ha\ra\seen\taa\space\meem\nun\dal\ra\jeem\alef\taa}}%
\def\plftitle{\FR{\lam\ya\seen\taa\ \alef\sheen\kaf\alef\lam}}%
\def\plttitle{\FR{\lam\ya\seen\taa\  \jeem\dal\alef\waw\lam}}%
%%
\def\mtctitle{\FR{\fa\ha\ra\seen\taa\space\meem\nun\dal\ra\jeem\alef\taa}}%
\def\mlftitle{\FR{\lam\ya\seen\taa\ \alef\sheen\kaf\alef\lam}}%
\def\mlttitle{\FR{\lam\ya\seen\taa\  \jeem\dal\alef\waw\lam}}%
%%
\def\stctitle{\FR{\fa\ha\ra\seen\taa\space\meem\nun\dal\ra\jeem\alef\taa}}%
\def\slftitle{\FR{\lam\ya\seen\taa\ \alef\sheen\kaf\alef\lam}}%
\def\slttitle{\FR{\lam\ya\seen\taa\  \jeem\dal\alef\waw\lam}}%
⟨/farsi3⟩
```

13.55 "Finnish" language: finnish.mld

The titles for the "finnish" language are taken from the finnish.dtx file (by Mikko Kanerva and Keranen Reino) in the babel package [38, 39, 62]. See also section 13.56 on the following page.

```
⟨*finnish⟩
\ProvidesFile{finnish.mld}[2006/03/20]
%% Finnish titles for minitoc.sty
%% from finnish.dtx (babel)
%% Kanerva, Mikko and Reino, Keranen
\def\ptctitle{Sis\"alt\"o}%
\def\plftitle{Kuvat}%
\def\plttitle{Taulukot}%
%%
\def\mtctitle{Sis\"alt\"o}%
\def\mlftitle{Kuvat}%
\def\mlttitle{Taulukot}%
%%
\def\stctitle{Sis\"alt\"o}%
\def\slftitle{Kuvat}%
\def\slttitle{Taulukot}%
⟨/finnish⟩
```

13.56 "Finnish2" language: finnish2.mld

The titles for the "finnish2" language are taken from a variant proposed by the finnish.dtx file (by Mikko Kanerva and Keranen Reino) in the babel package [38, 39, 62]. See also section 13.55 on the page before.

```
⟨*finnish2⟩
\ProvidesFile{finnish2.mld}[2006/01/13]
%% Finnish titles for minitoc.sty (variant)
%% from finnish.dtx (babel)
%% Kanerva, Mikko and Reino, Keranen
\def\ptctitle{Sis\"allys}%
\def\plftitle{Kuvat}%
\def\plttitle{Taulukot}%
%%
\def\mtctitle{Sis\"allys}%
\def\mlftitle{Kuvat}%
\def\mlttitle{Taulukot}%
%%
\def\stctitle{Sis\"allys}%
\def\slftitle{Kuvat}%
\def\slttitle{Taulukot}%
⟨/finnish2⟩
```

13.57 "Francais" language: francais.mld

The "francais"[23] language is a synonym for the "french" language, so we load the file french.mld (see section 13.58):

```
⟨*francais⟩
\ProvidesFile{francais.mld}[2004/12/14]
\mtcselectlanguage{french}%
⟨/francais⟩
```

13.58 "French" language: french.mld

The titles for the "french" language are taken from the frenchb.dtx file (by Daniel Flipo) in the babel package [38, 39, 57]. See also sections 13.1 on page 434, 13.2 on page 435, 13.32 on page 449, 13.57, 13.61 on page 463, 13.62 on page 464, and 13.63 on page 464.

```
⟨*french⟩
\ProvidesFile{french.mld}[2006/03/21]
```

[23] The right spelling is "français", but I did not dare using a cedilla in a file name.

```
%% French titles for minitoc.sty
%% from frenchb.dtx (babel)
%% Flipo, Daniel
\def\ptctitle{Table des mati\‘eres}%
\def\plftitle{Liste des figures}%
\def\plttitle{Liste des tableaux}%
%%
\def\mtctitle{Sommaire}%
\def\mlftitle{Figures}%
\def\mlttitle{Tableaux}%
%%
\def\stctitle{Sommaire}%
\def\slftitle{Figures}%
\def\slttitle{Tableaux}%
⟨/french⟩
```

13.59 “French1” language: `french1.mld`

The titles for the “`french1`” language are taken from the `frenchb.dtx` (by Daniel Flipo) file in the babel package [38, 39, 57], with some adaptations for the part-level titles.

```
⟨*french1⟩
\ProvidesFile{french1.mld}[2006/03/29]
%% French titles for minitoc.sty
%% from frenchb.dtx (babel) Flipo, Daniel
%% ptctitle, plftitle and plttitle modified (JPFD)
\def\ptctitle{\ifnum\value{part}=1\relax
  Sommaire de la premi\‘ere partie\relax
  \else Sommaire de la partie~\Roman{part}\fi}%
\def\plftitle{\ifnum\value{part}=1\relax
  Liste des figures de la premi\‘ere partie\relax
  \else Liste des figures de la partie~\Roman{part}\fi}%
\def\plttitle{\ifnum\value{part}=1\relax
  Liste des tableaux de la premi\‘ere partie\relax
  \else Liste des tableaux de la partie~\Roman{part}\fi}%
%%
\def\mtctitle{Sommaire}%
\def\mlftitle{Figures}%
\def\mlttitle{Tableaux}%
%%
\def\stctitle{Sommaire}%
\def\slftitle{Figures}%
\def\slttitle{Tableaux}%
⟨/french1⟩
```

13.60 "French2" language: french2.mld

The titles for the "french2" language are taken from the frenchb.dtx file (by Daniel Flipo) in the babel package [38, 39, 57], with some adaptations for the part-level titles[24]. See also section 9.5.6 on page 248, for the subtle distinction between "deuxième" and "seconde". See the mtc-2nd.tex example file in section 4.2 on page 88.

```
⟨*french2⟩
\ProvidesFile{french2.mld}[2006/07/07]
%% French titles for minitoc.sty
%% from frenchb.dtx (babel) Flipo, Daniel
%% ptctitle, plftitle and plttitle modified (JPFD)
\def\mtcFrenchIIpart{\ifcase\value{part}%
\or premi\‘ere partie\or
{\ifmtcsecondpart seconde\else deuxi\‘eme\fi} partie\or
troisi\‘eme partie\or quatri\‘eme partie\or cinqui\‘eme partie\or
sixi\‘eme partie\or septi\‘eme partie\or huiti\‘eme partie\or
neuvi\‘eme partie\or dixi\‘eme partie\or onzi\‘eme partie\or
douzi\‘eme partie\or treizi\‘eme partie\or quatorzi\‘eme partie\or
quinzi\‘eme partie\or seizi\‘eme partie\or dix-septi\‘eme partie\or
dix-huiti\‘eme partie\or dix-neuvi\‘eme partie\or
vingti\‘eme partie\else partie~\Roman{part}\fi}
\def\ptctitle{\ifnum\value{part}<1\relax
  Sommaire \else
  Sommaire de la \mtcFrenchIIpart\fi}
\def\plftitle{\ifnum\value{part}<1\relax
  Liste des figures\else
  Liste des figures de la \mtcFrenchIIpart\fi}
\def\plttitle{\ifnum\value{part}<1\relax
  Liste des tableaux\else
  Liste des tableaux de la \mtcFrenchIIpart}
%%
\def\mtctitle{Sommaire}%
\def\mlftitle{Figures}%
\def\mlttitle{Tableaux}%
%%
\def\stctitle{Sommaire}%
\def\slftitle{Figures}%
\def\slttitle{Tableaux}%
⟨/french2⟩
```

13.61 "Frenchb" language: frenchb.mld

The "frenchb" language is a synonym for the "french" language, so we load the french.mld file. See section 13.58 on page 461.

[24] This is an example of a .mld file needing some support from code in the minitoc package.

```
⟨*frenchb⟩
\ProvidesFile{frenchb.mld}[2003/02/11]
\mtcselectlanguage{french}%
⟨/frenchb⟩
```

13.62 "Frenchle" language: frenchle.mld

The "frenchle" language is a synonym for the "french" language, so we load the french.mld file. See section 13.58 on page 461.

```
⟨*frenchle⟩
\ProvidesFile{frenchle.mld}[2003/02/20]
\mtcselectlanguage{french}%
⟨/frenchle⟩
```

13.63 "Frenchpro" language: frenchpro.mld

The "frenchpro" language is a synonym for the "french" language, so we load the french.mld file. See section 13.58 on page 461.

```
⟨*frenchpro⟩
\ProvidesFile{frenchpro.mld}[2003/02/20]
\mtcselectlanguage{french}%
⟨/frenchpro⟩
```

13.64 "Galician" language: galician.mld

The titles for the "galician" language[25] are taken from the galician.dtx file (by Manuel Carriba) in the babel package [38, 39, 53]:

```
⟨*galician⟩
\ProvidesFile{galician.mld}[2006/01/13]
%% Galician titles for minitoc.sty
%% from galician.dtx (babel)
%% Carriba, Manuel
\def\ptctitle{\'Indice Xeral}%
\def\plftitle{\'Indice de Figuras}%
\def\plttitle{\'Indice de T\'aboas}%
%%
\def\mtctitle{\'Indice Xeral}%
```

[25] Spoken in Galice, in the north-west part of Spain, around Santiago de Compostela.

```
\def\mlftitle{\'Indice de Figuras}%
\def\mlttitle{\'Indice de T\'aboas}%
%%
\def\stctitle{\'Indice Xeral}%
\def\slftitle{\'Indice de Figuras}%
\def\slttitle{\'Indice de T\'aboas}%
⟨/galician⟩
```

13.65 "German" language: `german.mld`

The titles for the "`german`" language are taken from the babel package [38, 39]. See also the section 13.13 on page 439.

```
⟨*german⟩
\ProvidesFile{german.mld}[1999/03/16]
%% German titles for minitoc.sty
\def\ptctitle{Inhaltsangabe}%
\def\plftitle{Figuren}%
\def\plttitle{Tabellen}%
%%
\def\mtctitle{Inhaltsangabe}%
\def\mlftitle{Figuren}%
\def\mlttitle{Tabellen}%
%%
\def\stctitle{Inhaltsangabe}%
\def\slftitle{Figuren}%
\def\slttitle{Tabellen}%
⟨/german⟩
```

13.66 "Germanb" language: `germanb.mld`

The "`germanb`" language is a variant for the "german" language. The titles are taken from the `germanb.dtx` file (by Johannes Braams and Bernd Raichle) in the babel package [38, 39, 72]:

```
⟨*germanb⟩
\ProvidesFile{germanb.mld}[2006/01/13]
%% German titles (variant) for minitoc.sty
%% from germanb.dtx (babel)
%% Braams, Johannes and Raichle, Bernd
\def\ptctitle{Inhaltsverzeichnis}%
\def\plftitle{Abbildungsverzeichnis}%
\def\plttitle{Tabellenverzeichnis}%
%%
\def\mtctitle{Inhaltsverzeichnis}%
\def\mlftitle{Abbildungsverzeichnis}%
\def\mlttitle{Tabellenverzeichnis}%
```

```
%%
\def\stctitle{Inhalt}%
\def\slftitle{Abbildungen}%
\def\slttitle{Tabellen}%
⟨/germanb⟩
```

13.67 "Germanb2" language: germanb2.mld

The "germanb2" language is a variant for the "german" language, with short titles. See also section 13.66 on the preceding page. The titles are taken from the babel package [38, 39]:

```
⟨*germanb2⟩
\ProvidesFile{germanb2.mld}[2005/09/27]
%% German titles (variant) for minitoc.sty
\def\ptctitle{Inhalt}%
\def\plftitle{Abbildungen}%
\def\plttitle{Tabellen}%
%%
\def\mtctitle{Inhalt}%
\def\mlftitle{Abbildungen}%
\def\mlttitle{Tabellenverzeichnis}%
%%
\def\stctitle{Inhalt}%
\def\slftitle{Abbildungen}%
\def\slttitle{Tabellen}%
⟨/germanb2⟩
```

13.68 "Greek" language: greek.mld

The titles for the "greek" language (modern greek) are taken from the greek.dtx file (by Apostolos Syropoulos) in the babel package [38, 39, 80]. Greek fonts are required.

```
⟨*greek⟩
\ProvidesFile{greek.mld}[2006/01/13]
%% Greek titles for minitoc.sty
%% from greek.dtx (babel)
%% Syropoulos, Apostolos
%% Needs greek fonts.
\def\ptctitle{Perieq'omena}%
\def\plftitle{Kat'alogoc Sqhm'atwn}%
\def\plttitle{Kat'alogoc Pin'akwn}%
%%
\def\mtctitle{Perieq'omena}%
\def\mlftitle{Kat'alogoc Sqhm'atwn}%
\def\mlttitle{Kat'alogoc Pin'akwn}%
%%
```

```
\def\stctitle{Perieq'omena}%
\def\slftitle{Kat'alogoc Sqhm'atwn}%
\def\slttitle{Kat'alogoc Pin'akwn}%
⟨/greek⟩
```

13.69 "Greek-mono" language: greek-mono.mld

The titles for the "greek-mono" language[26] are taken from the omega-greek.ldf file (by Alexej M. Kryokov and Dmitry Ivanov) in the Antomega project [150]:

```
⟨*greek – mono⟩
\ProvidesFile{greek-mono.mld}[2005/02/08]
%% from omega-greek.ldf (Antomega project)
%% Needs Omega
%% Alexej M. Kryokov
%% Dmitry Ivanov
%%
\def\ptctitle{\localgreek%
{^^^^03a0^^^^03b5^^^^03c1^^^^03b9^^^^03b5^^^^03c7^^^^03cc^^^^03bc%
 ^^^^03b5^^^^03bd^^^^03b1}}%
\def\plftitle{\localgreek%
{^^^^039a^^^^03b1^^^^03c4^^^^03ac^^^^03bb^^^^03bf^^^^03b3^^^^03bf%
 ^^^^03c2 ^^^^03c3^^^^03c7^^^^03b7^^^^03bc^^^^03ac^^^^03c4^^^^03c9%
 ^^^^03bd}}%
\def\plftitle{\localgreek%
{^^^^039a^^^^03b1^^^^03c4^^^^03ac^^^^03bb^^^^03bf^^^^03b3^^^^03bf%
 ^^^^03c2 ^^^^03c0^^^^03b9^^^^03bd^^^^03ac^^^^03ba^^^^03c9^^^^03bd}}%
%%
\def\mtctitle{\localgreek%
{^^^^03a0^^^^03b5^^^^03c1^^^^03b9^^^^03b5^^^^03c7^^^^03cc^^^^03bc%
 ^^^^03b5^^^^03bd^^^^03b1}}%
\def\mlftitle%{\localgreek%
{^^^^039a^^^^03b1^^^^03c4^^^^03ac^^^^03bb^^^^03bf^^^^03b3^^^^03bf%
 ^^^^03c2 ^^^^03c3^^^^03c7^^^^03b7^^^^03bc^^^^03ac^^^^03c4^^^^03c9%
 ^^^^03bd}}%
\def\mlftitle{\localgreek%
{^^^^039a^^^^03b1^^^^03c4^^^^03ac^^^^03bb^^^^03bf^^^^03b3^^^^03bf%
 ^^^^03c2 ^^^^03c0^^^^03b9^^^^03bd^^^^03ac^^^^03ba^^^^03c9^^^^03bd}}%
%%
\def\stctitle{\localgreek%
{^^^^03a0^^^^03b5^^^^03c1^^^^03b9^^^^03b5^^^^03c7^^^^03cc^^^^03bc%
 ^^^^03b5^^^^03bd^^^^03b1}}%
\def\slftitle{\localgreek%
{^^^^039a^^^^03b1^^^^03c4^^^^03ac^^^^03bb^^^^03bf^^^^03b3^^^^03bf%
 ^^^^03c2 ^^^^03c3^^^^03c7^^^^03b7^^^^03bc^^^^03ac^^^^03c4^^^^03c9%
 ^^^^03bd}}%
\def\slftitle{\localgreek%
{^^^^039a^^^^03b1^^^^03c4^^^^03ac^^^^03bb^^^^03bf^^^^03b3^^^^03bf%
```

[26] Monotonic greek, from a recent (1982) but strongly contested – and contestable – reform of the greek language.

```
^^^^03c2 ^^^^03c0^^^^03b9^^^^03bd^^^^03ac^^^^03ba^^^^03c9^^^^03bd}}%
⟨/greek – mono⟩
```

13.70 "Greek-polydemo" language: greek-polydemo.mld

The titles for the "greek-polydemo" language [27] are taken from the file omega-greek.ldf (by Alexej M. Kryokov and Dmitry Ivanov) in the Antomega project [150]:

```
⟨*greek – polydemo⟩
\ProvidesFile{greek-polydemo.mld}[2005/02/08]
%% from omega-greek.ldf (Antomega project)
%% Needs Omega
%% Alexej M. Kryokov
%% Dmitry Ivanov
%%
\def\ptctitle{\localgreek%
{^^^^03a0^^^^03b5^^^^03c1^^^^03b9^^^^03b5^^^^03c7^^^^1f79^^^^03bc%
 ^^^^03b5^^^^03bd^^^^03b1}}%
\def\plftitle{\localgreek%
{^^^^039a^^^^03b1^^^^03c4^^^^1f71^^^^03bb^^^^03bf^^^^03b3^^^^03bf%
 ^^^^03c2 ^^^^03c3^^^^03c7^^^^03b7^^^^03bc^^^^1f71^^^^03c4^^^^03c9%
 ^^^^03bd}}%
\def\plttitle{\localgreek%
{^^^^039a^^^^03b1^^^^03c4^^^^1f71^^^^03bb^^^^03bf^^^^03b3^^^^03bf%
 ^^^^03c2 ^^^^03c0^^^^03b9^^^^03bd^^^^1f71^^^^03ba^^^^03c9^^^^03bd}}%
%%
\def\mtctitle{\localgreek%
{^^^^03a0^^^^03b5^^^^03c1^^^^03b9^^^^03b5^^^^03c7^^^^1f79^^^^03bc%
 ^^^^03b5^^^^03bd^^^^03b1}}%
\def\mlftitle{\localgreek%
{^^^^039a^^^^03b1^^^^03c4^^^^1f71^^^^03bb^^^^03bf^^^^03b3^^^^03bf%
 ^^^^03c2 ^^^^03c3^^^^03c7^^^^03b7^^^^03bc^^^^1f71^^^^03c4^^^^03c9%
 ^^^^03bd}}%
\def\mlttitle{\localgreek%
{^^^^039a^^^^03b1^^^^03c4^^^^1f71^^^^03bb^^^^03bf^^^^03b3^^^^03bf%
 ^^^^03c2 ^^^^03c0^^^^03b9^^^^03bd^^^^1f71^^^^03ba^^^^03c9^^^^03bd}}%
%%
\def\stctitle{\localgreek%
{^^^^03a0^^^^03b5^^^^03c1^^^^03b9^^^^03b5^^^^03c7^^^^1f79^^^^03bc%
 ^^^^03b5^^^^03bd^^^^03b1}}%
\def\slftitle{\localgreek%
{^^^^039a^^^^03b1^^^^03c4^^^^1f71^^^^03bb^^^^03bf^^^^03b3^^^^03bf%
 ^^^^03c2 ^^^^03c3^^^^03c7^^^^03b7^^^^03bc^^^^1f71^^^^03c4^^^^03c9%
 ^^^^03bd}}%
\def\slttitle{\localgreek%
{^^^^039a^^^^03b1^^^^03c4^^^^1f71^^^^03bb^^^^03bf^^^^03b3^^^^03bf%
 ^^^^03c2 ^^^^03c0^^^^03b9^^^^03bd^^^^1f71^^^^03ba^^^^03c9^^^^03bd}}%
⟨/greek – polydemo⟩
```

[27] Polytonic demotic (popular) greek, for classical greek.

13.71 "Greek-polykatha" language: greek-polykatha.mld

The titles for the "greek-polykatha" language [28] are taken from the omega-greek.ldf (by Alexej M. Kryokov and Dmitry Ivanov) in the Antomega project [150]:

```
⟨*greek – polykatha⟩
\ProvidesFile{greek-polykatha.mld}[2005/02/08]
%% from omega-greek.ldf (Antomega project)
%% Needs Omega
%% Alexej M. Kryokov
%% Dmitry Ivanov
%%
\def\ptctitle{\localgreek%
{^^^^03a0^^^^03b5^^^^03c1^^^^03b9^^^^03b5^^^^03c7^^^^1f79^^^^03bc%
 ^^^^03b5^^^^03bd^^^^03b1}}%
\def\plftitle{\localgreek%
{^^^^039a^^^^03b1^^^^03c4^^^^1f71^^^^03bb^^^^03bf^^^^03b3^^^^03bf%
 ^^^^03c2 ^^^^03c3^^^^03c7^^^^03b7^^^^03bc^^^^1f71^^^^03c4^^^^03c9%
 ^^^^03bd}}%
\def\plttitle{\localgreek%
{^^^^039a^^^^03b1^^^^03c4^^^^1f71^^^^03bb^^^^03bf^^^^03b3^^^^03bf%
 ^^^^03c2 ^^^^03c0^^^^03b9^^^^03bd^^^^1f71^^^^03ba^^^^03c9^^^^03bd}}%
%%
\def\mtctitle{\localgreek%
{^^^^03a0^^^^03b5^^^^03c1^^^^03b9^^^^03b5^^^^03c7^^^^1f79^^^^03bc%
 ^^^^03b5^^^^03bd^^^^03b1}}%
\def\mlftitle{\localgreek%
{^^^^039a^^^^03b1^^^^03c4^^^^1f71^^^^03bb^^^^03bf^^^^03b3^^^^03bf%
 ^^^^03c2 ^^^^03c3^^^^03c7^^^^03b7^^^^03bc^^^^1f71^^^^03c4^^^^03c9%
 ^^^^03bd}}%
\def\mlttitle{\localgreek%
{^^^^039a^^^^03b1^^^^03c4^^^^1f71^^^^03bb^^^^03bf^^^^03b3^^^^03bf%
 ^^^^03c2 ^^^^03c0^^^^03b9^^^^03bd^^^^1f71^^^^03ba^^^^03c9^^^^03bd}}%
%%
\def\stctitle{\localgreek%
{^^^^03a0^^^^03b5^^^^03c1^^^^03b9^^^^03b5^^^^03c7^^^^1f79^^^^03bc%
 ^^^^03b5^^^^03bd^^^^03b1}}%
\def\slftitle{\localgreek%
{^^^^039a^^^^03b1^^^^03c4^^^^1f71^^^^03bb^^^^03bf^^^^03b3^^^^03bf%
 ^^^^03c2 ^^^^03c3^^^^03c7^^^^03b7^^^^03bc^^^^1f71^^^^03c4^^^^03c9%
 ^^^^03bd}}%
\def\slttitle{\localgreek%
{^^^^039a^^^^03b1^^^^03c4^^^^1f71^^^^03bb^^^^03bf^^^^03b3^^^^03bf%
 ^^^^03c2 ^^^^03c0^^^^03b9^^^^03bd^^^^1f71^^^^03ba^^^^03c9^^^^03bd}}%
⟨/greek – polykatha⟩
```

[28] Polytonic greek, « kathaverousa » (purified) style, a form of the Greek language created during the early XIX-th century by Adamantios Korais, 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.

13.72 "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. The titles are taken from the guarani.ldf file by Javier Bezos [32]. A special input encoding (win-gn.def) is needed. These files are available on the CTAN archives.

```
⟨*guarani⟩
\ProvidesFile{guarani.mld}[2005/08/26]
%% Guarani titles for minitoc.sty
%% from guarani.ldf by Javier Bezos.
%% Input encoding win-gn.def is needed.
%%
\def\ptctitle{\'Indice general}%
\def\plftitle{\'Indice de figuras}%
\def\plttitle{\'Indice de cuadros}%
%%
\def\mtctitle{\'Indice general}%
\def\mlftitle{\'Indice de figuras}%
\def\mlttitle{\'Indice de cuadros}%
%%
\def\stctitle{\'Indice general}%
\def\slftitle{\'Indice de figuras}%
\def\slttitle{\'Indice de cuadros}%
⟨/guarani⟩
```

13.73 "Hangul1" language: hangul1.mld and hangul1.mlo

The Korean language was originally written using the Chinese characters; it is now mainly written in Hangûl, the Korean writing system, optionally incorporating Hanja to write Sino-Korean words [241]. See [124, page 150] and [205].

The titles for the "hangul1" language (korean in hangûl script, first variant) are taken from the file hangul.cap of the CJK system [167, 168] (by Werner Lemberg). Special fonts are needed, of course. See also sections 13.74 on the next page, 13.75 on the following page, 13.76 on page 472, 13.77 on page 473, 13.78 on page 473, 13.79 on page 474, and 13.80 on page 474.

The titles for the "hangul1" language contain characters that cannot be easily generated, hence we load hangul1.mlo.

```
⟨*hangul1⟩
\ProvidesFile{hangul1.mld}[2005/01/28]
%% From the file hangul.cap of the CJK package
%% for using Asian logographs (Chinese/Japanese/Korean) with LaTeX2e
%% created by Werner Lemberg <wl@gnu.org>
%%
```

```
%% Version 4.5.2 (28-Mar-2003)
%% Hangul captions
%% character set: KS X 1001:1992 (=KS C 5601-1992), encoding: EUC (=Wansung)
%%
\mtcloadmlo{hangul1}%
⟨/hangul1⟩
```

13.74 "Hangul2" language: hangul2.mld and hangul2.mlo

The titles for the "hangul2" language (korean in hangûl script, second variant) are taken from the file hangul.cpx of the CJK system [167, 168] (by Werner Lemberg). Special fonts are needed, of course. See also sections 13.73 on the preceding page, 13.75, 13.76 on the following page, 13.77 on page 473, 13.78 on page 473, 13.79 on page 474, and 13.80 on page 474.

The titles for the "hangul2" language contain characters that cannot be easily generated, hence we load hangul2.mlo.

```
⟨*hangul2⟩
\ProvidesFile{hangul2.mld}[2005/01/28]
%% From the file hangul.cpx of the CJK package
%% for using Asian logographs (Chinese/Japanese/Korean) with LaTeX2e
%% created by Werner Lemberg <wl@gnu.org>
%%
%% Version 4.5.2 (28-Mar-2003)
%% Hangul captions
%% character set: KS X 1001:1992 (=KS C 5601-1992),
%% encoding: EUC (=Wansung), preprocessed
%%
\mtcloadmlo{hangul2}%
⟨/hangul2⟩
```

13.75 "Hangul3" language: hangul3.mld and hangul3.mlo

The titles for the "hangul3" language (korean in hangûl script, third variant) are taken from the file hangul2.cap of the CJK system [167, 168] (by Werner Lemberg). Special fonts are needed, of course. See also sections 13.73 on the page before, 13.74, 13.76 on the next page, 13.77 on page 473, 13.78 on page 473, 13.79 on page 474, and 13.80 on page 474.

The titles for the "hangul3" language contain characters that cannot be easily generated, hence we load hangul3.mlo.

```
⟨*hangul3⟩
\ProvidesFile{hangul3.mld}[2005/01/28]
%% From the file hangul2.cap of the CJK package
%% for using Asian logographs (Chinese/Japanese/Korean) with LaTeX2e
%% created by Werner Lemberg <wl@gnu.org>
%%
%% Version 4.5.2 (28-Mar-2003)
%% Hangul captions set 2
%% character set: KS X 1001:1992 (=KS C 5601-1992), encoding: EUC (=Wansung)
%%
\mtcloadmlo{hangul3}%
⟨/hangul3⟩
```

13.76 "Hangul4" language: hangul4.mld and hangul4.mlo

The titles for the "hangul4" language (korean in hangûl script, fourth variant) are taken from the file hangul2.cpx of the CJK system [167, 168] (by Werner Lemberg). Special fonts are needed, of course. See also sections 13.73 on page 470, 13.74 on the page before, 13.75 on the preceding page, 13.77 on the next page, 13.78 on the following page, 13.79 on page 474, and 13.80 on page 474.

The titles for the "hangul4" language contain characters that cannot be easily generated, hence we load hangul4.mlo.

```
⟨*hangul4⟩
\ProvidesFile{hangul4.mld}[2005/01/28]
%% From the file hangul2.cpx of the CJK package
%% for using Asian logographs (Chinese/Japanese/Korean) with LaTeX2e
%% created by Werner Lemberg <wl@gnu.org>
%%
%% Version 4.5.2 (28-Mar-2003)
%% Hangul captions set 2, character set: KS X 1001:1992 (=KS C 5601-1992),
%% encoding: EUC (=Wansung), preprocessed
%%
\mtcloadmlo{hangul4}%
⟨/hangul4⟩
```

13.77 "Hangul-u8" language: hangul-u8.mld and hangul-u8.mlo

The titles for the "hangul-u8" language (korean in hangûl script, for *Lambda* **Λ**) are taken from the file u8hangul.tex of the HLaTeX system [146, in korean] by Koaunghi Un. Special fonts are needed, of course. Input encoding is UTF-8.

See also sections 13.73 on page 470, 13.74 on page 471, 13.75 on page 471, 13.76 on the preceding page, 13.78, 13.79 on the following page, and 13.80 on the next page. See [124, page 150] and [205].

The titles for the "hangul-u8" language contain characters that cannot be easily generated, hence we load hangul-u8.mlo.

```
⟨*hangul – u8⟩
\ProvidesFile{hangul-u8.mld}[2006/02/21]
%% From the file u8hangul.tex of the HLaTeX package
%% by Koaunghi Un (koaunghi@kornet.net)
%%
%% Hangul captions for Lambda
%%
\mtcloadmlo{hangul-u8}%
⟨/hangul – u8⟩
```

13.78 "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 [167, 168] (by Werner Lemberg). Special fonts are needed, of course. See also sections 13.73 on page 470, 13.74 on page 471, 13.75 on page 471, 13.76 on the page before, 13.77, 13.79 on the next page, and 13.80 on the following page.

The titles for the "hanja1" language contain characters that cannot be easily generated, hence we load hanja1.mlo.

```
⟨*hanja1⟩
\ProvidesFile{hanja1.mld}[2005/01/28]
%% From the file hanja.cpx of the CJK package
%% for using Asian logographs (Chinese/Japanese/Korean) with LaTeX2e
%% created by Werner Lemberg <wl@gnu.org>
%%
%% Version 4.5.2 (28-Mar-2003)
%% Hanja captions, character set: KS X 1001:1992 (=KS C 5601-1992),
%% encoding: EUC (=Wansung), preprocessed
%%
\mtcloadmlo{hanja1}%
```

⟨/hanja1⟩

13.79 "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 [167, 168] (by Werner Lemberg). Special fonts are needed, of course. See also sections 13.73 on page 470, 13.74 on page 471, 13.75 on page 471, 13.76 on page 472, 13.77 on the page before, 13.78 on the preceding page, and 13.80.

The titles for the "hanja2" language contain characters that cannot be easily generated, hence we load hanja2.mlo.

```
⟨*hanja2⟩
\ProvidesFile{hanja2.mld}[2005/01/28]
%% From the file hanja.cap of the CJK package
%% for using Asian logographs (Chinese/Japanese/Korean) with LaTeX2e
%% created by Werner Lemberg <a7971428@unet.univie.ac.at>
%%
%% Version 4.1.3 (20-Jun-1997)
%% Hanja captions, character set: KS X 1001:1992 (=KS C 5601-1992),
%% encoding: EUC (=Wansung)
%%
\mtcloadmlo{hanja2}%
⟨/hanja2⟩
```

13.80 "Hanja-u8" language: hanja-u8.mld and hanja-u8.mlo

The titles for the "hanja-u8" language (korean in hanja script, for *Lambda* **Λ**) are taken from the file u8hanja.tex of the HLaTeX system [146, in korean] by Koaunghi Un. Special fonts are needed, of course. Input encoding is UTF-8. See also section 13.73 on page 470, 13.74 on page 471, 13.75 on page 471, 13.76 on page 472, 13.77 on the preceding page, 13.78 on the page before, and 13.79. See [124, page 150] and [205].

The titles for the "hanja-u8" language contain characters that cannot be easily generated, hence we load hanja-u8.mlo.

```
⟨*hanja – u8⟩
\ProvidesFile{hanja-u8.mld}[2006/02/21]
%% From the file hanja-u8.tex of the HLaTeX package
%% by Koaunghi Un (koaunghi@kornet.net)
%%
%% Hanja captions for Lambda
```

```
%%
\mtcloadmlo{hanja-u8}%
⟨/hanja – u8⟩
```

13.81 "Hebrew" language: hebrew.mld

The titles for the "hebrew" language are taken from the ArabTEX package [154, 155], which should be used, with the associated fonts. See also section 13.82.

```
⟨*hebrew⟩
\ProvidesFile{hebrew.mld}[2001/02/28]
%% Hebrew titles for minitoc.sty
%% Need hebrew fonts (see arabtex documentation)
\def\ptctitle{\tav\vav\kaf\finalnun\
              \ayin\nun\yod\nun\yod\finalmem}%
\def\plftitle{\resh\shin\yod\mem\tav\
              \alef\yod\vav\resh\yod\finalmem}%
\def\plttitle{\resh\shin\yod\mem\tav\
              \tet\bet\lamed\alef\vav\tav}%
%%
\def\mtctitle{\tav\vav\kaf\finalnun\
              \ayin\nun\yod\nun\yod\finalmem}%
\def\mlftitle{\resh\shin\yod\mem\tav\
              \alef\yod\vav\resh\yod\finalmem}%
\def\mlttitle{\resh\shin\yod\mem\tav\
              \tet\bet\lamed\alef\vav\tav}%
%%
\def\stctitle{\tav\vav\kaf\finalnun\
              \ayin\nun\yod\nun\yod\finalmem}%
\def\slftitle{\resh\shin\yod\mem\tav\
              \alef\yod\vav\resh\yod\finalmem}%
\def\slttitle{\resh\shin\yod\mem\tav\
              \tet\bet\lamed\alef\vav\tav}%
⟨/hebrew⟩
```

13.82 "Hebrew2" language: hebrew2.mld

The titles for the "hebrew2" language are taken from the file hebrew.dtx (by Boris Lavva) in the babel package [38, 39, 68], which should be used, with the associated fonts and encodings. See also section 13.81.

```
⟨*hebrew2⟩
\ProvidesFile{hebrew2.mld}[2006/01/11]
%% From hebrew.dtx in the Babel package.
%% Boris Lavva (lavva@tx.technion.ac.il)
```

```
%% Need hebrew fonts.
  \def\ptctitle{\@ensure@R{%
    \hebtav\hebvav\hebkaf\hebfinalnun\ %
    \hebayin\hebnun\hebyod\hebyod\hebnun\hebyod\hebfinalmem}}%
  \def\plftitle{\@ensure@R{%
    \hebresh\hebshin\hebyod\hebmem\hebtav\ %
    \hebalef\hebyod\hebvav\hebresh\hebyod\hebfinalmem}}%
  \def\plttitle{\@ensure@R{%
    \hebresh\hebshin\hebyod\hebmem\hebtav\
    \hebtet\hebbet\heblamed\hebalef\hebvav\hebtav}}%
%%
  \def\mtctitle{\@ensure@R{%
    \hebtav\hebvav\hebkaf\hebfinalnun\ %
    \hebayin\hebnun\hebyod\hebyod\hebnun\hebyod\hebfinalmem}}%
  \def\mlftitle{\@ensure@R{%
    \hebresh\hebshin\hebyod\hebmem\hebtav\ %
    \hebalef\hebyod\hebvav\hebresh\hebyod\hebfinalmem}}%
  \def\mlttitle{\@ensure@R{%
    \hebresh\hebshin\hebyod\hebmem\hebtav\
    \hebtet\hebbet\heblamed\hebalef\hebvav\hebtav}}%
%%
  \def\stctitle{\@ensure@R{%
    \hebtav\hebvav\hebkaf\hebfinalnun\ %
    \hebayin\hebnun\hebyod\hebyod\hebnun\hebyod\hebfinalmem}}%
  \def\slftitle{\@ensure@R{%
    \hebresh\hebshin\hebyod\hebmem\hebtav\ %
    \hebalef\hebyod\hebvav\hebresh\hebyod\hebfinalmem}}%
  \def\slttitle{\@ensure@R{%
    \hebresh\hebshin\hebyod\hebmem\hebtav\
    \hebtet\hebbet\heblamed\hebalef\hebvav\hebtav}}%
⟨/hebrew2⟩
```

13.83 “Hindi” language: hindi.mld

The “hindi” language is just like “devanagari”, so we just load devanagari.mld (see section 13.41 on page 452):

```
⟨*hindi⟩
\ProvidesFile{hindi.mld}[2006/08/24]
\mtcselectlanguage{devanagari}%
⟨/hindi⟩
```

13.84 “Hindi-modern” language: hindi-modern.mld

The titles for the “hindi-modern” language are taken from the captions.dn file (by Anshuman Pandey, CV Radhakrishnan, Zdeněk Wagner, John Smith, Kevin Carmody,

Richard MAHONEY and Dominik WUJASTYK) in the Devanāgarī package [204] (Devanāgarī) after conversion. See also section 13.41 on page 452. Specific fonts are required. The home page of the package is http://devnag.ramovar.org.

```
⟨*hindi – modern⟩
\ProvidesFile{hindi-modern.mld}[2006/08/29]
%% Hindi modern titles for minitoc.sty
%% from captions.dn in ``Devanagari for TeX'' by
%% Pandey, Anshuman
%% Radhakrishnan, CV
%% Wagner, Zden\v{e}k
%% Smith, John
%% Carmody, Kevin
%% Mahoney, Richard
%% Wujastyk, Dominik
\def\ptctitle{{\dn Evqy{\rs -\re}\8{s}cF}}%
\def\plftitle{{\dn Ec/o{\qva} kF \8{s}cF}}%
\def\plttitle{{\dn tAElkAao\2 kF \8{s}cF}}%
%%
\def\mtctitle{{\dn Evqy{\rs -\re}\8{s}cF}}%
\def\mlftitle{{\dn Ec/o{\qva} kF \8{s}cF}}%
\def\mlttitle{{\dn tAElkAao\2 kF \8{s}cF}}%
%%
\def\stctitle{{\dn Evqy{\rs -\re}\8{s}cF}}%
\def\slftitle{{\dn Ec/o{\qva} kF \8{s}cF}}%
\def\slttitle{{\dn tAElkAao\2 kF \8{s}cF}}%
⟨/hindi – modern⟩
```

13.85 "Hungarian" language: hungarian.mld

The "hungarian" language is a synonym of the "magyar" language, so we load magyar.mld. See section 13.108 on page 488.

```
⟨*hungarian⟩
\ProvidesFile{hungarian.mld}[2004/12/14]
\mtcselectlanguage{magyar}%
⟨/hungarian⟩
```

13.86 "Icelandic" language: icelandic.mld

The titles for the "icelandic" language are taken from the icelandic.dtx file (by Einar ÁRNASON) in the babel package [38, 39, 46]:

```
⟨*icelandic⟩
\ProvidesFile{icelandic.mld}[2006/01/13]
```

```
%% Icelandic titles for minitoc.sty
%% from icelandic.dtx (babel)
%% Árnason, Einar
%% need inputenc with 8-bits encoding
\def\ptctitle{Efnisyfirlit}%
\def\plftitle{Myndaskrá}%
\def\plttitle{Töfluskrá}%
%%
\def\mtctitle{Efnisyfirlit}%
\def\mlftitle{Myndaskrá}%
\def\mlttitle{Töfluskrá}%
%%
\def\stctitle{Efnisyfirlit}%
\def\slftitle{Myndaskrá}%
\def\slttitle{Töfluskrá}%
⟨/icelandic⟩
```

13.87 "Indon" language: indon.mld

The "indon" language is just like "bahasai", so we just load bahasai.mld (see section 13.15 on page 440):

```
⟨*indon⟩
\ProvidesFile{indon.mld}[2006/01/13]
\mtcselectlanguage{bahasai}%
⟨/indon⟩
```

13.88 "Indonesian" language: indonesian.mld

The "indonesian" language is just like "bahasai", so we just load bahasai.mld (see section 13.15 on page 440):

```
⟨*indonesian⟩
\ProvidesFile{indonesian.mld}[2006/01/13]
\mtcselectlanguage{bahasai}%
⟨/indonesian⟩
```

13.89 "Interlingua" language: interlingua.mld

The titles for the "interlingua" language are taken from the interlingua.dtx file (by Peter Kleiweg) in the babel package [38, 39, 63]. Interlingua is an auxiliary language, built from the common vocabulary of Spanish/Portuguese, English, Italian and French, with some

normalisation of spelling. The grammar is very easy, more similar to English's than to neolatin languages. The site http://www.interlingua.com is mostly written in interlingua (as is http://interlingua.altervista.org), in case you want to read some sample of it. You can have a look at the grammar at http://www.geocities.com/linguablau

```
⟨*interlingua⟩
\ProvidesFile{interlingua.mld}[2006/01/13]
%% Interlingua titles for minitoc.sty
%% from interlingua.dtx (babel)
%% Kleiweg, Peter
\def\ptctitle{Contento}%
\def\plftitle{Lista de Figuras}%
\def\plttitle{Lista de Tabellas}%
%%
\def\mtctitle{Contento}%
\def\mlftitle{Figuras}%
\def\mlttitle{Tabellas}%
%%
\def\stctitle{Contento}%
\def\slftitle{Figuras}%
\def\slttitle{Tabellas}%
⟨/interlingua⟩
```

13.90 "Irish" language: irish.mld

The titles for the "irish" language come from the irish.dtx file (by Johannes Braams, Marion Gunn and Fraser Grant) in the babel package [38, 39, 42]:

```
⟨*irish⟩
\ProvidesFile{irish.mld}[2006/02/28]
%% Irish titles for minitoc.sty
%% from irish.dtx (babel)
%% Braams, Johannes and Gunn, Marion and Grant, Fraser
\def\ptctitle{Cl\'ar \'Abhair}%
\def\plftitle{L\'ear\'aid\'{\i}}%
\def\plttitle{T\'abla\'{\i}}%
%%
\def\mtctitle{Cl\'ar \'Abhair}%
\def\mlftitle{L\'ear\'aid\'{\i}}%
\def\mlttitle{T\'abla\'{\i}}%
%%
\def\stctitle{Cl\'ar \'Abhair}%
\def\slftitle{L\'ear\'aid\'{\i}}%
\def\slttitle{T\'abla\'{\i}}%
⟨/irish⟩
```

13.91 "Italian" language: italian.mld

The titles for the "italian" language come from the file italian.dtx (by Maurizio Codogno and Claudio Beccari) in the babel package [38, 39, 55]. See also section 13.92.

```
⟨*italian⟩
\ProvidesFile{italian.mld}[2006/01/13]
%% Italian titles for minitoc.sty
%% from italian.dtx (babel)
%% Maurizio Codogno, (mau@beatles.cselt.stet.it)
%% and Claudio Beccari, (beccari@polito.it)
\def\ptctitle{Indice}%
\def\plftitle{Elenco delle figure}%
\def\plttitle{Elenco delle tabelle}%
%%
\def\mtctitle{Indice}%
\def\mlftitle{Elenco delle figure}%
\def\mlttitle{Elenco delle tabelle}%
%%
\def\stctitle{Indice}%
\def\slftitle{Elenco delle figure}%
\def\slttitle{Elenco delle tabelle}%
⟨/italian⟩
```

13.92 "Italian2" language: italian2.mld

The titles for the "italian2" language are the same as for the "italian" language, except at the part level ("Contenuto"). See also section 13.91.

```
⟨*italian2⟩
\ProvidesFile{italian2.mld}[2006/01/13]
%% Italian titles for minitoc.sty. Variant.
%% from italian.dtx (babel)
%% Maurizio Codogno, (mau@beatles.cselt.stet.it)
%% and Claudio Beccari, (beccari@polito.it)
\def\ptctitle{Contenuto}%
\def\plftitle{Elenco delle figure}%
\def\plttitle{Elenco delle tabelle}%
%%
\def\mtctitle{Contenuto}%
\def\mlftitle{Elenco delle figure}%
\def\mlttitle{Elenco delle tabelle}%
%%
\def\stctitle{Contenuto}%
\def\slftitle{Elenco delle figure}%
\def\slttitle{Elenco delle tabelle}%
⟨/italian2⟩
```

13.93 "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[29]. See also sections 13.94, 13.95 on the next page, 13.96 on the following page, 13.97 on page 483, and 13.98 on page 483. The titles for the "japanese" language contain characters that cannot be easily generated, hence we load japanese.mlo.

```
⟨*japanese⟩
\ProvidesFile{japanese.mld}[2006/01/13]
%% Japanese titles for minitoc.sty
%% Needs japanese fonts (CJK) and special input encoding.
%% From Kumazawa Toshiki
%% kumazawa@biwako.shiga-u.ac.jp
%% http://www.biwako.shiga-u.ac.jp/sensei/kumazawa/tex/minitoc.html
\mtcloadmlo{japanese}%
⟨/japanese⟩
```

13.94 "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 [167, 168] (by Werner LEMBERG). Special fonts are needed, of course. See also sections 13.93, 13.95 on the following page, 13.96 on the next page, 13.97 on page 483, and 13.98 on page 483. The titles for the "japanese2" language contain characters that cannot be easily generated, hence we load japanese2.mlo.

```
⟨*japanese2⟩
\ProvidesFile{japanese2.mld}[2006/01/13]
%% From the file JIS.cap of the CJK package
%% for using Asian logographs (Chinese/Japanese/Korean) with LaTeX2e
%% created by Werner Lemberg <wl@gnu.org>
%%
%% Version 4.5.2 (28-Mar-2003)
%% Japanese captions, character set: JIS X 0208:1997 (or JIS X 0208-1990),
%% encoding: EUC
%%
\mtcloadmlo{japanese2}%
⟨/japanese2⟩
```

[29] http://www.biwako.shiga-u.ac.jp/sensei/kumazawa/tex/minitoc.html

13.95 "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 [167, 168] (by Werner Lemberg). Special fonts are needed, of course. See also sections 13.93 on the preceding page, 13.94 on the page before, 13.96, 13.97 on the next page, and 13.98 on the following page. The titles for the "japanese3" language contain characters that cannot be easily generated, hence we load japanese3.mlo.

```
⟨*japanese3⟩
\ProvidesFile{japanese3.mld}[2006/01/13]
%% From the file JIS.cpx of the CJK package
%% for using Asian logographs (Chinese/Japanese/Korean) with LaTeX2e
%% created by Werner Lemberg <wl@gnu.org>
%%
%% Version 4.5.2 (28-Mar-2003)
%% Japanese captions, character set: JIS X 0208:1997 (or JIS X 0208-1990)
%% encoding: EUC, preprocessed
%%
\mtcloadmlo{japanese3}%
⟨/japanese3⟩
```

13.96 "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 [167, 168] (by Werner Lemberg). Special fonts are needed, of course. See also sections 13.93 on the page before, 13.94 on the preceding page, 13.95, 13.97 on the following page, and 13.97 on the next page. The titles for the "japanese4" language contain characters that cannot be easily generated, hence we load japanese4.mlo.

```
⟨*japanese4⟩
\ProvidesFile{japanese4.mld}[2006/01/13]
%% From the file SJIS.cap of the CJK package
%% for using Asian logographs (Chinese/Japanese/Korean) with LaTeX2e
%% created by Werner Lemberg <wl@gnu.org>
%%
%% Version 4.5.2 (28-Mar-2003)
%% Japanese captions
%% character set: JIS X 0208:1997 (or JIS X 0208-1990), encoding: SJIS
%%
\mtcloadmlo{japanese4}%
⟨/japanese4⟩
```

13.97 "Japanese5" language: japanese5.mld and japanese5.mlo

The titles for the "japanese5" (japanese, fifth variant) language are taken from file SJIS.cpx of the CJK system [167, 168] (by Werner Lemberg). Special fonts are needed, of course. See also sections 13.93 on page 481, 13.94 on page 481, 13.95 on the preceding page, 13.96 on the page before, and 13.98. The titles for the "japanese5" language contain characters that cannot be easily generated, hence we load japanese5.mlo.

```
⟨*japanese5⟩
\ProvidesFile{japanese5.mld}[2006/01/13]
%% From the file SJIS.cpx of the CJK package
%% for using Asian logographs (Chinese/Japanese/Korean) with LaTeX2e
%% created by Werner Lemberg <wl@gnu.org>
%%
%% Version 4.5.2 (28-Mar-2003)
%% Japanese captions
%% character set: JIS X 0208:1997 (or JIS X 0208-1990),
%% encoding: SJIS, preprocessed
%%
\mtcloadmlo{japanese5}%
⟨/japanese5⟩
```

13.98 "Japanese6" language: japanese6.mld and japanese6.mlo

The titles for the "japanese6" (japanese, sixth variant) language have been found (by a Google search) on the Web site of Professor Toshiki Kumazawa[30]. See also sections 13.93 on page 481, 13.94 on page 481, 13.95 on the page before, 13.96 on the preceding page, and 13.97. The titles for the "japanese6" language contain characters that cannot be easily generated, hence we load japanese6.mlo.

```
⟨*japanese6⟩
\ProvidesFile{japanese6.mld}[2006/10/31]
%% Japanese6 titles for minitoc.sty
%% Needs japanese fonts (CJK) and special input encoding.
%% From Kumazawa Toshiki
%% kumazawa@biwako.shiga-u.ac.jp
%% http://www.biwako.shiga-u.ac.jp/sensei/kumazawa/tex/minitoc.html
\mtcloadmlo{japanese6}%
⟨/japanese6⟩
```

[30] http://www.biwako.shiga-u.ac.jp/sensei/kumazawa/tex/minitoc.html

13.99 “Kannada” language: kannada.mld

The Kannada (“kannada”) (or Kannara) language is a dravidian language spoken in the Karnataka state of India. The titles are taken from the kanlel.sty package file from the Kannadatex project[31] by C.S. Yogananda. Specific fonts are required.

```
⟨*kannada⟩
\ProvidesFile{kannada.mld}[2006/02/14]
%% from kanlel.sty of Kannadatex (C.S. Yogananda yoga@math.iisc.ernet.in)
%%
\def\ptctitle{pariviDi}
\def\plftitle{citarxgaLa paTiTx}
\def\plftitle{koVSaTxkagaLa paTiTx}
%%
\def\mtctitle{pariviDi}
\def\mlftitle{citarxgaLa paTiTx}
\def\mlftitle{koVSaTxkagaLa paTiTx}
%%
\def\stctitle{pariviDi}
\def\slftitle{citarxgaLa paTiTx}
\def\slftitle{koVSaTxkagaLa paTiTx}
⟨/kannada⟩
```

13.100 “Khalka” language: khalkha.mld

“khalkha” is a synomym for “xalx”, so we just load xalx.mld (see sections 13.173 on page 520, 13.174 on page 520, and 13.175 on page 521):

```
⟨*khalkha⟩
\ProvidesFile{khalkha.mld}[2005/11/16]
\mtcselectlanguage{xalx}%
⟨/khalkha⟩
```

13.101 “Latin” language: latin.mld

The titles for the “latin” language are taken from the latin.dtx file (by Claudio Beccari and Krzysztof Konrad Żelechowski) in the babel package [38, 39, 48]. See also section 13.102 on the next page.

```
⟨*latin⟩
\ProvidesFile{latin.mld}[2006/01/13]
%% Latin titles for minitoc.sty
```

[31] http://Sarovar.org/projects/kannadatex

```
%% from latin.dtx (babel)
%% Beccari, Claudio
\def\ptctitle{Index}%
\def\plftitle{Conspectus descriptionum}%
\def\plttitle{Conspectus tabularum}%
%%
\def\mtctitle{Index}%
\def\mlftitle{Conspectus descriptionum}%
\def\mlttitle{Conspectus tabularum}%
%%
\def\stctitle{Index}%
\def\slftitle{Conspectus descriptionum}%
\def\slttitle{Conspectus tabularum}%
⟨/latin⟩
```

13.102 "Latin2" language: latin2.mld

The titles for the "latin2" language (latin, abbreviated variant) are taken from the latin.dtx (by Claudio Beccari and Krzysztof Konrad Żelechowski) file in the babel package [38, 39, 48], but abbreviated. See also section 13.101 on the preceding page.

```
⟨*latin2⟩
\ProvidesFile{latin2.mld}[2006/01/13]
%% Latin titles (shortened) for minitoc.sty
%% from latin.dtx (babel)
%% Beccari, Claudio
\def\ptctitle{Index}%
\def\plftitle{Conspectus descriptionum}%
\def\plttitle{Conspectus tabularum}%
%%
\def\mtctitle{Index}%
\def\mlftitle{Descriptiones}%
\def\mlttitle{Tabulae}%
%%
\def\stctitle{Index}%
\def\slftitle{Descriptiones}%
\def\slttitle{Tabulae}%
⟨/latin2⟩
```

13.103 "Latvian" language: latvian.mld

The titles for the "latvian" language[32] come from the latvian.ldf file (by Alexej M. Kryokov and Dmitry Ivanov) in the Antomega project [150]. See also section 13.104 on the next page.

[32] Note that "latvian" is the original name for "letton".

```
⟨*latvian⟩
\ProvidesFile{latvian.mld}[2005/02/08]
%% from latvian.ldf (Antomega project)
%% Needs Omega
%% Alexej M. Kryokov
%% Dmitry Ivanov
%%
\def\ptctitle{\locallatvian{Saturs}}%
\def\plftitle{\locallatvian{Att^^^^0113lu saraksts}}%
\def\plttitle{\locallatvian{Tabulu saraksts}}%
%%
\def\mtctitle{\locallatvian{Saturs}}%
\def\mlftitle{\locallatvian{Att^^^^0113lu saraksts}}%
\def\mlttitle{\locallatvian{Tabulu saraksts}}%
%%
\def\stctitle{\locallatvian{Saturs}}%
\def\slftitle{\locallatvian{Att^^^^0113lu saraksts}}%
\def\slttitle{\locallatvian{Tabulu saraksts}}%
⟨/latvian⟩
```

13.104 "Letton" language: `letton.mld`

The "`letton`" language is a synonym for the "`latvian`" language, so we just load `latvian.mld`. See section 13.103 on the preceding page.

```
⟨*letton⟩
\ProvidesFile{letton.mld}[2005/02/08]
\mtcselectlanguage{latvian}%
⟨/letton⟩
```

13.105 "Lithuanian" language: `lithuanian.mld`

The titles for the "`lithuanian`" language are taken from the `lithuanian.ldf` file[33] (by Sigitas Tolusis) for the babel package [38, 39]:

```
⟨*lithuanian⟩
\ProvidesFile{lithuanian.mld}[2006/01/13]
%% Lithuanian titles for minitoc.sty
%% from lithuanian.ldf
%% in http://www.vtex.lt/tex/download/texinput/babel/babel.zip
%% Tolusis, Sigitas (sigitas@vtex.lt)
\def\ptctitle{Turinys}%
\def\plftitle{Paveiksl\protect\c u s\protect\c ara\protect\v sas}%
\def\plttitle{Lentel\protect\.es}%
```

[33] Found in http://www.vtex.lt/tex/download/texinput/babel/babel.zip.

```
%%
\def\mtctitle{Turinys}%
\def\mlftitle{Paveiksl\protect\c u s\protect\c ara\protect\v sas}%
\def\mlttitle{Lentel\protect\.es}%
%%
\def\stctitle{Turinys}%
\def\slftitle{Paveiksl\protect\c u s\protect\c ara\protect\v sas}%
\def\slttitle{Lentel\protect\.es}%
⟨/lithuanian⟩
```

13.106 "Lowersorbian" language: lowersorbian.mld

The titles for the "lowersorbian" language [34] are taken from the lsorbian.dtx file (by Eduard Werner) in the babel package [38, 39, 81]. See also section 13.167 on page 517. A shorter language name is lsorbian (see section 13.107).

```
⟨*lowersorbian⟩
\ProvidesFile{lowersorbian.mld}[2006/02/28]
%% Lower sorbian titles for minitoc.sty
%% from lsorbian.dtx (babel)
%% Werner, Eduard
\def\ptctitle{Wop\'simje\'se}%
\def\plftitle{Zapis wobrazow}%
\def\plttitle{Zapis tabulkow}%
%%
\def\mtctitle{Wop\'simje\'se}%
\def\mlftitle{Zapis wobrazow}%
\def\mlttitle{Zapis tabulkow}%
%%
\def\stctitle{Wop\'simje\'se}%
\def\slftitle{Zapis wobrazow}%
\def\slttitle{Zapis tabulkow}%
⟨/lowersorbian⟩
```

13.107 "Lsorbian" language: lsorbian.mld

The "lsorbian" language is a synonym for "lowersorbian", so we just load lowersorbian.mld. See section 13.106.

```
⟨*lsorbian⟩
\ProvidesFile{lsorbian.mld}[2006/01/23]
```

[34] 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 (VI-th century).

```
%% Lower sorbian titles for minitoc.sty
%% from lsorbian.dtx (babel)
%% Werner, Eduard
\selectlanguage{lowersorbian}%
⟨/lsorbian⟩
```

13.108 "Magyar" language: magyar.mld

The titles for the "magyar" language are taken from the magyar.dtx (by József Bérces and Árpád Bíró, with help from Attila Koppanyi) file in the babel package [38, 39, 49]. A synonym of "magyar" is "hungarian" (see section 13.85 on page 477). See also sections 13.109 and 13.110 on the following page for variants (it seems that magyar.dtx might have evolved).

```
⟨*magyar⟩
\ProvidesFile{magyar.mld}[2006/03/08]
%% Magyar titles for minitoc.sty
%% from magyar.dtx (babel)
%% Bíró, Árpád and Bérces, József
\def\ptctitle{Tartalom}%
\def\plftitle{\'Abr\'ak}%
\def\plttitle{T\'abl\'azatok}%
%%
\def\mtctitle{Tartalom}%
\def\mlftitle{\'Abr\'ak}%
\def\mlttitle{T\'abl\'azatok}%
%%
\def\stctitle{Tartalom}%
\def\slftitle{\'Abr\'ak}%
\def\slttitle{T\'abl\'azatok}%
⟨/magyar⟩
```

13.109 "Magyar2" language: magyar2.mld

The titles for the "magyar2" language are taken from a variant proposed in the babel package [38, 39]. See also sections 13.108 and 13.110 on the following page.

```
⟨*magyar2⟩
\ProvidesFile{magyar.mld}[2006/03/08]
%% Magyar2 titles for minitoc.sty (variant)
%% from magyar.dtx (babel)
%% Bíró, Árpád and Bérces, József
\def\ptctitle{Tartalom}%
\def\plftitle{\'Abr\'ak list\'aja}%
\def\plttitle{T\'abl\'azatok list\'aja}%
%%
```

```
\def\mtctitle{Tartalom}%
\def\mlftitle{\'Abr\'ak list\'aja}%
\def\mlttitle{T\'abl\'azatok list\'aja}%
%%
\def\stctitle{Tartalom}%
\def\slftitle{\'Abr\'ak list\'aja}%
\def\slttitle{T\'abl\'azatok list\'aja}%
⟨/magyar2⟩
```

13.110 "Magyar3" language: magyar3.mld

The titles for the "magyar3" language (third variant of magyar) are taken from the magyar.dtx file in the babel package [38, 39, 49]. See also sections 13.108 on the page before and 13.109 on the preceding page.

```
⟨*magyar3⟩
\ProvidesFile{magyar3.mld}[2006/03/08]
%% Magyar3 titles for minitoc.sty (variant)
%% from magyar.dtx (babel)
%% Bíró, Árpád and Bérces, József
\def\ptctitle{Tartalomjegyz\'ek}%
\def\plftitle{\'Abr\'ak jegyz\'eke}%
\def\plttitle{T\'abl\'azatok jegyz\'eke}%
%%
\def\mtctitle{Tartalomjegyz\'ek}%
\def\mlftitle{\'Abr\'ak jegyz\'eke}%
\def\mlttitle{T\'abl\'azatok jegyz\'eke}%
%%
\def\stctitle{Tartalomjegyz\'ek}%
\def\slftitle{\'Abr\'ak jegyz\'eke}%
\def\slttitle{T\'abl\'azatok jegyz\'eke}%
⟨/magyar3⟩
```

13.111 "Malay" language: malay.mld

The "malay" language is just like "bahasam", so we just load bahasam.mld (see section 13.16 on page 440):

```
⟨*malay⟩
\ProvidesFile{malay.mld}[2006/01/11]
\mtcselectlanguage{bahasam}%
⟨/malay⟩
```

13.112 "Malayalam-keli" language: malayalam-keli.mld

The titles for the "malayalam-keli" language[35], with the "Keli" fonts, are taken from the malayalam package [3] by Alex AJ. This language requires specific fonts. See also sections 13.114 on the following page and 13.115 on the next page.

```
⟨*malayalam – keli⟩
\ProvidesFile{malayalam-keli.mld}[2006/01/13]
%%
%% Malayalam: Keli fonts
%%
\def\ptctitle{\mm \X{\<68>}\X{\<197>}\X{\<83>}\X{\<161>}\<119>}%
\def\plftitle{\mm \X{\<78>\<110>}\X{\<123>\<88>}\X{\<167>}\X{\<196>}}%
\def\plttitle{\mm \X{\<116>\<83>}\X{\<95>\<110>}\X{\<102>\<112>}\X{\<73>}\X{\<196>}}%
%%
\def\mtctitle{\mm \X{\<68>}\X{\<197>}\X{\<83>}\X{\<161>}\<119>}%
\def\mlftitle{\mm \X{\<78>\<110>}\X{\<123>\<88>}\X{\<167>}\X{\<196>}}%
\def\mlttitle{\mm \X{\<116>\<83>}\X{\<95>\<110>}\X{\<102>\<112>}\X{\<73>}\X{\<196>}}%
%%
\def\stctitle{\mm \X{\<68>}\X{\<197>}\X{\<83>}\X{\<161>}\<119>}%
\def\slftitle{\mm \X{\<78>\<110>}\X{\<123>\<88>}\X{\<167>}\X{\<196>}}%
\def\slttitle{\mm \X{\<116>\<83>}\X{\<95>\<110>}\X{\<102>\<112>}\X{\<73>}\X{\<196>}}%
⟨/malayalam – keli⟩
```

13.113 "Malayalam-omega" language: malayalam-omega.mld and malayalam-omega.mlo

This is the Malayalam language implementation "malayalam-omega" based on *Lambda* (**Λ**) (the version of LaTeX for Omega) via the omal package [4] (by Alex AJ) of the Malayalam-Omega project[36]. As the titles contain characters in a special encoding, we must load a .mlo file. A lot of fonts are available via options of the omal package.

```
⟨*malayalam – omega⟩
\ProvidesFile{malayalam-omega.mld}[2006/02/13]
% from omal.sty (Alex A.J. indicTeX@gmail.com)
\mtcloadmlo{malayalam-omega}
%
⟨/malayalam – omega⟩
```

[35] 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.

[36] http://Sarovar.org/projects/malayalam

13.114 "Malayalam-rachana" language: malayalam-rachana.mld

The titles for the "malayalam-rachana" language, with the traditionnal "Rachana" fonts (old lipi), are taken from the malayalam package [3] by Alex AJ. This language requires specific fonts. See also sections 13.112 on the preceding page and 13.115.

```
⟨*malayalam – rachana⟩
\ProvidesFile{malayalam-rachana.mld}[2005/06/07]
%%
%% Malayalam: Rachana fonts, traditionnal.
%%
\def\ptctitle{\mm \X{\<68>}\X{\<201>}\X{\<83>}\X{\<183>}\<119>}%
\def\plftitle{\mm \X{\<78>\<111>}\X{{\C\<94>}}\X{\<186>}\X{\<179>}}%
\def\plttitle{\mm \X{\<117>\<83>}\X{\<95>\<111>}\X{{\F\<59>}}\X{\<73>}\X{\<179>}}%
%%
\def\mtctitle{\mm \X{\<68>}\X{\<201>}\X{\<83>}\X{\<183>}\<119>}%
\def\mlftitle{\mm \X{\<78>\<111>}\X{{\C\<94>}}\X{\<186>}\X{\<179>}}%
\def\mlttitle{\mm \X{\<117>\<83>}\X{\<95>\<111>}\X{{\F\<59>}}\X{\<73>}\X{\<179>}}%
%%
\def\stctitle{\mm \X{\<68>}\X{\<201>}\X{\<83>}\X{\<183>}\<119>}%
\def\slftitle{\mm \X{\<78>\<111>}\X{{\C\<94>}}\X{\<186>}\X{\<179>}}%
\def\slttitle{\mm \X{\<117>\<83>}\X{\<95>\<111>}\X{{\F\<59>}}\X{\<73>}\X{\<179>}}%
⟨/malayalam – rachana⟩
```

13.115 "Malayalam-rachana2" language: malayalam-rachana2.mld

The titles for the "malayalam-rachana2" language, with the reformed "Rachana" fonts (new lipi), are taken from the malayalam package [3] by Alex AJ. This language requires specific fonts. See also sections 13.112 on the preceding page and 13.114.

```
⟨*malayalam – rachana2⟩
\ProvidesFile{malayalam-rachana2.mld}[2006/01/13]
%%
%% Malayalam: Rachana fonts, reformed.
%%
\def\ptctitle{\mm \X{\<68>}\X{\<201>}\X{\<83>}\X{\<183>}\<119>}%
\def\plftitle{\mm \X{\<78>\<111>}\X{\<125>\<88>}\X{\<186>}\X{\<179>}}%
\def\plttitle{\mm \X{\<117>\<83>}\X{\<95>\<111>}\X{\<106>\<113>}\X{\<73>}\X{\<179>}}%
%%
\def\mtctitle{\mm \X{\<68>}\X{\<201>}\X{\<83>}\X{\<183>}\<119>}%
\def\mlftitle{\mm \X{\<78>\<111>}\X{\<125>\<88>}\X{\<186>}\X{\<179>}}%
\def\mlttitle{\mm \X{\<117>\<83>}\X{\<95>\<111>}\X{\<106>\<113>}\X{\<73>}\X{\<179>}}%
%%
\def\stctitle{\mm \X{\<68>}\X{\<201>}\X{\<83>}\X{\<183>}\<119>}%
\def\slftitle{\mm \X{\<78>\<111>}\X{\<125>\<88>}\X{\<186>}\X{\<179>}}%
```

```
\def\slttitle{\mm \X{\<117>\<83>}\X{\<95>\<111>}\X{\<106>\<113>}\X{\<73>}\X{\<179>}}%
⟨/malayalam – rachana2⟩
```

13.116 "Manju" language: manju.mld

The "manju" language is a synomym for "bithe", so we just load bithe.mld (see section 13.22 on page 444):

```
⟨*manju⟩
\ProvidesFile{manju.mld}[2005/11/16]
\mtcselectlanguage{bithe}%
⟨/manju⟩
```

13.117 "Meyalu" language: meyalu.mld

The "meyalu" language is just like "bahasam", so we just load bahasam.mld (see section 13.16 on page 440):

```
⟨*meyalu⟩
\ProvidesFile{meyalu.mld}[2006/01/13]
\mtcselectlanguage{bahasam}%
⟨/meyalu⟩
```

13.118 "Mongol" language: mongol.mld

The titles for the "mongol" language are taken from the MonTEX package [97, 100]. This language requires specific fonts. See also sections 13.19 on page 442, 13.20 on page 442, 13.21 on page 443, 13.22 on page 444, 13.29 on page 447, 13.30 on page 448, 13.173 on page 520, 13.174 on page 520, and 13.175 on page 521.

```
⟨*mongol⟩
\ProvidesFile{mongol.mld}[1999/03/16]
%% Mongol (xalx) titles for minitoc.sty
%% Needs mongol fonts
\def\ptctitle{{\mnr Garqig}}%
\def\plftitle{{\mnr Zurgi"in jagsaalt}}%
\def\plttitle{{\mnr X"usn"agti"in jagsaalt}}%
%%
\def\mtctitle{{\mnr Garqig}}%
\def\mlftitle{{\mnr Zurgi"in jagsaalt}}%
\def\mlttitle{{\mnr X"usn"agti"in jagsaalt}}%
```

```
%%
\def\stctitle{{\mnr Garqig}}%
\def\slftitle{{\mnr Zurgi"in jagsaalt}}%
\def\slttitle{{\mnr X"usn"agti"in jagsaalt}}%
⟨/mongol⟩
```

13.119 “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 just load the ngermanb.mld file. See also section 13.122 on the next page.

```
⟨*naustrian⟩
\ProvidesFile{naustrian.mld}[2004/12/14]
\mtcselectlanguage{ngermanb}%
⟨/naustrian⟩
```

13.120 “Newzealand” language: newzealand.mld

The “newzealand” language is just like “english”, so we just load english.mld (see section 13.43 on page 454):

```
⟨*newzealand⟩
\ProvidesFile{newzealand.mld}[2006/01/11]
\mtcselectlanguage{english}%
⟨/newzealand⟩
```

13.121 “Ngerman” language: ngerman.mld

The “ngerman” language is a synonym of the “ngermanb” language [37], so we just load the ngermanb.mld file. See also section 13.122 on the next page.

```
⟨*ngerman⟩
\ProvidesFile{ngerman.mld}[2004/12/14]
\mtcselectlanguage{ngermanb}%
⟨/ngerman⟩
```

[37] A revised version of the germanb variant of the german language.

13.122 "Ngermanb" language: ngermanb.mld

The titles for the "ngermanb" language [38] are taken from the file ngermanb.dtx file (by Bernd Raichle and Walter Schmidt) in the babel package [38, 39, 73]. See also sections 13.119 on the preceding page, and 13.121 on the page before.

```
⟨*ngermanb⟩
\ProvidesFile{ngermanb.mld}[2006/01/13]
%% New german (B) titles for minitoc.sty
%% from ngermanb.dtx (babel)
%% Raichle, Bernd and Schmidt, Walter
\def\ptctitle{Inhaltsverzeichnis}%    % oder nur: Inhalt
\def\plftitle{Abbildungsverzeichnis}%
\def\plttitle{Tabellenverzeichnis}%
%%
\def\mtctitle{Inhalt}%
\def\mlftitle{Abbildungsverzeichnis}%
\def\mlttitle{Tabellenverzeichnis}%
%%
\def\stctitle{Inhalt}%
\def\slftitle{Abbildungsverzeichnis}%
\def\slttitle{Tabellenverzeichnis}%
⟨/ngermanb⟩
```

13.123 "Ngermanb2" language: ngermanb2.mld

The titles for the "ngermanb2" language [39] are taken from the babel package [38, 39, 73]. See also section 13.122.

```
⟨*ngermanb2⟩
\ProvidesFile{ngermanb2.mld}[2005/09/27]
%% New german (B) short (2) titles for minitoc.sty
\def\ptctitle{Inhalt}%
\def\plftitle{Abbildungen}%
\def\plttitle{Tabellen}%
%%
\def\mtctitle{Inhalt}%
\def\mlftitle{Abbildungen}%
\def\mlttitle{Tabellen}%
%%
\def\stctitle{Inhalt}%
\def\slftitle{Abbildungen}%
\def\slttitle{Tabellen}%
⟨/ngermanb2⟩
```

[38] A variant of the german language, with revised spelling.
[39] A variant of the german language, with revised spelling and short titles.

13.124 "Norsk" language: norsk.mld

The titles for the "norsk" language (or *bokmål*, "language of the kingdom") are taken from the norsk.dtx file (by Johannes Braams, Haavard Helstrup, Alv Kjetil Holme, Per Steinar Iversen, Terje Engeset Petterst and Rune Kleveland) in the babel package [38, 39, 43], with help from Dag Langmyhr. See also section 13.126 on the next page.

```
⟨*norsk⟩
\ProvidesFile{norsk.mld}[2006/01/13]
%% Norsk titles for minitoc.sty
%% from noesk.dtx (babel)
%% Braams, Johannes and Helstrup, Haavard and Holme, Alv Kjetil and
%% Iversen, Per Steinar and  Petterst, Terje Engeset and  Kleveland, Rune
%% Thanks to Dag Langmyhr (dag@ifi.uio.no)
\def\ptctitle{Innhold}%
\def\plftitle{Figurer}%
\def\plttitle{Tabeller}%
%%
\def\mtctitle{Innhold}%
\def\mlftitle{Figurer}%
\def\mlttitle{Tabeller}%
%%
\def\stctitle{Innhold}%
\def\slftitle{Figurer}%
\def\slttitle{Tabeller}%
⟨/norsk⟩
```

13.125 "Norsk2" language: norsk2.mld

The titles for the "norsk2" language (or *bokmål*, "language of the kingdom") are taken from the babel package [38, 39, 43], with help from Dag Langmyhr. These are variants of the titles of section 13.124.

```
⟨*norsk2⟩
\ProvidesFile{norsk2.mld}[2005/09/27]
%% Short norsk titles for minitoc.sty
%% Thanks to Dag Langmyhr (dag@ifi.uio.no)
\def\ptctitle{Innhold}%
\def\plftitle{Figurliste}%
\def\plttitle{Tabelliste}%
%%
\def\mtctitle{Innhold}%
\def\mlftitle{Figurliste}%
\def\mlttitle{Tabelliste}%
%%
\def\stctitle{Innhold}%
\def\slftitle{Figurliste}%
\def\slttitle{Tabelliste}%
```

```
⟨/norsk2⟩
```

13.126 "Nynorsk" language: nynorsk.mld

The titles for the "nynorsk" language [40] are taken from norsk.dtx file (by Johannes Braams, Haavard Helstrup, Alv Kjetil Holme, Per Steinar Iversen, Terje Engeset Petterst and Rune Kleveland) in the babel package [38, 39, 43], with help from Dag Langmyhr. See also section 13.124 on the page before.

```
⟨*nynorsk⟩
\ProvidesFile{nynorsk.mld}[2006/01/13]
%% Nynorsk titles for minitoc.sty
%% from norsk.dtx (babel)
%% Braams, Johannes and Helstrup, Haavard and Holme, Alv Kjetil and
%% Iversen, Per Steinar and  Petterst, Terje Engeset and  Kleveland, Rune
%% Thanks to Dag Langmyhr (dag@ifi.uio.no)
\def\mtctitle{Innhald}%
\def\mlftitle{Figurar}%
\def\mlttitle{Tabellar}%
%%
\def\ptctitle{Innhald}%
\def\plftitle{Figurar}%
\def\plttitle{Tabellar}%
%%
\def\stctitle{Innhald}%
\def\slftitle{Figurar}%
\def\slttitle{Tabellar}%
⟨/nynorsk⟩
```

13.127 "Nynorsk2" language: nynorsk2.mld

The titles for the "nynorsk2" language are variants of the titles of the "nynorsk" language. See also section 13.126.

```
⟨*nynorsk2⟩
\ProvidesFile{nynorsk.mld}[1999/03/16]
%% Nynorsk titles for minitoc.sty
%% Thanks to Dag Langmyhr (dag@ifi.uio.no)
\def\mtctitle{Innhald}%
\def\mlftitle{Figurliste}%
\def\mlttitle{Tabelliste}%
%%
```

[40] Created around 1800 by Ivar Åssen to make a real independent and national norvegian language, in reaction to danish, from the various dialect spoken in the country. But nynorsk has never gained much popularity outside rural regions.

```
\def\ptctitle{Innhald}%
\def\plftitle{Figurliste}%
\def\plttitle{Tabelliste}%
%%
\def\stctitle{Innhald}%
\def\slftitle{Figurliste}%
\def\slttitle{Tabelliste}%
⟨/nynorsk2⟩
```

13.128 "Polish" language: polish.mld

The titles for the "polish" language are taken from the polish.dtx file (by Elmar Schalueck and Michael Janich) in the babel package [38, 39, 78]. See also sections 13.129 and 13.130 on the following page.

```
⟨*polish⟩
\ProvidesFile{polish.mld}[2006/01/13]
%% Polish titles for minitoc.sty
%% from polish.dtx (babel)
%% Schalueck, Elmar and Janich, Michael
\def\ptctitle{Spis rzeczy}%
\def\plftitle{Spis rysunk\'ow}%
\def\plttitle{Spis tablic}%
%%
\def\mtctitle{Spis rzeczy}%
\def\mlftitle{Spis rysunk\'ow}%
\def\mlttitle{Spis tablic}%
%%
\def\stctitle{Spis rzeczy}%
\def\slftitle{Spis rysunk\'ow}%
\def\slttitle{Spis tablic}%
⟨/polish⟩
```

13.129 "Polish2" language: polish2.mld

The titles for the "polish2" language[41] are taken from the omega-polish.ldf (by Alexej M. Kryokov and Dmitry Ivanov) in the Antomega project [150]. See also sections 13.128 and 13.130 on the following page.

```
⟨*polish2⟩
\ProvidesFile{polish2.mld}[2005/02/08]
%% from omega-polish.ldf (Antomega project)
%% Needs Omega
%% Alexej M. Kryokov
```

[41] "Polish2" is a variant of "polish".

```
%% Dmitry Ivanov
%%
\def\ptctitle{\localpolish{Spis tre^^^^00b1ci}}%
\def\plftitle{\localpolish{Spis rysunk^^^^00adw}}%
\def\plttitle{\localpolish{Spis tablic}}%
%%
\def\mtctitle{\localpolish{Spis tre^^^^00b1ci}}%
\def\mlftitle{\localpolish{Spis rysunk^^^^00adw}}%
\def\mlttitle{\localpolish{Spis tablic}}%
%%
\def\stctitle{\localpolish{Spis tre^^^^00b1ci}}%
\def\slftitle{\localpolish{Spis rysunk^^^^00adw}}%
\def\slttitle{\localpolish{Spis tablic}}%
⟨/polish2⟩
```

13.130 “Polski” language: polski.mld

The titles for the “polski” language (variant for polish) are taken from the polski.dtx file (by Mariusz Olko and Marcin Woliński) in the PLaTeX package [199, 247]. See also sections 13.128 on the page before and 13.129 on the preceding page.

```
⟨*polski⟩
\ProvidesFile{polski.mld}[2006/02/28]
%% Polski titles for minitoc.sty
%% from polski.dtx (PLaTeX)
%% Mariusz Olko, Marcin Woli\'nski.
%%
\def\ptctitle{Spis tre\'sci}%
\def\plftitle{Spis rysunk\'ow}%
\def\plttitle{Spis tabel}%
%%
\def\mtctitle{Spis rysunk\'ow}%
\def\mlftitle{Spis rysunk\'ow}%
\def\mlttitle{Spis tabel}%
%%
\def\stctitle{Spis rysunk\'ow}%
\def\slftitle{Spis rysunk\'ow}%
\def\slttitle{Spis tabel}%
⟨/polski⟩
```

13.131 “Portuges” language: portuges.mld

The name “portuges” is another spelling for “portuguese” (see section 13.132 on the next page), so we just load portuguese.mld:

```
⟨*portuges⟩
\ProvidesFile{portuges.mld}[2005/06/07]
\mtcselectlanguage{portuguese}%
⟨/portuges⟩
```

13.132 "Portuguese" language: portuguese.mld

The titles for the "portuguese" language are taken from the portuges.dtx file (by Jose Pedro Ramalhete) in the babel package [38, 39, 74].

See also section 13.23 on page 444, because the titles are different in Brazil, even if the language is also portuguese.

```
⟨*portuguese⟩
\ProvidesFile{portuguese.mld}[2006/01/13]
%% Portuguese titles for minitoc.sty
%% from portuges.dtx (babel)
%% Ramalhete, Jose Pedro
\def\ptctitle{Conte\'udo}%
\def\plftitle{Lista de Figuras}%
\def\plttitle{Lista de Tabelas}%
%%
\def\mtctitle{Conte\'udo}%
\def\mlftitle{Lista de Figuras}%
\def\mlttitle{Lista de Tabelas}%
%%
\def\stctitle{Conte\'udo}%
\def\slftitle{Lista de Figuras}%
\def\slttitle{Lista de Tabelas}%
⟨/portuguese⟩
```

13.133 "Romanian" language: romanian.mld

The titles for the "romanian" language come from the romanian.dtx file (by Horst Umstatter and Robert Juhasz) in the babel package [38, 39, 60].

```
⟨*romanian⟩
\ProvidesFile{romanian.mld}[2006/01/13]
%% Romanian titles for minitoc.sty
%% from romanian.dtx (babel)
%% Horst, Umstatter and Juhasz, Robert
\def\ptctitle{Cuprins}%
\def\plftitle{List\u{a} de figuri}%
\def\plttitle{List\u{a} de tabele}%
%%
```

```
\def\mtctitle{Cuprins}%
\def\mlftitle{List\u{a} de figuri}%
\def\mlttitle{List\u{a} de tabele}%
%%
\def\stctitle{Cuprins}%
\def\slftitle{List\u{a} de figuri}%
\def\slttitle{List\u{a} de tabele}%
⟨/romanian⟩
```

13.134 "Romanian2" language: romanian2.mld

The titles for the "romanian2" language come from the romanian.dtx file (by Adrian Rezuş and Bernd Raichle) in the RomanianTEX package [217]. See also sections 13.133 on the page before and 13.135. Alas, RomanianTEX is not compatible with the babel package [38, 39].

```
⟨*romanian2⟩
\ProvidesFile{romanian2.mld}[2006/08/03]
%% Romanian titles for minitoc.sty
%% from RomanianTeX (romanian.dtx)
%% Adrian Rezus (adriaan@cs.kun.nl)
%% Bernd Raichle
\def\ptctitle{Cuprins}%
\def\plftitle{Lista de figuri}%
\def\plttitle{Lista de tabele}%
%%
\def\mtctitle{Cuprins}%
\def\mlftitle{Lista de figuri}%
\def\mlttitle{Lista de tabele}%
%%
\def\stctitle{Cuprins}%
\def\slftitle{Lista de figuri}%
\def\slttitle{Lista de tabele}%
⟨/romanian2⟩
```

13.135 "Romanian3" language: romanian3.mld

The titles for the "romanian3" language come from the romanian.dtx file (by Adrian Rezuş and Bernd Raichle) in the RomanianTEX package [217]. See also sections 13.133 on the page before and 13.134. Alas, RomanianTEX is not compatible with the babel package [38, 39].

```
⟨*romanian3⟩
\ProvidesFile{romanian3.mld}[2006/08/03]
%% Romanian titles for minitoc.sty
%% from RomanianTeX (romanian.dtx) variant.
%% Adrian Rezus (adriaan@cs.kun.nl)
```

```
%% Bernd Raichle
\def\ptctitle{Tabla de materii}%
\def\plttitle{Indice de figuri}%
\def\plttitle{Tabele}%
%%
\def\mtctitle{Tabla de materii}%
\def\mlttitle{Indice de figuri}%
\def\mlttitle{Tabele}%
%%
\def\stctitle{Tabla de materii}%
\def\slttitle{Indice de figuri}%
\def\slttitle{Tabele}%
⟨/romanian3⟩
```

13.136 "Russian" language: russian.mld

The titles for the "russian" language are taken from the babel package [38, 39]. Specific cyrillic fonts are required.

```
⟨*russian⟩
\ProvidesFile{russian.mld}[1999/03/16]
%% Russian titles for minitoc.sty
\def\ptctitle{Oglavlenie}%
\def\plftitle{Pere{\cz}en{\mz} risunkov}%
\def\plttitle{Pere{\cz}en{\mz} tablic}%
%%
\def\mtctitle{Oglavlenie}%
\def\mlftitle{Pere{\cz}en{\mz} risunkov}%
\def\mlttitle{Pere{\cz}en{\mz} tablic}%
%%
\def\stctitle{Oglavlenie}%
\def\slftitle{Pere{\cz}en{\mz} risunkov}%
\def\slttitle{Pere{\cz}en{\mz} tablic}%
⟨/russian⟩
```

13.137 "Russian2m" language: russian2m.mld

The titles for the "russian2m" language ("russian2m" is a modern variant of "russian") are taken from the russian2m.ldf file (by Alexej M. Kryokov and Dmitry Ivanov) in the Antomega project [150]. Specific cyrillic fonts are required. See also section 13.136.

```
⟨*russian2m⟩
\ProvidesFile{russian2m.mld}[2005/02/08]
%% from russian2m.ldf (Antomega project, russian modern)
%% Needs Omega and cyrillic fonts
```

```
%% Alexej M. Kryokov
%% Dmitry Ivanov
%%
\def\ptctitle{\localrussian%
{^^^^041e^^^^0433^^^^043b^^^^0430^^^^0432^^^^043b^^^^0435^^^^043d%
 ^^^^0438^^^^0435}}%
\def\plftitle{\localrussian%
{^^^^0421^^^^043f^^^^0438^^^^0441^^^^043e^^^^043a ^^^^0438^^^^043b%
 ^^^^043b^^^^044e^^^^0441^^^^0442^^^^0440^^^^0430^^^^0446^^^^0438%
 ^^^^0439}}%
\def\plttitle{\localrussian%
{^^^^0421^^^^043f^^^^0438^^^^0441^^^^043e^^^^043a ^^^^0442^^^^0430%
 ^^^^0431^^^^043b^^^^0438^^^^0446}}%
%%
\def\mtctitle{\localrussian%
{^^^^041e^^^^0433^^^^043b^^^^0430^^^^0432^^^^043b^^^^0435^^^^043d%
 ^^^^0438^^^^0435}}%
\def\mlftitle{\localrussian%
{^^^^0421^^^^043f^^^^0438^^^^0441^^^^043e^^^^043a ^^^^0438^^^^043b%
 ^^^^043b^^^^044e^^^^0441^^^^0442^^^^0440^^^^0430^^^^0446^^^^0438%
 ^^^^0439}}%
\def\mlttitle{\localrussian%
{^^^^0421^^^^043f^^^^0438^^^^0441^^^^043e^^^^043a ^^^^0442^^^^0430%
 ^^^^0431^^^^043b^^^^0438^^^^0446}}%
%%
\def\stctitle{\localrussian%
{^^^^041e^^^^0433^^^^043b^^^^0430^^^^0432^^^^043b^^^^0435^^^^043d%
 ^^^^0438^^^^0435}}%
\def\slftitle{\localrussian%
{^^^^0421^^^^043f^^^^0438^^^^0441^^^^043e^^^^043a ^^^^0438^^^^043b%
 ^^^^043b^^^^044e^^^^0441^^^^0442^^^^0440^^^^0430^^^^0446^^^^0438%
 ^^^^0439}}%
\def\slttitle{\localrussian%
{^^^^0421^^^^043f^^^^0438^^^^0441^^^^043e^^^^043a ^^^^0442^^^^0430%
 ^^^^0431^^^^043b^^^^0438^^^^0446}}%
⟨/russian2m⟩
```

13.138 “Russian2o” language: russian2o.mld

The titles for the “russian2o” language (“russian2o” is an old variant of “russian”) are taken from the omega-russian.ldf file (by Alexej M. Kryokov and Dmitry Ivanov) in the Antomega project [150]. Specific cyrillic fonts are required. See also section 13.136 on the preceding page.

```
⟨*russian2o⟩
\ProvidesFile{russian2o.mld}[2005/02/08]
%% from russian2o.mld (Antomega project - russian old)
%% Needs Omega and cyrillic fonts
%% Alexej M. Kryokov
%% Dmitry Ivanov
```

```
%%
\def\ptctitle{\localrussian%
{^^^^041e^^^^0433^^^^043b^^^^0430^^^^0432^^^^043b^^^^0435^^^^043d%
 ^^^^0456^^^^0435}}%
\def\plftitle{\localrussian%
{^^^^0421^^^^043f^^^^0438^^^^0441^^^^043e^^^^043a^^^^044a ^^^^0438%
 ^^^^043b^^^^043b^^^^044e^^^^0441^^^^0442^^^^0440^^^^0430^^^^0446%
 ^^^^0456^^^^0439}}%
\def\plttitle{\localrussian%
{^^^^0421^^^^043f^^^^0438^^^^0441^^^^043e^^^^043a^^^^044a ^^^^0442%
 ^^^^0430^^^^0431^^^^043b^^^^0438^^^^0446^^^^044a}}%
%%
\def\mtctitle{\localrussian%
{^^^^041e^^^^0433^^^^043b^^^^0430^^^^0432^^^^043b^^^^0435^^^^043d%
 ^^^^0456^^^^0435}}%
\def\mlftitle{\localrussian%
{^^^^0421^^^^043f^^^^0438^^^^0441^^^^043e^^^^043a^^^^044a ^^^^0438%
 ^^^^043b^^^^043b^^^^044e^^^^0441^^^^0442^^^^0440^^^^0430^^^^0446%
 ^^^^0456^^^^0439}}%
\def\mlttitle{\localrussian%
{^^^^0421^^^^043f^^^^0438^^^^0441^^^^043e^^^^043a^^^^044a ^^^^0442%
 ^^^^0430^^^^0431^^^^043b^^^^0438^^^^0446^^^^044a}}%
%
\def\stctitle{\localrussian%
{^^^^041e^^^^0433^^^^043b^^^^0430^^^^0432^^^^043b^^^^0435^^^^043d%
 ^^^^0456^^^^0435}}%
\def\slftitle{\localrussian%
{^^^^0421^^^^043f^^^^0438^^^^0441^^^^043e^^^^043a^^^^044a ^^^^0438%
 ^^^^043b^^^^043b^^^^044e^^^^0441^^^^0442^^^^0440^^^^0430^^^^0446%
 ^^^^0456^^^^0439}}%
\def\slttitle{\localrussian%
{^^^^0421^^^^043f^^^^0438^^^^0441^^^^043e^^^^043a^^^^044a ^^^^0442%
 ^^^^0430^^^^0431^^^^043b^^^^0438^^^^0446^^^^044a}}%
⟨/russian2o⟩
```

13.139 “Russianb” language: russianb.mld

The titles for the “russianb” language (“russianb” is a variant of “russian”) are taken from the russianb.dtx file (by Olga G. Lapko, Vladimir Volovich et Werner Lemberg) in the babel package [38, 39, 66, 160]. Specific cyrillic fonts are required. See also section 13.136 on page 501. The parttoc title varies depending if chapters are defined or not by the document class.

```
⟨*russianb⟩
\ProvidesFile{russianb.mld}[2006/02/15]
%% Russian (russianb) titles for minitoc.sty
%% from russianb.dtx (babel)
%% Lapko, Olga and Volovitch, Vladimir and Lemberg, Werner
\expandafter\ifx\csname chapter\endcsname\relax
\def\ptctitle{%
```

```
  {\cyr\CYRS\cyro\cyrd\cyre\cyrr\cyrzh\cyra\cyrn\cyri\cyre}}%
\else
\def\ptctitle{%
  {\cyr \CYRO\CYRg\CYRl\CYRa\CYRv\CYRl\CYRe\CYRn\CYRi\CYRe}}%
\fi
\def\plftitle{%
  {\cyr \CYRS\CYRp\CYRi\CYRs\CYRo\CYRk\space
    \CYRi\CYRl\CYRl\CYRyu\CYRs\CYRt\CYRr\CYRa\CYRc\CYRi\CYRishrt}}%
\def\plttitle{%
  {\cyr \CYRS\CYRp\CYRi\CYRs\CYRo\CYRk\space
  \CYRt\CYRa\CYRb\CYRl\CYRi\CYRc}}%
%%
\def\mtctitle{%
  {\cyr \CYRO\CYRg\CYRl\CYRa\CYRv\CYRl\CYRe\CYRn\CYRi\CYRe}}%
\def\mlftitle{%
  {\cyr \CYRS\CYRp\CYRi\CYRs\CYRo\CYRk\space
    \CYRi\CYRl\CYRl\CYRyu\CYRs\CYRt\CYRr\CYRa\CYRc\CYRi\CYRishrt}}%
\def\mlttitle{%
  {\cyr \CYRS\CYRp\CYRi\CYRs\CYRo\CYRk\space
  \CYRt\CYRa\CYRb\CYRl\CYRi\CYRc}}%
%%
\def\stctitle{%
  {\cyr \CYRO\CYRg\CYRl\CYRa\CYRv\CYRl\CYRe\CYRn\CYRi\CYRe}}%
\def\slftitle{%
  {\cyr \CYRS\CYRp\CYRi\CYRs\CYRo\CYRk\space
    \CYRi\CYRl\CYRl\CYRyu\CYRs\CYRt\CYRr\CYRa\CYRc\CYRi\CYRishrt}}%
\def\slttitle{%
  {\cyr \CYRS\CYRp\CYRi\CYRs\CYRo\CYRk\space
  \CYRt\CYRa\CYRb\CYRl\CYRi\CYRc}}%
⟨/russianb⟩
```

13.140 “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 [97, 100]. Specific cyrillic fonts are required. See also section 13.136 on page 501.

```
⟨*russianc⟩
\ProvidesFile{russianc.mld}[1999/03/16]
%% Russian titles for minitoc.sty
%% Needs cyrillic fonts
\def\ptctitle{\xalx{Oglawlenie}}%
\def\plftitle{\xalx{Spisok risunkow}}%
\def\plttitle{\xalx{Spisok tablic}}%
%%
\def\mtctitle{\xalx{Soderjanie}}%
\def\mlftitle{\xalx{Spisok risunkow}}%
\def\mlttitle{\xalx{Spisok tablic}}%
%%
\def\stctitle{\xalx{Soderjanie}}%
```

```
\def\slftitle{\xalx{Spisok risunkow}}%
\def\slttitle{\xalx{Spisok tablic}}%
⟨/russianc⟩
```

13.141 "Russian-cca" language: russian-cca.mld and russian-cca.mlo

They are several variants for the russian titles with the cmcyralt fonts. The titles for a first variant of the "russian-cca" are taken from the russian.sty (by Victor Boyko and Vadim Maslov) file in the cmcyralt package [36].

The titles for the "russian-cca" language contain characters that cannot be easily generated, hence we load russian-cca.mlo.

```
⟨*russian – cca⟩
\ProvidesFile{russian-cca.mld}[2006/03/08]
%% Russian-cca titles for minitoc.sty
%% From russian.sty in the cmcyralt package
%% Vadim Maslov (vadik@cs.umd.edu)
%% Victor Boyko (vb1890@cs.nyu.edu)
%% Needs cmcyralt fonts and special input encoding.
\mtcloadmlo{russian-cca}%
⟨/russian – cca⟩
```

13.142 "Russian-cca1" language: russian-cca1.mld and russian-cca1.mlo

They are several variants for the russian titles with the cmcyralt fonts. The titles for the "russian-cca1" language are taken from the cmcyralt.sty file (by Vadim Maslov, Alexander Harin and Vadim V. Zhytnikov) in the cmcyralt package[130].

The titles for the "russian-cca1" language contain characters that cannot be easily generated, hence we load russian-cca1.mlo.

```
⟨*russian – cca1⟩
\ProvidesFile{russian-cca1.mld}[2006/03/08]
%% Russian-cca1 titles for minitoc.sty
%% From cmcyralt.sty in the cmcyralt package
%% cmcyr fonts in alt encoding
%% Vadim Maslov (vadik@cs.umd.edu)
%% Alexander Harin (harin@lourie.und.ac.za)
%% and  Vadim V. Zhytnikov (vvzhy@phy.ncu.edu.tw)
\mtcloadmlo{russian-cca1}%
⟨/russian – cca1⟩
```

13.143 "Russian-lh" language: russian-lh.mld and russian-lh.mlo

The russian titles for the LH fonts ("russian-lh" language) are taken from the russian.sty file (by Sergei O. Naumov) in the LH package [194].

The titles for the "russian-lh" language contain characters that cannot be easily generated, hence we load russian-lh.mlo.

```
⟨*russian – lh⟩
\ProvidesFile{russian-lh.mld}[2006/03/08]
%% Russian-lh titles for minitoc.sty
%% From russian.sty in the LH package
%% LH fonts in special encoding
%% Sergei O. Naumov (serge@astro.unc.edu)
\mtcloadmlo{russian-lh}%
⟨/russian – lh⟩
```

13.144 "Russian-lhcyralt" language: russian-lhcyralt.mld and russian-lhcyralt.mlo

The russian titles for the LHCYRALT fonts ("russian-lhcyralt" language) are taken from the lhcyralt.sty file (by Vadim V. Zhytnikov) in the lhcyr package [262].

The titles for the "russian-lhcyralt" language contain characters that cannot be easily generated, hence we load russian-lhcyralt.mlo. The input encoding is ALT (code page CP866).

```
⟨*russian – lhcyralt⟩
\ProvidesFile{russian-lhcyralt.mld}[2006/03/10]
%% Russian-lhcyralt titles for minitoc.sty
%% From lhcyralt.sty in the LHCYR package
%% LHCYRALT fonts in special encoding ALT (CP866)
%% Vadim V. Zhytnikov (vvzhy@td.lpi.ac.ru)
\mtcloadmlo{russian-lhcyralt}%
⟨/russian – lhcyralt⟩
```

13.145 "Russian-lhcyrkoi" language: russian-lhcyrkoi.mld and russian-lhcyrkoi.mlo

The russian titles for the LHCYRKOI fonts ("russian-lhcyrkoi" language) are taken from the lhcyrkoi.sty file (by Vadim V. Zhytnikov) in the lhcyr package [262].

The titles for the "russian-lhcyrkoi" language contain characters that cannot be easily generated, hence we load russian-lhcyrkoi.mlo. The input encoding is KOI-8.

```
⟨*russian – lhcyrkoi⟩
\ProvidesFile{russian-lhcyrkoi.mld}[2006/03/13]
%% Russian-lhcyrkoi titles for minitoc.sty
%% From lhcyrkoi.sty in the LHCYR package
%% LHCYRKOI fonts in special encoding KOI-8
%% Vadim V. Zhytnikov (vvzhy@td.lpi.ac.ru)
\mtcloadmlo{russian-lhcyrkoi}%
⟨/russian – lhcyrkoi⟩
```

13.146 "Russian-lhcyrwin" language: russian-lhcyrwin.mld and russian-lhcyrwin.mlo

The russian titles for the LHCYRWIN fonts ("russian-lhcyrwin" language) are taken from the lhcyrwin.sty file (by Vadim V. Zhytnikov) in the lhcyr package [262].

The titles for the "russian-lhcyrwin" language contain characters that cannot be easily generated, hence we load russian-lhcyrwin.mlo. The input encoding is the code page CP1251.

```
⟨*russian – lhcyrwin⟩
\ProvidesFile{russian-lhcyrwin.mld}[2006/03/13]
%% Russian-lhcyrwin titles for minitoc.sty
%% From lhcyrwin.sty in the LHCYR package
%% LHCYRWIN fonts in special encoding CP1251
%% Vadim V. Zhytnikov (vvzhy@td.lpi.ac.ru)
\mtcloadmlo{russian-lhcyrwin}%
⟨/russian – lhcyrwin⟩
```

13.147 "Samin" language: samin.mld

The titles for the "samin" language [42] come from the samin.dtx file (by Regnor Jernsletten) in the babel package [38, 39, 61]. Specific fonts are required.

```
⟨*samin⟩
\ProvidesFile{samin.mld}[2006/01/13]
%% North Sámi (samin) titles for minitoc.sty
%% from samin.dtx (babel)
%% Jernsletten, Regnor
\def\ptctitle{Sisdoallu}%
\def\plftitle{Govvosat}%
\def\plttitle{Tabeallat}%
%%
\def\mtctitle{Sisdoallu}%
\def\mlftitle{Govvosat}%
\def\mlttitle{Tabeallat}%
%%
\def\stctitle{Sisdoallu}%
\def\slftitle{Govvosat}%
\def\slttitle{Tabeallat}%
⟨/samin⟩
```

13.148 "Scottish" language: scottish.mld

The titles for the "scottish" language come from the scottish.dtx file (by Fraser Grant) in the babel language [38, 39, 58]:

```
⟨*scottish⟩
\ProvidesFile{scottish.mld}[2006/01/13]
%% Scottish titles for minitoc.sty
%% from scottish.dtx (babel)
%% Grant, Fraser
\def\ptctitle{Cl\`ar-obrach}%
\def\plftitle{LiostaDhealbh}%
\def\plttitle{LiostaChl\`ar}%
%%
\def\mtctitle{Cl\`ar-obrach}%
\def\mlftitle{LiostaDhealbh}%
\def\mlttitle{LiostaChl\`ar}%
%%
\def\stctitle{Cl\`ar-obrach}%
\def\slftitle{LiostaDhealbh}%
\def\slttitle{LiostaChl\`ar}%
```

[42] Several Sámi 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 Sámi. Hence the (artificial) name samin.dtx (and not sami.dtx or the like) in the North Sámi case. This note is copied and adapted from the samin.dtx file. These dialects and languages are part of the Finnic group.

```
⟨/scottish⟩
```

13.149 "Serbian" language: serbian.mld

The titles for the "serbian" (serbocroatian) language are taken from the serbian.dtx file (by Dejan MUHAMEDAGIĆ and Jankovic SLOBODAN) in the babel package [38, 39, 70]. Serbocroatian is spoken by Serbs, Croats and Chernogors, but only Serbs and Chernogors use the cyrillic alphabet (a variant). See also section 13.150.

```
⟨*serbian⟩
\ProvidesFile{serbian.mld}[2006/01/13]
%% Serbian titles for minitoc.sty
%% from serbian.dtx (babel)
%% Muhamedagi\'{c}, Dejan and Slobodan, Jankovic
\def\ptctitle{Sadr\v{z}aj}%
\def\plftitle{Slike}%
\def\plttitle{Tabele}%
%%
\def\mtctitle{Sadr\v{z}aj}%
\def\mlftitle{Slike}%
\def\mlttitle{Tabele}%
%%
\def\stctitle{Sadr\v{z}aj}%
\def\slftitle{Slike}%
\def\slttitle{Tabele}%
⟨/serbian⟩
```

13.150 "Serbianc" language: serbianc.mld

The titles for the "serbianc" language[43] have been gently provided by Marko ÈEHAJA and Frank KÜSTER. Cyrillic fonts are required. Serbocroatian is spoken by Serbs, Croats and Chernogors, but only Serbs and Chernogors use the cyrillic alphabet (a variant). See also section 13.149.

```
⟨*serbianc⟩
\ProvidesFile{serbianc.mld}[2006/01/13]
%% Provides titles for minitoc.sty in Serbian Cyrillic
%%
%% Marko Èehaja Internut@Thetaworld.Org
%% Frank Küster, Biozentrum der Univ. Basel, frank@kuesterei.ch
%% Abt. Biophysikalische Chemie
\def\ptctitle{%
  {\cyr\CYRS\cyra\cyrd\cyrr\cyrzh\cyra\cyrje}}%
```

[43] The "serbianc" language is written with cyrillic characters.

```
\def\plftitle{%
  {\cyr\CYRS\cyrl\cyri\cyrk\cyre}}%
\def\plttitle{\CYRT\cyra\cyrb\cyrl\cyri\cyrc\cyre}%
%%
\def\mtctitle{%
  {\cyr\CYRS\cyra\cyrd\cyrr\cyrzh\cyra\cyrje}}%
\def\mlftitle{%
  {\cyr\CYRS\cyrl\cyri\cyrk\cyre}}%
\def\mlttitle{\cyr\CYRT\cyra\cyrb\cyrl\cyri\cyrc\cyre}%
%%
\def\stctitle{%
  {\cyr\CYRS\cyra\cyrd\cyrr\cyrzh\cyra\cyrje}}%
\def\slftitle{%
  {\cyr\CYRS\cyrl\cyri\cyrk\cyre}}%
\def\slttitle{\CYRT\cyra\cyrb\cyrl\cyri\cyrc\cyre}%
⟨/serbianc⟩
```

13.151 “Slovak” language: slovak.mld

The titles for the “slovak” language are taken from the slovak.dtx file (by Jana Chlebikova and Tobias Schlemmer) in the babel package [38, 39, 54]. Using the T1 font encoding is recommended.

```
⟨*slovak⟩
\ProvidesFile{slovak.mld}[2006/01/13]
%% Slovak titles for minitoc.sty
%% from slovak.dtx (babel)
%% Chlebikova, Jana and Schlemmer, Tobias
\def\ptctitle{Obsah}%
\def\plftitle{Zoznam obr\'azkov}%
\def\plttitle{Zoznam tabuliek}%
%%
\def\mtctitle{Obsah}%
\def\mlftitle{Zoznam obr\'azkov}%
\def\mlttitle{Zoznam tabuliek}%
%%
\def\stctitle{Obsah}%
\def\slftitle{Zoznam obr\'azkov}%
\def\slttitle{Zoznam tabuliek}%
⟨/slovak⟩
```

13.152 “Slovene” language: slovene.mld

The titles for the “slovene” language come from the slovene.dtx file (by Danilo Zavrtanik and Leon Žlajpah) in the babel package [38, 39, 84]:

```
⟨*slovene⟩
\ProvidesFile{slovene.mld}[2006/01/13]
%% Slovene titles for minitoc.sty
%% from slovene.dtx (babel)
%%  Zavrtanik, Danilo and \v{Z}lajpah, Leon
\def\ptctitle{Kazalo}%
\def\plftitle{Slike}%
\plttitle{Tabele}%
%%
\def\mtctitle{Kazalo}%
\def\mlftitle{Slike}%
\def\mlttitle{Tabele}%
%%
\def\stctitle{Kazalo}%
\def\slftitle{Slike}%
\def\slttitle{Tabele}%
⟨/slovene⟩
```

13.153 "Spanish" language: spanish.mld

The titles for the "spanish" language are taken from the spanish.dtx file (by Javier Bezos, initialy by Julio Sánchez) in the babel package [38, 39, 31]. Note that the "spanish" language is in fact "castillan" (see section 13.33 on page 449). Other languages are spoken in Spain: "basque" (section 13.18 on page 441), "catalan" (section 13.35 on page 449), and "galician" (section 13.64 on page 464). Note that "spanish2" is a version of "spanish" with shorter titles (see section 13.154 on the next page). And "spanish3" (see section 13.155 on the following page) is a version for the Antomega [150] project; some titles are differents. And "spanish4" is a variant of "spanish" where \ptctitle is shorter for articles (section 13.156 on page 513).

```
⟨*spanish⟩
\ProvidesFile{spanish.mld}[2006/02/15]
%% Spanish titles for minitoc.sty
%% from spanish.dtx (babel)
%% Bezos, Javier
\expandafter\ifx\csname chapter\endcsname\relax
\def\ptctitle{\'Indice}%
\else
\def\ptctitle{\'Indice General}%
\fi
\def\plftitle{\'Indice de Figuras}%
\def\plttitle{\'Indice de Tablas}%
%%
\def\mtctitle{\'Indice}%
\def\mlftitle{\'Indice de Figuras}%
\def\mlttitle{\'Indice de Tablas}%
%%
\def\stctitle{\'Indice}%
\def\slftitle{\'Indice de Figuras}%
```

```
\def\slttitle{\'Indice de Tablas}%
⟨/spanish⟩
```

13.154 "Spanish2" language: spanish2.mld

The titles for the "spanish2" language are taken from the spanish.dtx file in the babel package [38, 39, 31], but made shorter for chapter and section levels. See section 13.153 on the preceding page.

```
⟨*spanish2⟩
\ProvidesFile{spanish2.mld}[2005/03/31]
%% Spanish titles for minitoc.sty
\def\ptctitle{\'Indice General}%
\def\plftitle{\'Indice de Figuras}%
\def\plttitle{\'Indice de Tablas}%
%%
\def\mtctitle{Contenido}%
\def\mlftitle{Figuras}%
\def\mlttitle{Tablas}%
%%
\def\stctitle{Contenido}%
\def\slftitle{Figuras}%
\def\slttitle{Tablas}%
⟨/spanish2⟩
```

13.155 "Spanish3" language: spanish3.mld

The titles for the "spanish3" language are taken from the omega-spanish.ldf file of the Antomega project [150]. See section 13.153 on the page before.

```
⟨*spanish3⟩
\ProvidesFile{spanish3.mld}[2005/09/06]
%% Spanish titles for minitoc.sty
%% from omega-spanish.ldf of the \pack{Antomega} project.
\def\ptctitle{\localspanish{^^^^00cdndice general}}%
\def\plftitle{\localspanish{^^^^00cdndice de figuras}}%
\def\plttitle{\localspanish{^^^^00cdndice de cuadros}}%
%%
\def\mtctitle{\localspanish{^^^^00cdndice general}}%
\def\mlftitle{\localspanish{^^^^00cdndice de figuras}}%
\def\mlttitle{\localspanish{^^^^00cdndice de cuadros}}%
%%
\def\stctitle{\localspanish{^^^^00cdndice general}}%
\def\slftitle{\localspanish{^^^^00cdndice de figuras}}%
\def\slttitle{\localspanish{^^^^00cdndice de cuadros}}%
⟨/spanish3⟩
```

13.156 "Spanish4" language: spanish4.mld

The titles for the "spanish4" language are taken from the spanish.dtx file (by Javier Bezos) from the CervanTEX package [30]. The title of the parttocs is shorter for articles. See also section 13.153 on page 511.

```
⟨*spanish4⟩
\ProvidesFile{spanish4.mld}[2006/01/19]
%% Spanish titles for minitoc.sty
%% from spanish.dtx (CervanTeX)
%% Bezos, Javier
\expandafter\ifx\csname chapter\endcsname\relax
\def\ptctitle{\'Indice}%
\else
\def\ptctitle{\'Indice general}%
\fi
\def\plftitle{\'Indice de figuras}%
\def\plttitle{\'Indice de cuadros}%
%%
\def\mtctitle{\'Indice}%
\def\mlftitle{\'Indice de figuras}%
\def\mlttitle{\'Indice de cuadros}%
%%
\def\stctitle{\'Indice}%
\def\slftitle{\'Indice de figuras}%
\def\slttitle{\'Indice de cuadros}%
⟨/spanish4⟩
```

13.157 "Swedish" language: swedish.mld

The titles for the "swedish" language are taken from the swedish.dtx file (by Sten Hellman and Erik Österhols, with a correction by Jan Michel Rynning) in the babel package [38, 39, 59]. See also section 13.158 on the next page.

```
⟨*swedish⟩
\ProvidesFile{swedish.mld}[2006/01/13]
%% Swedish titles for minitoc.sty
%% from swedish.dtx (babel)
%% Hellman, Sten and Österhols, Erik
\def\ptctitle{Inneh\csname aa\endcsname ll}%
\def\plftitle{Figurer}%
\def\plttitle{Tabeller}%
%%
\def\mtctitle{Inneh\csname aa\endcsname ll}%
\def\mlftitle{Figurer}%
\def\mlttitle{Tabeller}%
%%
\def\stctitle{Inneh\csname aa\endcsname ll}%
```

```
\def\slftitle{Figurer}%
\def\slttitle{Tabeller}%
⟨/swedish⟩
```

13.158 "Swedish2" language: swedish2.mld

The titles for the "swedish2" language (variant for swedish) are taken from the rapport.doc file (by Sven Mattisson) in the SlaTeX package [181]. See also section 13.157 on the preceding page.

```
⟨*swedish2⟩
\ProvidesFile{swedish2.mld}[2006/04/04]
%% Swedish2 titles for minitoc.sty
%% from rapport.doc (slatex)
%% Mattisson, Sven
\def\ptctitle{Inneh\csname aa\endcsname ll}%
\def\plftitle{Figurf\"orteckning}%
\def\plttitle{Tabellf\"orteckning}%
%%
\def\mtctitle{Inneh\csname aa\endcsname ll}%
\def\mlftitle{Figurf\"orteckning}%
\def\mlttitle{Tabellf\"orteckning}%
%%
\def\stctitle{Inneh\csname aa\endcsname ll}%
\def\slftitle{Figurf\"orteckning}%
\def\slttitle{Tabellf\"orteckning}%
⟨/swedish2⟩
```

13.159 "Thai" language: thai.mld and thai.mlo

The titles for the "thai" language come from the thaicjk.ldf file (by Werner Lemberg) and use fonts of the CJK system [167, 168]. The thailatex package [183] (by Surapant Meknavin, Theppitak Karoonboonyanan, Chanop Silpa-Anan and Veerathanabutr Poonlap) provides the same titles in its thai.ldf file.

The titles for the "thai" language contain characters that cannot be easily generated, hence we load thai.mlo.

```
⟨*thai⟩
\ProvidesFile{thai.mld}[2005/01/28]
%% from thaicjk.ldf CJK 4.5.2 Thai support for the babel system
%% by Werner Lemberg <wl@gnu.org>
%%
\mtcloadmlo{thai}%
⟨/thai⟩
```

13.160 "Turkish" language: turkish.mld

The titles for the "turkish" language are taken from the turkish.dtx file (by Mustafa Burc, Pierre Mackay and Turgut Uyar) in the babel package [38, 39, 51]:

```
⟨*turkish⟩
\ProvidesFile{turkish.mld}[2006/01/13]
%% Turkish titles for minitoc.sty
%% from turkish.dtx (babel)
%% Burc, Mustafa
\def\ptctitle{\.I\c cindekiler}%
\def\plftitle{\c Sekiller Listesi}%
\def\plttitle{Tablolar\in Listesi}%
%%
\def\mtctitle{\.I\c cindekiler}%
\def\mlftitle{\c Sekiller Listesi}%
\def\mlttitle{Tablolar\in Listesi}%
%%
\def\stctitle{\.I\c cindekiler}%
\def\slftitle{\c Sekiller Listesi}%
\def\slttitle{Tablolar\in Listesi}%
⟨/turkish⟩
```

13.161 "Uighur" language: uighur.mld

The "uighur" and "bicig" languages are synonyms, so we just load the bicig.mld file (see section 13.19 on page 442):

```
⟨*uighur⟩
\ProvidesFile{uighur.mld}[2006/05/31]
\mtcselectlanguage{bicig}%
⟨/uighur⟩
```

13.162 "Uighur2" language: uighur2.mld

The "uighur2" and "bicig2" languages are synonyms, so we just load the bicig2.mld file (see section 13.20 on page 442):

```
⟨*uighur2⟩
\ProvidesFile{uighur2.mld}[2006/05/31]
\mtcselectlanguage{bicig2}%
⟨/uighur2⟩
```

13.163 "Uighur3" language: uighur3.mld

The "uighur3" and "bicig3" languages are synonyms, so we just load the bicig3.mld file (see section 13.19 on page 442):

```
⟨*uighur3⟩
\ProvidesFile{uighur3.mld}[2006/05/31]
\mtcselectlanguage{bicig3}%
⟨/uighur3⟩
```

13.164 "UKenglish" language: UKenglish.mld

The "UKenglish" language is just like "english" ("UK" is for "United Kingdom"), so we just load english.mld (see section 13.43 on page 454):

```
⟨*UKenglish⟩
\ProvidesFile{UKenglish.mld}[2005/07/11]
\mtcselectlanguage{english}%
⟨/UKenglish⟩
```

13.165 "Ukraineb" language: ukraineb.mld

The "ukraineb" language is a synonym for "ukrainian", so we just load ukrainian.mld. See section 13.166.

```
⟨*ukraineb⟩
\ProvidesFile{ukraineb.mld}[2006/12/19]
%% Ukranian titles for minitoc.sty
%% from ukraineb.dtx (babel)
%% Andrij Shvaika, Olga Lapko
\selectlanguage{ukrainian}%
⟨/ukraineb⟩
```

13.166 "Ukrainian" language: ukrainian.mld

The titles for the "ukrainian" language come from the ukraineb.dtx file (by Olga G. Lapko and Andrij M. Shvaika) in the babel package [38, 39, 79]. Cyrillic fonts are required. Another language name is ukraineb (see section 13.165).

```
⟨*ukrainian⟩
\ProvidesFile{ukrainian.mld}[2006/01/33]
%% Ukraine titles for minitoc.sty
%% from ukraineb.dtx (babel)
%% Shvaika, Andrij and Lapko, Olga
%% Needs cyrillic fonts
\def\mtctitle{{\cyr\CYRZ\cyrm\cyrii\cyrs\cyrt}}%
\def\mlftitle{{\cyr\CYRP\cyre\cyrr\cyre\cyrl\cyrii\cyrk
        \ \cyrii\cyrl\cyryu\cyrs\cyrt\cyrr\cyra\cyrc\cyrii\cyrishrt}}%
\def\mlttitle{{\cyr\CYRP\cyre\cyrr\cyre\cyrl\cyrii\cyrk
        \ \cyrt\cyra\cyrb\cyrl\cyri\cyrc\cyrsftsn}}%
%%
\def\ptctitle{{\cyr\CYRZ\cyrm\cyrii\cyrs\cyrt}}%
\def\plftitle{{\cyr\CYRP\cyre\cyrr\cyre\cyrl\cyrii\cyrk
        \ \cyrii\cyrl\cyryu\cyrs\cyrt\cyrr\cyra\cyrc\cyrii\cyrishrt}}%
\def\plttitle{{\cyr\CYRP\cyre\cyrr\cyre\cyrl\cyrii\cyrk
        \ \cyrt\cyra\cyrb\cyrl\cyri\cyrc\cyrsftsn}}%
%%
\def\stctitle{{\cyr\CYRZ\cyrm\cyrii\cyrs\cyrt}}%
\def\slftitle{{\cyr\CYRP\cyre\cyrr\cyre\cyrl\cyrii\cyrk
        \ \cyrii\cyrl\cyryu\cyrs\cyrt\cyrr\cyra\cyrc\cyrii\cyrishrt}}%
\def\slttitle{{\cyr\CYRP\cyre\cyrr\cyre\cyrl\cyrii\cyrk
        \ \cyrt\cyra\cyrb\cyrl\cyri\cyrc\cyrsftsn}}%
⟨/ukrainian⟩
```

13.167 “Uppersorbian” language: uppersorbian.mld

The titles for the “uppersorbian” language[44] are taken from the usorbian.dtx file (by Eduard Werner) in the babel package [38, 39, 82]. See also section 13.106 on page 487. A shorter language name is usorbian (see section 13.169 on the following page).

```
⟨*uppersorbian⟩
\ProvidesFile{uppersorbian.mld}[2006/02/38]
%% Upper sorbian titles for minitoc.sty
%% from usorbian.dtx (babel)
%% Werner, Eduard
%% Needs cyrillic fonts
\def\ptctitle{Wobsah}%
\def\plftitle{Zapis wobrazow}%
\def\plttitle{Zapis tabulkow}%
%%
\def\mtctitle{Wobsah}%
\def\mlftitle{Zapis wobrazow}%
\def\mlttitle{Zapis tabulkow}%
%%
\def\stctitle{Wobsah}%
```

[44] 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 (VI-th century).

```
\def\slftitle{Zapis wobrazow}%
\def\slttitle{Zapis tabulkow}%
⟨/uppersorbian⟩
```

13.168 "USenglish" language: USenglish.mld

The "USenglish" language ("US" is for "United States (of America))" is just like "english"[45], so we just load english.mld (see section 13.43 on page 454):

```
⟨*USenglish⟩
\ProvidesFile{USenglish.mld}[2005/07/11]
\mtcselectlanguage{english}%
⟨/USenglish⟩
```

13.169 "Usorbian" language: usorbian.mld

The "usorbian" language is a synonym for "uppersorbian", so we just load uppersorbian.mld. See section 13.167 on the preceding page.

```
⟨*usorbian⟩
\ProvidesFile{usorbian.mld}[2006/01/23]
%% Upper sorbian titles for minitoc.sty
%% from usorbian.dtx (babel)
%% Werner, Eduard
\selectlanguage{uppersorbian}%
⟨/usorbian⟩
```

13.170 "Vietnam" language: vietnam.mld

The titles for the "vietnam" language are taken from the vietnam package. Vietnamese fonts are required. See also section 13.171 on the next page.

```
⟨*vietnam⟩
\ProvidesFile{vietnam.mld}[1999/03/16]
%% vietnamese titles for minitoc.sty
%%
\def\ptctitle{M\d{u}c l\d{u}c}%
\def\plftitle{Danh s\'ach h\`inh v\~e}%
\def\plttitle{Danh s\'ach b\h{a}ng}%
```

[45] It should be true for the mini-table titles; the languages themselves have some differences, like the hyphenation rules.

```
%%
\def\mtctitle{M\d{u}c l\d{u}c}%
\def\mlftitle{Danh s\'ach h\`inh v\~e}%
\def\mlttitle{Danh s\'ach b\h{a}ng}%
%%
\def\stctitle{M\d{u}c l\d{u}c}%
\def\slftitle{Danh s\'ach h\`inh v\~e}%
\def\slttitle{Danh s\'ach b\h{a}ng}%
⟨/vietnam⟩
```

13.171 "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 13.170 on the preceding page.

```
⟨*vietnamese⟩
\ProvidesFile{vietnamese.mld}[2004/12/14]
\mtcselectlanguage{vietnam}%
⟨/vietnamese⟩
```

13.172 "Welsh" language: welsh.mld

The titles for the "welsh" language come from the welsh.dtx file (by Johannes Braams) in the babel package [38, 39, 44]:

```
⟨*welsh⟩
\ProvidesFile{welsh.mld}[1999/12/06]
%% Welsh titles for minitoc.sty
%% from welsh.dtx (babel)
%% Braams, Johannes
\def\ptctitle{Cynnwys}%
\def\plftitle{Rhestr Ddarluniau}%
\def\plttitle{Rhestr Dablau}%
%%
\def\mtctitle{Cynnwys}%
\def\mlftitle{Rhestr Ddarluniau}%
\def\mlttitle{Rhestr Dablau}%
%%
\def\stctitle{Cynnwys}%
\def\slftitle{Rhestr Ddarluniau}%
\def\slttitle{Rhestr Dablau}%
⟨/welsh⟩
```

13.173 "Xalx" language: xalx.mld

The titles for the "xalx" language are taken from the MonTEX package [97, 100]. Xalx or Khalkha is the name of the Mongolian nationality residing in Mongolia proper. Their dialect forms the basis of Mongolian written with Cyrillic letters. See also sections 13.100 on page 484, 13.174, and 13.175 on the next page.

```
⟨*xalx⟩
\ProvidesFile{xalx.mld}[2005/11/16]
%% Mongol (xalx) titles for minitoc.sty
\def\ptctitle{{\mnr Garqig}}%
\def\plftitle{{\mnr Zurgi"in jagsaalt}}%
\def\plttitle{{\mnr X"usn"agti"in jagsaalt}}%
%%
\def\mtctitle{{\mnr Garqig}}%
\def\mlftitle{{\mnr Zurgi"in jagsaalt}}%
\def\mlttitle{{\mnr X"usn"agti"in jagsaalt}}%
%%
\def\stctitle{{\mnr Garqig}}%
\def\slftitle{{\mnr Zurgi"in jagsaalt}}%
\def\slttitle{{\mnr X"usn"agti"in jagsaalt}}%
⟨/xalx⟩
```

13.174 "Xalx2" language: xalx2.mld

The titles for the "xalx2" language are taken from the MonTEX package [97, 100]. This is a variant for the "xalx" language (see section 13.173).

```
⟨*xalx2⟩
\ProvidesFile{xalx2.mld}[2006/03/31]
%% Mongol (xalx2) titles for minitoc.sty
\def\ptctitle{{\mnr Aguulga}}%
\def\plftitle{{\mnr Zurgi"in jagsaalt}}%
\def\plttitle{{\mnr X"usn"agti"in jagsaalt}}%
%%
\def\mtctitle{{\mnr Aguulga}}%
\def\mlftitle{{\mnr Zurgi"in jagsaalt}}%
\def\mlttitle{{\mnr X"usn"agti"in jagsaalt}}%
%%
\def\stctitle{{\mnr Aguulga}}%
\def\slftitle{{\mnr Zurgi"in jagsaalt}}%
\def\slttitle{{\mnr X"usn"agti"in jagsaalt}}%
⟨/xalx2⟩
```

13.175 “Xalx3” language: xalx3.mld

The titles for the “xalx3” language are taken from the MonTEX package [97, 100]. This is an other variant for the “xalx” language (see section 13.173 on the page before).

```
⟨*xalx3⟩
\ProvidesFile{xalx3.mld}[2006/03/31]
%% Mongol (xalx3) titles for minitoc.sty
\def\ptctitle{\xalx{Soderjanie}}%
\def\plftitle{\xalx{Spisok risunkow}}%
\def\plttitle{\xalx{Spisok tablic}}%
%%
\def\mtctitle{\xalx{Soderjanie}}%
\def\mlftitle{\xalx{Spisok risunkow}}%
\def\mlttitle{\xalx{Spisok tablic}}%
%%
\def\stctitle{\xalx{Soderjanie}}%
\def\slftitle{\xalx{Spisok risunkow}}%
\def\slttitle{\xalx{Spisok tablic}}%
⟨/xalx3⟩
```

Part III

Complements

Contents of the Third Part

Bibliography

[1] Paul W. Abrahams, Karl Berry, and Kathryn A. Hargreaves. *TEX for the Impatient*, September 2003. mailto:impatient@tug.org. Available from: http://www.dante.de/tex-archive/info/impatient/book.pdf. 524

[2] Paul W. Abrahams, Karl Berry, and Kathryn A. Hargreaves. *TEX pour l'Impatient*, September 2004. French translation of [1] by Marc Chaudemanche. Available from: http://www.dante.de/tex-archive/info/impatient/fr/fbook.pdf.

[3] Alex AJ. *Typesetting Malayalam Using LaTeX 2ε*, April 2003. mailto:alexaj@myrealbox.com. Available from: http://www.dante.de/tex-archive/language/malayalam/doc/mm-usr.pdf. 41, 191, 434, 490, 491

[4] Alex AJ. *Typesetting Malayalam Using Ω*, January 2006. mailto:indicTeX@gmail.com. Available from: http://www.dante.de/tex-archive/macros/omega/latex/contrib/malayalam-omega/doc/manual-malayalam.pdf. 41, 191, 434, 490

[5] Mark Alford. *How to create Japanese language documents under GNU/Linux using LaTeX* [online], March 2006. Available from: http://www.physics.wustl.edu/~alford/tex/japanese_latex.html. mailto:alford@wuphys.wustl.edu.

[6] American Mathematical Society. *Instructions for Preparation of Papers and Monographs: AMS-LaTeX*. Providence, Rhode Island, 1999. Available from: http://www.dante.de/tex-archive/macros/latex/required/amslatex/classes/instr-l.pdf. 85

[7] Anonymous. *About Kanji* [online], June 1999. Available from: http://www2.gol.com/users/jpc/Japan/Kanji/. mailto:jpc@gol.com.

[8] Walter Appel, Céline Chevalier, Emmanuel Cornet, Sébastien Desreux, Jean-Julien Fleck, and Paul Pichaureau. *LaTeX pour l'impatient*. In Céline Chevalier, editor, *Technique & Pratique*. H & K, 68, boulevard de Port-Royal, 75005 Paris, July 2005. mailto:contact@H-K.fr, see http://www.h-k.fr/liens/tp/latex_pour_l_impatient.html.

[9] Donald Arseneau. *The* `notoccite` *package*, July 2000. mailto:asnd@triumf.ca. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/misc/notoccite.sty. 49, 85, 214, 230, 249, 274, 277, 280, 375, 408

[10] Donald Arseneau. *The* `placeins` *package*, June 2002. mailto:asnd@triumf.ca. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/placeins/. 29, 85, 202, 204, 216, 230, 385, 391, 407

[11] Donald Arseneau. *The* `wrapfig` *package*, January 2003. mailto:asnd@triumf.ca. Available from: http://www.dante.de/tex-archive/contrib/macros/latex/contrib/wrapfig/. 130, 211, 221

[12] Donald Arseneau. `chapterbib.sty`, February 2004. mailto:asnd@reg.triumf.ca. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/cite/chapterbib.sty. 57, 196, 211

[13] Helmer Aslaksen. *Chinese TEX Using the CJK LATEX Package, Unicode TrueType Fonts and pdfTEX under Windows* [online], February 2006. Available from: http://www.math.nus.edu.sg/aslaksen/cs/cjk.html. mailto:aslaksen@math.nus.edu.sg.

[14] Helmer Aslaksen. *Reading and Writing Chinese Characters and Pinyin on the Web Using Unicode* [online], January 2006. Available from: http://www.math.nus.edu.sg/aslaksen/read.html. mailto:aslaksen@math.nus.edu.sg.

[15] Iain Menzies Banks. *Against a Dark Background*. Bantam Books, 1993. 94

[16] Pierre Basso and Stephan Ulrich. `bibtopic.sty`, August 2002. mailto:Pierre.Basso@lim.univ-mrs.fr, mailto:stefanulrich@users.sourceforge.net. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/bibtopic/bibtopic.pdf. 57, 195

[17] David Bausum. *TEX: Reference and Examples*. Kluwer Academic Publishers, mailto:davidb@jvlnet.com, April 2002. See [18]. 525

[18] David Bausum. *TEX: Reference and Examples* [online], January 2006. Available from: http://www.tug.org/utilities/plain/trm.html. mailto:davidb@jvlnet.com, see [17]. 525

[19] Benjamin Bayart. *Joli manuel pour LATEX 2ε*. Guide local pour l'ESIEE, December 1995. Available from: http://www.dante.de/tex-archive/info/JMPL.ps.gz. 46

[20] Benjamin Bayart, Thierry Bayet, Prakash Countcham, Éric Depardieu, Jean-Pierre F. Drucbert, Mathieu Goutelle, Yvon Henel, Florence Henry, Loïc Joly, Christophe Jorssen, Erwan Le Pennec, Nicolas Markey, Françoise Marre-Fournier, Sébastien Mengin, Josselin Noirel, Bruno Piguet, Éric Streit, Thomas van Oudenhove de Saint Géry, and Damien Wyart. *Foire aux questions du groupe* `fr.comp.text.tex` *(version 3.0α)* [online], November 2004. Available from: http://faqfctt.fr.eu.org/. mailto:bayartb@edgard.fdn.fr.

[21] Claudio Beccari and Apostolos Syropoulos. *New Greek Fonts and the* `greek` *option of the* `babel` *package*. *TUGboat*, 19(4):419–425, December 1998. Available from: http://www.tug.org/TUGboat/Articles/tb19-4/tb61becc.pdf.

[22] Emmanuel Beffara. *Rubber Manual 1.0* [online], July 2005. Available from: http://www.pps.jussieu.fr/~beffara/soft/rubber/doc.html. 58, 217

[23] Alexander Berdnikov and Olga A. Grineva. *Some Problems with Accents in TEX: Letters with Multiple Accents and Accents Varying for Uppercace/Lowercase Letters*. In *Congrès EuroTEX'98*, volume 28-29 of *Cahiers GUTenberg*, pages 44–55, St. Malo, March 1998. Available from: http: www.gutenberg.eu.org/pub/GUTenberg/publicationsPDF/28-29-berdnikovc.pdf.

[24] Alexander Berdnikov, Olga G. Lapko, Mikhail Kolodin, Andrew Janishevsky, and Alexey Burykin. *Alphabets Necessary for Various Cyrillic Writing Systems (Towards X2 and T2 Encodings)*. In *Congrès EuroTEX'98*, volume 28-29 of *Cahiers GUTenberg*, pages 33–43, St. Malo, March 1998. Available from: http: www.gutenberg.eu.org/pub/GUTenberg/publicationsPDF/28-29-berdnikovb.pdf.

[25] Alexander Berdnikov, Olga G. Lapko, Mikhail Kolodin, Andrew Janishevsky, and Alexey Burykin. *Cyrillic encoding for LaTeX 2ε multi-language documents*. *TUGboat*, 19(4):403–416, 1998. Available from: http://www.tug.org/TUGboat/Articles/tb19-4/tb61berd.pdf.

[26] Alexander Berdnikov, Olga G. Lapko, Mikhail Kolodin, Andrew Janishevsky, and Alexey Burykin. *The Encoding Paragigm in LaTeX 2ε and the Projected X2 Encoding for Cyrillic Texts*. In *Congrès EuroTEX*, volume 28-29 of *Cahiers GUTenberg*, pages 17–31, March 1998. Available from: http: www.gutenberg.eu.org/pub/GUTenberg/publicationsPDF/28-29-berdnikova.pdf.

[27] Karl Berry. *Filenames for fonts*. *TUGboat*, 11(4):517–520, November 1990. Available from: http://www.tug.org/TUGboat/Articles/tb11-4/tb30berry.pdf. 539

[28] Karl Berry. *Fontname*, September 2005. Available from: http://www.tug.org/fontname/fontname.pdf.

[29] Berhanu Beyene, Manfred Kudlek, Olaf Kummer, and Jochen Metzinger. *Ethiopian Language Support for the Babel Package*. Universität Aamburg, December 1998. Available from: http://www.dante.de/tex-archive/language/ethiopia/ethiop/doc/ethiodoc.ps. 41, 200, 434, 457, 458

[30] Javier Bezos. *Estilo spanish para el sistema babel*, September 2003. mailto:jbezos@wanadoo.es. Available from: http://filemon.mecanica.upm.es/CervanTeX/spanish.dtx. 41, 513

[31] Javier Bezos. *Spanish support from the babel system*, February 2004. mailto:jbezos@wanadoo.es or mailto:babel@braams.cistron.nl. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/spanish.dtx. 511, 512

[32] Javier Bezos. *Typesetting Guaraní with TEX*, July 2004. Available from: http://www.dante.de/tex-archive/language/guarani/guarani.pdf. 41, 203, 434, 470

[33] Javier Bezos. *The `titlesec` and `titletoc` packages*, January 2005. http://www.texytipografia.com/contact.php. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/titlesec/titlesec.pdf. 85, 219, 384

[34] Giuseppe Bilotta. *The ℵ (Aleph) Project*. *TUGboat*, 25(1):105–107, July 2004. Available from: http://www.tug.org/TUGboat/Articles/tb25-1/bilotta.pdf. 215

[35] Denis Bitouzé and Jean-Côme Charpentier. *LaTeX*. In *Collection Synthex*. Pearson Education France, September 2006.

[36] Victor Boyko. *The `russian.sty` file*, January 1995. Available from: ftp://ftp.dante.de/tex-archive/macros/latex/contrib/cmcyralt/russian.sty. 505

[37] Johannes Braams. *Babel, a multilingual style-option system for use with LaTeX's standard document styles*. *TUGboat*, 12(2):291–301, June 1991. Available from: http://www.tug.org/TUGboat/Articles/tb12-2/tb32braa.pdf. 434

[38] Johannes Braams. *Babel, a multilingual package for use with LaTeX's standard document classes*, June 2004. mailto:babel@bramms.cistron.nl. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/babel.pdf. 36, 41, 71, 194, 195, 200, 228, 434, 435, 436, 440, 441, 444, 445, 446, 447, 449, 451, 452, 453, 454, 455, 456, 460, 461, 462, 463, 464, 465, 466, 475, 477, 478, 479, 480, 484, 485, 486, 487, 488, 489, 494, 495, 496, 497, 499, 500, 501, 503, 508, 509, 510, 511, 512, 513, 515, 516, 517, 519

[39] Johannes Braams. *Babel, a multilingual package for use with LaTeX's standard document classes (user)*, June 2004. mailto:babel@bramms.cistron.nl. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/user.pdf. 36, 41, 71, 194, 195, 200, 228, 434, 435, 436, 440, 441, 444, 445, 446, 447, 449, 451, 452, 453, 454, 455, 456, 460, 461, 462, 463, 464, 465, 466, 475, 477, 478, 479, 480, 484, 485, 486, 487, 488, 489, 494, 495, 496, 497, 499, 500, 501, 503, 508, 509, 510, 511, 512, 513, 515, 516, 517, 519

[40] Johannes Braams. *Dutch language support from the babel system*, March 2005. mailto:babel@bramms.cistron.nl. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/dutch.dtx. 435, 453

[41] Johannes Braams. *English support from the babel system*, March 2005. mailto:babel@bramms.cistron.nl. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/english.dtx. 454, 455

[42] Johannes Braams. *Irish support from the babel system*, March 2005. mailto:babel@bramms.cistron.nl. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/irish.dtx. 479

[43] Johannes Braams. *Norsk support from the babel system*, March 2005. mailto:babel@bramms.cistron.nl. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/norsk.dtx. 495, 496

[44] Johannes Braams. *Welsh support from the babel system*, March 2005. mailto:babel@bramms.cistron.nl. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/welsh.dtx. 519

[45] Johannes Braams, Juan M. Aguirregabiria, and Julio Sanchez. *Basque support from the babel system*, March 2005. mailto:babel@bramms.cistron.nl, mailto:lg.ehu.es. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/basque.dtx. 441

[46] Johannes Braams and Einar Árnason. *Icelandic support from the babel system*, March 2005. mailto:babel@bramms.cistron.nl, mailto:einar@lif.hi.is. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/icelandic.dtx. 477

[47] Johannes Braams and Goncal Badenes. *Catalan language support from the babel system*, March 2005. mailto:babel@bramms.cistron.nl, mailto:badenes@imec.be. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/catalan.dtx. 449

[48] Johannes Braams and Claudio Beccari. *Latin support from the babel system*, November 2005. mailto:babel@bramms.cistron.nl, mailto:claudio.beccari@polito.it. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/latin.dtx. 484, 485

[49] Johannes Braams, Árpád Bíró, and József Bérces. *Magyar support from the babel system*, March 2005. mailto:babel@bramms.cistron.nl. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/magyar.dtx. 488, 489

[50] Johannes Braams and Georgi Boshnakov. *Bulgarian language support from the babel system*, March 2005. mailto:babel@bramms.cistron.nl, mailto:georgi.boshnakov@umist.ac.uk. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/bulgarian.dtx. 446

[51] Johannes Braams and Mustafa Burc. *Turkish support from the babel system*, March 2005. mailto:babel@bramms.cistron.nl, mailto:rz6001@rziris01.rrz.uni-hamburg.de. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/turkish.dtx. 515

[52] Johannes Braams, David Carlisle, Alan Jeffrey, Leslie Lamport, Frank Mittelbach, Chris Rowley, and Rainer Schöpf. *The LaTeX 2ε Sources*, 2003. Available from: http://www.dante.de/tex-archive/macros/latex/base.

[53] Johannes Braams and Manuel Carriba. *Galician support from the babel system*, March 2005. mailto:babel@bramms.cistron.nl, mailto:mcarriba@eunetcom.net. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/galician.dtx. 464

[54] Johannes Braams, Jana Chlebikova, and Tobias Schlemmer. *Slovak support from the babel system*, March 2005. mailto:babel@bramms.cistron.nl, mailto:chlebikj@mff.uniba.cs, mailto:Tobias.Schlemmer@web.de. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/slovak.dtx. 510

[55] Johannes Braams, Maurizio Codogno, and Claudio Beccari. *Italian support from the babel system*, November 2005. mailto:babel@bramms.cistron.nl, mailto:mau@beatles.cselt.stet.it, mailto:beccari@polito.it. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/italian.dtx. 480

[56] Johannes Braams, Victor Eijkhout, and Nico Poppelier. *The development of national LaTeX styles*. *TUGboat*, 10(3):401–406, November 1989. Available from: http://www.tug.org/TUGboat/Articles/tb10-3/tb25braams.pdf. 434

[57] Johannes Braams and Daniel Flipo. *French support from the babel system*, February 2005. mailto:babel@bramms.cistron.nl, mailto:Daniel.Flipo@univ-lille1.fr. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/frenchb.dtx. 461, 462, 463

[58] Johannes Braams and Fraser Grant. *Scottish support from the babel system*, March 2005. mailto:babel@bramms.cistron.nl, mailto:fraser@cernvm. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/scottish.dtx. 508

[59] Johannes Braams and Sten Hellman. *Swedish support from the babel system*, March 2005. mailto:babel@bramms.cistron.nl, mailto:hellman@cernvm.cern.ch. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/swedish.dtx. 513

[60] Johannes Braams, Umstatter Horst, and Robert Juhasz. *Romanian support from the babel system*, March 2005. mailto:babel@bramms.cistron.nl, mailto:hhu@cernvm.cern.ch, mailto:robertj@uni-paderborn.de. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/romanian.dtx. 499

[61] Johannes Braams and Regnor Jernsletten. *North Sami support from the babel system*, February 2004. mailto:babel@bramms.cistron.nl, mailto:Regnor.Jernsletten@sami.uit.no or mailto:Regnor.Jernsletten@eunet.no. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/samin.dtx. 508

[62] Johannes Braams, Mikko Kanerva, and Keranen Reino. *Finnish support from the babel system*, March 2005. mailto:babel@bramms.cistron.nl, mailto:kanerva@cernvm, mailto:keranen@cernvm. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/finnish.dtx. 460, 461

[63] Johannes Braams and Peter Kleiweg. *Interlingua support from the babel system*, March 2005. mailto:babel@bramms.cistron.nl, mailto:kleiweg@let.rug.nl. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/interlingua.dtx. 478

[64] Johannes Braams, Jörg Knappen, and Terry Mart. *Bahasa Indonesia support from the babel system*, November 2005. mailto:babel@bramms.cistron.nl, mailto:joerg.knappen@alpha.ntp.springer.de, mailto:mart@vkpmzd.kph.uni-mainz.de. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/bahasa.dtx. 440

[65] Johannes Braams, Jörg Knappen, Terry Mart, and Bob Margolis. *Bahasa Malaysia support from the babel system*, November 2005. mailto:babel@bramms.cistron.nl, mailto:joerg.knappen@alpha.ntp.springer.de, mailto:mart@vkpmzd.kph.uni-mainz.de, mailto:bob.margolis@ntlworld.com. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/bahasam.dtx. 440

[66] Johannes Braams, Olga G. Lapko, Vladimir Volovich, and Werner Lemberg. *Russian support from the babel system*, March 2005. mailto:babel@bramms.cistron.nl, mailto:cyrtug@mir.msk.su, mailto:TeX@vvv.vsu.ru et mailto:wl@gnu.org. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/russianb.dtx. 447, 503

[67] Johannes Braams and Henning Larsen. *Danish language support from the babel system*, March 2005. mailto:babel@bramms.cistron.nl, mailto:larsen@cernvm.cern.ch. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/danish.dtx. 452

[68] Johannes Braams and Boris Lavva. *Hebrew language support from the babel system*, March 2005. mailto:babel@bramms.cistron.nl. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/hebrew.dtx. 475

[69] Johannes Braams and Milos Lokajicek. *Czech language support from the babel system*, March 2005. mailto:babel@bramms.cistron.nl, mailto:lokajick@cernvm. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/czech.dtx. 451

[70] Johannes Braams, Dejan Muhamedagić, and Jankovic Slobodan. *Serbocroatian support from the babel system*, March 2005. mailto:babel@bramms.cistron.nl, mailto:dejan@yunix.com, mailto:slobodan@archimed.filfak.ac.ni.yu. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/serbian.dtx. 509

[71] Johannes Braams and Alan Paić. *Croatian language support from the babel system*, March 2005. mailto:babel@bramms.cistron.nl, mailto:paica@cernvm.cern.ch. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/croatian.dtx. 451

[72] Johannes Braams and Bernd Raichle. *German support from the babel system*, February 2004. mailto:babel@bramms.cistron.nl, mailto:azu.Informatik.Uni-Stuttgart.de. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/germanb.dtx. 465

[73] Johannes Braams, Bernd Raichle, and Walter Schmidt. *New German support from the babel system*, February 2004. mailto:babel@bramms.cistron.nl, mailto:raichle@azu.Informatik.Uni-Stuttgart.de. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/ngermanb.dtx. 494

[74] Johannes Braams and Jose Pedro Ramalhete. *Portuguese support from the babel system*, March 2005. mailto:babel@bramms.cistron.nl, mailto:jramalhe@cernvm or mailto:Jose-Pedro_Ramalhete@macmail. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/portuges.dtx. 444, 499

[75] Johannes Braams and Christian Rolland. *Breton language support from the babel system*, March 2005. mailto:babel@bramms.cistron.nl, mailto:Christian.Rolland@univ-brest.fr. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/breton.dtx. 445

[76] Johannes Braams, Marti Ruiz-Altaba, and Jörg Knappen. *Esperanto support from the babel system*, March 2005. mailto:babel@bramms.cistron.nl, mailto:ruizaltb@cernvm.cern.ch, mailto:knappen@vkpmzd.kph.uni-mainz.de. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/esperanto.dtx. 455

[77] Johannes Braams and Enn Saar. *Estonian support from the babel system*, March 2005. mailto:babel@bramms.cistron.nl, mailto:saar@aai.ee. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/estonian.dtx. 456

[78] Johannes Braams, Elmar Schalueck, and Michael Janich. *Polish support from the babel system*, March 2005. mailto:babel@bramms.cistron.nl, mailto:elmar@uni-paderborn.de, mailto:massa@uni-paderborn.de. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/polish.dtx. 497

[79] Johannes Braams and Andrij Shvaika. *Ukrainian support from the babel system*, March 2005. mailto:babel@bramms.cistron.nl, mailto:ashv@icmp.lviv.ua. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/ukraineb.dtx. 516

[80] Johannes Braams and Apostolos Syropoulos. *Greek support from the babel system*, March 2005. mailto:babel@bramms.cistron.nl, mailto:apostolo@platon.ee.duth.gr or mailto:apostolo@obelix.ee.duth.gr. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/greek.dtx. 466

[81] Johannes Braams and Eduard Werner. *Lower Sorbian support from the babel system*, March 2005. mailto:babel@bramms.cistron.nl, mailto:edi@kaihh.hanse.de. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/lsorbian.dtx. 487

[82] Johannes Braams and Eduard Werner. *Upper Sorbian support from the babel system*, March 2005. mailto:babel@bramms.cistron.nl, mailto:edi@kaihh.hanse.de. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/usorbian.dtx. 517

[83] Johannes Braams and Adi Zaimi. *Albanian support from the babel system*, November 2005. mailto:babel@bramms.cistron.nl, mailto:zami1st@yahoo.com. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/albabian.dtx. 436

[84] Johannes Braams, Danilo Zavrtanik, and Leon Žlajpah. *Slovene support from the babel system*, March 2005. mailto:babel@bramms.cistron.nl, mailto:leon.zlajpah@ijs.si. Available from: http://www.dante.de/tex-archive/macros/latex/required/babel/slovene.dtx. 510

[85] Felix Braun. *Die Documentklasse* `jura`, July 1998. mailto:fbraun@atdot.org. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/jura/. 85, 193, 204, 383

[86] Jim Breen. *Jim* Breen*'s Japanese Page* [online], September 2006. Available from: http://www.csse.monash.edu.au/~jwb/japanese.html. mailto:jwb@csse.monash.edu.au.

[87] Peter Breitenlohner. *The ε-TEX manual (version 2)*. The NTS Team, Max-Planck-Institut für Physik, München, February 1998. Available from: http://www.dante.de/tex-archive/systems/e-tex/v2/doc/etex_man.pdf. 199

[88] David Carella. *Règles typographiques et normes: Mise en pratique avec LaTeX*. Vuibert, Paris, January 2006.

[89] David Carlisle. *The* `xr` *package*, May 1994. mailto:carlisle@cs.man.ac.uk. Available from: http:/www.dante.de/tex-archive/macros/required/tools/xr.dtx. 26, 221, 287

[90] David Carlisle. *The* `xr-hyper` *package*, March 2000. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/hyperref/xr-hyper.sty. 221

[91] David Carlisle. *The* `color` *package*, November 2005. mailto:carlisle@cs.man.ac.uk. Available from: http:/www.dante.de/tex-archive/macros/required/graphics/color.dtx. 112

[92] Otfried Cheong. *Using TrueType Fonts in PdfLaTeX* [online], September 2005. Available from: http://ipe.comp.geom.org/pdftex.html. mailto:otfried@tclab.kaist.ac.kr.

[93] Pai H. Chou. *How to make LaTeX (teTeX) handle unicode and CJK in MacOSX* [online], December 2004. Available from: http://www.ece.uci.edu/~chou/unicode-tex.html. mailto:chou@ece.uci.edu. 200

[94] Steven Douglas Cochran. *The `subfigure` package*, March 1995. mailto:sds@cs.cmu.edu, mailto:cochran@ieee.org. Available from: http://www.dante.de/tex-archive/obsolete/macros/latex/contrib/subfigure/subfigure.pdf. 31, 85, 132, 211, 219

[95] Steven Douglas Cochran. *The `captcont` package*, February 2002. mailto:sds@cs.cmu.edu, mailto:cochran@ieee.org. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/captcont/. 50, 73, 85, 195, 233, 252

[96] Steven Douglas Cochran. *The `subfig` Package*, January 2004. mailto:sds@cs.cmu.edu, mailto:cochran@ieee.org. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/subfig/subfig.pdf. 31, 85, 138, 195, 211, 218, 219, 366

[97] Oliver Corff. *MonTeX – A Quick Guide* (draft), July 2002. mailto:corff@zedat.fu-berlin.de. Available from: http://www.dante.de/tex-archive/language/mongolian/montex/doc/mlsquick.pdf. 41, 191, 208, 434, 442, 443, 444, 447, 448, 492, 504, 520, 521

[98] Oliver Corff. *Some Notes on the Pentaglot Dictionary* [online], April 2002. Available from: http://userpage.fu-berlin.de/~corff/im/Buch.Pentaglot.html. mailto:corff@zedat.fu-berlin.de. 209

[99] Oliver Corff. *Infosystem Mongolei* [online], July 2004. Available from: http://userpage.fu-berlin.de/~corff/infomong.html. An Internet-based Journal on Mongolian Affairs. 208

[100] Oliver Corff and Dorjpalam Dorj. *MonTeX – Mongolian for LaTeX 2ε, Implementation Level System Documentation*, July 2002. mailto:corff@zedat.fu-berlin.de. Available from: http://www.dante.de/tex-archive/language/mongolian/montex/doc/montex.pdf. 41, 191, 208, 434, 442, 443, 444, 447, 448, 492, 504, 520, 521

[101] Serguei D'Achian, Arnak Dalalyan, and Vardan Akopian. *ArmTeX: a System for Writing in Armenian with TeX and LaTeX (written in armenian)*, June 1999. mailto:Serguei.Dachian@univ-lemans.fr. Available from: http://www.dante.de/tex-archive/language/armtex/v2.0/manual.ps. 41, 194, 434, 438

[102] Bernard Desgraupes. *METAFONT, Guide pratique*. Vuibert, Paris, March 1999. 198

[103] Bernard Desgraupes. *LaTeX, Apprentissage, guide et référence*. Vuibert, Paris, second edition, March 2003.

[104] Bernard Desgraupes. *Passeport pour Unicode*. Vuibert, Paris, April 2005. 200

[105] Jean-Pierre F. Drucbert. *The* `shorttoc` *package*, August 2002. mailto:jean-pierre.drucbert@onera.fr. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/shorttoc/shorttoc.dtx. 19, 94, 218, 547

[106] Jean-Pierre F. Drucbert. *Le paquetage* `minitoc`, January 2007. mailto:jean-pierre.drucbert@onera.fr. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/minitoc/fminitoc.pdf. 94, 413

[107] Jean-Pierre F. Drucbert. *The* `minitoc` *package*, January 2007. mailto:jean-pierre.drucbert@onera.fr. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/minitoc/minitoc.pdf. 93, 413

[108] Victor Eijkhout. *The* `comment` *package*, October 1999. mailto:victor@eijkhout.net. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/comment/.

[109] Behdad Esfahbod and Roozbeh Pournader. *FarsiTEX and the Iranian TEX Community*. *TUGboat*, 22(3):41–44, 2003. Available from: http://www.tug.org/TUGboat/Articles/tb23-1/farsitex.pdf. 41, 201, 434, 459

[110] Mike Fabian. *CJK Support in SuSe Linux*, March 2005. Available from: http://www.suse.de/~mfabian/suse-cjk.pdf.

[111] Robin Fairbairns. *The UK TEX FAQ. Your 396 Questions Answered*, November 2005. Version 3.15a. Available from: ftp://cam.ctan.org/tex-archive/help/uk-tex-faq/newfaq.pdf.

[112] Michael J. Ferguson. *Report on multilingual activities*. *TUGboat*, 11(4):514–516, November 1990.

[113] Peter Flynn. *Formatting information. A beginner's introduction to typesetting with LATEX*, March 2005. mailto:peter.silmaril.ie. Available from: http://www.dante.de/tex-archive/tex/info/beginlatex/beginlatex-3.6.pdf.

[114] Maarten Gelderman. *A short introduction to font characteristics*. *TUGboat*, 20(2):96–104, June 1999. Available from: http://www.tug.org/TUGboat/Articles/tb20-2/tb63geld.pdf.

[115] Chuck Genschte and John Warnock. *PDF Reference*, November 2004. Available from: http://partners.adobe.com/public/developer/en/pdf/PDFReference16.pdf. 216

[116] Helen Gilhooly and Rozenn Etienne. *Lire et écrire le japonais*. Langues orientales. Larousse, Paris, March 2006. Translation and adaptation to french by Cécile Desprairies.

[117] Michel Goossens, Sebastian Rahtz, Eitan M. Gurai, Ross Moore, and Robert S. Sutor. *The LATEX Web Companion: Integrating TEX, HTML, and XML*. Tools and Techniques for Computer Typesetting. Addison-Wesley, Reading, Massachusetts, June 1999. With Eitan M. Gurari, Ross Moore and Robert S. Sutor.

[118] Norman Gray. *The* `urlbst` *package*, June 2005. http://www.astro.gla.ac.uk/users/norman/. Available from: http://www.tug.org/tex-archive/macros/latex/contrib/urlbst/urlbst.pdf. 200, 203, 220, 233, 567

[119] Hans Hagen. *Aleph* ℵ, July 2004. Available from: http://www,pragma-ade.com/general/manuals/aleph.pdf. 215

[120] Hans Hagen. *Typographic Programming*, 2004. Version préliminaire. Available from: http://www,pragma-ade.com/general/manuals/style.pdf.

[121] Thorsten Hansen. *The* `bibunits` *package*, May 2004. mailto:thorsten.hansen@psychol.uni-giessen.de. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/bibunits/bibunits.pdf. 57, 195, 211

[122] Thorsten Hansen. *The* `multibib` *package*, January 2004. mailto:thorsten.hansen@psychol.uni-giessen.de. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/multibib/multibib.pdf. 57, 211

[123] Patrick Happel. *The* `lipsum` *package*, January 2005. mailto:patrick.happel@rub.de. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/lipsum/lipsum.pdf. 206, 570

[124] Yannis Haralambous. *Fontes et codages*. O'Reilly France, April 2004. mailto:xavier@editions-oreilly.fr. Available from: http://www.oreilly.fr/catalogue/284177273X.desc.html. 470, 473, 474

[125] Yannis Haralambous and John Plaice. *First applications of* Ω*: Greek, Arabic, Khmer, Poetica, ISO 10646/Unicode, etc*. *TUGboat*, 15(3):344–352, 1994. 214

[126] Yannis Haralambous and John Plaice. Ω*, a TEX extension including Unicode and featuring lex-like filtering processes.* In Wlodek Bzyl and Plata-Przechlewski, editors, *Proceedings of the European TEX Conference*, pages 153–166, Gdnask, Poland, 1994. GUST. 214, 534

[127] Yannis Haralambous and John Plaice. *The Design and Use of a Multiple-Alphabet Font with Omega*. *MAPS*, 27:27–37, 2000. Available from: http://www.ntg.nl/maps/pdf/24_7.pdf.

[128] Yannis Haralambous and John Plaice. *Traitement automatique des langues et compositions sous Omega*. *Cahiers GUTenberg*, 39-40:139–166, May 2001. Available from: http://www.gutenberg.eu.org/publications/cahiers/r25-cahiers39-40/131-yannis.html.

[129] Yannis Haralambous, John Plaice, and Éric Picheral. Ω*, une extension de TEX incluant UNICODE et des filtres de type lex*. *Cahiers GUTenberg*, 20:55–80, June 1995. French translation of [126]. 214

[130] Alexander Harin and Vadim V. Zhytnikov. *The* `cmcyralt.sty` *file*, December 1994. Available from: ftp://ftp.dante.de/tex-archive/macros/latex/contrib/cmcyralt/cmcyralt.sty. 505

[131] Stephan Hennig. *The* `mcaption` *package*, September 2005. mailto:stephanhennig@arcor.de. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/mcaption/mcaption.pdf. 51, 69, 85, 206, 253

[132] Alan Hoenig. *TEX Unbound*. Oxford University Press, Oxford New York, 1998. LATEX & TEX Strategies for Fonts, Graphics , & More.

[133] Hiroya Ikeda. *pLaTeX 2ε*, September 2001. In japanese, mailto:ike@kobitosan.net. Available from: http://www.kobitosan.net/ike/latex/handhtml/index.html. 216

[134] Tetsuo Iwakuma and Tetsuo Furukawa. *A Guide to use Macros and Style Files in LaTeX*[46], August 1994. mailto:bear@hashi1.civil.tohoku.ac.jp and furakawa@hagi.ces.kyutech.ac.jp. Available from: http://www.moivre.usherbrooke.ca/Intranet/Doc/LATEX_HOWTO/styleuse.pdf.

[135] Youssef Jabri. *Typesetting Arabic and Farsi with the Arabi package. The Users Guide*, February 2006. mailto:yjabri@ensa.univ-oujda.ac.ma, École des sciences appliquées, Boîte 696, Oujda, Maroc. Available from: http://www.dante.de/tex-archive/language/arabic/arabi/doc/latex/arabi/user_guide.pdf. 36, 41, 194, 437, 460

[136] Alan Jeffrey. *PostScript font support in LaTeX 2ε*. TUGboat, 15(3):263–268, September 1994. Available from: http://www.tug.org/TUGboat/Articles/tb15-3/tb44jeff.pdf.

[137] Zhuhan Jiang. *The* `vruler` *package*, October 1996. mailto:zhuhan@neumann.une.edu.au, and look at http://www.dante.de/tex-archive/help/Catalogue/entries/vruler.html. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/misc/vruler.sty.

[138] David M. Jones. `hangcaption.sty`, August 1992. mailto:dmjones@theory.lcs.mit.edu. Available from: http://www.dante.de/tex-archive/macros/latex209/contrib/misc/hangcaption.sty. 50, 76, 85, 203, 234, 251

[139] Ki-Joo Kim. *A BibTeX Guide via Examples*, April 2004. mailto:kijoo2000@yahoo.com. Available from: http://www.geocities.com/bibtex_guide.pdf.

[140] Ki-Joo Kim. *How to Create PDF from LaTeX*, April 2004. mailto:kijoo2000@yahoo.com. Available from: http://www.geocities.com/latex2pdf.pdf.

[141] Ki-Joo Kim. *LaTeX Fonts*, November 2005. mailto:kijoo2000@yahoo.com. Available from: http://www.geocities.com/latex_fonts.pdf.

[142] Donald E. Knuth. *Computer Modern Typefaces*, volume E of *Computers and Typesetting*. Addison-Wesley, Reading, Massachusetts, 1986. 34, 198

[143] Donald E. Knuth. *The TeXbook*, volume A of *Computers and Typesetting*. Addison-Wesley, Reading, Massachusetts, 16th edition, 1989. Revised to cover TeX3, 1991. 196, 205, 219, 220, 535

[144] Donald E. Knuth. *The METAFONTbook*, volume C of *Computers and Typesetting*. Addison-Wesley, Reading, Massachusetts, 1993. With the final corrections of 1993. 198

[145] Donald E. Knuth. *Le TeXbook, Composition informatique*. Vuibert, Paris, 2003. French translation of [143], by Jean-Côme Charpentier. 205

[46] This document is somewhat obsolete.

[146] Un Koaunghi. *HLaTeX: a LaTeX package for the Korean language support*, April 2005. In korean. Available from: http://www.dante.de/tex-archive/language/korean/HLaTeX/hlguide.pdf. 191, 203, 434, 473, 474

[147] Markus Kohm and Jens-Uwe Morawski. *KOMA-Script, ein wandelbares LaTeX 2ε-Paket*, July 2006. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/koma-script/scrguide.pdf. 39, 47, 59, 71, 85, 134, 151, 191, 195, 204, 211, 218, 232, 233, 383, 565

[148] Helmut Kopka and Patrick W. Daly. *Guide to LaTeX*. Tools and Techniques for Computer Typesetting. Addison Wesley, Reading, Massachusetts, fourth edition, February 2004. 215

[149] Siep Kroonenberg. *NFSS: using font families in LaTeX 2ε*. MAPS, 11:52–54, 1999. mailto:siepo@cybercomm.nl. Available from: http://www.ntg.nl/maps/pdf/22_13.pdf.

[150] Alexej M. Kryukov. *Typesetting multilingual documents with ANTOMEGA*, September 2003. Available from: http://www.dante.de/tex-archive/systems/omega/contrib/antomega/doc/omega/antomega/antomega.pdf. 41, 194, 434, 467, 468, 469, 485, 497, 501, 502, 511, 512

[151] Markus G. Kuhn. *Effective scientific electronic publishing* [online], June 2006. Available from: http://www.cl.cam.ac.uk/~mgk25/publ-tips.html. mailto:mgk25@cl.cam.ac.uk.

[152] Frank Küster. `alnumsec.sty`: *Using alphanumeric section numbering with standard sectioning commands*, February 2005. mailto:frank@kuesterei.ch, Biozentrum der Univ. Basel, Abt. Biophysikalische Chemie. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/alnumsec/. 50, 72, 85, 193, 233, 252

[153] Thomas Lachand-Robert. *La maîtrise de TeX et LaTeX*. Masson, Paris, December 1997.

[154] Klaus Lagally. *ArabTeX—Typetting Arabic with vowels and ligatures*. In *Proceedings of the 7th European TeX Conference*, pages 153–172, Prague, 1992. CsTUG. mailto:lagallyk@acm.org. Available from: http://www.ntg.nl/maps/pdf/20_22.pdf. 36, 41, 194, 434, 436, 437, 475

[155] Klaus Lagally. *ArabTeX Version 3*, October 2003. mailto:lagallyk@acm.org. Available from: http://www.dante.de/tex-archive/language/arabtex/doc/html/arabtex.htm. 36, 41, 194, 434, 436, 437, 475

[156] Leslie Lamport. *LaTeX: A Document Preparation System — User's Guide and Reference Manual*. Addison-Wesley, Reading, Massachusetts, second edition, June 1994. 54, 205

[157] Leslie Lamport, Frank Mittelbach, and Johannes Braams. *Standard Document Classes for LaTeX version 2e*, September 1999. Available from: http://www.dante.de/tex-archive/macros/latex/base/classes.dtx. 191

[158] Olga G. Lapko. *Full Cyrillic: How Many Languages?* In Mimi Burbank and Christina Thiele, editors, *Proceedings of TUG'96*, pages 164–170, Dubna, 1996. JINR. Available from: http://www.tug.org/TUGboat/Articles/tb17-2/tb51olga.pdf.

[159] Olga G. Lapko. *The `floatrow` package*, July 2005. mailto:Lapko.O@g23.relcom.ru. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/floatrow/floatrow.pdf. 29

[160] Olga G. Lapko and Irina A. Makhovaya. *The Style `russianb` for Babel: Problems and solutions*. *TUGboat*, 16(4):364–372, 1995. Available from: http://www.tug.org/TUGboat/Articles/tb16-4/tb49olga.pdf. 36, 447, 503

[161] LaTeX3 Project Team. *Default docstrip headers*. *TUGboat*, 19(2):137–138, June 1998. Available from: http://www.tug.org/TUGboat/Articles/tb19-2/tb59ltdocstrip.pdf.

[162] LaTeX3 Project Team. *LaTeX 2ε Kernel (Output Routine)*, July 2000. Available from: http://www.dante.de/tex-archive/macros/latex/base/ltoutput.dtx. 29, 216, 407

[163] LaTeX3 Project Team. *LaTeX 2ε for authors*, July 2001. Available from: http://www.dante.de/tex-archive/macros/latex/doc/usrguide.pdf. 191

[164] LaTeX3 Project Team. *LaTeX 2ε for Class and Package writers*, December 2003. Available from: http://www.dante.de/tex-archive/macros/latex/doc/clsguide.pdf. 191

[165] LaTeX3 Project Team. *LaTeX 2ε font selection*, February 2004. Available from: http://www.dante.de/tex-archive/macros/latex/doc/fntguide.pdf. 191, 212, 270, 348, 352

[166] LaTeX3 Project Team. *LaTeX font encodings*, January 2006. Available from: http://www.dante.de/tex-archive/macros/latex/doc/encguide.pdf. 200, 212

[167] Werner Lemberg. *The CJK package for LaTeX 2ε — Multilingual support beyond babel*. *TUGboat*, 18(3):214–224, 1997. Available from: http://www.tug.org/TUGboat/Articles/tb18-3/cjkintro600.pdf. 36, 41, 197, 218, 219, 434, 450, 470, 471, 472, 473, 474, 481, 482, 483, 514

[168] Werner Lemberg. *The CJK package* [online], August 2005. Available from: http://cjk.ffii.org/. 36, 41, 197, 218, 219, 434, 450, 470, 471, 472, 473, 474, 481, 482, 483, 514

[169] Knut Lickert. *LaTeX 2ε for people in associations: `minutes.sty`*, January 2001. mailto:knut@lickert.net. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/minutes/. 208, 270, 373

[170] Ulf A. Lindgren. *FncyChap V1.33*, August 2005. mailto:ulf.a.lindgren@ericsson.com. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/fncychap/fncychap.pdf. 50, 72, 85, 202, 233, 250

[171] Anselm Lingnau. *An Improved Environment for Floats*, November 2001. mailto:anselm@strathspey.org. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/float/float.dtx. 29

[172] Maurizio Loreti. *The* `sfheaders` *package*, September 1997. mailto:loreti@padova.infn.it and http://wwwcdf.pd.infn.it/~loreti/mlo.html, University of Padova – Department of Physics, Via F. Marzolo, 8 – 35131 PADOVA – Italy. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/sfheaders/. 50, 72, 85, 218, 233, 251

[173] Tim P. Love. *Advanced LaTeX*, September 1994. mailto:tpl@eng.cam.ac.uk. Available from: http://www.moivre.usherbrooke.ca/Intranet/Doc/LATEX_HOWTO/latex-advanced.pdf.

[174] Tim P. Love. *Producing HTML and PDF files with LaTeX* [online], March 2006. Available from: http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/makingWWWdocs.html. mailto:tpl@eng.cam.ac.uk.

[175] Vincent Lozano. *Tout ce que vous avez toujours voulu savoir sur LaTeX sans jamais oser le demander*, September 2006. mailto:lozano@enise.fr. Available from: http://cours.enise.fr/info/latex/guide-local.pdf.

[176] Ken Lunde. *Understanding Japanese Information Processing*. O'Reilly & Associates, Inc, September 1993. 191, 198, 538

[177] Ken Lunde. *Online companion of "Understanding Japanese Information Processing"*, 1996. See [176]. Available from: ftp://ftp.ora.com/pub/examples/nutshell/ujip/doc/cjk.inf. 191, 198

[178] Ken Lunde. *CJKV Information Processing*. O'Reilly & Associates, Inc, January 1999.

[179] Nicolas Markey. *Split your bibliography into categories*, December 2005. mailto:markey@lsv.ens-cachan.fr. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/splitbib/splitbib.pdf. 57, 218

[180] Nicolas Markey. *Tame the BeaST (The B to X of* BibTeX*)*, June 2005. mailto:markey@lsv.ens-cachan.fr. Available from: http://www.dante.de/tex-archive/info/bibtex/tamethebeast/ttb_en.pdf. 191, 195

[181] Sven Mattisson. *SLaTeX: Swedish speaking version of LaTeX* [online], November 1989. Available from: http://ftp.dante.de/tex-archive/language/swedish/slatex/slatex.1. mailto:sven@tde.lu.se. 41, 218, 514

[182] Rowland McDonnell. *The* `sectsty` *package v2.0.2*, April 2002. mailto:rowland.mcdonnell@physics.org. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/sectsty/. 50, 67, 85, 125, 218, 232, 250

[183] Surapant Meknavin, Theppitak Karoonboonyanan, Chanop Silpa-Anan, and Poonlap Veerathanabutr. *The* `thailatex` *package*, March 2006. mailto:surapan@nectec.or.th. Available from: http://www.dante.de/tex-archive/language/thailatex/. 219, 514

[184] Frank Mittelbach. *ε-TeX: Guidelines for future TeX*. TUGboat, 11(3):337–345, September 1990. Available from: http://www.tug.org/TUGboat/Articles/tb11-3/tb29mitt.pdf.

[185] Frank Mittelbach. *Comments on "Filenames for Fonts" [27]*. TUGboat, 13(1):51–53, April 1992. Available from: http://www.tug.org/TUGboat/Articles/tb13-1/tb34mittfont.pdf.

[186] Frank Mittelbach. *An environment for multicolumn output*, July 2000. mailto:Frank.Mittelbach@latex-project.org. Available from: http:/www.dante.de/tex-archive/macros/required/tools/multicol.pdf. 87, 112

[187] Frank Mittelbach. *The* `doc` *and* `shortverb` *packages*, January 2004. Available from: http://www.tug.org/tex-archive/macros/latex/base/doc.dtx.

[188] Frank Mittelbach, Denys Duchier, Johannes Braams, Marcin Woliński, and Mark Wooding. *The* `docstrip` *program*, January 2004. Available from: http://www.tug.org/tex-archive/macros/latex/base/docstrip.dtx.

[189] Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, and Chris Rowley. *The LaTeX Companion 2*. Tools and Techniques for Computer Typesetting. Addison-Wesley Professional, Reading, Massachusetts, second edition, April 2004. 29, 138, 191, 206, 211, 216, 407

[190] Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, and Chris Rowley. *Der LaTeX-Begleiter*. Pearson Studium, second edition, October 2005. German version.

[191] Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, and Chris Rowley. *LaTeX Companion*. Pearson Education France, Paris, second edition, 2005. French translation by Jacques André, Benoît Belet, Jean-Côme Charpentier, Jean-Michel Hufflen, and Yves Soulet. 561

[192] Frank Mittelbach and Chris Rowley. *Language Information in Structured Documents: A Model for Mark-up and Rendering*. TUGboat, 18(3):199–205, September 1997. Available from: http://www.tug.org/TUGboat/Articles/tb18-3/tb56lang.pdf.

[193] Young Joon Moon. *How to make and view a Japanese PDF file*, June 2002. mailto:director@research-j.org. Available from: http://e-japanese-online.com/english/japanese-computing/platex/Free-Japanese-PDF-authoring-EV003.pdf.

[194] Sergei O. Naumov. *LaTeX2e support for LH family of fonts*, April 1995. Available from: http://web.mit.edu/texsrc/source/latex/LH/russian.sty. 506

[195] Frank Neukam, Markus Kohm, Axel Kielhorn, and Jens-Uwe Morawski. *The KOMA-Script Bundle*, March 2005. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/koma-script/scrguien.pdf. 39, 47, 59, 71, 85, 134, 151, 191, 195, 204, 211, 218, 232, 233, 383, 565

[196] Heiko Oberdiek. *PDF information and navigation elements with hyperref, pdfTeX, and thumbpdf*. In *EuroTeX'99 Proceedings*. TUG, 1999. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/hyperref/doc/paper.pdf. 85

[197] Tobias Oetiker, Hubert Partl, Helene Hyna, and Elisabeth Schlegl. *Une courte (?) introduction à LaTeX 2ε*. Cahiers GUTenberg, special issue number 2, February 2003. French translation of [198] by Matthieu Herrb, adaptation and complements by Daniel Flipo. Available from: http://www.dante.de/tex-archive/info/lshort/french/flshort-3.20.pdf. 191

[198] Tobias Oetiker, Hubert Partl, Helene Hyna, and Elisabeth Schlegl. *The Not So Short Introduction to LaTeX 2ε*, May 2006. Available from: http://www.dante.de/tex-archive/info/lshort/english/lshort.pdf. 191, 198, 205, 539

[199] Mariuz Olko and Marcin Woliński. *LaTeX 2ε po polsku czyli PLaTeX (wersja 1.3.1)*, May 2003. Available from: http://www.tug.org/tex-archive/macros/latex/contrib/platex/polski.dtx. 41, 216, 498

[200] Tanguy Ortolo and Jeanne Legrand. *LaTeX à 200%*. O'Reilly France, November 2006.

[201] Scott Pakin. *How to Package Your LaTeX Package*, November 2004. mailto:scott+dtx@pakin.org. Available from: http://www.dante.de/tex-archive/info/dtxtut/dtxtut.pdf.

[202] Palash Baran Pal. *Bangtex: a package for typesetting documents in Bangla using the TeX/LaTeX systems* [online], January 2001. Available from: http://www.saha.ac.in/theory/palashbaran.pal/bangtex/bangtex.html. 41, 195, 434, 441

[203] Anshuman Pandey. *Typesetting Bengali in TeX*. TUGboat, 20(2):119–126, 1999. mailto:apandey@u.washington.edu. Available from: http://www.tug.org/TUGboat/Articles/tb20-2/tb63pand.pdf.

[204] Anshuman Pandey. *Devanāgarī for TeX*, May 2006. mailto:apandey@u.washington.edu. Available from: http://www.dante.de/tex-archive/language/devanagari/velthuis/doc/deneric/velthuis/manual.pdf. 41, 200, 203, 434, 452, 477

[205] Minje Byeng-sen Park. *Histoire de l'imprimerie coréenne des origines à* 1910. Maison-Neuve & Larose, Paris, September 2003. 470, 473, 474

[206] Oren Patashnik. BibTeX*ing*, February 1988. Documentation for general BibTeX users. Available from: http://www.dante.de/tex-archive/biblio/bibtex/contrib/doc/btxdoc.pdf. 195

[207] Oren Patashnik. *Designing* BibTeX *Styles*, February 1988. The part of BibTeX's documentation that's not meant for general users. Available from: http://www.dante.de/tex-archive/biblio/bibtex/contrib/doc/btxhak.pdf. 195

[208] Jonathan Pechta, Federico Zenith, Holger Danielsson, and Jeroen Wijnhout. *CJK Support* [online], March 2006. Available from: http://sourceforge.net/Documentation/html/cjk.html. Part of [210].

[209] Jonathan Pechta, Federico Zenith, Holger Danielsson, and Jeroen Wijnhout. *Document Encoding* [online], March 2006. Available from: http://sourceforge.net/Documentation/html/lang.html. Part of [210].

[210] Jonathan Pechta, Federico Zenith, Holger Danielsson, and Jeroen Wijnhout. *The Kile Handbook* [online], March 2006. Available from: http://sourceforge.net/Documentation/html/. 540

[211] Karel Píška. *Cyrillic Alphabets*. In Mimi Burbank and Christina Thiele, editors, *Proceedings of TUG'96*, pages 1–7, Dubna, 1996. JINR. Available from: http://www.tug.org/TUGboat/Articles/tb17-2/tb51pisk.pdf.

[212] John Plaice and Yannis Haralambous. *The latest developments in* Ω. *TUGboat*, 17(2):181–183, June 1996. Available from: http://omega.enstb.org/papers/latest-omega9papers/latest-omega96.pdf. 214

[213] John Plaice and Yannis Haralambous. *Typesetting French, German and English in* Ω. In *Congrès EuroTEX'98*, volume 28-29 of *Cahiers GUTenberg*, St. Malo, March 1998. 214

[214] Sebastian Rahtz and Heiko Oberdiek. *The* `hyperref` *package* [online], August 2003. Available from: http://www.tug.org/applications/hyperref. This page points to the package documentation and its distribution. 59, 69, 85, 203, 207, 221, 230, 249, 388

[215] Sebastian Rahtz and Heiko Oberdiek. *Hypertext marks in LaTeX: a manual for* `hyperref`, February 2004. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/hyperref/doc/manual.pdf. 85, 113

[216] Bernd Raichle, Rolf Niepraschk, and Thomas Hafner. *Fragen und Antworten (FAQ) über das Textsatzsystem TEX und DANTE, Deutschsprachige Anwendervereinigung TEX e.V*, September 2003. Version 72. Available from: http://www.dante.de/faq/de-tex-faq/de-tex-faq.pdf.

[217] Adrian Rezuş and Bernd Raichle. *Romanian TEX*, October 1996. mailto:adriaan@cs.kun.nl. Available from: http://www.dante.de/tex-archive/language/romanian/RomanianTeX/romanian.dtx. 500

[218] Christian Rolland. *LaTeX par la pratique*. O'Reilly France, October 1999. mailto:Rolland.Christian@wanafoo.fr.

[219] Chris Rowley. *Models and languages for formatted documents*. *TUGboat*, 20(3):189–195, September 1999. Available from: http://www.tug.org/TUGboat/Articles/tb29-3/tb64rowl.pdf.

[220] Young Ryu. *The TX Fonts*, December 2000. Available from: http://www.dante.de/tex-archive/fonts/txfonts/doc/txfontsdocA4.pdf. 34

[221] Eddie Saudrais. *Et les fontes,* man *? Gestion des fontes avec LaTeX* 2_ε, June 2001. mailto:eddie.saudrais@wanadoo.fr. Available from: http://perso.orange.fr/eddie.saudrais/latex/fontedoc.zip.

[222] Eddie Saudrais. *Le petit typographe rationnel*, 2005. mailto:eddie.saudrais@wanadoo.fr. Available from: http://perso.orange.fr/eddie.saudrais/prepa/typo.pdf.

[223] Axel Sommerfeldt. *The* `caption2` *package*, November 2005. mailto:caption@sommerfee.de. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/caption/caption2.pdf. 51, 69, 85, 196, 252

[224] Axel Sommerfeldt. *The* `caption` *package*, January 2006. mailto:caption@sommerfee.de. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/caption/caption.pdf. 51, 69, 85, 196, 252

[225] Axel Sommerfeldt. *Setzen von Abbildungs- und Tabellenbeschriftungen mit dem* `caption`*-Paket*, January 2006. mailto:caption@sommerfee.de. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/caption/anleitung.pdf. 85

[226] D. P. Story. *Using LaTeX to Create Quality PDF Documents for the World Wide Web* [online], November 1999. Available from: http://www.math.uakron/~dpstory/latx2pdf.html. mailto:dpstory@uakron.edu.

[227] Apostolos Syropoulos, Antonis Tsolomitis, and Nick Sofroniou. *Digital Typography Using LaTeX*. Springer, October 2002. Available from: http://ocean1.ee.duth.gr/LaTeXBook/.

[228] Daniel Taupin†. *The* `varsects` *package*, November 1998. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/taupin/varsects.sty. 50, 69, 85, 220, 232, 250

[229] Hàn Thế Thành. *The pdfTeX Program*. In *EuroTeX'98 Proceedings*, volume 28-29 of *Cahiers GUTenberg*, pages 197–219, March 1998. Available from: http:www.gutenberg.eu.org/pub/GUTenberg/publicationsPDF/28-29-han.pdf.

[230] Hàn Thế Thành. *Micro-typographic extensions to the TeX typesetting system*. *TUGboat*, 21(4):317–434, October 2000. Doctoral dissertation (Faculty of Informatics, Masarik University, Brno, Czech Republic). Available from: http://www.tug.org/TUGboat/Articles/tb21-4/tb69thanh.pdf.

[231] Hàn Thế Thành, Sebastian Rahtz, Hans Hagen, Hartmut Henkel, and Paweł Jackowski. *The pdfTeX user manual*, February 2006. Available from: http://www.dante.de/tex-archive/systems/pdftex/manual/pdftex-a.pdf.

[232] Karsten Tinnefeld. *The* `quotchap` *document style*, February 1998. mailto:tinnefeld@irb.cs.uni-dortmund.de. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/quotchap/. 50, 72, 85, 217, 233, 251

[233] Mark Trettin. *Une liste des péchés des utilisateurs de LaTeX 2ε (ou Commandes et extensions obsolètes, et quelques autres erreurs)*. mailto:Mark@Trettin@gmx.de, mailto:juergen.fen@gmx.de and mailto:Yvon.Henel@wanadoo.fr, June 2004. English translation by Jürgen Fenn, french translation by Yvon Henel. Available from: http://www.dante.de/tex-archive/info/l2tabu/french/l2tabufr-heavy.pdf.

[234] Mark Trettin. *An essential guide to LaTeX 2ε usage (Obsolete commands and packages)*. mailto:Mark@Trettin@gmx.de and mailto:juergen.fen@gmx.de, June 2006. English translation by Jürgen Fenn. Available from: http://www.dante.de/tex-archive/info/l2tabu/english/l2tabuen.pdf.

[235] Groupe de travail TWS-TDS[47]. *TDS: une structure de répertoires pour les fichiers TeX*, June 2004. French translation of [236] by Jean-Côme Charpentier, with a postface by Fabrice Popineau. Available from: http://www.gutenberg.eu.org/publications/publicationsPDF/44-twg-tds.pdf. 219, 225

[236] TUG Working Group on a TeX Directory Structure (TWG-TDS). *A Directory Structure for TeX Files*, June 2004. mailto:tds@tug.org. Available from: http://www.tug.org/tds/tds.pdf. 219, 225, 542

[47] TWS-TDS = *TUG Working Group on a TeX Directory Structure (TWG-TDS)*.

[237] Hideo Umeki. *The `geometry` package*, July 2002. mailto:hideo.umeki@toshiba.co.jp. Available from: http://www.dante.de/tex-archive/contrib/macros/latex/contrib/geometry/manual.pdf. 112

[238] *The Unicode Standard* [online], February 2006. Available from: http://unicode.org. 200

[239] Piet van Oostrum. *TEX Implementations*, November 2002. mailto:piet@cs.uu.nl, Department of Computer Science, Padualaan 14, De Uithof, PO Box 80.089, 3508 TB, Utrecht, The Netherlands. Available from: http://www.cs.uu.nl/~piet/teximpl.pdf.

[240] Thomas van Oudenhove de Saint Géry. *Bibliographie avec* BibTEX, 2003. Available from: http://www.enstimac.fr/~vanouden/LaTeX/contrib/Pres_Bibtex.tar.gz.

[241] Suki K. Venkatesan. *Moving from bytes to words to semantics*. TUGboat, 26(2):165–169, August 2005. Proceedings of the 2005 Annual Meeting. 470

[242] André Violante. *LATEX: Notions de base*. Dunod, January 2007.

[243] Carl F. Voegelin and Florence M. Voegelin. *Classification and Index of the World Languages*. Elsevier, Amsterdam, December 1977.

[244] Vladimir Volovich, Werner Lemberg, and LATEX3 Project Team. *Cyrillic language support in LATEX*, March 1999. Available from: http://www.dante.de/tex-archive/macros/latex/doc/cyrguide.pdf. 191

[245] Stephan P. von Bechtolsheim. *TEX in practice: Comments on a 4-volumes, 1400-pages series on TEX*. TUGboat, 11(3):409–412, September 1990. http://www.tug.org/TUGboat/Articles/tb11-3/tb29bechtolsheim.pdf.

[246] Stephan P. von Bechtolsheim. *TEX 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.

(Une approche progressive des secrets de TEX). 254

[247] Staszek Wawrykiewicz. *PLATEX i pdfPLATEX*, May 2005. In polish, mailto:staw@gust.org.pl. Available from: http://www.gust.org.pl/doc/poczatki/platex.html. 41, 216, 498

[248] Peter R. Wilson. *The `stdclsdv` package*, July 2000. mailto:herries.press@earthlink.net. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/stdclsdv/stdclsdv.pdf. 217, 218

[249] Peter R. Wilson. *The `needspace` package*, January 2003. mailto:herries.press@earthlink.net. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/misc/needspace.sty. 54, 211

[250] Peter R. Wilson. *The* `tocloft` *package*, September 2003. mailto:herries.press@earthlink.net. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/tocloft/tocloft.pdf. 42, 61, 68, 75, 85, 112, 125, 210, 211, 220, 384

[251] Peter R. Wilson. *The* `abstract` *package*, April 2004. mailto:herries.press@earthlink.net. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/abstract/abstract.pdf. 50, 67, 85, 193, 231, 387

[252] Peter R. Wilson. *The* `appendix` *package*, April 2004. mailto:herries.press@earthlink.net. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/appendix/appendix.pdf. 60, 85, 194, 231, 383

[253] Peter R. Wilson. *The* `tocbibind` *package*, May 2004. mailto:herries.press@earthlink.net. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/tocbibind/tocbibind.pdf. 47, 85, 89, 94, 98, 99, 136, 210, 211, 219, 230, 383

[254] Peter R. Wilson. *The* `tocvsec2` *package*, May 2004. mailto:herries.press@earthlink.net. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/tocvsec2/tocvsec2.pdf.

[255] Peter R. Wilson. *The* `ccaption` *package*, March 2005. mailto:herries.press@earthlink.net. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/ccaption/ccaption.pdf. 51, 69, 85, 196, 253

[256] Peter R. Wilson. *The* `docmfp` *package*, March 2005. mailto:herries.press@earthlink.net. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/docmfp/docmfp.pdf.

[257] Peter R. Wilson. *The Memoir Class*, September 2005. mailto:herries.press@earthlink.net. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/memoir/memman.pdf. 62, 69, 85, 207, 230, 232, 253, 262, 386

[258] Peter R. Wilson. *The Memoir Class (Addendum)*, December 2005. mailto:herries.press@earthlink.net. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/memoir/memmanadd.pdf. 62, 69, 85, 207, 230, 232, 253, 262, 386

[259] Peter R. Wilson. *The* `romannum` *package*, May 2005. mailto:herries.press@earthlink.net. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/romannum/romannum.pdf. 50, 72, 85, 217, 233, 251

[260] Peter R. Wilson and Donald Arseneau. *The* `iftmarg` *package*, March 2000. mailto:herries.press@earthlink.net, mailto:asnd@triumf.ca. Available from: http://www.dante.de/tex-archive/macros/latex/contrib/misc/ifmtarg.sty. 204, 254

[261] Charlie S. Zender. *LaTeX Cheat Sheet*, August 2006. mailto:zender@uci.edu, Department of Earth Science, University of California, Irvine, CA 92697-3100. Available from: http://www.dust.ess.uci.edu/doc/ltx/ltx.pdf.

[262] Vadim V. Zhytnikov. *The LHCYR collection* [online], December 1998. Available from: http://ftp.dante.de/tex-archive/macros/latex/contrib/supported/lhcyr/. mailto:vvzhy@td.lpi.ac.ru. 506, 507

Changes History

⋆ **version 00**

- 1990/10/01
  - Original version, by Nigel WARD.
- 1991/11/01
  - Revised to reuse `\chapter`, `\section`, `\subsection` commands transparently, generate toc-file-name automatically, assorted other cleanup (Dan JURAFSKY).

⋆ **version 01**

- 1993/06/01
  - New design, to avoid allocating a newwrite, or file descriptor, for each chapter (a deadly sin!) (Jean-Pierre F. DRUCBERT).
  - 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).
  - Replaced the `minipage` environment by a `verse` environment, to allow a minitoc being split across pages.
  - All the layout of the minitoc is in the `\minitableofcontents` command, so if someboby wants to redefine that layout, he has just to rewrite it (and only it).
  - 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).
  - Problems: you *must* have `\chapterend` to terminate each chapter with a minitoc. How about avoiding this constraint?
  - The depth of the minitoc is user-adjustable with the counter `minitocdepth` (similar to `tocdepth` for the table of contents).
  - 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).
  - 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?).

- Some mods added to work with xr (external references). xr version 5 is much more tolerant.

⋆ **version 02**

- 1993/07/05
  - Added compatibility with hangcaption (the package hangcaption (if present) must be loaded *before* the minitoc package). *Beware* to options modifying `\@caption`.

※ **version 03**

- 1993/07/09
  - Version 3 not released (buggy).

⋆ **version 04**

- 1993/07/09
  - Added `\if@realch` to avoid contents lines from pseudo-chapters to go into the toc.
  - The package mtcoff allows you to use a LaTeX document with minitoc commands and to make them transparent: just replace the minitoc package by mtcoff.

⋆ **version 05**

- 1993/07/13
  - Added a selection mechanism to not write spurious things in the minitocs.

⋆ **version 06**

- 1993/07/15
  - Fixed problems about chapters in the toc, removed obsolete `\caption` stuff (filters are better) added compatibility with toch (toch makes a table of chapters; if used, must be loaded *before* minitoc; in fact, it is the ancestor of the shorttoc [105] package).

⋆ **version 07**

- 1993/07/22
  - (*major differences*) Completely rewritten, using tricks from xr (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.

⋆ **version 08**

- 1993/07/29
  - Spacing adjustements.

⋆ **version 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!).

⋆ **version 10**

- 1993/08/05
  - Works now with appendices. Detects now the obsolete versions of `latex.tex` (`\@inputcheck` or `\reset@font` not defined).

⋆ **version 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.

⋆ **version 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` environnement.
  - Removed old commented out lines from previous versions.

⋆ **version 13**

- 1993/12/17
  - Added `minilof` and `minilot` stuff. For MS-DOS, uncomment the definition of `\SHORTEXT`.

⋆ **version 14**

- 1994/01/03
  - Corrected space under `minitoc/lof/lot` and added a `\raggedright` setting to avoid "underfull" warnings.
  - Corrected some spacing problems (avoiding ~'s). `\mtifont` is changed from `\normalsize\bf` to `\large\bf`.
  - 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`.
  - Removed the setting of `\clubpenalty` and `\widowpenalty` to `10 000` (done by `\samepage`), and `\noindent`.
  - Simplified processing of optional argument in `\minitoc`, `\minilof` and `\minilot`.

⋆ **version 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.

- 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.

⋆ version 16

- 1994/02/02
  - Bug fixes (typos).

⋆ version 17

- 1994/06/23
  - '`n`' (null) synonym of '`e`' (empty) in the optional argument of `\minitoc`, `\dominitoc`, and siblings.
  - Compatibility with "LaTeX 2_ε". Thanks to Denis B. Roegel (who found the problem) and Frank Mittelbach (who gave the hints to solve).

⋆ version 18

- 1994/06/26
  - Make minitoc really compatible with LaTeX 2_ε.
  - 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).

⋆ version 19

- 1994/08/16
  - Added stuff for numbering of chapters (parts, sections) not starting at 1. `\firstchapteris` etc. commands added.
  - `\mtcrule`, `\nomtcrule` etc. commands added.
  - Corrected a bug in `\c@mti`.
  - Corrected `mtcswedish.sty` (Jan Michel Rynning).
  - Corrected appendix in articles.

⋆ version 20

- 1994/08/25
  - Corrected spacing before and after minitocs and siblings.
  - 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 vertical spacing.
  - Corrected a problem with chapters numbered with uppercase roman numbers.

⋆ version 21

- 1994/09/07

- Corrected typos in minitoc and `minitoc.tex`.

⋆ **version 22**

- 1994/10/10
  - Corrected typos in minitoc.

⋆ **version 23**

- 1994/11/08
  - Added a missing line in `\sectlof`.
  - Works with document classes resetting chapter (or section) number at each part (thanks to Denis B. Roegel).
  - Added the notion of "absolute numbering" for the mini-tables.
  - Removed stuff for `\firstchapteris` and co. These commands are obsolete.
  - Removed appendix stuff.

⋆ **version 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 LaTeX 2_ε release of December 1994. Thanks to Denis B. Roegel who did the work.

⋆ **version 25**

- 1996/09/13
  - Updated `mtcnorsk.sty` and added `mtcnynorsk.sty` on a suggestion from Dag Langmyhr.

⋆ **version 26**

- 1996/11/14
  - Language specific files are now named *language*`.mld` (replacing `mtc`*language*`.sty`) because they are not packages and it makes shorter names.
  - Added breton, estonian, germanb, greek, irish, russianb, scottish, lower and upper sorbian; renamed "esperanto" by "esperant" like in the babel package.

⋆ **version 27**

- 1996/12/20
  - Corrections for starred sectionning commands.
  - `english.mld` loaded as default language.
  - Added `vietnam.mld` and `arab.mld`.
  - Renamed minitocoff into mtcoff to keep the name short.

⋆ **version 28**

- 1997/10/29

  - Added the afrikaan(s), brazil, and ethiopia(n) languages.
  - Added autoconfiguration of extensions.
  - Added the `shortext` package option.
  - Added *coffee* stuff.
  - Added `\addstarred` stuff (for starred chapter stuff).
  - Fixed bug in parttocs.
- 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.
- 1998/12/03
  - Added the `tight` and `loose` package options.

⋆ **version 29**

- 1999/03/16
  - Added the bicig, buryat, mongol and russianc languages.
- 1999/06/28
  - Added the armenian language (from ArmTEX).
- 1999/07/23
  - Added the `dotted`/`undotted` package options (default: `dotted`).
- 1999/07/29
  - Added the lithuanian language.

⋆ **version 30**

- 1999/12/06
  - Added the basque, ngermanb, serbian, ukraineb, and welsh languages.
  - Corrected a bug in `\sltname` definition (`mlt` should be `slt`).

⋆ **version 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.

⋆ **version 32**

- 2000/08/08
  - 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.
  - `\nomtcrule` corrected.

⋆ version 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.
- 2000/12/08
  - Corrected a feature in `\mtcaddchapter` and co. with a blank optional argument.

⋆ version 34

- 2000/12/13
  - Added in the documentation a section for use with the tocbibind package.
  - Added `.mld` files for alternate names of languages: so, `american.mld` just loads `english.mld`, which contains the real definitions.

⋆ version 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`.
- 2001/02/26
  - Added `bulgarian.mld`, `hebrew.mld`, `icelandic.mld`, `latin.mld`, and `samin.mld`.
- 2001/03/09
  - Added `\mtcselectlanguage`.
- 2001/06/01
  - Fixed the `estonian` package option (missing).
- 2001/07/04
  - Added the interlingua language.

⋆ version 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).
- 2002/02/18
  - Corrected a spacing problem with empty titles (problem signalled by Frank Mittelbach).
  - Workaround for the `\parttoc`-`\chapter*` problem.
- 2002/02/19

  - Added `\mtcskip` and `\mtcskipamount`.
- 2002/02/27
  - Fixed test for empty files.
- 2002/03/13
  - Added the `bangla` language.
- 2002/03/15
  - Reduced depth of `\mtc@strutbox`.

※ version 37

- 2003/01/24
  - Version #37 dropped.

⋆ version 38

- 2003/01/24
  - `pt` becomes `\@pt` and `0pt` becomes `\z@`.
  - `\hrule` and `\vrule` replaced by `\rule` (LaTeX).
  - Added `mtc@zrule` for zero-dims rules.
  - Added the frenchb language (synonym of french).
- 2003/01/30
  - Changed the test for empty titles.
  - Added the `flsection` and `flsectionb` package options.
- 2003/01/31
  - The `tight` and `loose` package options are applied to `\parttoc` (Thomas Leonhardt).
- 2003/02/07
  - Package options `flsection` and `flsectionb` removed and replaced by the `insection` package option (like `flsectionb`).
- 2003/02/11
  - Corrected numbering of SLF, SLT.
- 2003/02/20
  - Added `frenchle` and `frenchpro` language options (synonyms of `french`).
  - Corrected secttocs, at least.
- 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.

⋆ version 39

- 2003/04/09
  - Modern font commands for compatibility with the memoir class.
  - `\nomtcpagenumbers` and memoir class.
- 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).
- 2004/09/08
  - Added language options and `.mld` files for dialects: `canadian` (`english`), `acadian`, `acadien`, `canadien` (`french`), `naustrian`, `ngerman` (`ngermanb`).
  - Added comments in `.mld` files using special fonts.
  - Documentation: added a paragraph about making a TOC for appendices, eventually not listed in the main TOC.
- 2004/09/17
  - Corrections in the documentation; corrections about rules.

⋆ version 40

- 2004/12/09
  - 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.
  - Added a figure in `minitoc.tex` about the need of three compilations.
  - Added some infos in `minitoc.bug`.
  - Added a paragraph about a problem with the appendix package.
- 2004/12/13
  - Updated `fminitoc.bib` and `minitoc.bib`.
- 2004/12/14
  - Added the `hints` package option. This option is still experimental; your advice is welcome.
- 2004/12/20
  - Added `fminitoc.pdf` (french documentation in PDF format).

⋆ version 41

- 2005/01/05
  - Corrections in documentation.
  - Message added if some sectionning commands are not available.
  - Replaced `\typeout` commands in minitoc 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 *document*`.log` file, not on the terminal).
- 2005/01/06
  - Added the `\mtcsetfont` (Benjamin Bayart) and `\mtcsettitlefont` commands, with a much simpler syntax.
- 2005/01/10
  - Added bibliography.
- 2005/01/11

  - $\mathcal{A}\mathcal{M}\mathcal{S}$ classes: amsart and amsproc are incompatible with minitoc, amsbook needs precautions.
- 2005/01/12
  - Added `\mtcsetformat`.
- 2005/01/18
  - Added `\mtcsettitle`.
  - Added a hint for recommending the `insection` package option.
- 2005/01/19
  - Added a hint about the presence of `\dominitoc` and co.
  - Added a hint about consistency of `\dominitoc/\minitoc` and co.
  - Improved documentation about hints.
- 2005/01/20
  - Added a hint about using short extensions with more that 99 parts or 99 chapters or 99 sections.
- 2005/01/25
  - `\ptifont`: `\Huge\bfseries` becomes `\LARGE\bfseries`.
- 2005/01/26
  - Added `\mtcsetpagenumbers`.
- 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`.
- 2005/02/02
  - Added `\mtcsetrules`.
- 2005/02/03
  - Added `\plfrule`, `\noplfrule`, `\mlfrule`, `\nomlfrule`, `\slfrule`, `\noslfrule`, `\pltrule`, `\nopltrule`, `\mltrule`, `\nomltrule`, `\sltrule`, `\nosltrule`.
- 2005/02/04
  - Added the `mtchideinmaintoc` environment.
- 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.
- 2005/02/09
  - Added the `mtchideinmainlof` and `mtchideinmainlot` environments.
- 2005/02/10
  - Added tests on the `mtchideinmain*` environments.
- 2005/02/14
  - Added `\mtcfixindex`.

※ version 42

- 2005/02/14
  - Version 42 not released.
  - Replaced “language” by “langue” in the french documentation.
- 2005/02/15
  - Fixed a minor typo.
- 2005/02/16
  - Upgraded `\mtcfixindex`.
- 2005/02/21
  - Added `\mtcsettitle`, forgotten to be inserted in v41.

⋆ version 43

- 2005/02/21
  - Version 43: consolidation of v40, v41 and v42.
- 2005/02/24
  - Fixed a big bug in `\mtcsetformat`.
  - Fixed a bug in `mtcoff.sty` about `\mtcfixindex`.
- 2005/03/02
  - Fixed the `\mtcset...` macros.
  - Moved history to the end of package code.
  - Added the `INSTALL` file and a chapter about installation.
- 2005/03/07
  - Fixed a typo (Benjamin Bayart).
  - Completed the hint about consistency of `\dominitoc/\minitoc` and co.
- 2005/03/08
  - Added a hint about consistency of `\minitoc` and `\tableofcontents`.
- 2005/03/09
  - Added comments about fonts.
- 2005/03/10
  - Corrections in documentation.
- 2005/03/11
  - Added `\mtcsetfeature`.
- 2005/03/14
  - Added `bulgarianb.mld` (upper bulgarian).
- 2005/03/15
  - Added `*[-\baselineskip]` after the `\\` after the top rule of each part level mini-table.
- 2005/03/16
  - Corrections in the arguments of `\mtcsetfeature`.

- 2005/03/18
  - Removed `\markboth` for minitocs (...) and secttocs (...).
- 2005/03/21
  - Added `spanish2.mld`.
- 2005/03/22
  - Added a hint for the abstract package.
- 2005/04/07
  - Corrected the `stc@verse` environment.
  - Added `finnish2.mld`, `latin2.mld`, and `magyar2.mld`.
- 2005/04/08
  - Renamed `portuges.mld` as `portugues.mld`.
- 2005/04/12
  - Correction in `\mtcskip`.
  - First version in `.dtx` format.
- 2005/04/14
  - Removed `\ypart`, `\ychapter`, `\ysection`, and stuff; unused.
- 2005/05/11
  - Corrected a typo in `\@dosectlot`.
  - Added `\mtcfixglossary`.
  - Print the documentation with "`oneside`" to have all marginal notes on left. Added the (extended to 54 floats) code of morefloats (Don Hosek) to allow more marginpars and floats.
  - Added `minitoc.ist` to format the index correctly.
- 2005/05/26
  - Fixed rules in parttocs, partlofs and partlots.
- 2005/05/30
  - Fixed chapter-level entries in parttocs, when page numbers must be removed.
  - Added a hint about the sectsty package (must be loaded *before* minitoc).
- 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.
- 2005/06/02
  - Added the notion of depth for mini-tables of figures/tables.
  - Added `\mtcsetdepth`.
  - The `hints` option is the default and no more considered as experimental.
- 2005/06/03
  - Added an error message in `\mtcsetdepth` if the counter is not available.
- 2005/06/06

  - Added `portuges.mld`, which loads `portugues.mld`.
- 2005/06/07
  - Added three variants for the malayalam language: `malayalam-keli.mld`, `malayalam-rachana.mld`, and `malayalam-rachana2.mld`.
- 2005/06/14
  - Added method for bilingual documentation.
- 2005/06/15
  - Added `fminitoc.ist` to format correctly the index in french.
- 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.
- 2005/06/17
  - The file `fminitoc.dtx` is now generated by `minitoc.ins`.
- 2005/06/21
  - Added "OUI", "NON", "oui", "non", "O", and "o" as true/false keywords.
  - Compacted the code about detection of short/long extensions.
- 2005/06/22
  - Added "VRAI", "FAUX", "vrai", "faux", "V", and "v" as true/false keywords.
- 2005/06/23
  - Correctly set the `\ifFTR` flag to have the names of months in the right language in the bibliography.
- 2005/06/29
  - Set the flag `\mtcoffwarn@true` in mtcoff if a command `\mtcadd...` is found.
- 2005/07/01
  - Added `castillian.mld`.
  - Renamed `portugues.mld` as `portuguese.mld`.
- 2005/07/11
  - Added `brazilian.mld`, `british.mld`, `UKenglish.mld`, and `USenglish.mld`.
- 2005/07/12
  - Suppressed "General:" in the changes history.
- 2005/07/13
  - Replaced some `\PackageWarning` commands by `\PackageInfo`.
- 2005/07/18
  - Restoring the correspondence of each language option with a `.mld` file.
- 2005/07/20
  - Improving the `mtchideinmainlof` and `mtchideinmainlot` environments.
- 2005/07/21

  - Removing unused some flags `\if@mtc@setpagenumbers@act@` and `\if@mtc@setrules@act@`.
  - Added the `\decrementptc`, `\decrementmtc`, and `\decrementstc` commands.
- 2005/07/22
  - Corrected a bug in mtcoff.
  - Improved some messages in mtcoff.
  - Added a test on the version of the placeins package.
- 2005/08/23
  - Added a note about `\FloatBarrier`.
- 2005/08/24
  - Added a note about an alignment problem in the minitocs. Updated `minitoc.bug`.
  - Made two versions of the `mtchideinmainlof` and `mtchideinmainlot` environments, depending of the presence of the corresponding depth counter.
  - The memoir class is incompatible if too recent.
- 2005/08/25
  - Added a comment about the position of the `\do...` preparation commands.
  - Corrections in the `mtchideinmainlof` and `mtchideinmainlot` environments.
- 2005/08/26
  - Added `guarani.mld`.
- 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.
- 2005/09/02
  - Added a patch for the recent version of the memoir class.
- 2005/09/06
  - Added `spanish3.mld`.
- 2005/09/08
  - Use `\mtcselectlanguage` in language options and in "secondary" `.mld` files.
- 2005/09/09
  - Added `\mtcloadmlo` to be used in some `.mld` files to load a `.mlo` file.
- 2005/09/12
  - Added a test to forbid direct calls of `\mtcloadmlo` by the user.
- 2005/09/13
  - Added `farsi1.mld`, `farsi1.mlo`, `farsi2.mld`, and `farsi2.mlo`.
  - Added a note about the `rubber` tool.
- 2005/09/15

  - Added `mtcglo.ist` to format the glossary.
- 2005/09/16
  - Removed the page numbers in the glossary. Done in the `*mk` scripts.

⋆ **version 44**

- 2005/09/26
  - Changes history (glossary) typeset in `RaggedRight`.
- 2005/09/27
  - Added `germanb2.mld`, `ngermanb2.mld`, `norsk2.mld`, and `nynorsk2.mld`.
- 2005/09/28
  - New method for history: embedded lists on 3 levels.
  - Removed `mtcglo.ist`.
- 2005/09/29
  - Cleaned the `*mk` scripts.
  - Added the `listfiles` package option.
- 2005/09/30
  - Corrected typos.
  - Added the name of the `.maf` file in the message of the `listfiles` package option.
  - Improved the cleaning in the `*mk` scripts, using a `.maf` file.
- 2005/10/03
  - Load the patch for the memoir class only if necessary; do not load it if memoir is dated after 2005/09/25.
  - Added a remark in the FAQ chapter (and `minitoc.bug`) about precautions to take with the starred sectionning commands.
- 2005/10/04
  - Added the `nolistfiles` package option.
  - Added a hint about the caption, caption2, ccaption, and mcaption packages (they must be loaded *before* minitoc).
- 2005/10/05
  - Fixed typos in the documentation.
  - Fixed some marginal notes in the commented code.
- 2005/10/06
  - Minor corrections in the documentation.
  - Use the `xargs` Unix command in the `*mk` scripts to remove the auxiliary files.
- 2005/10/07
  - Minor corrections in the documentation.
  - Added a short intro to the "Frequently Asked Questions" chapter and to `minitoc.bug`.
- 2005/11/02

  - Minor corrections in the documentation.
- 2005/11/04
  - Minor corrections in the documentation.
- 2005/11/07
  - Begin adding the "Jargon" chapter.
- 2005/11/08
  - Added the french LaTeX Companion [191].
- 2005/11/09
  - Continuing the "Jargon" chapter.
  - Adding `minitoc.pre` in `minitoc.l`.
  - Adding a note about the need of running `imk` before `emk` or `fmk`.
- 2005/11/10
  - Fixed typos in the documentation.
  - Added a note about a problem with minitoc, hyperref and memoir.
  - Continuing the "Jargon" chapter.
- 2005/11/14
  - Fixed typos in the documentation.
  - Continuing the "Jargon" chapter.
  - Improve the notes about the memoir class.
- 2005/11/15
  - Continuing the "Jargon" chapter.
  - Improve the notes about the memoir class.
  - Added `\plfSfont`, `\pltSfont`, `\mlfSfont`, `\mltSfont`, `\slfSfont`, and `\sltSfont` for subfigures and subtables entries in the mini-tables.
- 2005/11/16
  - Continuing the "Jargon" chapter.
  - Fixed a bug about fonts for subfigures and subtables entries in the mini-tables.
  - Added `bicig2.mld`, `bithe.mld`, `manju.mld`, `xalx.mld`, and `khalkha.mld`.
- 2005/11/17
  - Continuing the "Jargon" chapter.
  - Added testing via internal *quarks* commands in `\mtcsetfont`.
- 2005/11/18
  - Continuing the "Jargon" chapter.
  - Fixed typos in the documentation.
- 2005/11/21
  - Continuing the "Jargon" chapter.
- 2005/11/22
  - Continuing the "Jargon" chapter.
- 2005/11/23

  - Continuing the “Jargon” chapter.
  - Updating the bibliography.
- 2005/11/24
  - Continuing the “Jargon” chapter.
- 2005/11/25
  - Continuing the “Jargon” chapter.
  - Changed “table” into “tableau” in the french doc, where necessary.
  - Updating the bibliography.
- 2005/11/28
  - Continuing the “Jargon” chapter.
  - Fixed typos in the documentation.
- 2005/11/29
  - Continuing the “Jargon” chapter.
  - Fixed typos in the documentation.
  - Fixed typos in the bibliography.
  - Updating the bibliography.
- 2005/11/30
  - Continuing the “Jargon” chapter.
  - Avoid some warnings “Token not allowed” from `pdftex`.
- 2005/12/01
  - Continuing the “Jargon” chapter.
- 2005/12/02
  - Continuing the “Jargon” chapter.
  - Reordering a long sequence of citations.
  - Added `mailto:` in the mailing URLs.
- 2005/12/05
  - Fixed typos in the documentation.
  - Continuing the “Jargon” chapter.
  - Added a hint about the varsects package (must be loaded *before* minitoc).
- 2005/12/06
  - Continuing the “Jargon” chapter.
  - Correcting an hyperlink in the bibliography (for the xr package).
  - Attempting to avoid broken URLs, using `quote`, footnotes and `\par`.
- 2005/12/07
  - Continuing the “Jargon” chapter.
  - Updating the bibliography.
  - Corrections of layout (some headers, a table).
  - In the warning message of the hint about a number of mini-tables greater than 99 (if short extensions), give the effective number.
  - Reduce the width of some info, warning or error messages.

- 2005/12/08
  - Corrections of layout (some headers).
- 2005/12/09
  - Corrections of french quotes.
  - Added some PDF options.
  - Continuing the "Jargon" chapter.
  - Corrected an URL to the $\mathcal{A}\mathcal{M}\mathcal{S}$ in the bibliography.
- 2005/12/19
  - Made some messages shorter (mainly by removing stars).
- 2005/12/21
  - Correction of typos.
  - Added some labels.
  - Added a chapter with the (explained) messages. Not yet sorted.
  - The documentation needs 4 LaTeX runs.
- 2005/12/22
  - Made some messages shorter.
  - Corrections in the list of messages.
  - Updating the bibliography.
- 2005/12/23
  - Improving the placement of floats on pages of floats: to the top.
- 2006/01/03
  - Corrections in the documentation (thanks to Markus Gleiszner).
  - Added `addsec.tex`
- 2006/01/04
  - Corrected the flag `\ifundottedmtc`.
  - Correction to make `addsec.tex` work.
- 2006/01/05
  - Added "*" as keyword for the first argument of `\mtcsetpagenumbers` and `\mtcsetrules` (asked by Markus Gleiszner).
  - Removed "`\MessageBreak`" from the index.
- 2006/01/06
  - Continuing the "Jargon" chapter.
  - Corrected the bibliography entry about BangTeX.
  - Updated the bibliography.
  - Used the afterpage package in the documentation to fix a float positionning problem.
- 2006/01/09
  - Corrections in the documentation.
  - Fixing a float positionning problem.
- 2006/01/10

  - Corrections in the documentation.
  - Continuing the "Jargon" chapter.
  - Updated the bibliography.
  - Added the bahasam language.
  - Added the albanian language.
  - Added the hebrew2 language.
- 2006/01/11
  - Updated the bibliography.
  - Updated the documentation for the albanian, bahasa, bahasam, and hebrew2 languages.
  - Updated `french.mld` (removing abusive uppercase letters).
  - Corrected the `italian.mld` file. Added the italian2 language.
  - Added the australian and newzealand languages (english).
  - Renamed the bahasa language as bahasai; bahasa is synonym of bahasai.
  - Added the malay and meyalu languages, synonyms of bahasam.
  - Added the indon and indonesian languages, synonyms of bahasai.
- 2006/01/12
  - Updated the bibliography.
  - Updated the acknowledgements.
  - Added references to the new bibliographic entries.
- 2006/01/13
  - Fixed an instability in page breaks in the documentation of `japanese3.mld`.
  - Added comments in some `.mld` files.
  - Added `magyar3.mld`.
  - Updated `lithuanian.mld`.
- 2006/01/16
  - Correction in `\mtcaddsection`.
- 2006/01/17
  - Correction in `\mtcfixindex` and `\mtcfixglossary`.
  - Updated the bibliography.
  - Limitation of the initial depth of displayed bookmarks.
- 2006/01/18
  - Added some comments in point 34 of the FAQ (and in `minitoc.bug`) about the initialization of fonts.
  - Added `romanian2.mld` and `romanian3.mld`.
  - Updated the bibliography.
- 2006/01/19
  - Updated the bibliography.
  - Load some packages before hyperref.
  - Added `spanish4.mld`.
- 2006/01/23

  - Corrected the table about default titles.
  - Corrected the keywords for `\mtcsetfont`.
  - Added `lowersorbian.mld`, `uppersorbian.mld`, and `ukrainian.mld`.
- 2006/01/24
  - Updated documentation for `lowersorbian.mld`, `uppersorbian.mld`, and `ukrainian.mld`.
- 2006/01/25
  - Corrections in the documentation.
  - Updated the bibliography.
- 2006/01/26
  - Added a hint about the KOMA-Script classes [147, 195], and an entry in the FAQ chapter (and in `minitoc.bug`).
- 2006/01/27
  - Updated the bibliography.
  - Added a note in documentation of `serbian.mld` and `serbianc.mld`.
- 2006/01/30
  - Added `ethiopian2.mld` (for Omega).
- 2006/01/31
  - Simplifications in the "Messages" chapter.
  - Corrections in the "Jargon" chapter.
- 2006/02/01
  - Corrections in the documentation.
  - Added the "Postface" chapter.
- 2006/02/02
  - Corrections in the "Postface" chapter.
  - Updated the bibliography.
- 2006/02/06
  - Corrections in the documentation.
  - Updated the bibliography.
  - Added package dblaccnt for the "The pdfTEX Program" entry in the bibliography. Its author's first name needs a double accent (Hàn Thế Thành).
- 2006/02/07
  - Corrections in the documentation.
  - Updated the bibliography.
- 2006/02/09
  - Corrections in the documentation.
- 2006/02/10
  - Corrections in the documentation.
  - Updated the bibliography.
- 2006/02/13

  - Added `malayalam-omega.mld` and `malayalam-omega.mlo`.
  - Updated the bibliography.
- 2006/02/14
  - Added `kannada.mld`.
  - Updated the bibliography.
- 2006/02/15
  - Corrections in `russianb.mld` and `spanish.mld`.
  - Corrections in the documentation and the bibliography.
  - Place `\mtcfixglossary` before `\mtcfixindex`.
- 2006/02/16
  - Added a citation from Donald Arseneau.
  - Updated the bibliography.
  - Updated the acknowlegments.
- 2006/02/17
  - Updated the bibliography.
  - Updated the jargon.
- 2006/02/20
  - Added `u8hangul.mld`, `u8hangul.mlo`, `u8hanja.mld`, and `u8hanja.mlo`.
- 2006/02/21
  - Renamed languages u8hangul and u8hanja into hangul-u8 and hanja-u8.
  - Updated the bibliography.
- 2006/02/22
  - Added a hint about repeated preparation commands.
  - Moved up the declaration of some flags relative to the `hints` option.
  - Added `\mtcprepare`.
- 2006/02/23
  - Updated the bibliography.
- 2006/02/24
  - Updated the bibliography.
- 2006/02/27
  - Corrections in the documentation.
  - Added `minitoc.pre` to class 6.
- 2006/02/28
  - Corrections in the documentation.
  - Corrected the position of tables in the "Jargon" chapter.
  - Corrected `irish.mld`, `lsorbian.mld` and `usorbian.mld`.
  - Added `polski.mld`.
- 2006/03/01
  - Hints about the jura class and the alphanum package, incompatible with minitoc.

- 2006/03/02
  - Use bibliographic styles with an URL field, built with the help of `urlbst` [118].
- 2006/03/06
  - Update the bibliography.
- 2006/03/08
  - Corrections in `magyar.mld`, `magyar2.mld`, and `magyar3.mld`.
  - Added `russian-cca.mld`, `russian-cca1.mld`, and `russian-lh.mld`, with their `.mlo` files.
- 2006/03/09
  - Update the bibliography.
- 2006/03/10
  - Update the bibliography.
  - Added `russian-lhcyralt.mld`, `russian-lhcyrkoi.mld`, and `russian-lhcyrwin.mld`, with their `.mlo` files.
- 2006/03/13
  - Corrections in the documentation.
- 2006/03/14
  - Added the mtcmess package.
- 2006/03/16
  - The messages are now numbered.
  - Update the bibliography.
- 2006/03/20
  - Corrections in the documentation.
- 2006/03/21
  - Update the bibliography.
- 2006/03/22
  - Update the jargon.
- 2006/03/28
  - Corrections in the documentation.
  - Update the jargon.
- 2006/03/29
  - Added FAQ 37 about `.mld` files and babel.
  - Added `french1.mld` and `french2.mld`.
  - Update the jargon.
- 2006/03/30
  - Added `english1.mld` and `english2.mld`.

⋆ **version** 45

- 2006/03/31
  - Suppression of the PostScript versions of the documentation.

  - Added `arab2.mld`, `bicig3.mld`, `buryat2.mld`, `xalx2.mld`, and `xalx3.mld`.
- 2006/04/03
  - Corrections in the documentation.
- 2006/04/04
  - Added `swedish2.mld`.
  - The `insection` package option loads also the flafter package.
- 2006/04/05
  - Corrections in the documentation.
  - Added `lamed.eps` and `lamed.pdf` as images for the *Lamed* logo (built from `lamed.tex`).
  - Reordering of the chapters in the user's manual (part I).
- 2006/04/06
  - Use sectsty to better format section titles.
- 2006/04/07
  - Corrections in the documentation.
  - Added `\ifmtcsecondpart` to check if the document has exactly 2 parts.
- 2006/04/10
  - Corrections in the documentation.
- 2006/04/11
  - Corrections in the documentation.
- 2006/04/12
  - Corrections in the documentation.
- 2006/04/13
  - In the `insection` package option, load the flafter package *before* the placeins package.
  - Added a figure about the float barriers.
- 2006/04/27
  - Added notes in FAQ 20, about the use with the appendix package.
  - Added comments about the `insection` option.
  - Update the bibliography.
  - Begin correction of the `mtchideinmaintoc` environment.
- 2006/05/02
  - Added notes in `minitoc.bug`, point 20, about the use with the appendix package.
  - End correction of the `mtchideinmaintoc` environment.
  - Analogous corrections in the `mtchideinmainlof` and `mtchideinmainlot` environments.
  - Update the bibliography.
- 2006/05/03

  - Added notes about the `mtchideinmainlof` and `mtchideinmainlot` environments.
  - Added `hide1.tex` and `hide2.tex`.
  - Added a hint about the fncychap package (must be loaded *before* minitoc).
  - Added a hint about the quotchap package (must be loaded *before* minitoc).
  - Update the bibliography.
- 2006/05/04
  - Update the bibliography.
  - Added a hint about the romannum package (must be loaded *before* minitoc).
  - Added a hint about the sfheaders package (must be loaded *before* minitoc).
  - Added a hint about the alnumsec package (must be loaded *before* minitoc).
  - Corrections in the documentation.
- 2006/05/05
  - Corrections in the documentation.
- 2006/05/24
  - Corrections in the documentation.
  - Renamed `hide.tex` to `hide1.tex`.
  - Update the bibliography.
- 2006/05/30
  - Corrections in the documentation.
  - Use `\MakeUpperCase` in `\markboth` for page styles.
- 2006/05/31
  - Update the bibliography.
  - Added a hint about the captcont package (must be loaded *before* minitoc).
  - Corrections in the documentation.
  - Added `uighur.mld`, `uighur2.mld`, and `uighur3.mld` (as synonyms for the bicig variants).
- 2006/06/01
  - Corrections in the documentation.
  - Added description of MonTEX in the jargon.
  - Added a comment about the `imk` script in `INSTALL` and the "Installation" chapter.
  - Added an entry about "package" in the jargon.
  - Added the `mtc-apx.tex` example file.
  - Added FAQ 44 and the `\mtcgapbeforeheads` and `\mtcgapafterheads` commands.
- 2006/06/02
  - Corrections in the documentation.
  - Update the bibliography.
  - Added the `gaps.tex` example file.
- 2006/06/05

  - Corrections in the documentation.
  - Update the bibliography.
- 2006/06/06
  - Corrections in the documentation.
  - Update the bibliography.
- 2006/06/08
  - Corrections in the documentation.
  - Spacing correction in `french2.mld`.

⋆ **version** 46

- 2006/06/09
  - Corrections in the documentation and the bibliography.
- 2006/06/21
  - Update the bibliography.
  - Comment about the thailatex package.
- 2006/06/22
  - Generate some example files with `minitoc.dtx`/`minitoc.ins`.
  - Added chapter "Example files".
- 2006/06/23
  - Renamed chapter "Example files" as "Examples of documents".
  - Use the lipsum package [123] in some of the examples of documents.
  - Update the bibliography.
- 2006/06/27
  - Update the examples of documents.
  - Added the `second.tex` example file.
- 2006/06/29
  - Added the `amem.tex`, `mem.tex` and `mem1.tex` example files.
- 2006/06/30
  - Added the `fo1.tex`, `fo2.tex` and `scr.tex` example files.
- 2006/07/03
  - Added the `subf.tex` example file.
  - Corrections about the depth of minilofs, minilots and siblings.
- 2006/07/04
  - Added the `tsfc.tex` and `tbi.tex` example files.
  - Corrections in the bibliography.
- 2006/07/07
  - Corrections of typos.
  - Corrections in the bibliography.
  - Added the `2c.tex` and `mtc-bo.tex` example files.
  - Correction in `french2.mld`.

- 2006/07/10
  - Correction in minilots and minilofs (and siblings) about depth.
  - Added the `hop.tex` and `cri.tex` example files.
  - Update the bibliography.
- 2006/07/11
  - Added the `livre.tex`, `ch0.tex`, `tlc.tex` and `mu.tex` example files.
  - Update the bibliography.
- 2006/07/12
  - Update the jargon.
- 2006/07/13
  - Corrections in the documentation.
  - The not released versions are flagged by ※ in place of ⋆.
- 2006/07/17
  - The "About this document" section becomes a starred first chapter.
- 2006/07/18
  - Added the `hir.tex` and `hia.tex` example files.
- 2006/07/19
  - Update the bibliography.
  - Corrections in `add.tex` and `addsec.tex` for the index.
  - Added the `xmk` script to typeset the examples into PDF documents.
  - Updated the scripts to treat the examples.
- 2006/07/20
  - Do not forget `\jobname.mtc1` in the list of files.
  - In the scripts, the backup directory (`OLD`) is now `/tmp/`\`whoami`\`/OLD`.
  - In the scripts, the repartition directories (`CL[0-9]`) are now `/tmp/`\`whoami`\`/CL[0-9]`.

⋆ version 47

- 2006/07/26
  - Corrections in the documentation and the bibliography.
  - Update the bibliography.
- 2006/07/27
  - Added `arabi.mld` and `farsi3.mld` (from the Arabi system).
  - Update the bibliography.
- 2006/07/28
  - Update the jargon.
- 2006/07/31
  - Fixed `\l@xsection`.
  - Fixed some spacings in mini-tables.
- 2006/08/01
  - Added a `\kernafter...` vertical kern between each minitable and its bottom rule.

  - Added point 45 of the FAQ.
- 2006/08/03
  - Minor correction in warning message `F0008`.
  - Update the bibliography.
  - Fixed a bug in `romanian2.mld` and `romanian3.mld`.
  - Shortened the result of some example documents by using the report class in place of the book class (hence using one side printing).

⋆ **version** 48

- 2006/08/04
  - Fixed typos.
- 2006/08/22
  - Update the bibliography.
  - No preamble in `add.bib`.
- 2006/08/23
  - Corrections in the TOC formatting.
  - Increasing `\textwidth`.
  - Correction of the preamble problem in `add.bib` and all generated files.
- 2006/08/24
  - Remove comments about spurious lines in preamble of generated files.
  - Added `devanagari.mld` and `hindi.mld`.
- 2006/08/25
  - Update the bibliography.
  - Added `hindi-modern.mld`.
  - Corrected the `\name` macro (for the documentation).
- 2006/08/28
  - Corrections in the bibliography.
  - Correction (conversion) in `hindi-modern.mld`.
- 2006/08/29
  - Added error `E0036` if `english.mld` is not found to set the default titles.
- 2006/08/31
  - Update the bibliography.
  - Modified the `plainurl.bst` to have family names of authors and editors in small caps and years in old style digits. Titles are in emphasis. The `frplain1.bst` style is also updated.
- 2006/09/01
  - Update the bibliography.
  - The bibliographic styles `plainurl.bst` and `frplain1.bst` are renamed `en-mtc.bst` and `fr-mtc.bst`.
- 2006/09/05
  - Update the bibliography.

  - Renamed `add.bib` to `mtc-add.bib`.
  - Renamed `add.tex` to `mtc-add.tex`.
  - Renamed `addsec.tex` to `mtc-ads.tex`.
  - Renamed `2c.tex` to `mtc-2c.tex`.
  - The `listfiles` package option is now active by default.
- 2006/09/07
  - Renamed `app-mem.tex` to `mtc-amm.tex`.
  - Renamed `apx.tex` to `mtc-apx.tex`.
  - Renamed `bo.tex` to `mtc-bo.tex`.
  - Renamed `ch0.tex` to `mtc-ch0.tex`.
  - Renamed `cri.tex` to `mtc-cri.tex`.
  - Renamed `fo1.tex` to `mtc-fo1.tex`.
  - Renamed `fo2.tex` to `mtc-fo2.tex`.
  - Renamed `gaps.tex` to `mtc-gap.tex`.
  - Renamed `hia.tex` to `mtc-hia.tex`.
  - Renamed `hir.tex` to `mtc-hir.tex`.
  - Renamed `hide1.tex` to `mtc-hi1.tex`.
  - Renamed `hide2.tex` to `mtc-hi2.tex`.
  - Renamed `hop.tex` to `mtc-hop.tex`.
  - Renamed `livre.tex` to `mtc-liv.tex`.
  - Renamed `mem.tex` to `mtc-mem.tex`.
  - Renamed `mem1.tex` to `mtc-mm1.tex`.
  - Renamed `mini-art.tex` to `mtc-art.tex`.
  - Renamed `minitoc-ex.tex` to `mtc-bk.tex`.
  - Renamed `mu.tex` to `mtc-mu.tex`.
  - Renamed `scr.tex` to `mtc-scr.tex`.
  - Renamed `second.tex` to `mtc-2nd.tex`.
  - Renamed `subf.tex` to `mtc-sbf.tex`.
  - Renamed `tbi.tex` to `mtc-tbi.tex`.
  - Renamed `tlc.tex` to `mtc-tlc.tex`.
  - Renamed `tsfc.tex` to `mtc-tsf.tex`.
- 2006/09/08
  - Updated the bibliography (added the Pentaglot).
  - Corrected the format of two tables about NFSS.
  - Example documents in alphabetical order in their chapter.
- 2006/09/11
  - Updated the bibliography.
- 2006/09/12
  - Added a figure about systems derived from TEX and LATEX.
- 2006/09/13
  - Added the `mtc-syn.tex` example document file.

⋆ **version** 49

- 2006/09/14
  - Slightly modified the layout of the list of files (“Installation” chapter).
  - Simplifications in the scripts.
  - Updated the bibliography.
- 2006/09/18
  - Updated the bibliography.
  - Added point 46 in the FAQ and example file `mtc-tlo.tex`.
- 2006/09/26
  - Updated the bibliography.
  - Corrections in the bibliography and the bibliographic styles.
- 2006/09/29
  - Better error messages about undefined preparation and insertion commands.
  - Updated the bibliography.
  - Added “`+`” and “`-`” as synonyms for “`on`” and “`off`”, respectively.
- 2006/10/20
  - Corrections in the bibliography.
  - Fixed typos.
  - Updated the bibliography.
  - Added a table of some encodings.
- 2006/10/31
  - Suppressed the “Summary” entry in the summary, but added it in the Table of Contents.
  - Improving some tables.
  - Added the `japanese6.mld` and `japanese6.mlo` files.
  - Updated the bibliography.
- 2006/11/03
  - Corrections in the bibliography.
  - Corrections in formatting a citation from Donald Arseneau.
  - Combine four figures in one (with sub-figures).
  - Added (in the memento) a table of the classes and packages which are incompatible or need precautions with minitoc.
  - Added a hint about the hangcaption package (must be loaded *before* minitoc).
- 2006/11/06
  - Completed the list of the standard classes.
- 2006/11/09
  - Added a validation of the language options with the presence of the `.mld` and `.mlo` files.
  - Added notes about the mandatory presence of the `english.mld` file.
- 2006/11/13

- The validation of the language options writes only informative messages in the *document*`.log` file and, if necessary, gives only one warning message.

⋆ **version** 50

- 2006/11/17
  - Removed old examples of documents: `mtc-adds.tex`, `mtc-amem.tex`, `mtc-book.tex`, `mtc-gaps.tex`, `mtc-mem1.tex`, `mtc-subf.tex`, and `mtc-tsfc.tex`.
  - Updated the bibliography.
  - Added the `tmk` script and a table describing a TDS structure for minitoc.
  - Added an item about the TDS in the jargon.
  - Updated the `INSTALL` file and the "Installation" chapter.
- 2006/11/29
  - Added the warning message `W0094` with the list of the missing minitoc languages files (`.mld` and `.mlo`).
  - Corrections in the bibliography.
  - Updated the `INSTALL` file and the "Installation" chapter.
  - Changed the names of the scratch directories in some scripts.
  - Updated the bibliography.
  - Added the file `minitoc-texmf.zip` (a ZIP archive of a TDS-compliant hierarchy of all files of the package) to the distribution.

⋆ **version** 51

- 2006/12/18
  - Improving the index: packages and classes, scripts, tools, names, examples, extensions, options, language options.
- 2006/12/20
  - Improving the index: names.
  - Updated some `.mld` files with names of the authors of titles.
- 2007/01/09
  - Miscellaneous corrections.
  - The names of some internal macros are shortened to fit into the margin.
  - Added a `\ProvidesFile` command to the example files.
  - Indexing the environments (not perfect).
  - Indexing the files.
  - Renamed the file "`catalog`" into "`CATALOG`".
  - Indexing the counters and depth counters.
  - The example files are in their own directory in the (proposed) TDS hierarchy.

Index

Symbols

A

B

C

F

G

H

I

L

M

N

O

Q

R

S

T

U

V