

## CONTENTS

Starting with 1.08h, about 25 code samples (especially the longer ones) are incorporated into the PDF file (thanks to SCOTT PAKIN's `attachfile` package) as *file attachment annotations*. Each file is represented by an icon after the verbatim rendering of the corresponding code. Clicking suitably on the icon will open a dialog to open or save the file contents. This is much better than copy-paste which loses indentation. There is also in some viewer a *Comment pane* which can be opened on the right side of the main window and which provides a convenient clickable list of all these *attachments annotations*.

PDF viewers unable to make these attachments accessible to the user may replace the clickable icons by some blank space.

---

## TABLE OF CONTENTS

<b>Abstract</b>	p. 6
<b>Foreword</b>	1, p. 6
<b>License</b>	2, p. 7

### Part I. Overview

<b>Do I need to be a geek to use <code>etoc</code>?</b>	3, p. 8
Limitations in the use of list environments for tables of contents.	
<b>Line styles and toc display style</b>	4, p. 9
<code>\etocsetstyle</code> for the line styles – <code>\etocsettocstyle</code> for the toc display – Compatibility mode.	
<b>A first example</b>	5, p. 10
<b>A second example</b>	6, p. 12
<b>A Beautiful Thesis example</b>	7, p. 13
<b>Linked list of the main package commands</b>	8, p. 15

### Part II. Arbitrarily many TOCs, and local ones too

<b>Labeling and reusing elsewhere</b>	9, p. 16
<b>A powerful functionality of <code>etoc</code>: the re-assignment of levels with <code>\etocsetlevel</code></b>	10, p. 17
<b>The <code>\etocsettocdepth</code> and <code>\etocsetnexttocdepth</code> commands</b>	11, p. 18
The <code>hyperref</code> option <code>bookmarksdepth</code> .	

---

Documentation (2017/10/23) generated from the source file with timestamp "Time-stamp: <23-10-2017 at 12:03:59 CEST>".

<b>The command <code>\etocsettocdepth.toc</code></b> . . . . .	<b>12, p. 19</b>
The commands <code>\etocobeytoctocdepth</code> and <code>\etocignoretoctocdepth</code> .	
<b>The commands <code>\etocdepthtag.toc</code> and <code>\etocsettagdepth</code></b> . . . . .	<b>13, p. 20</b>
The commands <code>\etocobeydepthtags</code> and <code>\etocignoredepthtags</code> .	
<b>The commands <code>\etocglobaldefs</code> and <code>\etoclocaldefs</code></b> . . . . .	<b>14, p. 21</b>
<b>Not displayed empty TOCs</b> . . . . .	<b>15, p. 21</b>
The <code>\etoccheckemptiness</code> command – The <code>\etocnotocifnotoc</code> command – The <code>\etocifwasempty</code> command.	
<b>Adding commands to the .toc file</b> . . . . .	<b>16, p. 22</b>
The <code>hyperref</code> option <code>hidelinks</code> .	

### Part III. Examples

<b>Testing the compatibility mode</b> . . . . .	<b>17, p. 24</b>
<b>Another compatibility mode</b> . . . . .	<b>18, p. 25</b>
<b>Emulating the book class</b> . . . . .	<b>19, p. 27</b>
<b>A framed display</b> . . . . .	<b>20, p. 31</b>
<b>Another TOC with a background color</b> . . . . .	<b>21, p. 31</b>
<b>A (crazy) inline display</b> . . . . .	<b>22, p. 33</b>

### Part IV. Surprising uses of `etoc`

<b>The TOC of TOCs</b> . . . . .	<b>23, p. 36</b>
<b>Arbitrary “Lists Of...”, <code>\etoccontentsline</code></b> . . . . .	<b>24, p. 38</b>
<b>A TOC with a fancy layout</b> . . . . .	<b>25, p. 39</b>
<b>The TOC as a tree</b> . . . . .	<b>26, p. 40</b>
<b>The TOC as a molecule</b> . . . . .	<b>27, p. 43</b>
<b>The TOC as a TikZ mind map</b> . . . . .	<b>28, p. 45</b>
<b>The TOC as a table</b> . . . . .	<b>29, p. 49</b>
<b>A TOC self-adjusting widths for its typesetting</b> . . . . .	<b>30, p. 54</b>

### Part V. Commands for the toc line styles

<b>The <code>\etocsetstyle</code>, <code>\etocname</code> and <code>\etocpage</code> commands</b> . . . . .	<b>31, p. 57</b>
<b>The <code>\etocskipfirstprefix</code> and <code>\etociffirst</code> commands</b> . . . . .	<b>32, p. 59</b>

## CONTENTS

The <code>\etocnumber</code> command . . . . .	33, p. 59
The <code>\etocifnumbered</code> switch . . . . .	34, p. 59
The <code>\etocthenname</code> , <code>\etocthenumber</code> , and <code>\etocthepage</code> commands . . . . .	35, p. 60
The <code>\etoclink</code> command . . . . .	36, p. 60
The <code>\etocthelinkedname</code> , <code>\etocthelinkednumber</code> , <code>\etocthelinkedpage</code> and <code>\etoc-thelink</code> commands . . . . .	37, p. 60
The <code>\etocsavedsectiontocline</code> , etc... commands. . . . .	38, p. 61
The <code>\etocsetlevel</code> command . . . . .	39, p. 62
Scope of commands added to the <code>.toc</code> file . . . . .	40, p. 63
Testing the scope—This is a (pale) red subsection for illustrative purposes.	
Am I also red? . . . . .	41, p. 63

## Part VI. Commands for the toc display style

Specifying the toc display style . . . . .	42, p. 64
The command <code>\etocsettocstyle</code> —The commands <code>\etocmulticolstyle</code> , <code>\etocmulticol</code> , and <code>\etoc-localmulticol</code> —The command <code>\etocstocstyle</code> —The commands <code>\etocruledstyle</code> , <code>\etocruled</code> and <code>\etoclocalruled</code> —The commands <code>\etocframedstyle</code> , <code>\etocframed</code> , and <code>\etoclocalframed</code> —Headings, titles, <code>\etocoldpar</code> , <code>\etocinnertopsep</code> —The compatibility mode <code>\etocstandarddisplaystyle</code> —The command <code>\etocinline</code> .	
Starred variants of the <code>\tableofcontents</code> etc... commands . . . . .	43, p. 67

## Part VII. Using and customizing the `etoc` own styles

Summary of the main style commands . . . . .	44, p. 69
Setting up local styles—Setting up toc display styles—Displaying tables of contents—Labels and references.	
The package default line styles: <code>\etocdefaultlines</code> . . . . .	45, p. 70
Customizing <code>etoc</code> . . . . .	46, p. 75
Customizing the <code>etoc</code> pre-defined line styles—Customizing the toc display styles.	
One more example of colored TOC layout . . . . .	47, p. 77

## Part VIII. Tips

The <code>\etocsetlocaltop.toc</code> command . . . . .	48, p. 80
Hacking framed parboxes . . . . .	49, p. 81
Interverting the levels . . . . .	50, p. 82
All subsections of this document.	

Displaying statistics . . . . .	51, p. 83
Using depth tags . . . . .	52, p. 85
Typesetting the TOC as a table (the old way) . . . . .	53, p. 87

## Part IX. **etoc** and the outside world

<b>Compatibility with other packages</b> . . . . .	<b>54, p. 89</b>
Generalities—Compatibility with beamer—Compatibility with Babel—Compatibility with hyperref—Compatibility with multicol—Compatibility with tocloft—Compatibility with the memoir class—Compatibility with package tocvsec2—Compatibility with package tableof—Compatibility with package tocstyle.	
<b>T<sub>E</sub>Xnical matters</b> . . . . .	<b>55, p. 91</b>
<b>Errors and catastrophes</b> . . . . .	<b>56, p. 92</b>

## Part X. The code

<b>Timestamp</b> . . . . .	<b>57, p. 93</b>
<b>Change history</b> . . . . .	<b>58, p. 93</b>
<b>Implementation</b> . . . . .	<b>59, p. 96</b>

## 1. Foreword

### Abstract

The **etoc** package gives to the user complete control on how the entries of the table of contents should be constituted from the *name*, *number*, and *page number* of each sectioning unit. This goes via the definition of *line styles* for each sectioning level used in the document. The package provides its own custom line styles. Simpler ones are given as examples in the documentation. The simplest usage will be to take advantage of the layout facilities of packages dealing with list environments.

Regarding the *global toc display*, **etoc** provides pre-defined styles based on a multi-column format, with, optionally, a ruled title or framed contents.

The `\tableofcontents` command may be used arbitrarily many times and it has a variant `\localtableofcontents` which prints tables of contents ‘local’ to the current surrounding document unit. An extension of the `\label/\ref` syntax allows to reproduce (with another layout) a local table of contents defined somewhere else in the document.

Via “depth tags”, one gets an even finer control for each table of contents of which sectioning units it should, or not, display.

The formatting inherited (and possibly customized by other packages) from the document class will be used when in *compatibility mode*.

The assignment of levels to the sectioning units can be changed at any time, and **etoc** can thus be used in a quite general manner to create custom “lists of”, additionally to the tables of contents related to the document sectioning units. No auxiliary file is used additionally to the standard `.toc` file.

## 1. Foreword

Popular packages dealing with TOCs include `tocloft`, `titletoc` and `minitoc`. Why another one? I started **etoc** for my own use, and found out only later about these mentioned packages.

As is well explained in the `tocloft` package documentation, the standard L<sup>A</sup>T<sub>E</sub>X layout for the Table of Contents is buried in the class definitions. In particular, most of the lengths therein are hardcoded, and the only way for the user to change them is to recopy the class definitions into the document and then change them to obtain what is desired (within suitable `\makeatletter` and `\makeatother`). The more reasonable alternative is to use a dedicated package such as `tocloft` or to use another flexible document class.

However, although now things are hopefully not hard-coded, one still has to go through the package or class interface. This means one has to memorize a (possibly large) number of macros which will serve only to this task, and one will always be constrained to customizing one initially given layout.

The spirit of **etoc** is something else. The user will deal with the *name*, the *number* and the *page number* corresponding to each document sectional division (and found in a line of the `.toc` file) in a completely arbitrary manner: they are made available via the `\etocname`, `\etocnumber`, and `\etocpage` commands.

**etoc** appears to be (at least partially) compatible with the `article`, `book`, `report`, `scrartcl`, `scrbook`, `scrreprt` and `memoir` classes.

## 2. License

```
% Package: etoc
% Version: 1.081-2017/10/23
% License: LPPL 1.3c
% Copyright (C) 2012-2017 Jean-Francois Burnol <jfbu at free dot fr>
% Copyright (C) 2014-2017 Christine Roemer <Christine_Roemer at t-online dot de>
% and collaborators for the translation into German of the documentation
%
% This Work may be distributed and/or modified under the
% conditions of the LaTeX Project Public License, in its
% version 1.3c. This version of this license is in
%   http://www.latex-project.org/lppl/lppl-1-3c.txt
% and 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 2005/12/01 or later.
%
% The Authors of this Work are:
% Jean-Francois Burnol <jfbu at free dot fr>
%   for the source code and English documentation, and
% Christine Roemer <Christine_Roemer at t-online dot de> and collaborators
%   for the translation into German of the documentation.
%
% This Work consists of the main source file etoc.dtx and the derived
% files etoc.sty, etoc.ins, etoc.tex, etoc-DE.tex, etoc.pdf, etoc-DE.pdf,
% etoc.dvi, etoc-DE.dvi, README.md.
%
% Running etex (or latex or pdflatex) on etoc.dtx extracts etoc.sty,
% etoc.ins, etoc.tex, etoc-DE.tex, and README.md. See README.md for
% further instructions.
```

### 3. Do I need to be a geek to use **etoc**?

## Part I. Overview

Here are some statistics for this part: it contains 6 sections and 4 subsections. The name of the first section is “Do I need to be a geek to use **etoc**?” and the corresponding number is “3”. The name of the last section is “Linked list of the main package commands” and its number is “8”. The name of the first subsection is “Limitations in the use of list environments for tables of contents” and the corresponding number is “3.1”. The name of the last subsection is “Compatibility mode” and its number is “4.3”.

### 3. Do I need to be a geek to use **etoc**?

Not quite. The simplest way is to use `enumerate` environments, with the customizing facilities of packages such as `enumitem`,<sup>1</sup> to display the data fetched by **etoc** from the `.toc` file. The data consists of the *name* (`\etocname`), *number* (`\etocnumber`), and *page number* (`\etocpage`) as extracted from the `.toc` file.

This is illustrated [at the start of Part V](#). More sophisticated examples would use more sophisticated `enumitem` options. One may say then that again the user has to memorize some customizing! indeed, but the syntax and option names to memorize are in no way related only to matters of tables of contents, hence an economy of use of the poor brain.

Next in ease of use, perhaps, is the method explained [later in this part \(section 5\)](#). For this some knowledge of `\leftskip`, `\rightskip`, etc... is necessary. And a slight elaboration of this method, whose code is to be found in [section 52](#), allows to mimick very well, if so desired, the standard looks. An even closer emulation of the book class design is now included in this documentation as [section 19](#).

As will be amply illustrated in this manual, **etoc** is quite versatile (especially as it allows to re-define at any point in the document the hierarchy of sectioning units) and one can achieve surprising effects with it: [Part IV](#) is devoted to this, and some more is to be found in [Part VIII](#).

A very important aspect of **etoc** is that it is geared towards making many TOCs in the same document, *using only one .toc file*! The present documentation contains 43 visible tables of contents (and a few invisible ones) and uses only one `.toc` file!<sup>2</sup> So one should think twice before adding manually extra commands to the `.toc` file (see [section 16](#)).

#### 3.1. Limitations in the use of list environments for tables of contents

There are some limitations to the use of list environments for typesetting TOCs. One of them is intrinsic to the scope limitations created by the groups associated to the environ-

<sup>1</sup><http://ctan.org/pkg/enumitem>

<sup>2</sup>and the counting itself has been achieved by a table of contents which was inserted in the framed paragraph! ... the technique for this kind of effect will be explained later.



ments: the .toc file may contain, besides the information to be typeset in the TOCs, some other commands, such as language changing commands, which do not expect to see their scope limited in this way inside a group (L<sup>A</sup>T<sub>E</sub>X’s environments create groups).

Therefore the built-in “line styles” proposed by **etoc** as an example (and which are illustrated<sup>3</sup> by the [main table of contents](#) in this document) do not make use of environments. Actually, in this user manual, only the [table of contents](#) at the start of [Part V](#), the [subsection 50.1](#) (which is a TOC!) and examples from [section 37](#) have their line styles expressed in terms of enumerate or itemize environments.

## 4. Line styles and toc display style

A distinction shall be made between the *line styles*, *i.e.* the way the name, number and page numbers are used at each level, and the *toc display style* (for lack of a better name) which tells how the title should be set, whether an entry in the .toc file should be made, whether the contents should be typeset with multiple columns, etc... the latter is governed by the command `\etocsettocstyle` (or some higher-level commands) and the former by the command `\etocsetstyle`.

### 4.1. `\etocsetstyle` for the line styles

The command to inform **etoc** of what to do with `\etocname`, `\etocnumber`, and `\etocpage` is called `\etocsetstyle`. It has five mandatory arguments. The first one is the name of the sectional unit: a priori known names are `book`, `part`, `chapter`, `section`, `subsection`, `subsubsection`, `paragraph`, and `subparagraph`, and any other name can be declared and assigned to a (numeric) level via the `\etocsetlevel` command.<sup>4</sup>

The four other arguments of `\etocsetstyle` specify: 1) *what to do when this level is first encountered, down from a more general one*, then 2) & 3) (two arguments, a ‘prefix’ and a ‘contents’) *what to do when a new entry of that type is found*, and 4) *the last argument is the code to execute when a division unit of higher importance is again hit upon*.

### 4.2. `\etocsettocstyle` for the toc display

The `\etocsettocstyle` command allows to specify what should be done before and after the line entries of the TOC are typeset, and in particular how the title should be printed. It has two arguments, the first one is executed before the TOC contents (typically it will print “Contents” and define suitable marks for the page headings) and the second is executed after the TOC contents.

**etoc** provides four (customizable) higher level toc styles: `\etocmulticolstyle`, `\etoctocstyle`, `\etocruledstyle`, and `\etocframedstyle`. All use the `multicol` package with a default of two columns (single-column mode is obtained with the optional argument [1]).

These commands must be followed either by `\tableofcontents` or `\localtableofcontents`.

---

<sup>3</sup>with a twist, subsections having been downgraded to the subsubsection style...

<sup>4</sup>under the memoir class, **etoc** knows appendix as a sectioning name.

## 5. A first example

### 4.3. Compatibility mode

Both for the “line styles” and the “toc display style”, it is possible to switch into a compatibility mode which uses the defaults from the document class.<sup>5</sup> This is activated by:

```
\etocstandardlines          % ‘line entries’ as without \usepackage{etoc}  
\etocstandarddisplaystyle  % ‘toc display’ as without \usepackage{etoc}
```

If the command `\etocsetstyle` has *not been used in the preamble* the package will be at `\begin{document}` in this compatibility mode: hence just adding `\usepackage{etoc}` should hopefully not change anything to the look of a previously existing document, under the article, book, report, scrartcl, scrbook, scrreprt and memoir classes.

Any use of `\etocsetstyle` in the *preamble or body* of the document turns off from that point on the compatibility mode for line styles, but maintains the compatibility mode for the TOC title. One re-activates the compatibility mode for line styles with `\etocstandardlines`; and `\etococlines` will re-activate the line styles as defined with the help of `\etocsetstyle`, if their scope was not limited to a group or environment.

The command `\etocdefaultlines` resets the line styles as pre-defined internally by **etoc** and described in [section 45](#).

Even if `\etocsetstyle` has been used, the global display style remains initially as defined by the document class (or the `tocloft` package); one needs to use the command `\etocsettocstyle` or its variants to exit from this compatibility mode at the “toc display style” level. It will be re-activated if use if made of `\etocstandarddisplaystyle`.

See further [section 18](#).

## 5. A first example

Let us present a first example of specification for line styles. Immediately after the start of [Part I](#) we inserted in the source file:

```
\invisiblelocaltableofcontents \label{toc:overview}
```

This sets-up the label `toc:overview`, and we can use it at any location in the document:

```
\tableofcontents \ref{toc:overview}
```

And as we used `\invisible...`,<sup>6</sup> the local TOC will exist only through its clones elsewhere in the document.

---

### My first **etoc**: TOC of [Part I \(Overview\)](#)

---

<b>3. Do I need to be a geek to use <b>etoc</b>?</b>	8
3.1 Limitations in the use of list environments for tables of contents	8
<b>4. Line styles and toc display style</b>	9
4.1 <code>\etocsetstyle</code> for the line styles	9
4.2 <code>\etocsettocstyle</code> for the toc display	9
4.3 Compatibility mode	10
<b>5. A first example</b>	10
<b>6. A second example</b>	12

---

<sup>5</sup>for the “toc display style” **etoc** checks if it knows the class, and if not defaults to the article class layout. It will also check if `tocloft` has customized the TOC title.

<sup>6</sup>this is a shortcut for setting temporarily the `tocdepth` to `-3`, which has the effect to tell **etoc** not to print the TOC, and not even the heading.

<b>7. A Beautiful Thesis example</b>	13
<b>8. Linked list of the main package commands</b>	15

We could have used the line styles defined by `etoc`, with `\etocdefaultlines`, or the default document class styles with `\etocstandardlines`, but we were a bit more ambitious here and wanted to design our own. The technique is a simple one: each heading is in its own paragraph, which may extend on multiple lines; it is responsible for setting its own `\leftskip`.

This is a simple design which does not make provisions for page breaks which should be discouraged in-between a section and a subsection etc. . . as we only used it for the table of contents of this part, thus with sections as top levels, there was no need to specify a style for parts too (we defined a subsubsection line style but as it turns out there are no subsubsections in this part). The two commands used are `\etocsetstyle` for specifying the line styles, and `\etocruledstyle` for the TOC global style.

The `\rightskip` is shared by all, and creates space where the page numbers get printed. For an elaboration of this technique see the next [section 6](#) as well as [section 52](#) which provides a TOC with parts and paragraphs. Both allow multi-line headings and employ a technique for putting page numbers in the right margin which was inspired from what L<sup>A</sup>T<sub>E</sub>X2e's `\@dottedtocline` macro does.

Here is how it was produced:

```
\begingroup\parindent 0pt \parfillskip 0pt \leftskip 0cm \rightskip 1cm
\etocsetstyle {section}
  {}
  {\leavevmode\leftskip 0cm\relax}
  {\bfseries\normalsize\makebox[.5cm][l]{\etocnumber.}%
   \etocname\nobreak\hfill\nobreak
   \rlap{\makebox[1cm]{\mdseries\etocpage}}\par}
  {}
\etocsetstyle {subsection}
  {}
  {\leavevmode\leftskip .5cm\relax }
  {\mdseries\normalsize\makebox[1cm][l]{\etocnumber}%
   \etocname\nobreak\hfill\nobreak
   \rlap{\makebox[1cm]{\etocpage}}\par}
  {}
\etocsetstyle {subsubsection}
  {}
  {\leavevmode\leftskip 1.5cm\relax }
  {\mdseries\normalsize\makebox[1cm][l]{\etocnumber}%
   \etocname\nobreak\hfill\nobreak
   \rlap{\makebox[1cm]{\etocpage}}\par}
  {}
\etocruledstyle[1]{\bfseries \Large My first \etoc: TOC of
  \autoref{part:overview} (\nameref{part:overview})}
\tableofcontents \ref {toc:overview}
\endgroup
```

## 6. A second example

This second example displays only the contents from [Part V](#) and [Part VI](#). This selection is done via the technique of *depth tags*, described in [section 13](#) and [section 52](#). Its layout is a bit like the one of the [main document TOC](#), although the line styles are coded very differently.

### CONTENTS

#### PART V – COMMANDS FOR THE TOC LINE STYLES

31. THE <code>\ETOCSETSTYLE</code> , <code>\ETOCNAME</code> AND <code>\ETOCPAGE</code> COMMANDS .....	57
32. THE <code>\ETOCSKIPFIRSTPREFIX</code> AND <code>\ETOCIFFIRST</code> COMMANDS .....	59
33. THE <code>\ETOCNUMBER</code> COMMAND.....	59
34. THE <code>\ETOCIFNUMBERED</code> SWITCH .....	59
35. THE <code>\ETOCTHENNAME</code> , <code>\ETOCTHENNUMBER</code> , AND <code>\ETOCTHEPAGE</code> COMMANDS .....	60
36. THE <code>\ETOCLINK</code> COMMAND.....	60
37. THE <code>\ETOCHELINKEDNAME</code> , <code>\ETOCHELINKEDNUMBER</code> , <code>\ETOCHELINKEDPAGE</code> AND <code>\ETOCHELINK</code> COMMANDS .....	60
38. THE <code>\ETOCsavedsectiontocline</code> , ETC... COMMANDS.....	61
39. THE <code>\ETOCSETLEVEL</code> COMMAND.....	62
40. SCOPE OF COMMANDS ADDED TO THE <code>.TOC</code> FILE .....	63
<i>Testing the scope (40.1, p. 63). This is a (pale) red subsection for illustrative purposes (40.2, p. 63).</i>	
41. AM I ALSO RED? .....	63

#### PART VI – COMMANDS FOR THE TOC DISPLAY STYLE

42. SPECIFYING THE TOC DISPLAY STYLE .....	64
<i>The command <code>\etocsettocstyle</code> (42.1, p. 64). The commands <code>\etocmulticolstyle</code>, <code>\etocmulticol</code>, and <code>\etoclocalmulticol</code> (42.2, p. 65). The command <code>\etocstyle</code> (42.3, p. 65). The commands <code>\etocruledstyle</code>, <code>\etocruled</code> and <code>\etoclocalruled</code> (42.4, p. 66). The commands <code>\etocframedstyle</code>, <code>\etocframed</code>, and <code>\etoclocalframed</code> (42.5, p. 66). Headings, titles, <code>\etocoldpar</code>, <code>\etocinnertopsep</code> (42.6, p. 66). The compatibility mode <code>\etocstandarddisplaystyle</code> (42.7, p. 67). The command <code>\etocinline</code> (42.8, p. 67).</i>	
43. STARRED VARIANTS OF THE <code>\TABLEOFCONTENTS</code> ETC... COMMANDS .....	67
The code:	

```

\begingroup
\newcommand*{\DotsAndPage}
{\nobreak\leaders\hbox{\bfseries\normalsize\hbox to .75ex {\hss.\hss}}%
\hfill\nobreak
\makebox[\rightskip][r]{\bfseries\normalsize\etocpage}\par}

\etocsetstyle {part}
{\parindent 0pt
\nobreak
\etocskipfirstprefix}
{\pagebreak[3]\bigskip}

```

```

{\large\rmfamily\bfseries\scshape\centering
\etocifnumbered{Part \etocnumber{} - }{}\etocname\par}
{}}

\etocsetstyle {section}
{\leftskip 0pt \rightskip .75cm \parfillskip-\rightskip
\nobreak\medskip
\etocskipfirstprefix}
{\leftskip 0pt \rightskip .75cm \parfillskip-\rightskip
\pagebreak[1]\smallskip}
{\normalsize\rmfamily\bfseries\scshape
\etocnumber. \etocname\DotsAndPage }
{\parfillskip 0pt plus 1fil\relax }

\etocsetstyle {subsection}
{\leftskip1cm\rightskip .75cm \parfillskip 0pt plus 1fil\relax
\nobreak\smallskip}
{}}
{\footnotesize\sffamily\mdseries\itshape
\etocname{} (\etocnumber, p. \etocpage). }
{\par\medskip}

\etocsettagdepth {preamble} {none}
%\etocsettagdepth {overview} {none}% not needed explicitly, keeps value
%\etocsettagdepth {arbitrarily}{none}
%\etocsettagdepth {examples} {none}
%\etocsettagdepth {surprising} {none}
\etocsettagdepth {linestyles} {subsection}
\etocsettagdepth {globalcmds} {subsection}
\etocsettagdepth {custom} {none}
%\etocsettagdepth {tips} {none}
%\etocsettagdepth {etocandworld}{none}
%\etocsettagdepth {code} {none}

\etocsettocstyle {\centering\LARGE\textsc{\contentsname}\par\nobreak\medskip}{}
\etocsetnexttocdepth {subsection}
\tableofcontents
\endgroup

```

## 7. A Beautiful Thesis example

Here is a relatively simple example of use of the package functionalities. Let us set up some line styles. We choose a style for sections and sub-sections which would be suitable for, respectively, sections and sub-sections in an average length memoir. The line style specifications have some redundancy for clarity, and do not care about what to do at possible page breaks. Also, they do not worry about potential multi-column use.

\_\_\_\_\_ *My Beautiful Thesis*

## Chapter 3 Do I need to be a geek to use

	<b>etoc?</b>	<b>8</b>
3.1	<i>Limitations in the use of list environments for tables of contents</i>	8
<b>Chapter 4</b>	<b>Line styles and toc display style</b>	<b>9</b>
4.1	<i>\etocsetstyle for the line styles</i>	9
4.2	<i>\etocsettocstyle for the toc display</i>	9
4.3	<i>Compatibility mode</i>	10
<b>Chapter 5</b>	<b>A first example</b>	<b>10</b>
<b>Chapter 6</b>	<b>A second example</b>	<b>12</b>
<b>Chapter 7</b>	<b>A Beautiful Thesis example</b>	<b>13</b>
<b>Chapter 8</b>	<b>Linked list of the main package commands</b>	<b>15</b>

```

\beginngroup % we start a group to keep the style changes local
\newlength{\tocleftmargin} \setlength{\tocleftmargin}{5cm}
\newlength{\tocrightmargin} \setlength{\tocrightmargin}{1cm}

\etocsetstyle{section} % will pretend to be a Chapter
{\addvspace{1ex}\parfillskip0pt
 \leftskip\tocleftmargin % (already done in title)
 \rightskip\the\tocrightmargin plus 1fil
 \parindent0pt\color{cyan}} % (already done)
{\bfseries\LARGE\upshape\addvspace{1ex}\leavevmode}
{\llap{Chapter\hspace{.5em}}{\etocnumber}\hspace{.75cm}}{\etocname}
\hfill\makebox[-\tocrightmargin][l]{\makebox[0pt]{\etocpage}}\par}
{}

\etocsetstyle{subsection} % will pretend to be a Section
{}
{\mdseries\large\addvspace{.5ex}\leavevmode}
{\llap{\etocnumber\hspace{.75cm}}{\textit{\etocname}}%
 \hfill\makebox[-\tocrightmargin][l]{\makebox[0pt]{\etocpage}}\par}
{}

\def\tmptitle{My Beautiful Thesis}
\etocsettocstyle{\color{cyan}\parindent0pt \leftskip\tocleftmargin
 \leavevmode\leaders\hrule height 1pt\hfill\
 \huge\textit{\tmptitle}\par}{\bigskip}

\tableofcontents \ref{toc:overview}
\endngroup

```

## 8. Linked list of the main package commands

<code>\etocaftercontentshook</code>	<code>\etoclocalmulticol</code>	<code>\etocsettocdepth</code>
<code>\etocaftertitlehook</code>	<code>\etoclocalruled</code>	<code>\etocsettocstyle</code>
<code>\etocaftertochhook</code>	<code>\etocmulticolstyle</code>	<code>\etocskipfirstprefix</code>
<code>\etocbeforetitlehook</code>	<code>\etocmulticol</code>	<code>\etocstandarddisplaystyle</code>
<code>\etocdefaultlines</code>	<code>\etocname</code>	<code>\etocstandardlines</code>
<code>\etocdepthtag.toc</code>	<code>\etocnumber</code>	<code>\etocthelinkedname</code>
<code>\etocdisplay</code>	<code>\etocobeydepthtags</code>	<code>\etocthelinkednumber</code>
<code>\etocframedstyle</code>	<code>\etocobeytoctocdepth</code>	<code>\etocthelinkedpage</code>
<code>\etocframed</code>	<code>\etocpage</code>	<code>\etocthelink</code>
<code>\etociffirst</code>	<code>\etocruledstyle</code>	<code>\etocthename</code>
<code>\etocifnumbered</code>	<code>\etocruled</code>	<code>\etocthenumber</code>
<code>\etocignoredepthtags</code>	<code>\etocsetlevel</code>	<code>\etocthepage</code>
<code>\etocignoretoctocdepth</code>	<code>\etocsetnexttocdepth</code>	<code>\etoc toc contentsline</code>
<code>\etocinline</code>	<code>\etocsetstyle</code>	<code>\etoc toclines</code>
<code>\etoclink</code>	<code>\etocsettagdepth</code>	<code>\localtableofcontents</code>
<code>\etoclocalframed</code>	<code>\etocsettocdepth.toc</code>	<code>\tableofcontents</code>

## Part II.

# Arbitrarily many TOCs, and local ones too

Here are some statistics for this part: it contains 8 sections and 7 subsections. The name of the first section is “Labeling and reusing elsewhere” and the corresponding number is “9”. The name of the last section is “Adding commands to the .toc file” and its number is “16”. The name of the first subsection is “The hyperref option *bookmarksdepth*” and the corresponding number is “11.1”. The name of the last subsection is “The hyperref option *hidelinks*” and its number is “16.1”.

<b>9. Labeling and reusing elsewhere</b>	<b>16</b>
<b>10. A powerful functionality of <i>etoc</i>: the re-assignment of levels with <code>\etocsetlevel</code></b>	<b>17</b>
<b>11. The <code>\etocsettocdepth</code> and <code>\etocsetnexttocdepth</code> commands</b>	<b>18</b>
11.1. The hyperref option <i>bookmarksdepth</i> . . . . .	19
<b>12. The command <code>\etocsettocdepth.toc</code></b>	<b>19</b>
12.1. The commands <code>\etocobeytoctocdepth</code> and <code>\etocignoretoctocdepth</code> .	20
<b>13. The commands <code>\etocdepthtag.toc</code> and <code>\etocsettagdepth</code></b>	<b>20</b>
13.1. The commands <code>\etocobeydepthtags</code> and <code>\etocignoredepthtags</code> . . . .	21
<b>14. The commands <code>\etocglobaldefs</code> and <code>\etoclocaldefs</code></b>	<b>21</b>
<b>15. Not displayed empty TOCs</b>	<b>21</b>
15.1. The <code>\etocchecksemtiness</code> command . . . . .	21
15.2. The <code>\etocnotocifnotoc</code> command . . . . .	22
15.3. The <code>\etocifwasempty</code> command . . . . .	22
<b>16. Adding commands to the .toc file</b>	<b>22</b>
16.1. The hyperref option <i>hidelinks</i> . . . . .	23

## 9. Labeling and reusing elsewhere

*etoc* allows arbitrarily many `\tableofcontents` commands in your document. The line styles and the toc display style may of course be changed in-between. Furthermore `\localtableofcontents` will print local tables of contents: *i.e.* all sections and sub-units inside a given chapter, or all subsections and lower inside a given subsection, etc . . .

*etoc* allows the labeling of a TOC with (for example) `\label{toc:A}` and will re-display it elsewhere when told `\tableofcontents\ref{toc:A}`. The actual layout (title inclusive) used for the cloned TOC will be decided locally. The line styles and toc display



style (including the title) will be the current ones and the current value of the `tocdepth` counter is obeyed. As an example the table of contents of [Part VII](#) is in a `float` which appears on this page.

<b>I am from far away</b>	
<b>44. Summary of the main style commands</b>	<b>69</b>
44.1. Setting up local styles . . . . .	69
44.2. Setting up toc display styles . . . . .	70
44.3. Displaying tables of contents . . . . .	70
44.4. Labels and references . . . . .	70
<b>45. The package default line styles: <code>\etocdefaultlines</code></b>	<b>70</b>
<b>46. Customizing <code>etoc</code></b>	<b>75</b>
46.1. Customizing the <code>etoc</code> pre-defined line styles . . . . .	75
46.2. Customizing the toc display styles . . . . .	76
<b>47. One more example of colored TOC layout</b>	<b>77</b>

We used this:

```
\begin{figure}[ht!]
\centering
\begingroup
\etocstandardlines
\renewcommand{\etocbkgcolorcmd}{\color{green!5}}
\renewcommand{\etocbelowtocskip}{0pt\relax}
\fbboxseplex
\etocframedstyle [1]{\fbbox{\makebox[.5\linewidth]{\etocfontminusone
\hyperref[toc:c]{I am from far away}}}}
\tableofcontents \label{toc:d} \ref{toc:c}
\endgroup
\end{figure}
```

Important: one should not use elsewhere `\tableofcontents \ref{toc:d}`. To clone again, one must use a reference to the original label: `\tableofcontents \ref{toc:c}`.

## 10. A powerful functionality of `etoc`: the re-assignment of levels with `\etocsetlevel`

The intrinsic levels manipulated by `etoc` are numeric: from -2 (which corresponds to book in the memoir class) down (from the big to the small) to 5 (subparagraph). But the assignment of a numeric level to a given name can be modified at any time with the command `\etocsetlevel{\<level_name>}{\<number>}`. In conjunction with the use of the L<sup>A</sup>T<sub>E</sub>X `tocdepth` counter, this has powerful applications: `\<level_name>` does not have to coincide with an actual document sectioning command, and `etoc` can be used to print

## 11. The `\etocsettocdepth` and `\etocsetnexttocdepth` commands

arbitrary “lists of things”, using no other auxiliary file than the `.toc` file. This is explained further in [Part IV](#).

It is often said that in the standard classes, the sectioning level of `\part` is 0 in the classes not having a `\chapter` command, and -1 in classes having a `\chapter` command. This is *correct* for what regards the *automatic numbering*, as is governed by the value of the `secnumdepth` counter; but it is *wrong* for what regards the effect of the `tocdepth` counter: setting the `tocdepth` to -1 in the `article` class just before `\tableofcontents` does *not* prevent Parts from appearing in the Table of Contents. One has to set it to -2 for that, whether in the `article` or in the `book` class.

The canonical levels, a priori known to **etoc**, are those of relevance to the **tocdepth** counter in the standard classes and are recapitulated in this table:

(memoir class) book	-2
part	-1
chapter	0
section	1
subsection	2
subsubsection	3
paragraph	4
subparagraph	5

With **etoc**, the user can easily print a local table of contents inside a given subsection, where subsubsections will be printed in the style of sections, paragraphs in the style of subsections, and subparagraphs in the style of subsubsections, if so desired. One can also decide to set everything to be at the level 6 (never displayed by **etoc**), except for example paragraphs, promoted to be at level 1, and then one obtains a nice table of contents of all the paragraphs from the document! (`tocdepth` at least 1)<sup>7</sup>

## 11. The `\etocsettocdepth` and `\etocsetnexttocdepth` commands

The `tocdepth` counter has no bearing on what gets written to the `.toc` file; its action is only on the actual typesetting of the table of contents.<sup>8</sup> In the standard classes there is only one `\tableofcontents` possible, whereas with **etoc**, arbitrarily many are allowed, so one may change `tocdepth` to the appropriate value (which decides the finest sectioning level displayed) again and again each time a table of contents needs to be typeset.

**etoc** provides `\etocsettocdepth{<level>}` whose mandatory argument is either numeric (from -3 to 5) or a division name such as `subsection` or `subsubsection` or any name previously declared to **etoc** with `\etocsetlevel` (the keywords `all` and `none` are recognized, although not corresponding to a document division). This does the appropriate `\setcounter{tocdepth}{numeric_level}`.

<sup>7</sup>and one should naturally not print this TOC of paragraphs in compatibility mode, which would insist on inserting a gigantic left margin.

<sup>8</sup>In the standard classes (at least), it also influences the `\listoftables` and `\listoffigures`, via `\@dottedtocline`.

As is explained in the next subsection, `tocdepth` is used by `hyperref`, and one must take steps to prevent its changes from influencing the bookmarks, too. So, `etoc` has `\etocsetnexttocdepth{<level>}` whose influence ceases immediately after the next table of contents. Thus, `\invisibletableofcontents` is essentially `\etocsetnexttocdepth{none}\tableofcontents`.

The simplest course is thus to have after `\begin{document}` and before the first `\tableofcontents` a single instance of the `\etocsettocdepth` command, with argument the deepest level (or most commonly used deepest level) among the tables of contents of the document, and to use locally, where needed, `\etocsetnexttocdepth` before `\tableofcontents` or `\localtableofcontents`.

`\etocsetnexttocdepth` used to modify the `tocdepth` counter immediately. In case two or more such commands were issued in a row with no TOC being typeset, this could break the correct restoring of the `tocdepth` counter after the TOC. Starting with 1.08h, the `tocdepth` counter is modified only at the time the TOC is typeset, not earlier, thus fixing that issue.<sup>9</sup>

### 11.1. The `hyperref` option `bookmarksdepth`

When modifying the counter `tocdepth` for the purposes of multiple uses of `\tableofcontents` or `\localtableofcontents`, one should be aware that package `hyperref` by default takes into account the *current* value of the `tocdepth` counter to decide whether the pdf file will contain a bookmark corresponding to sectioning commands encountered in the source file. Thus, one typically needs to reset `tocdepth` to its previous value after having temporarily modified it for a given table of contents.

Or, there is the `bookmarksdepth=n` option of package `hyperref`, with  $n$  the desired document bookmarks maximal depth, which can be numeric or the name of a level known to `hyperref`. This documentation previously passed `bookmarksdepth=3` as option to `hyperref`, so even if `tocdepth` was left to 1 by inadvertance after printing a certain table of contents this did not modify the bookmark tree of the pdf file. Now that `\etocsetnexttocdepth` has been added to the package, we have used it systematically and there was no need for `bookmarksdepth=3` anymore.

## 12. The command `\etocsettocdepth.toc`

This command `\etocsettocdepth.toc` implements some functionality of `tocvsec2`<sup>10</sup>, a package which however was incompatible with `etoc` (it can still be used for its `secnumdepth`-related commands, but its `toc`-related activities will get canceled by `etoc`) and more-or-less designed for a single table of contents.

The action of `\etocsettocdepth.toc` is totally different than the one of `\etocsettocdepth`. Rather than modifying the `tocdepth` counter immediately, it adds a line to the `.toc` file which, when executed inside a table of contents will enact this change.

The command `\etocsettocdepth.toc`, like `\etocsettocdepth`, accepts both numeric and named arguments. In the case of a named argument, the actual numeric value to

<sup>9</sup>Thanks to Denis Brouzé for signaling the problem.

<sup>10</sup><http://ctan.org/pkg/tocvsec2>. I thank Denis Brouzé for drawing my attention to the incompatibility of this package with `etoc`.

### 13. The commands `\etocdepthtag.toc` and `\etocsettagdepth`

be used is not yet decided at the time the `.toc` file is created; it will be the value currently specified for the named level at the time each table of contents (not having done `\etocignoretoctocdepth`) is typeset.

The `tocdepth` counter will never be set to a value finer than its initial value at the start of the table of contents: so adding commands `\etocsettocdepth.toc` in the document is a way to *restrict* locally the depth of the table of contents. For example to prevent inclusion in the tables of contents of the sub-sub-sections of a given chapter.

This gets executed in ALL tables of contents.

#### 12.1. The commands `\etocobeytoctocdepth` and `\etocignoretoctocdepth`

So `\etocignoretoctocdepth` is provided to cancel the `\etocsettocdepth.toc` mechanism when needed; and `\etocobeytoctocdepth` will re-activate it. The package does initially `\etocobeytoctocdepth`.

### 13. The commands `\etocdepthtag.toc` and `\etocsettagdepth`

As mentioned above, when the argument to `\etocsettocdepth.toc` isn't numeric but a named level the actual numeric level used is decided at the time the TOC gets typeset, thus with the *shuffling abilities of etoc*, one may obtain a very flexible control in the following way: one first declares with `\etocsetlevel` some dummy named level, say `partIII`, assigning it the invisible numeric level 6. Then before the actual third `\part` command in the source one inserts `\etocsettocdepth.toc{partIII}`. At the location where the effect should cease one inserts in the source `\etocsettocdepth.toc{all}`. It is now possible to specify, each time one wants to typeset a TOC, what will be the finest level for the entries originating in the third part: one just has to do `\etocsetlevel{partIII}{1}` for example. This will limit to sections (if `tocdepth` was at least at that level at the time `\tableofcontents` is encountered).

This method has some limitations: (i) the second argument to `\etocsetlevel` must be numeric, (ii) it can not be set to `-3` (which would be necessary if one wanted to exclude an entire Book from a TOC in a memoir document with multiple such Books ... admittedly a possibly rare case!), (iii) it is a bit of a hack as `partIII` is not a real division unit, but just a *tag*.

Release 1.07h has a command `\etocdepthtag.toc` which makes it easier to achieve this variable local control of the `tocdepth` at the time of typesetting TOCs.

It is used as `\etocdepthtag.toc{<tag_name>}`, where the `<tag_name>` is anything, and this will put the tag in the `.toc` file. When typesetting a TOC, one issues a series of commands `\etocsettagdepth{<tag_name>}{<level>}` where the `<level>` may be either numeric (from `-3` to `5`) or the name of a division unit known to *etoc*, or `none` or `all`. The effect of the tag inside the `.toc` file will then be to set the `tocdepth` counter to the desired value, in real time (as mentioned already, this can not get finer than the initial value of `tocdepth` at the start of the TOC).

The added flexibility is thus that `\etocsetlevel` has not been used in a kind of hacky way, that one may use named level depths, and the keywords `none` and `all`.

### 13.1. The commands `\etocobeydepthtags` and `\etocignoredepthtags`

As usual, once the tag depths have been set, they remain in effect until getting redefined or seeing their scope expire via the closing of a group or of a surrounding environment. For an example, see [section 52](#).

### 13.1. The commands `\etocobeydepthtags` and `\etocignoredepthtags`

After `\etocignoredepthtags`, the `.toc` depth tags are ignored (but `\etocdepthtag.toc` still works). The package does initially `\etocobeydepthtags` which makes `etoc` react to the found tags in the `.toc` file.

## 14. The commands `\etocglobaldefs` and `\etoclocaldefs`

In  $\text{\LaTeX}$  the meaning of a command defined via `\newcommand\foo{...}` inside an environment (or group) vanishes from  $\text{\TeX}$ 's memory on exit from this environment (or group). At times however it is needed to make definitions with global scope, for this  $\text{\TeX}$  has the primitive prefix `\global`.

By default `etoc`'s definitions of `\etocname` etc... are local. This causes problems in certain contexts such as TOC as tables ([section 29](#), [section 53](#)) and also with `enumitem` *inline* variants of its standard environments, because the command `\item` then closes a group (see [section 37](#)).

After `\etocglobaldefs`, `etoc` will make its definitions of `\etocname` etc... have global scope. For normal use this is not necessary. It does not hurt either to activate it systematically.

To return to the default, use `\etoclocaldefs`. Note that both `\etocglobaldefs` and `\etoclocaldefs` actions are local to the environment or group where they are used.

## 15. Not displayed empty TOCs

### 15.1. The `\etoccheckemptiness` command

The user needs to issue `\etoccheckemptiness` to tell `etoc` to check whether local tables of contents are empty and in case of emptiness to print nothing at all.<sup>11</sup> This can be useful to authors of  $\text{\LaTeX}$  classes who for example wish to have a `\chapter` command doing systematically a `\localtableofcontents`, or for people producing files via automatic conversions and some of those might have sectioning commands and others not.

«Emptiness» means that no `\contentsline` command would get executed within the scope of the local table of contents — empty line styles by themselves do not make the TOC empty. `etoc` always executes the `\etocaftertoohook` command; and the test for emptiness itself executes everything else found in the `.toc` file. See [section 16](#) in this context.

1. the `\etocifwasempty` command discussed below can be used from inside `\etoc-aftertoohook`, and even from inside `\etocbeforetitlehook`.
2. there is also `\etocdoesnotcheckemptiness` (since 1.08i.)

---

<sup>11</sup>Thanks to Paul Gaborit who asked for such a feature.

## 16. Adding commands to the .toc file

The suppression of the heading (more precisely of the toc display style elements) may be effective only for the final  $\LaTeX$  runs. For example in the situation of a `\tableofcontents\ref{foo}` where the label `foo` is not yet recognized, the heading (but not the contents) is printed and the TOC is declared non-empty. Or, if one adds a `\localtableofcontents` to a document, on the next run, the test for emptiness will in fact apply to the next one, and the last local TOC of the document will have its contents temporarily unknown to **etoc**, hence will be declared non empty, and the heading will be printed.

For a finalized document compiled with initially no auxiliary files, the first  $\LaTeX$  run will declare all local TOCs non empty and print for each of them a heading (and no contents naturally). The second  $\LaTeX$  run will then correctly decide which local TOC is empty or not.

### 15.2. The `\etocnotocifnotoc` command

The user can then extend the emptiness-checking to the global TOCs with `\etocnotocifnotoc`. May I respectfully give the advice then to rather do none of `\usepackage{etoc}` nor `\tableofcontents?` ; -). Well, there is always the case of batch conversions of documents having or not sectioning units.

### 15.3. The `\etocifwasempty` command

The command `\etocifwasempty{<YES>}{<NO>}` executes `<YES>` if the previous TOC was found to be empty and `<NO>` if its was not so. This may serve to act appropriately after a truly empty TOC. If `\etocchecksempiness` has not been issued, this conditional always executes the `<NO>` branch.

This command is robust, and `\etocxifwasempty` is its expandable version.

Do not forget the second argument: at least an empty pair of braces must be present.

This conditional may wrongly say that the local TOC is empty or not empty until  $\LaTeX$  compilations stabilize. But if it says that a local TOC is empty, this does mean that **etoc** considered the just encountered local table of contents to be empty (for that run) and thus printed nothing (not even a `\par`).

## 16. Adding commands to the .toc file

We described above `\etocsettocdepth.toc` and `\etocdepthtag.toc` which both insert commands inside the .toc file. An even more general mechanism of adding “action tags” to the .toc file could be envisioned, but this would just be a wrapper for direct use of `\addtocontents{toc}{\something}`.

One should be cautious when adding in this way things to the .toc file. For example, inserting `\addtocontents{toc}{\string\clearpage}` just before a `\part` to fix the problem when some part entry (in the table of contents) is isolated at the bottom of one page, will cause problems with multiple TOCs: this `\clearpage` will be executed by **etoc** each time a `\tableofcontents` or `\localtableofcontents` command is encountered! The more prudent thing is to do rather: `\addtocontents{toc}{\string\myclearpage}`, to have a `\let\myclearpage\relax` at the top level of the document and to use where needed something like:

```
\let\myclearpage\clearpage
\tableofcontents
\let\myclearpage\relax
```

The memoir class has the command `\settocdepth` which writes a `\changetocdepth` command inside the `.toc` file. This will impact the typesetting by **etoc** of *all* tables of contents, with (possibly) unexpected results: imagine the document has `\settocdepth{chapter}` at some point to avoid having the sections from subsequent chapters be listed in the main table of contents. Then a local table of contents in one of these chapters will print a title but will be without any entry.

As the memoir class by itself allows multiple `\tableofcontents` these issues already arise there, independently of **etoc**, see page 170 of the memoir manual.

For this specific issue, the commands `\etocsettocdepth.toc`, `\etocignoretoc-tocdepth` and `\etocobeytoctocdepth` are the way to go; or their variants `\etocdepth-tag.toc` and `\etocsettagdepth`.

As an aside, any `\setcounter{tocdepth}{n}` command added directly to the `.toc` file sees its effect (since release 1.07g) canceled at the end of each table of contents, which automatically does a `\setcounter{tocdepth}{previous_value}` with the value active on entering the table of contents.

## 16.1. The hyperref option *hidelinks*

The colored links (and also the rectangle links) are a bit annoying when used in tables of contents, especially when the document uses **etoc** and has plenty of them! One may wish for having colored links, *except* for those within table of contents! Indeed, why would things in TOCs need to be either framed in rectangles or colored, when the user *already expects them to be links*?

I use the following trick: either in the preamble using `\AtBeginDocument`, or right after `\begin{document}`, I have the command

```
\addtocontents{toc}{\protect\hypersetup{hidelinks}}
```

All TOCs typeset by **etoc** have their contents done within a group (as if enclosed in an environment). So the command `\hypersetup{hidelinks}` will be executed by *each* TOC, but its effect will be limited to that TOC.

I found out experimentally that the option `hidelinks` could indeed be set many times with `\hypersetup` (this is not the case of all hyperref options).



## Part III.

# Examples

Here are some statistics for this part: it contains 6 sections and 0 subsection. The name of the first section is “Testing the compatibility mode” and the corresponding number is “17”. The name of the last section is “A (crazy) inline display” and its number is “22”.

<b>17. Testing the compatibility mode</b>	<b>24</b>
<b>18. Another compatibility mode</b>	<b>25</b>
<b>19. Emulating the book class</b>	<b>27</b>
<b>20. A framed display</b>	<b>31</b>
<b>21. Another TOC with a background color</b>	<b>31</b>
<b>22. A (crazy) inline display</b>	<b>33</b>

We present some additional examples. To understand all code snippets in detail, one will possibly need to have first browsed through [Part V](#) and [Part VI](#).

### 17. Testing the compatibility mode

As a further example we now print the local table of contents of [Part VI](#). First we will test the compatibility mode.<sup>12</sup> The original was invisibly defined with a label at the beginning of [Part VI](#).

```
\begingroup % to keep in particular toc=left with local effect
\KOMAOptions{toc=left}
\etocstandarddisplaystyle % necessary for the display to obey toc=left
\etocstandardlines
\tableofcontents \ref{toc:globalcmds}
\endgroup
```

### Contents

<b>42. Specifying the toc display style</b>	<b>64</b>
42.1. The command <code>\etocsettocstyle</code> . . . . .	64
42.2. The commands <code>\etocmulticolstyle</code> , <code>\etocmulticol</code> , and <code>\etoclocalmulticol</code> . . . . .	65
42.3. The command <code>\etoclocstyle</code> . . . . .	65
42.4. The commands <code>\etocruledstyle</code> , <code>\etocruled</code> and <code>\etoclocalruled</code> . . .	66
42.5. The commands <code>\etocframedstyle</code> , <code>\etocframed</code> , and <code>\etoclocalframed</code> .	66

<sup>12</sup>the present document uses the `scrartcl` class, and we check here that the `etoc` compatibility mode does respect the customizing done via the class commands.



42.6. Headings, titles, <code>\etocoldpar</code> , <code>\etocinnertopsep</code> . . . . .	66
42.7. The compatibility mode <code>\etocstandarddisplaystyle</code> . . . . .	67
42.8. The command <code>\etocinline</code> . . . . .	67
<b>43. Starred variants of the <code>\tableofcontents</code> etc... commands</b>	<b>67</b>

## 18. Another compatibility mode

As explained in [subsection 4.3](#), the commands `\etocstandardlines` and `\etocstandarddisplaystyle` tell **etoc** to, essentially, act as an observer. The document class layout for the table of contents is then perfectly obeyed. There is no way to customize this standard layout (change fonts, margins, vertical spacings, etc...) from within the package. For this, use some package dedicated to this task; because **etoc** either is (temporarily perhaps) in compatibility mode with no customization on its part possible, or the user has specified the layout in `\etocsetstyle` commands (and `\etocsettocstyle`) and is (supposedly...) in complete control.

Well, there is actually an alternative. It is possible to use the `\etocsetstyle` commands to recreate an artificial compatibility mode, in order to achieve effects like the following, all things being otherwise equal to the document class defaults:

1. get the hyperref link to encapsulate only the names, but not the numbers of each entry of the table of contents,
2. use the document class style for chapters and sections, but modify it only for subsections,
3. do either of the above only for some portions of the table of contents.

Here is how to proceed.

This has become easier since release 1.08k as now **etoc** preserves automatically in `\etocsavedsectiontocline`, `\etocsavedchaptertocline` etc... the meaning of the  $\text{\LaTeX}$  `\l@section`, `\l@chapter`, etc... macros as they were at the time the TOC is typeset.

```
\makeatletter
\newcommand{\MyStandardTOC}{%
  \begingroup
  % for the book or article classes:
  %\etocsetstyle{part}{}{}
  % {\etocsavedparttocline{\etocnumber\hspace{1em}\etocname}{\etocpage}}{}%
  % for the scrbook or scrartcl classes:
  \etocsetstyle{part}{}{}
  {\etocsavedparttocline{\numberline{\etocnumber}\etocname}{\etocpage}}{}%
  % following is identical in book/article/scrbook/scrartcl classes:
  \etocsetstyle{chapter}{}{} %% only for book and scrbook
  {\etocsavedchaptertocline{\numberline{\etocnumber}\etocname}{\etocpage}}{}%
  \etocsetstyle{section}{}{}
  {\etocsavedsectiontocline{\numberline{\etocnumber}\etocname}{\etocpage}}{}%
  \etocsetstyle{subsection}{}{}
  {\etocsavedsubsectiontocline{\numberline{\etocnumber}\etocname}{\etocpage}}{}%
  % etc... if further sectioning units are used
  \etocstandarddisplaystyle % this is for the title, page-marks, etc...
  \tableofcontents
  \endgroup}
\makeatother
```

One can add to the above arbitrary text formatting commands, for example replace `\etocpage` by `\textcolor{blue}{\etocpage}`.

If the document has only one table of contents then there is no need to put the commands inside a macro, or even inside a group.<sup>13</sup> With these commands `etoc` will construct a TOC completely identical to what would have been done by one of the document class: `article`, `book`, `scrartcl`, `scrbook`.<sup>14</sup> The number and the name of each entry are each separately an `hyperref` link, as is always the case with `etoc`, when not in compatibility mode. Replacing `\etocnumber` with `\etocthenumber` will give a TOC where the numbers are not links anymore, but the names still are. Or one may decide to use `\etocthename` and keep an hyperlinked number with `\etocnumber`.

Here is a subtler example where one only marginally modifies the sections (adding color to the number and removing the `hyperref` link) and keeps the subsections as in the default, *except* for those of one specific section, for which the layout is completely modified:

## Contents

<b>3. Do I need to be a geek to use <code>etoc</code>?</b>	<b>8</b>
3.1. Limitations in the use of list environments for tables of contents . . . . .	8
<b>4. Line styles and toc display style</b>	<b>9</b>
<i>\etocsetstyle for the line styles (4.1) – \etocsettocstyle for the toc display (4.2) – Compatibility mode (4.3).</i>	
<b>5. A first example</b>	<b>10</b>
<b>6. A second example</b>	<b>12</b>
<b>7. A Beautiful Thesis example</b>	<b>13</b>
<b>8. Linked list of the main package commands</b>	<b>15</b>

This example only has sections and subsections, and the code used in `\MyStandardTOC` was:

```
\etocsetstyle{section}{}
{\ifnum\etocthenumber=4
  \etocsetstyle{subsection}
    {\par\nopagebreak\begin{group}
      \leftskip1.5em \rightskip\@tocrmarg \parfillskip\@flushglue
      \parindent 0pt \normalfont\normalsize\rmfamily\itshape
      % \columnseplem
```

<sup>13</sup>and if moreover one just wants to keep the same layout as in the default, one may question why using `etoc`... there is *one* good reason: numbers and names are separately `hyperref` links, whereas normally there is only one link holding both the number and the name corresponding to one toc entry.

<sup>14</sup>For the memoir class, one needed prior to `etoc` 1.08k a bit more: each of the command `\booknumberline`, `\partnumberline` and `\chapternumberline` had to be saved. They can now be used directly in the line styles, because their meanings are not modified anymore by `etoc` during its TOC typesetting.

```

% \begin{minipage}{\dimexpr\linewidth-\leftskip-\rightskip\relax}%
% \begin{multicols}{2}%
\etocskipfirstprefix
{\allowbreak\-,}\,
{\etocname\ \textup{(\etocnumber)}}
{.\par\endgroup}%
% {.\par\end{multicols}\end{minipage}\par\endgroup}%
\else
\etocsetstyle{subsection}
{}{}
{\etocsavedsubsectiontocline{\numberline{\etocnumber}\etocname}{\etocpage}}
{}%
\fi}
{\etocsavedsectiontocline{\numberline{{\color{cyan}\etocthenumber}}\etocname}{\etocpage}}
{}%

```

Notice the page head-mark added by this standard TOC. Sections and subsections are printed exactly as in the default (except for the subsections of one specific user-chosen section and except for the color of the section numbers), with no need to specify explicitly any length, font or other formatting instructions. But we had to examine the `scrartcl` sources to determine what to use for `\leftskip` and `\rightskip` for our customized section entries.

## 19. Emulating the book class

As explained in [subsection 4.3](#): without explicit use of an `\etocsetstyle` command the package will leave to the document class the hand regarding the “toc line styles”. It is sometimes asked by users (for example those using `etoc` for its `\localtableofcontents`) how to stay close to but not completely identical with the design implemented by the standard classes, such as `book`. I can recommend package `tocloft` for this, as it is compatible with `etoc` (see [subsection 54.6](#)) and thus `etoc` will obey the `tocloft` customizations (as long as no use has been made of `\etocsetstyle`). It is also possible to modify only the style for, say, sections and leave the parts, chapters, subsections as in the document class, via the technique from [section 18](#).

But for complete control, here is a translation of the `book` class code into `etoc` lingua. It is then easy to modify the relevant lengths or adjust the used fonts. I thank DENIS BROUZÉ for prompting me to include this in the `etoc` manual, as it resulted from some conversation we had about this. The code is not 100% faithful to the `book` class, and particularly its rendering of (multi-line) non-numbered units differs (... I think, as I copied pasted as is the code from where I had stored it and did not do much thinking about it again). Some proficiency in low-level `TEX` and `LATEX` macros is needed to understand what the code says, but for modifying fonts or some lengths such in-depth understanding is not needed.

With some extra code one can *automatically adjust the widths* assigned to typesetting sectioning numbers in order to prevent overflows, even with for example XXXVIII; but this is a more advanced feature which I have moved to [section 30](#).

First we set up some lengths. I use macro registers, not real `LATEX` lengths. When using `em`’s however, this means that one must pay attention to when the actual dimension assignment is made, as this will then depend upon the current font settings. In the code below, at the location where the `\TOCnumwidthB` and `\TOCnumwidthC` will be used, the `1em` from

## 19. Emulating the book class

their specification will be matched to the normal medium series font, not the bold font; this is deliberate so that one can compare more readily with the other dimensions; besides, with the `\TOCcomputenumwidths` from [section 30](#) these macros will actually hold a dimension using pt as dimensional unit.

```
% it will be easy to globally shift the TOC horizontally if needed
\def\TOCleftmargin      {0pt}
\def\TOCrightmargin    {2.55em}% like LaTeX's \@tocrmarg

% this is for dotted leaders
\newbox\TOCleaderbox
\def\TOCleaderboxwidth {0.7777em}% about like what standard classes do

% vertical spacing
\def\TOCverysmallvskip {0pt plus .2pt}
\def\TOCmedvskip       {1em plus 1pt}
\def\TOCbigvskip       {2.25em plus 1pt}

% the "numwidths" for typesetting the numbering of division units.
% I don't recall exactly how (and for which fonts) these figures were chosen.
% They quickly prove too small if using Roman numerals (as do too the book
% class defaults even though they are a bit larger).
\def\TOCnumwidthB {1.5em} % chapter
\def\TOCnumwidthC {2.278em}% section, I think default is 2.3em
\def\TOCnumwidthD {3.056em}% analog in standard class is 3.2em
\def\TOCnumwidthE {3.833em}% analog in standard class is 4.1em
\def\TOCnumwidthF {4.611em}% analog in standard class is 5em
\def\TOCnumwidthG {5.389em}% analog in standard class is 6em

% The code for the "global toc style".

\newcommand*\TOCglobalstyle {%
\etocsettocstyle
  {\if@twocolumn \@restonecoltrue \onecolumn \else \@restonecolfalse \fi
   \parindent\z@ \leftskip\z@skip \rightskip \z@skip
   \setbox\TOCleaderbox\hbox to \TOCleaderboxwidth{\hss.\hss}%
   \chapter *{\noindent\kern\TOCleftmargin\relax % uses "pt"...
    \contentsname
    \@mkboth {\MakeUppercase \contentsname}{\MakeUppercase \contentsname}}%
   \rightskip \TOCrightmargin\relax
   \parfillskip -\rightskip % or a smaller value if desired
   \leftskip \TOCleftmargin \relax }
  {\if@restonecol \twocolumn \fi\cleardoublepage}%
%
\etocsetstyle{part}
{}
{\addpenalty {-\@highpenalty}%
 \addvspace \TOCbigvskip
 \leavevmode
 {\large \bfseries % use a group to limit font change
  \interlinepenalty\@M
  \etocifnumbered{\etocnumber\hspace{1em}}{}}%
  \etocname
  \nobreak\hfil\makebox[-\parfillskip][r]{\etocpage}}\par
\nobreak
}
{}}
```

```

    {}%
%
\etocsetstyle{chapter}
{\advance\leftskip\TOCnumwidthB\relax}
{\addpenalty {-\@highpenalty }%
\vskip \TOCmedvskip\relax
\leavevmode
{\interlinepenalty\@M
\etocifnumbered
{\llap{\makebox[\TOCnumwidthB][l]{\bfseries\etocnumber}}}
{\advance\leftskip-\TOCnumwidthB\relax}%
\bfseries\etocname
\nobreak\hfil\makebox[-\parfillskip][r]{\etocpage}\par }%
\penalty \@highpenalty
}
}
{\advance\leftskip-\TOCnumwidthB\relax}%
%
\TOCsetlinestyle {section}      {\TOCnumwidthC}%
\TOCsetlinestyle {subsection}   {\TOCnumwidthD}%
\TOCsetlinestyle {subsubsection}{\TOCnumwidthE}%
\TOCsetlinestyle {paragraph}    {\TOCnumwidthF}%
\TOCsetlinestyle {subparagraph} {\TOCnumwidthG}%
}% end of \TOCglobalstyle

%The common code for line styles is abstracted into a macro:

\newcommand\TOCsetlinestyle [2]{% #1= unit, #2= numwidth as macro
\etocsetstyle{#1}
{\advance\leftskip#2\relax}
{\vskip \TOCverysmallvskip\relax
\leavevmode
{\interlinepenalty\@M
\etocifnumbered
{\llap{\makebox[#2][l]{\etocnumber}}}{\advance\leftskip-#2\relax}%
\etocname
\nobreak\leaders \copy\TOCleaderbox
\hfil\makebox[-\parfillskip][r]{\etocpage}%
\par }%
}
}
{\advance\leftskip-#2\relax}%
}
\makeatother

```

Nota Bene: the code deliberately handles the non-numbered sectioning units unlike the way of the standard document classes (particularly regarding the alignment of multi-line headings.)

The whole thing was encapsulated in `\TOCglobalstyle`, because we also want a `\TOClocalstyle` for local tables of contents which typically will want to use `\section*` rather than `\chapter*` and not insert page marks in the headers. The `\TOClocalstyle` is to be issued once, after the main document TOC, or rather before using `\localtableofcontents`. If one wants a full TOC at end of document one will naturally have to issue again `\TOCglobalstyle` there.

## 19. Emulating the book class

```
\makeatletter
\newcommand*{\TOClocalstyle}{%
\etocsettocstyle
  {\if@twocolumn \@restonecoltrue \onecolumn \else \@restonecolfalse \fi
   \setbox\TOCleaderbox\hbox to \TOCleaderboxwidth{\hss.\hss}%
   \parindent\z@
   \dimen@ 2.25em % for left indenting
   \section *{\kern\dimen@ % use of \dimen@ works here by sheer luck
    \contentsname
    % un-comment this if marks are wanted:
    %\@mkboth {\MakeUppercase \contentsname}{\MakeUppercase \contentsname}%
    }% end of \section
   \parskip \z@skip
   \vspace{-1.25\baselineskip}% somewhat ad hoc
   \leftskip 2.25em
   \rightskip 4.5em
   \advance\rightskip-\TOCrightmargin\relax
   \leavevmode\leaders\hrule\@height\p@\hfill\kern\z@\par
   \rightskip 4.5em
   \parfillskip -\TOCrightmargin\relax }
  {\nobreak\vskip-.5\baselineskip
   \leavevmode\leaders\hrule\@height\p@\hfill\kern\z@\par
   \bigskip
   \if@restonecol \twocolumn \fi }%
%
\etocsetstyle{section}{
{\advance\leftskip\TOCnumwidthC\relax}
{\addpenalty \@secpenalty
\etociffirst}{\addvspace{\TOCmedvskip}}}%
\leavevmode
{\interlinepenalty\@M
\bfseries\etocifnumbered
  {\llap{\makebox[\TOCnumwidthC][l]{\etocnumber}}}
  {\advance\leftskip-\TOCnumwidthC}%
\etocname\nobreak\hfil\makebox[-\parfillskip][r]{\etocpage}\par }%
\penalty \@highpenalty }
{}
{\advance\leftskip-\TOCnumwidthC\relax}%
% the rest is identical with code for global tocs:
\TOCsetlinestyle {subsection} {\TOCnumwidthD}%
\TOCsetlinestyle {subsubsection}{\TOCnumwidthE}%
\TOCsetlinestyle {paragraph} {\TOCnumwidthF}%
\TOCsetlinestyle {subparagraph} {\TOCnumwidthG}%
}% end of \TOClocalstyle
\makeatother
```

As mentioned previously, this handles non-numbered (multi-line) sectioning units somewhat differently from what happens in the standard document classes.

For some reason this code has some hard-coded 2.25em and 4.5em which were not abstracted into macros or lengths. The code inserts horizontal rules above and below the TOC contents in a non-separable by pagebreak way.

See [section 30](#) for more.

## 20. A framed display

We now opt for a “framed” style, using the package default line styles and some colors added (it has been put in a float which appears on the next page).

```
\etocdefaultlines
\begin{group}
\renewcommand{\etoccolumnsep}{2em}
\renewcommand{\etocinnerleftsep}{1.5em}
\renewcommand{\etocinnerrightsep}{1.5em}
% specify a background color for the toc contents
\renewcommand{\etocbkgcolorcmd}{\color{yellow!10}}
% set up the top and bottom rules
\renewcommand{\etoctoprule}{\hrule height 1pt}
\renewcommand{\etoctoprulecolorcmd}{\color{red!25}}
\renewcommand{\etocbottomrule}{\hrule height 1pt}
\renewcommand{\etocbottomrulecolorcmd}{\color{red!25}}
% set up the left and right rules
\renewcommand{\etocleftrule}{\vrule width 5pt}
\renewcommand{\etocrightrule}{\vrule width 5pt}
\renewcommand{\etocleftrulecolorcmd}{\color{red!25}}
\renewcommand{\etocrightrulecolorcmd}{\color{red!25}}
% use \fcolorbox to set up a colored frame for the title
\fbboxrule1pt
\renewcommand{\etocbelowtocskip}{0pt\relax}
\etocframedstyle {\normalsize\rmfamily\itshape
  \fcolorbox{red}{white}{\parbox{.8\linewidth}{\centering
    This is a table of contents \‘a la \etoc, but just for
    the sections and subsections in this part. As it is put
    in a frame, it has to be small enough to fit on
    one page. It has the label |toc:b|.}}}
\begin{figure}[ht!]
  \centering
\tableofcontents \label{toc:b} \ref{toc:globalcmds}
\end{figure}
\end{group}
```

## 21. Another TOC with a background color

Let us now try out some more sophisticated line styles. The display will use the `\etocframedstyle` package command, which requires that the produced table of contents fits on a single page. We wrap it up in a [figure environment](#) showing up on page 34.

This design uses the **etoc** ‘framed’ style with a background color. The frame borders have been set to have the same color as the one serving as background for the entire thing. It would be advantageous to use rather inside `\etocsettocstyle` commands from a package like `tcolorbox` as this allows sophisticated breakable boxes (with `TikZ/pgf` for decoration.)

The details of the line styles used here are a bit involved, they were written by the author at some early stage of this documentation and have only been slightly revised to use more  $\LaTeX$ -commands and less  $\TeX$ -primitives. Similar code is used also for [this other toc](#).

## 21. Another TOC with a background color

*This is a table of contents à la **etoc**, but just for the sections and subsections in this part. As it is put in a frame, it has to be small enough to fit on one page. It has the label `toc:b`.*

### Specifying the toc display style

42, p. 64

The command `\etocsettocstyle`

42.1, p. 64

The commands `\etocmulticolstyle`, `\etocmulticol`, and `\etoclocalmulticol` . . . 42.2, p. 65

The command `\etocstyle` . . . 42.3, p. 65

The commands `\etocruledstyle`, `\etocruled` and `\etoclocalruled` . . . 42.4, p. 66

The commands `\etocframedstyle`, `\etocframed`, and `\etoclocalframed`

42.5, p. 66

Headings, titles, `\etocoldpar`, `\etocinnertopsep` . . . 42.6, p. 66

The compatibility mode `\etocstandarddisplaystyle` . . . 42.7, p. 67

The command `\etocinline` . . . 42.8, p. 67

### Starred variants of the `\tableofcontents` etc... commands

43, p. 67

```
\begin{figure}[htbp!]\centering
\colorlet{subsecnum}{black}
\colorlet{secbackground}{green!30}
\colorlet{tocbackground}{red!20!green!20}

\renewcommand{\etocbkgcolorcmd}{\color{tocbackground}}
\renewcommand{\etocleftrulecolorcmd}{\color{tocbackground}}
\renewcommand{\etocrightrulecolorcmd}{\color{tocbackground}}
\renewcommand{\etocbottomrulecolorcmd}{\color{tocbackground}}
\renewcommand{\etoctoprerulecolorcmd}{\color{tocbackground}}

\renewcommand{\etocleftrule}{\vrule width 3cm}
\renewcommand{\etocrightrule}{\vrule width 1cm}
\renewcommand{\etocbottomrule}{\hrule height 12pt}
\renewcommand{\etoctoprerule}{\hrule height 12pt}

\renewcommand{\etocinnertopsep}{0pt}
\renewcommand{\etocinnerbottomsep}{0pt}
\renewcommand{\etocinnerleftsep}{0pt}
\renewcommand{\etocinnerrightsep}{0pt}

\newcommand\shiftedwhiterule[2]{%
  \hbox to \linewidth{\color{white}%
    \hskip#1\leaders\vrule height1pt\hfil}\nointerlineskip
  \vskip#2}

\etocsetstyle{subsubsection}
{\etocskipfirstprefix}
{\shiftedwhiterule{\leftskip}{6pt}}
{\sffamily\footnotesize
  \leftskip2.3cm\hangindent1cm\rightskip.5cm\relax
  \makebox[1cm][l]{\color{subsecnum}\etocnumber}%
  \color{black}\etocname
  \nobreak\leaders\hbox to.2cm{\hss.}\hfill}
```



```

\rlap{\makebox[.5cm][r]{\etocpage\hspace{.1cm}}}\par
\nointerlineskip\vskip3pt}
{}

\etocsetstyle{subsection}
{\etocskipfirstprefix}
{\shiftedwhiterule{1.5cm}{6pt}}
{\sffamily\small
\leftskip1.5cm\hangindent.8cm\rightskip.5cm\relax
\makebox[.75cm][l]{\color{subsecnum}\etocnumber}%
\color{black}\etocname
\nobreak\leaders\hbox to.2cm{\hss.}\hfill
\rlap{\makebox[.5cm][r]{\etocpage\hspace{.1cm}}}\par
\nointerlineskip\vskip3pt}
{}

\newcommand{\coloredstuff}[2]{%
\leftskip0pt\rightskip0pt\parskip0pt
\fbboxsep0pt % \colorbox uses \fbboxsep also when no frame!
\noindent\colorbox{secbackground}
{\parbox{\linewidth}{%
\vskip5pt
{\noindent\color{#1}#2\par}\nointerlineskip
\vskip3pt}}%
\par\nointerlineskip}

\etocsetstyle{section}
{\coloredstuff{blue}{\hfil \bfseries\large Contents of Part One\hfil}}
{\vskip3pt\sffamily\small}
{\coloredstuff{blue}
{\leftskip1.5cm\rightskip.5cm\parfillskip-\rightskip
\makebox[0pt][r]{\makebox[.5cm][l]{\etocnumber}}%
\etocname\nobreak\hfill\makebox[.5cm][r]{\etocpage\hspace{.1cm}}}%
\vskip6pt}
{}

\etocframedstyle[1]{
\tableofcontents \label{toc:floating} \ref{toc:overview}
\vspace{-\baselineskip}
\centeredline{|\tableofcontents \ref{toc:overview}|
(\emph{cf.} \hyperref[toc:clone]{this other toc})}
\end{figure}

```

## 22. A (crazy) inline display

Let us finally make some crazy inline display of the table of contents of this entire document. We will typeset the subsections as footnotes... This kind of style is suitable for a hyperlinked document, probably not for print! (although I like it, but my personal tastes in many matters do not seem to be widely shared).

Here is the inline table of contents. *Abstract, Foreword, License. Overview: Do I need*

Contents of Part One	
3	Do I need to be a geek to use <b>etoc</b> ? 8
3.1	Limitations in the use of list environments for tables of contents . . . . . 8
4	Line styles and toc display style 9
4.1	\etocsetstyle for the line styles . . . . . 9
4.2	\etocsettocstyle for the toc display . . . . . 9
4.3	Compatibility mode . . . . . 10
5	A first example 10
6	A second example 12
7	A Beautiful Thesis example 13
8	Linked list of the main package commands 15

`\tableofcontents \ref{toc:overview}` (cf. [this other toc](#))

to be a geek to use **etoc**?<sup>15</sup>, Line styles and toc display style<sup>16</sup>, A first example, A second example, A Beautiful Thesis example, Linked list of the main package commands. **Arbitrarily many TOCs, and local ones too:** Labeling and reusing elsewhere, A powerful functionality of **etoc**: the re-assignment of levels with `\etocsetlevel`, The `\etocsettocdepth` and `\etocsetnexttocdepth` commands<sup>17</sup>, The command `\etocsettocdepth.toc`<sup>18</sup>, The commands `\etocdepthtag.toc` and `\etocsettagdepth`<sup>19</sup>, The commands `\etocglobaldefs` and `\etoclocaldefs`, Not displayed empty TOCs<sup>20</sup>, Adding commands to the `.toc file`<sup>21</sup>. **Examples:** Testing the compatibility mode, Another compatibility mode, Emulating the book class, A framed display, Another TOC with a background color, A (crazy) inline display. **Surprising uses of etoc:** The TOC of TOCs, Arbitrary “Lists Of...”, `\etoc toc contents line`, A TOC with a fancy layout, The TOC as a tree, The TOC as a molecule, The TOC as a TikZ mind map, The TOC as a table, A TOC self-adjusting widths for its typesetting. **Commands for the toc line styles:** The `\etocsetstyle`, `\etocname` and `\etocpage` commands, The `\etocskipfirstprefix` and `\etociffirst` commands, The `\etocnumber` command, The `\etocifnumbered` switch, The `\etocthename`, `\etocthenumber`, and `\etocthepage` commands, The `\etoclink`

<sup>15</sup>Limitations in the use of list environments for tables of contents.

<sup>16</sup>`\etocsetstyle` for the line styles; `\etocsettocstyle` for the toc display; Compatibility mode.

<sup>17</sup>The `hyperref` option `bookmarksdepth`.

<sup>18</sup>The commands `\etocobeytocdepth` and `\etocignoretocdepth`.

<sup>19</sup>The commands `\etocobeydepthtags` and `\etocignoredepthtags`.

<sup>20</sup>The `\etocchecksempiness` command; The `\etocnotocifnotoc` command; The `\etocifwasempty` command.

<sup>21</sup>The `hyperref` option `hidelinks`.

command, The `\etocthelinkedname`, `\etocthelinkednumber`, `\etocthelinkedpage` and `\etocthelink` commands, The `\etocsavedsectiontocline`, etc... commands., The `\etocsetlevel` command, Scope of commands added to the `.toc` file<sup>22</sup>, Am I also red?. **Commands for the toc display style:** Specifying the toc display style<sup>23</sup>, Starred variants of the `\tableofcontents` etc... commands. **Using and customizing the `etoc` own styles:** Summary of the main style commands<sup>24</sup>, The package default line styles: `\etocdefaultlines`, Customizing `etoc`<sup>25</sup>, One more example of colored TOC layout. **Tips:** The `\etocsetlocaltop.toc` command, Hacking framed parboxes, Intervverting the levels<sup>26</sup>, Displaying statistics, Using depth tags, Typesetting the TOC as a table (the old way). **`etoc` and the outside world:** Compatibility with other packages<sup>27</sup>, T<sub>E</sub>Xnical matters, Errors and catastrophes. **The code:** Timestamp, Change history, Implementation.

The code used:

```
\begingroup
\newsavebox{\forsubsections}
\etocsetstyle{part}{\upshape. \etocskipfirstprefix}
    {\. \upshape}
    {\bfseries\etocname:~~}
    {}
\etocsetstyle{section}{\itshape\etocskipfirstprefix}
    {\, }
    {\mdseries\etocname}
    {}
\etocsetstyle{subsection}
    {\begin{lrbox}{\forsubsections}\upshape\etocskipfirstprefix}
    {\; }
    {\etocname}
    {\end{lrbox}\footnote{\unhbox\forsubsections.}}
\etocsetstyle{subsubsection}
    {\itshape\etocskipfirstprefix}
    {\, }
    {\etocname}
    {\/\upshape)}
\etocsettocstyle{Here is the inline table of contents. }{.\par}
\tableofcontents \label{toc:crazyinline}
\endgroup
```

<sup>22</sup>Testing the scope; This is a (pale) red subsection for illustrative purposes.

<sup>23</sup>The command `\etocsettocstyle`; The commands `\etocmulticolstyle`, `\etocmulticol`, and `\etoclocalmulticol`; The command `\etoclocalmulticol`; The command `\etoclocalruled`; The commands `\etocruledstyle`, `\etocruled` and `\etoclocalruled`; The commands `\etocframedstyle`, `\etocframed`, and `\etoclocalframed`; Headings, titles, `\etocoldpar`, `\etocinnertopsep`; The compatibility mode `\etocstandarddisplaystyle`; The command `\etocinline`.

<sup>24</sup>Setting up local styles; Setting up toc display styles; Displaying tables of contents; Labels and references.

<sup>25</sup>Customizing the `etoc` pre-defined line styles; Customizing the toc display styles.

<sup>26</sup>All subsections of this document.

<sup>27</sup>Generalities; Compatibility with beamer; Compatibility with Babel; Compatibility with hyperref; Compatibility with multicol; Compatibility with tocloft; Compatibility with the memoir class; Compatibility with package tocvsec2; Compatibility with package tableof; Compatibility with package tocstyle.

## Part IV.

# Surprising uses of **etoc**

Here are some statistics for this part: it contains 8 sections and 0 subsection. The name of the first section is “The TOC of TOCs” and the corresponding number is “23”. The name of the last section is “A TOC self-adjusting widths for its typesetting” and its number is “30”.

<b>23. The TOC of TOCs</b>	<b>36</b>
<b>24. Arbitrary “Lists Of...”, <code>\etoccontentsline</code></b>	<b>38</b>
<b>25. A TOC with a fancy layout</b>	<b>39</b>
<b>26. The TOC as a tree</b>	<b>40</b>
<b>27. The TOC as a molecule</b>	<b>43</b>
<b>28. The TOC as a TikZ mind map</b>	<b>45</b>
<b>29. The TOC as a table</b>	<b>49</b>
<b>30. A TOC self-adjusting widths for its typesetting</b>	<b>54</b>

## 23. The TOC of TOCs

Here is the numbered and linked list of all tables of contents which are displayed within this document:<sup>28</sup> [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#), [9](#), [10](#), [11](#), [12](#), [13](#), [14](#), [15](#), [17](#), [16](#), [18](#), [19](#), [20](#), [21](#), [22](#), [23](#), [24](#), [25](#), [26](#), [27](#), [28](#), [29](#), [30](#), [31](#), [32](#), [33](#), [34](#), [35](#), [36](#), [37](#), [38](#), [39](#), [40](#), [41](#), [42](#), [43](#). And to obtain it here we just wrote:

```
Here is the numbered and linked list of all tables of contents
which are displayed within this document: \tableofcontents.
```

The preparatory work was the following. First, we defined a counter `visibletoc` whose vocation is to get incremented at each displayed toc. **etoc** has its own private counter but it counts all TOCs, even those not displayed because the `tocdepth` value was `-2` or `-3`.

We could have added manually `\refstepcounter{visibletoc}` and `\label` commands at all suitable locations in the document source, and we would then have used here `\ref` commands, but this imposes heavy manual editing of the source.

There is a much better way: there is a hook `\etocaftertitlehook` and we told it to increment the `visibletoc` counter and to write a line to the `.toc` file, in a manner analogous to what sectioning commands such as `chapter`, `section`, or `subsection` do. As **etoc** increments its own private counter even before typesetting the title of a table of contents, this provides (most of the time) a better link destination than any counter manipulated from inside `\etocaftertitlehook` (for which the link would target the area just

<sup>28</sup>The TOCs put in floats may change the order: the numbers are listed in the order the TOCs are typeset in the document; but the numbering itself is from the order of the TOCs in the *source* of this document...

after the title). So, rather than including `\refstepcounter{visibletoc}` inside `\etoc-aftertitlehook`, we just put there `\stepcounter{visibletoc}` followed by the command `\etoccontentsline{visibletoc}{\thevisibletoc}`. This **etoc** command `\etoccontentsline{<level_name>}{<name>}` has the same effect as:

```
\addcontentsline{toc}{<level_name>}{<name>}
```

but its usefulness is to circumvent<sup>29</sup> the patching for automatic creation of bookmarks done to `\addcontentsline` by the `hyperref` package, as pdf bookmarks don't make much sense here (and would elicit a complaint of `hyperref` that the bookmark level is 'unknown').<sup>30</sup>

Finally, the preamble of the document did `\etocsetlevel{visibletoc}{6}`. The level 6 (or anything with a higher number) is ignored, even if `tocdepth` has value 10 for example; this is independently of whether **etoc** uses the document class default line styles or its own line styles, or the ones defined by the user with the `\etocsetstyle` command. So there is no need to worry that something could go wrong.

Then, only here we have set `\etocsetlevel{visibletoc}{0}`. And to display only this kind of entries we assign temporarily to part and chapter level 1 (or anything higher than zero) and set `tocdepth` to the value 0. We also did `\etocsetstyle{visibletoc}{\etocskipfirstprefix}{, }{\etocname}{}` which defines an inline display with the comma as separator. Finally, as **etoc** issues `\par` automatically by default just before typesetting a table of contents, we used the command `\etocinline` (also known as `\etocnopar`) which turns off this behavior.

Here are the implementation details:

```
< in the preamble >
\newcounter{visibletoc}
\renewcommand{\etocaftertitlehook}
{ \stepcounter{visibletoc} \etoccontentsline{visibletoc}{\thevisibletoc} }
\etocsetlevel{visibletoc}{6}
\begin{document}
  < document body >
\subsection{Surprising uses of etoc}
\begin{group}
  \etocinline
  \etocsetlevel{part}{1}
  % \etocsetlevel{chapter}{1} % (no chapters in scrartcl class)
  \etocsetlevel{visibletoc}{0}
  \etocsetstyle{visibletoc}
    { \etocskipfirstprefix}{, }{ \color{niceone} \etocname }{ }
  \etocsettocstyle{}{} % don't set any title, rules or frame or multicol!
  \etocsetnexttocdepth{visibletoc} % display only the 'visibletoc' entries from .toc
\end{group}
```

Here is the numbered and linked list of all tables of contents which are displayed within this document: `\tableofcontents`.

<sup>29</sup>using `\addtocontents` rather than `\addcontentsline`

<sup>30</sup>The package provides a starred variant `\etoccontentsline*`, which does allow the creation of bookmarks and has a third mandatory argument which is the Level to be used by these bookmarks; depending on the context the starred as well as the non-starred variants may be profitably preceded by `\phantomsection`.

## 24. Arbitrary “Lists Of...”, `\etoccontentsline`

After `\etocsetstyle{visibletoc}{...}{...}{...}`, all future TOCs (not in compatibility mode) will use the defined style for level 0 (which is normally the level for chapters). To keep these changes strictly local the simplest manner is to put everything inside a group.

The [section 50](#) gives another use of the shuffling of levels.

## 24. Arbitrary “Lists Of...”, `\etoccontentsline`

This idea of interverting the levels is very powerful and allows to let **etoc** display lists of arbitrary things contained in the document. All of that still using nothing else than the `.toc` file! Example: imagine a document with dozens of exercises, perhaps defined as `\newtheorem{exercise}{}[section]`. Let us explain how to instruct **etoc** to display an hyperlinked list of all these exercises. For this we put in the preamble:

```
\newtheorem{exerci}{}[section]
% the exercise number will be recoverable via \etocname: v--here--v
\newcommand*\exercisetotoc{\etoccontentsline{exercise}{\theexerci}}
\newenvironment{exercise}{\begin{exerci}\exercisetotoc}\end{exerci}}
\etocsetlevel{exercise}{6}
```

In this way, `\etocname` will give the exercise number (but `\etocnumber` will be empty). Had we used instead

```
\newcommand*\exercisetotoc{
  {\etoccontentsline{exercise}{\protect\numberline{\theexerci}}}
```

the exercise number would then have been available via `\etocnumber`, and `\etocname` would have been empty. It doesn't matter which one of the two methods is used. The **etoc** command `\etoccontentsline{...}{...}` is provided as a substitute to `\addcontentsline{toc}{...}{...}`: this is to avoid the patching which is done by `hyperref` to `\addcontentsline` in its process of creation of bookmarks. If one wants to authorize `hyperref` to create bookmarks at a specific level  $\langle n \rangle$ , one can use (here with  $\langle n \rangle = 2$ ) the starred variant `\etoccontentsline*` which has an additional argument:

```
\newcommand{\exercisetotoc}{\etoccontentsline*{exercise}{\theexerci}{2}}
```

The counter `exerci` is already incremented by the `exerci` theorem environment, and provides the correct destination for the link added by package `hyperref`. The command `\exercisetotoc` adds for each exercise a line to the `.toc` file, corresponding to a fictitious document unit with name ‘exercise’. A four-column list, including the sections, can then be typeset with the following code:

```
\etocsetnexttocdepth{2} % sections are at level 1 and will show up
\begingroup
\etocsetlevel{exercise}{2} % but:
\etocsetlevel{chapter}{3} % no chapters
\etocsetlevel{subsection}{3} % no subsections
\etocsetlevel{part}{3} % no parts
\etocsetstyle{exercise}{} % \etocname = exercise number
  {\noindent\etocname\strut\leaders\etoclineleaders\hfill\etocpage\par}
  {\pagebreak[2]\vskip\baselineskip}
\etocsetstyle{section}{}{}
```

```

{\noindent\strut{\bfseries\large\etocnumber\hskip.5em\etocname}\par
\nopagebreak[3]}}
\etocruledstyle[4]{\Large\bfseries List of the exercises}
\setlength{\columnseprule}{.4pt}
\tableofcontents
\endgroup

```

## 25. A TOC with a fancy layout

Here is a table of contents where the sections from a given chapter are to be represented by a number range (like 18–22 for a given chapter, 42–49 for another one ... of course to be inserted automatically in the TOC).

This is not an image inclusion, the TOC is produced from its original tex source inserted in this document after replacement of `part`, `chapter` or `section` with `dummypart`, `dummychapter` and `dummysection` (and there is also a dummy page count). We copied the line styles used in the original and displayed the table of contents following:

```

\etocsetlevel{dummypart} {-1} \etocsetlevel{part} {2}
\etocsetlevel{dummychapter}{0} \etocsetlevel{chapter}{2}
\etocsetlevel{dummysection}{1} \etocsetlevel{section}{2}
\etocsetnexttodepth{dummysection}

```

Each chapter displays the numbers of only the first and the last sections it contains. See the source file `etoc.dtx` for the details of how this is done.

Another technique for doing this kind of things is explained in [section 51](#).

# TABLE OF CONTENTS

---

## PART I

				SECTIONS.	PAGE.
Introductory	...	...	...	1—8	5

### Concord.

#### LESSON.

1.	Concord of Subject and Verb	...	...	9—17	7
2.	Concord of Substantive and Adjective	...	...	18—22	9
	Concord of Relative and its Antecedent	...	...	23—25	

## PART II

### Government.

3.	The Accusative Case	...	...	...	17
	General uses	...	...	26—30	
	Particular uses	...	...	31—37	
4.	Verbs governing two Accusatives	...	...	38—41	23
5.	The Causal	...	...	42—49	25
6.	The Instrumental Case	...	...	...	27
	General uses	...	...	50—54	
	Particular uses	...	...	55—59	
7.	The Dative Case	...	...	...	33
	General uses	...	...	60—65	
	Particular uses	...	...	66—71	
8.	The Ablative Case	...	...	...	39
	General uses	...	...	72—75	
	Particular uses	...	...	76—86	
9.	The Locative Case	...	...	...	45
	General uses	...	...	87—92	
	Particular uses	...	...	93—100	

## 26. The TOC as a tree

Using `tikz`<sup>31</sup> and the package `forest`<sup>32</sup> we shall display the table of contents of this part as a tree. The technique is to use the `etoc` modified command `\tableofcontents` not for typesetting, but to prepare a macro, or rather here a *token list* variable, with all the instructions to be executed later. LESLIE LAMPORT's book has no mention whatsoever of token lists, and L<sup>A</sup>T<sub>E</sub>X gives the impression to not really expect the general user to ever hear about them (or delimited macros); this whole section and the next are thus for advanced users.

Putting the `\etocnumber` and `\etocname` commands in `\treetok` would be of no use: to which number or name would they then refer to, in a delayed execution?

We need to store, not the macro names, but the macro contents. And also we wish to maintain the correct `hyperref` hyperlinks.<sup>33</sup> The commands `\etocname`, etc. . . , are robust, it is easier to work with `\etocthelinkednumber`, `\etocthelinkedname`, and `\etocthelinkedpage` which contain the same information in an easier accessible form.

For this `forest` tree we have designed very special `etoc` styles for sections and subsections. They use a token list register called `\treetok` and a macro `\appendtotok` whose rôle is to append to a given token list variable the contents of a macro given as second argument. All this will happen in reaction to a `\tableofcontents` command, but *nothing* has yet been printed in the process.<sup>34</sup> This is the later job of a `forest` environment which

<sup>31</sup><http://ctan.org/pkg/pgf>

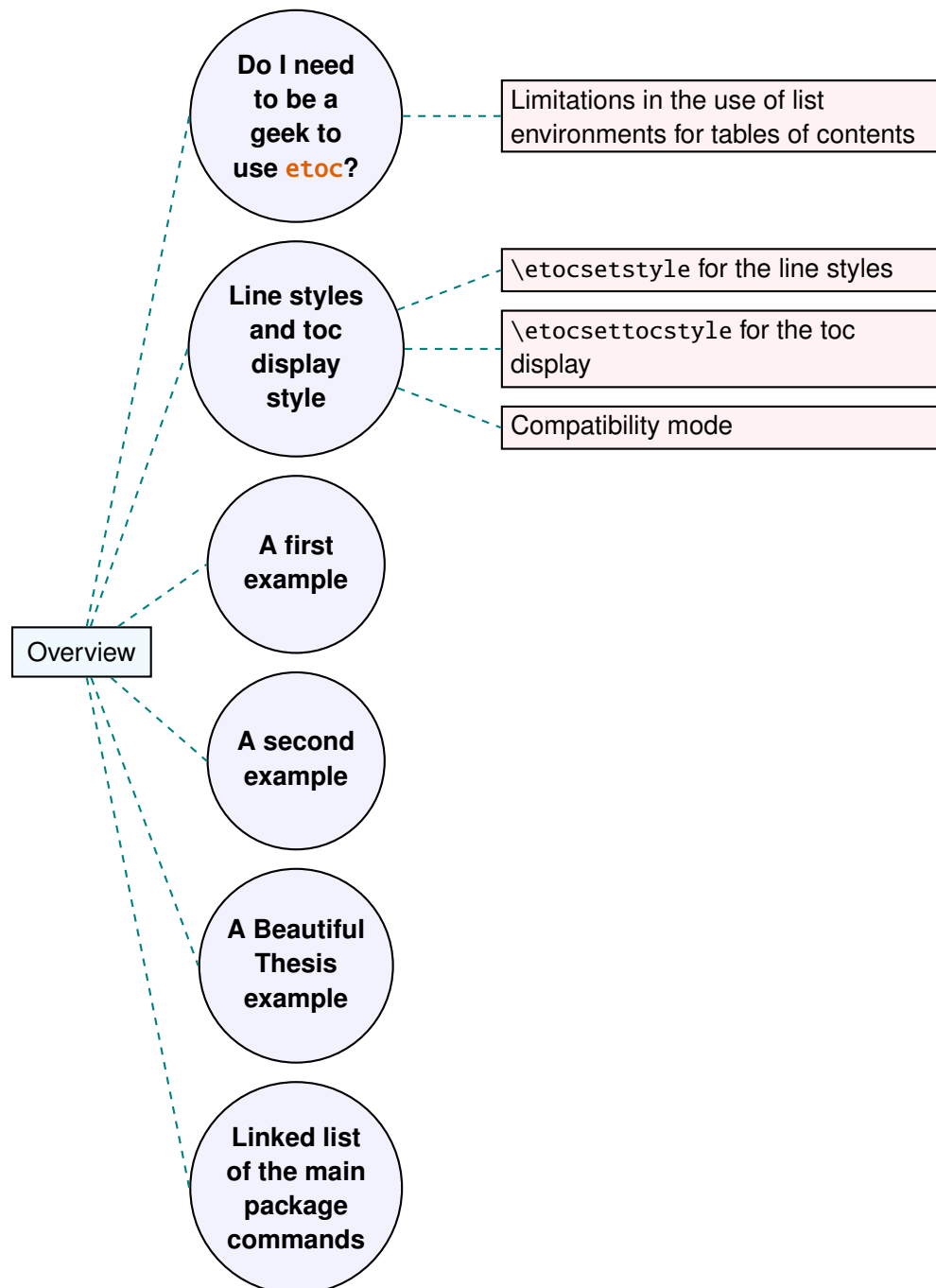
<sup>32</sup><http://ctan.org/pkg/forest>

<sup>33</sup>This manual up to the version of May 14, 2013 used package `tikz-qtrees` but there were problems with hyperlinks. No such problem arises either when using `forest` or the native `tikz` syntax for trees (which will be illustrated in the next section).

<sup>34</sup>There is always a `\par`, which here is not a problem, but can be suppressed if need be via the command `\etocinline` or its synonym `\etocnopar`.



will be given the contents of `\treetok`.



The resulting tree has been put in a [float](#), which appears above. Here is the code used for its production:

```

% \newtoks\treetok % put this (uncommented) preferably in the preamble
% \newtoks\tmptok % (idem)

\newcommand*\appendtotok[2]{% #1=toks variable, #2=macro, expands once #2
  #1\expandafter\expandafter\expandafter

```

## 26. The TOC as a tree

```

{\expandafter\the\expandafter #1#2}}

\newcommand*\PrepareSectionNode{%
  \tmptok {\centering\bfseries}%
  \appendtotok\tmptok\etocthelinkedname
  \edef\foresttreenode{ [{\noexpand\parbox{2cm}{\the\tmptok}}}%
}

\newcommand*\PrepareSubsectionNode{%
  \tmptok {\raggedright}%
  \appendtotok\tmptok\etocthelinkedname
  \edef\foresttreenode{ [{\noexpand\parbox{6cm}{\the\tmptok}}}%
}

\etocsetstyle{section}
  {\etocskipfirstprefix}
  {\appendtotok\treetok{ }}
  {\PrepareSectionNode \appendtotok\treetok\foresttreenode}
  {\appendtotok\treetok{ }}

\etocsetstyle{subsection}
  {\etocskipfirstprefix}
  {\appendtotok\treetok{ }}
  {\PrepareSubsectionNode \appendtotok\treetok\foresttreenode}
  {\appendtotok\treetok{ }}

\etocsettocstyle
  {\treetok{[{\hyperref[part:overview]{Overview}}]}}
  {\global\appendtotok\treetok{ }}

% forest does not like @\the\treetok if \treetok is empty. On first latex
% run, this will be the case because the TOC style defined above will not
% have been executed, as the label {toc:overview} does not refer to a valid
% TOC yet. So we must give a safe default value to \treetok
\treetok{[{\run latex again}]}

\begin{figure}[th!]\centering
  \etocsetnexttocdepth{subsection}
  \tableofcontents \label{toc:forest}\ref{toc:overview}
  \hypersetup{hidelinks}%
  \bracketset{action character=@}
  \begin{forest}
    for tree={anchor=center,child anchor=west,
      grow'=east,draw,thick,
      edge={draw,thick,dashed,color=teal}},
    where={level()==1}{circle,thick,fill=blue!5,
      before computing xy={l=3cm}}{ },
    where={level()==2}{fill=red!5,
      before computing xy={l=6cm}}{ },
    rectangle, thick, fill=cyan!5, inner sep=6pt,
    @\the\treetok
  \end{forest}
\end{figure}

```

Why `\hypersetup{hidelinks}`? as explained in [subsection 16.1](#), I prefer the links in TOCs not to be colored, nor framed, so this document inserts a command `\hypersetup`

`{hidelinks}` in the `.toc` file. But at the time the `\treetok` contents are unpacked the `\hyperlink` commands originating in `\etocthelinkedname`, etc... will be executed in the normal environment for links (which, in this document, is to colorize them). Rather than having `etoc`'s code try to guess what the current "style" for links is (a concept not really provided by `hyperref` it seems) and store it in `\etocthelinkedname`, etc..., I opted for the simpler solution to leave it up to the user to recreate whatever conditions are desired. So here it is necessary to re-issue `\hypersetup{hidelinks}` in the figure environment.

There are some other examples in this documentation where `\tableofcontents` is used to prepare material for later typesetting:

- printing the statistics at the start of each Part (see [section 51](#)) is done using save boxes (so the problem of the appearance of the links does not arise then).
- the typesetting of the TOC as a table in the pre-1.08 way (see [section 53](#)); there we also have to issue `\hypersetup{hidelinks}` after having collected the names, numbers and page numbers in a token list register.
- and the two additional tree examples in the next section.

## 27. The TOC as a molecule

It is also possible to construct a TOC tree obeying the TikZ syntax for trees: but this is a more complicated task for the `etoc` line styles for reasons related to the way braces are handled by  $\TeX$  (they need, when filling up the token list to be always balanced at each step, else complicated tricks must be employed.)

The simplest strategy is to allocate a token list (or use a macro) for each level used: we may need a `\parttok`, a `\chaptertok`, a `\sectiontok` and a `\subsectiontok`, to help in the task of filling up the total `\treetok`. As we are interested here in the table of contents of this (or another) document part, only a `\sectiontok` and a `\subsectiontok` will be needed.

```
% \newtoks\treetok % put this (uncommented) preferably in the preamble
% \newtoks\sectiontok
% \newtoks\subsectiontok
\newcommand*{\treenode}{}

\newcommand*{\appendchildtree}[2]{% token list t1 becomes: t1 child {t2}
  \edef\tmp{\the#1 child {\the#2}}%
  #1\expandafter{\tmp}%
}
\newcommand*{\preparetreenode}{%
  \tmptok\expandafter{\etocthelinkednumber}% expanded one time (mandatory)
  \edef\treenode{node {\the\tmptok}}%
}

\etocsetstyle{section}
{
  \etocskipfirstprefix
  {\appendchildtree\treetok\sectiontok}
  {\preparetreenode \sectiontok\expandafter{\treenode}}
  {\appendchildtree\treetok\sectiontok}
}
```

## 27. The TOC as a molecule

```

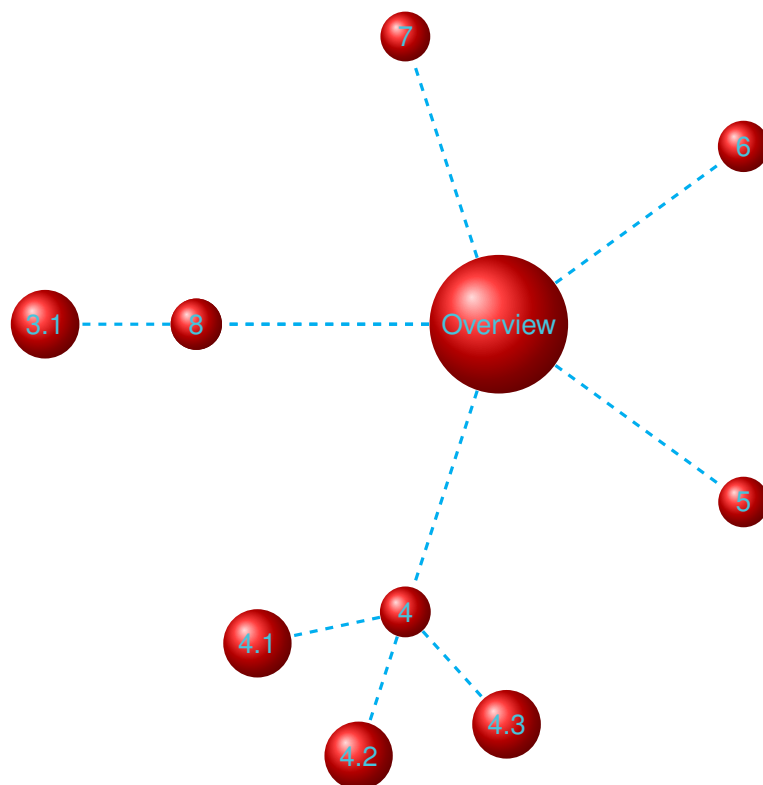
\etocsetstyle{subsection}
{
  \etocskipfirstprefix
  {\appendchildtree\sectiontok\subsectiontok}
  {\preparetreenode \subsectiontok\expandafter{\treenode}}
  {\appendchildtree\sectiontok\subsectiontok}
}

\etocsettocstyle
{
  \treetok{\node {\hyperref[part:overview]{Overview}}}}
{
  \global\appendtotok\treetok{ ;}}

\begin{figure}[thbp!]\centering
  \etocsetnexttocdepth{subsection}
  \tableofcontents \label{toc:molecule} \ref{toc:overview}
  \hypersetup{hidelinks}%
  \begin{tikzpicture}
    [grow cyclic,
     level 1/.style={level distance=4cm,sibling angle=72},
     level 2/.style={level distance=2cm,sibling angle=60},
     every node/.style={ball color=red,fill=red,stroke=red,stroke width=1pt},
     edge from parent path={([dashed,very thick,color=cyan]
       (\tikzparentnode) -(\tikzchildnode))}]

    \the\treetok
  \end{tikzpicture}
\end{figure}

```

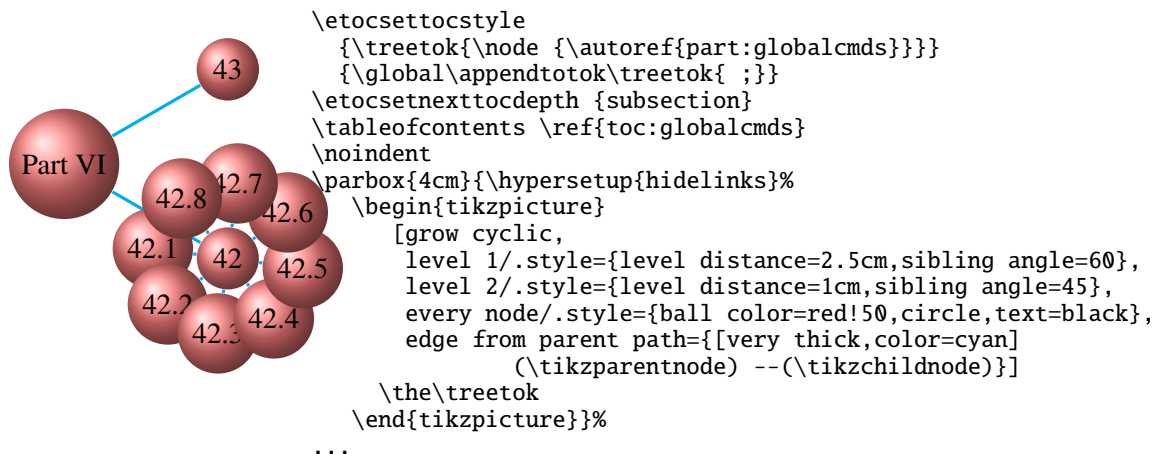


The `\tableofcontents` command appears just above the `tikzpicture` environment in a figure float (to make sure that the label of the table of contents refers to the same page as

the one where the picture will actually be printed). We thus get the table of contents as a “molecule”, which appears on the facing page.

This **TikZ TOC** is fully hyperlinked, like the previous **Forest TOC**.

On the side, the (fully hyperlinked) table of contents of **Part VI**.



## 28. The TOC as a TikZ mind map

This is in the same spirit as the “molecule” example. The use of the  $\varepsilon$ -TeX primitive `\unexpanded` will simplify the code.<sup>35</sup>

It is difficult to get everything to fit on one page. However `\resizebox` comes to the rescue. And it preserves hyperlinks. Nevertheless for this example I excluded some sections from the display, using the technique of the **etoc depth tags**. The fully hyperlinked TOC appears on the next page.

```

\begingroup
% \newtoks\treetok % done in preamble
% \newtoks\parttok

\newcommand*\partnode {} % just to check we don't overwrite something
\newcommand*\childnode {}

\newcommand*\tmprotate {} % just to check we don't overwrite something
\newcommand*\tmpoption {} % just to check we don't overwrite something
\newcommand*\tmpstuff {} % just to check we don't overwrite something

\newcommand*\appendtotok[2]{% #1=toks variable, #2=macro, expands once #2
  #1\expandafter\expandafter\expandafter{\expandafter\the\expandafter #1#2}}

\newcommand*\appendchildtree[3]{%
% this is to construct "t1 child [#3]{t2}" from #1=t1 and #2=t2
% t1 and t2 are two toks variable (not macros)
% #3 = for example teal!60
  \edef\tmpstuff {\the#1 child [#3]{\the#2}}%
  #1\expandafter {\tmpstuff }%

```

<sup>35</sup>The “molecule” example was added to this documentation on 2013/03/03. The “mindmap” example was motivated on 2015/03/11 by <http://tex.stackexchange.com/a/232584/4686>. Further help was then obtained via <http://tex.stackexchange.com/q/232816/4686> and this led to the [title page](#) which is a further example.

## 28. The TOC as a TikZ mind map



```

}

\newcounter{partco}

% 1,2,3,4,5,... -> 1,2,3,1,2,3,1,2,3
\def\pseudomodthree #1{\numexpr #1 + 3 - 3*((#1+1)/3)\relax}

\etocsetstyle{part}
{
  \etocskipfirstprefix
  % This updates the global tree with the data from the previous
  % part and all its children sections. Moved here because for some parts the
  % sections are not displayed due to depth tags.
  {\ifnum\value{partco}=3
    \appendchildtree\treetok\parttok {branch color= green!50,level distance=10cm}%
  \else
\ifcase\pseudomodthree{\value{partco}}%
  \or \appendchildtree\treetok\parttok {branch color= teal!60}% first
  \or \appendchildtree\treetok\parttok {branch color= yellow!80}% second
  \else\appendchildtree\treetok\parttok {branch color= green!50}% third and next ...
  \fi\fi
}
  {\stepcounter{partco}%
  % customize manually some TikZ set-up (should be done inside the TikZ thing I guess)
  \def\tmpoption {}%
  \def\tmprotate {}% first
  %\ifnum\value{partco}=5 \def\tmprotate {[counterclockwise from =-40]}\fi
  %\ifnum\value{partco}=8 \def\tmprotate {[counterclockwise from =-50]}\fi
  % define the part node
  \edef\partnode{node \tmpoption
    {\unexpanded\expandafter{\etocthelinkednumber}.
    \unexpanded\expandafter{\etocthelinkedname}}\tmprotate}%
  % this is a starting point which will be filled it by the section children
  \parttok\expandafter{\partnode}
  {\ifcase\pseudomodthree{\value{partco}}%
    \or \appendchildtree\treetok\parttok {branch color= teal!60}% first
    \or \appendchildtree\treetok\parttok {branch color= yellow!80}% second
    \else\appendchildtree\treetok\parttok {branch color= green!50}% third and next ...
    \fi
  }
}

\etocsetstyle{section}
{
  {}
  {}
  {% define the section node
  \edef\childnode{child {node {\unexpanded\expandafter{\etocthelinkednumber}
    \unexpanded\expandafter{\etocthelinkedname}}}%
  % append it to the current \parttok
  \appendtotok\parttok\childnode
  }
  {}
}

\etocsettocstyle
{
  {\setcounter{partco}{0}%
  \treetok{node [root concept]{\textbf{The \etoc documentation}}}}
  {\global\appendtotok\treetok{ ;}}
  % The \global above is mandatory because etoc always typesets TOC inside a group

```

## 28. The TOC as a TikZ mind map

```
\etocsetnexttocdepth{section}
% use of depth tags to cut out some sections.
\etocsettagdepth {preamble} {none}
\etocsettagdepth {overview} {part}
\etocsettagdepth {arbitrarily}{part}
\etocsettagdepth {examples} {section}
\etocsettagdepth {surprising} {part}
\etocsettagdepth {linestyles} {part}
\etocsettagdepth {globalcmds} {part}
\etocsettagdepth {custom} {section}
\etocsettagdepth {tips} {part}
\etocsettagdepth {etocandworld}{part}
\etocsettagdepth {code} {section}

\tikzset{
  branch color/.style={
    concept color=#1!white,
    every child/.append style={concept color=#1!white!30!white, font=\normalsize},
  }
}%

\begin{figure}[thp!]
\tableofcontents\label{toc:mindmap}%
\centeredline{\resizebox{.85\paperwidth}{!}%
{\begin{tikzpicture}[mindmap,
  grow cyclic,
  text width=2cm,
  align=flush center,
  nodes={concept},
  concept color=orange!60,
  root concept/.append style={text width=4cm, font=\Large},
  level 1/.append style={level distance=5cm,sibling angle=40, text width=3cm},
  level 2/.append style={level distance=7cm,sibling angle=30, text width=3cm},
  level 1 concept/.append style={font=\normalsize},
]}
\the\treetok
\end{tikzpicture}}}
\end{figure}
\endgroup
```

An interesting alternative is to use **etoc** rather to convert the entire TOC into a TikZ tree (perhaps excluding some parts) and print it out to a file from which it can be recovered and manipulated directly by the author of the document. Things written to the .log file get broken into lines. Here is a technique to get non-broken output. Once the `\treetok` has been computed by **etoc** (as in the **molecule** example, or the current example), this code snippet will write it out to file with extension `.toctree`:

```
\newwrite\TOCasTree
\immediate\openout\TOCasTree=\jobname.toctree
\immediate\write\TOCasTree{\the\treetok}%
```

The author can then copy it from there and customize it manually to get a suitable tikz picture. See also

<http://tex.stackexchange.com/a/232792>

for an elaboration of this.



## 29. The TOC as a table

With release 1.08 it is easier to typeset a TOC as a table. It is possible to open a tabular in the title part of the TOC (first argument to `\etocsettocstyle`) and then close it after the contents (second argument to `\etocsettocstyle`), and specify in the line styles how to use the tabulation & and tabular end of row `\\`. There are some conditions and a few caveats:

1. it is mandatory to issue `\etocglobaldefs` for `etoc`'s definitions to have global scope,
2. it is impossible to start one of the  $\langle start \rangle$ ,  $\langle prefix \rangle$ ,  $\langle contents \rangle$  or  $\langle finish \rangle$  specification with a sole `\hline`, *i.e.* one not preceded by a `\\` (it is however possible to put `\\` at the end of  $\langle prefix \rangle$  and the `\hline` at the start of  $\langle contents \rangle$ ).
3. as is explained next, it is recommended to put the `\\` at the start of the  $\langle prefix \rangle$  or  $\langle contents \rangle$  specifications in order to close the *previous* row, rather than at the end with the idea to close the *current* row; and when the TOC is a partial one (a `\localtableofcontents`) this is (in almost all situations) mandatory.

Here is an example of a TOC as a `longtable`, as is possible relatively simply now with 1.08. The code follows.

TABLE OF CONTENTS		
	Abstract	6
1	Foreword	6
2	License	7
I	<b>Overview</b>	8
3	Do I need to be a geek to use <code>etoc</code> ?	8
	3.1 <i>Limitations in the use of list environments for tables of contents</i>	8
4	Line styles and toc display style	9
	4.1 <i>\etocsetstyle for the line styles</i>	9
	4.2 <i>\etocsettocstyle for the toc display</i>	9
	4.3 <i>Compatibility mode</i>	10
5	A first example	10
6	A second example	12
7	A Beautiful Thesis example	13
8	Linked list of the main package commands	15
II	<b>Arbitrarily many TOCs, and local ones too</b>	16
9	Labeling and reusing elsewhere	16
10	A powerful functionality of <code>etoc</code> : the re-assignment of levels with <code>\etocsetlevel</code>	17
11	The <code>\etocsettocdepth</code> and <code>\etocsetnext-tocdepth</code> commands	18
	11.1 <i>The hyperref option</i> <code>bookmarksdepth</code>	19
12	The command <code>\etocsettocdepth.toc</code>	19

## 29. The TOC as a table

	12.1 <i>The commands <code>\etocobeytoc-tocdepth</code> and <code>\etocignoretoc-tocdepth</code></i>	20
<b>13</b>	The commands <code>\etocdepthtag.toc</code> and <code>\etocsettagdepth</code>	20
	13.1 <i>The commands <code>\etocobeydepth-tags</code> and <code>\etocignoredepthtags</code></i>	21
<b>14</b>	The commands <code>\etocglobaldefs</code> and <code>\etoclocaldefs</code>	21
<b>15</b>	Not displayed empty TOCs	21
	15.1 <i>The <code>\etocchecksempiness</code> command</i>	21
	15.2 <i>The <code>\etocnotocifnotoc</code> command</i>	22
	15.3 <i>The <code>\etocifwasempty</code> command</i>	22
<b>16</b>	Adding commands to the .toc file	22
	16.1 <i>The <code>hyperref</code> option <code>hidelinks</code></i>	23
<b>III</b>	<b>Examples</b>	24
<b>17</b>	Testing the compatibility mode	24
<b>18</b>	Another compatibility mode	25
<b>19</b>	Emulating the book class	27
<b>20</b>	A framed display	31
<b>21</b>	Another TOC with a background color	31
<b>22</b>	A (crazy) inline display	33
<b>IV</b>	<b>Surprising uses of <code>etoc</code></b>	36
<b>23</b>	The TOC of TOCs	36
<b>24</b>	Arbitrary “Lists Of...”, <code>\etoccontentline</code>	38
<b>25</b>	A TOC with a fancy layout	39
<b>26</b>	The TOC as a tree	40
<b>27</b>	The TOC as a molecule	43
<b>28</b>	The TOC as a TikZ mind map	45
<b>29</b>	The TOC as a table	49
<b>30</b>	A TOC self-adjusting widths for its typesetting	54
<b>V</b>	<b>Commands for the toc line styles</b>	57
<b>31</b>	The <code>\etocsetstyle</code> , <code>\etocname</code> and <code>\etocpage</code> commands	57
<b>32</b>	The <code>\etocskipfirstprefix</code> and <code>\etocif-first</code> commands	59
<b>33</b>	The <code>\etocnumber</code> command	59
<b>34</b>	The <code>\etocifnumbered</code> switch	59
<b>35</b>	The <code>\etocthename</code> , <code>\etocthenumber</code> , and <code>\etocthepage</code> commands	60
<b>36</b>	The <code>\etoclink</code> command	60

37	The <code>\etocthelinkedname</code> , <code>\etocthelinkednumber</code> , <code>\etocthelinkedpage</code> and <code>\etocthelink</code> commands	60
38	The <code>\etocsavedsectiontocline</code> , etc... commands.	61
39	The <code>\etocsetlevel</code> command	62
40	Scope of commands added to the <code>.toc</code> file	63
	40.1 <i>Testing the scope</i>	63
	40.2 <i>This is a (pale) red subsection for illustrative purposes</i>	63
41	Am I also red?	63
<b>VI</b>	<b>Commands for the toc display style</b>	64
42	Specifying the toc display style	64
	42.1 <i>The command <code>\etocsettocstyle</code></i>	64
	42.2 <i>The commands <code>\etocmulticolstyle</code>, <code>\etocmulticol</code>, and <code>\etoclocalmulticol</code></i>	65
	42.3 <i>The command <code>\etoc tocstyle</code></i>	65
	42.4 <i>The commands <code>\etocruledstyle</code>, <code>\etocruled</code> and <code>\etoclocalruled</code></i>	66
	42.5 <i>The commands <code>\etocframedstyle</code>, <code>\etocframed</code>, and <code>\etoclocalframed</code></i>	66
	42.6 <i>Headings, titles, <code>\etocoldpar</code>, <code>\etocinnertopsep</code></i>	66
	42.7 <i>The compatibility mode <code>\etocstandarddisplaystyle</code></i>	67
	42.8 <i>The command <code>\etocinline</code></i>	67
43	Starred variants of the <code>\tableofcontents</code> etc... commands	67
<b>VII</b>	<b>Using and customizing the <code>etoc</code> own styles</b>	69
44	Summary of the main style commands	69
	44.1 <i>Setting up local styles</i>	69
	44.2 <i>Setting up toc display styles</i>	70
	44.3 <i>Displaying tables of contents</i>	70
	44.4 <i>Labels and references</i>	70
45	The package default line styles: <code>\etocdefaultlines</code>	70
46	Customizing <code>etoc</code>	75
	46.1 <i>Customizing the <code>etoc</code> pre-defined line styles</i>	75
	46.2 <i>Customizing the toc display styles</i>	76
47	One more example of colored TOC layout	77
<b>VIII</b>	<b>Tips</b>	80
48	The <code>\etocsetlocaltop.toc</code> command	80
49	Hacking framed parboxes	81

## 29. The TOC as a table

<b>50</b>	Interverting the levels	82
	50.1 <i>All subsections of this document</i>	82
<b>51</b>	Displaying statistics	83
<b>52</b>	Using depth tags	85
<b>53</b>	Typesetting the TOC as a table (the old way)	87
<b>IX</b>	<b>etoc and the outside world</b>	89
<b>54</b>	Compatibility with other packages	89
	54.1 <i>Generalities</i>	89
	54.2 <i>Compatibility with beamer</i>	90
	54.3 <i>Compatibility with Babel</i>	90
	54.4 <i>Compatibility with hyperref</i>	90
	54.5 <i>Compatibility with multicol</i>	90
	54.6 <i>Compatibility with tocloft</i>	90
	54.7 <i>Compatibility with the memoir class</i>	91
	54.8 <i>Compatibility with package tocvsec2</i>	91
	54.9 <i>Compatibility with package tableof</i>	91
	54.10 <i>Compatibility with package tocstyle</i>	91
<b>55</b>	T <sub>E</sub> Xnical matters	91
<b>56</b>	Errors and catastrophes	92
<b>X</b>	<b>The code</b>	93
<b>57</b>	Timestamp	93
<b>58</b>	Change history	93
<b>59</b>	Implementation	96

```

\begingroup
\etocglobaldefs % necessary for \etocname etc... to survive &

% observe the locations of the \\
\etocsetstyle{part}
{
  {}
  {\hline}
  {\strut\etocnumber &\bfseries\etocname&\etocpage }
  {}
}

\etocsetstyle{section}
{
  {}
  {\etociffirst{\etocifnumbered{\hline}{\hline\hline}}{\}}
  {\etocnumber&\etocname &\etocpage }
  {}
}

\etocsetstyle{subsection}
{
  {}
  {}
  {\makebox[1cm][c]{\etocnumber}%
   \parbox[t]{\dimexpr6cm-\tabcolsep\relax}{\sloppy\itshape\etocname\strut}%
   &\itshape\etocpage }
  {}
}

\etocsettocstyle

```

```

{\hypersetup{hidelinks}%
 \begin{longtable}{|>{\bfseries}c|p{7cm}|r|}
 \hline
 \multicolumn{3}{|c|}{\Large\bfseries\strut\strut TABLE OF CONTENTS}%
 }
 {\hline\end{longtable}}

\etocsetnexttocdepth {subsection}

\tableofcontents
\endgroup

```

Examining the code above the reader will wonder why the `\hline` are always given first in  $\langle prefix+contents \rangle$  and not, as is more intuitive, rather last. In some favorable cases (but almost never for local tables of contents) one may indeed construct TOC-as-tables with the `\hline` located at the end of the style specifications. The problem in the previous example was with the positioning of the `\hline`'s.

Due to technical aspects of how  $\text{\TeX}$  constructs alignments any definition or assignment done after an `\hline` starts a new row, and thus makes `\hline` an illegal token (this shows as a misplaced `\noalign` error.) Not only does `etoc` have to do such definitions to construct `\etocname` etc..., it is furthermore the case that some packages put things in the `.toc` file and as a result there is never any guarantee that between two `\contentsline` there will not be such a token like `\relax` which in the contexts of alignments forces  $\text{\TeX}$  to start a cell and thus makes it impossible then to insert an `\hline`.

The safest way is thus to start with an `\hline` each line style specification in order to close the *previous* table row. We had a little problem with the fact that we wanted parts not only to have a rule above them (easy, they do `\hline`) but also below them: after each part there is a section, and it is these sections which are used to insert the missing `\hline` (this is done with the help of the `\etociffirst` conditional).

The attentive reader will notice one last subtlety: the title was supposed to be followed not by one, but by two `\hline`'s. To solve this we could have used a one-time macro redefining itself, but we noticed that the title was followed by the unnumbered Abstract, thus the `\etocifnumbered` switch came to the rescue.

Last technical note: because we put the `\hline` inside the branches, there was no need to employ the expandable variants `\etocxiffirst` and `\etocxifnumbered`.

For the hardliner's old way of obtaining the exact same result, see [section 53](#).

Here is also a much simpler example. It is a local table of contents.

### 30. A TOC self-adjusting widths for its typesetting

Section	number	page
Do I need to be a geek to use <b>etoc</b> ?	3	8
Line styles and toc display style	4	9
A first example	5	10
A second example	6	12
A Beautiful Thesis example	7	13
Linked list of the main package commands	8	15

```

\begin{center}
\etocsetstyle{section}
{
  {}
  {\etociffirst{\\hline\hline}{\\hline}}
  {\etocname & \etocnumber & \etocpage }
  {}
}

\etocsettocstyle
{
  \hypersetup{hidelinks}\begin{tabular}{|p{4cm}|c|c|}\hline
  \multicolumn{1}{|c|}{\bfseries Section}&
  \bfseries number&
  \bfseries page}
  {\\hline\end{tabular}}

\etocglobaldefs % MANDATORY !!
\etocsetnexttocdepth{1}

\tableofcontents\ref{toc:overview}
\end{center}

```

### 30. A TOC self-adjusting widths for its typesetting

This is a continuation of [section 19](#). The goal is to adjust automatically the “numwidths” used for typesetting the unit numbers in the (local) tables of contents.

```

\makeatletter
\newcommand*{\TOCcompute@numwidths [2]{% #1=empty/"local", #2=minimal indent
  \begingroup
    \def\TOCnumwidthB {0pt}%
    \def\TOCnumwidthC {0pt}%
    \def\TOCnumwidthD {0pt}%
    \def\TOCnumwidthE {0pt}%
    \def\TOCnumwidthF {0pt}%
    \def\TOCnumwidthG {0pt}%
    \etocsetstyle{part}{}{}{}{}%
    \etocsetstyle{chapter}{}
      {\setbox0\hbox{\bfseries\etocthenumber\kern#2}}
      {\ifdim\wd0>\TOCnumwidthB\edef\TOCnumwidthB{\the\wd0}\fi}}%

```

```

\etocsetstyle{section}{}
  {\setbox0\hbox{\bfseries\etocthenumber\kern#2}}
  {\ifdim\wd0>\TOCnumwidthC\edef\TOCnumwidthC{\the\wd0}\fi}{}%
\etocsetstyle{subsection}{}
  {\setbox0\hbox{\etocthenumber\kern#2}}
  {\ifdim\wd0>\TOCnumwidthD\edef\TOCnumwidthD{\the\wd0}\fi}{}%
\etocsetstyle{subsubsection}{}
  {\setbox0\hbox{\etocthenumber\kern#2}}
  {\ifdim\wd0>\TOCnumwidthE\edef\TOCnumwidthE{\the\wd0}\fi}{}%
\etocsetstyle{paragraph}{}
  {\setbox0\hbox{\etocthenumber\kern#2}}
  {\ifdim\wd0>\TOCnumwidthF\edef\TOCnumwidthF{\the\wd0}\fi}{}%
\etocsetstyle{subparagraph}{}
  {\setbox0\hbox{\etocthenumber\kern#2}}
  {\ifdim\wd0>\TOCnumwidthG\edef\TOCnumwidthG{\the\wd0}\fi}{}%
%
\etocsettocstyle{}
  {\global\let\TOCnumwidthB\TOCnumwidthB
  \global\let\TOCnumwidthC\TOCnumwidthC
  \global\let\TOCnumwidthD\TOCnumwidthD
  \global\let\TOCnumwidthE\TOCnumwidthE
  \global\let\TOCnumwidthF\TOCnumwidthF
  \global\let\TOCnumwidthG\TOCnumwidthG}%
\etocnepar
\csname #1tableofcontents\endcsname
\typeout{Next TOCs will use \TOCnumwidthB\space for chapter number width}%
\typeout{Next TOCs will use \TOCnumwidthC\space for section number width}%
\typeout{Next TOCs will use \TOCnumwidthD\space for subsection number width}%
\typeout{Next TOCs will use \TOCnumwidthE\space for subsubsection number width}%
\typeout{Next TOCs will use \TOCnumwidthF\space for paragraph number width}%
\typeout{Next TOCs will use \TOCnumwidthG\space for subparagraph number width}%
\endgroup
}%
\newcommand*\TOCcomputenumwidths [1][0.5em]{%
  \TOCcompute@numwidths {}{#1}%
}%
\newcommand*\TOCcomputelocalnumwidths [1][0.5em]{%
  \TOCcompute@numwidths {local}{#1}%
}%
\makeatother

```

The optional parameter to `\TOCcomputenumwidths` specifies the minimal indent. In case nothing is numbered you may wish a higher value than 0.5em. For each local table of contents to have its own width computations, the macro `\TOCcomputelocalnumwidths` is provided. As the code makes global assignments, either use (once) `\TOCcomputenumwidths` or do `\TOCcomputelocalnumwidths` for each local table of contents.

```

\TOCcomputelocalnumwidths % may use optional argument to replace 0.5em
\localtableofcontents

```

Notes:

1. naturally these are only suggestions. For example one could put everything in single macros `\TOCtoc` and `\TOClocaltoc` to simultaneously compute the numwidths and then typeset the (local) table of contents.

### 30. A TOC self-adjusting widths for its typesetting

2. if you want to adjust the `tocdepth` recall from [subsection 11.1](#) that it influences `hyperref` hence you may need to use a group `\begingroup . . . \endgroup`. Or, one can use `\etocsetnexttocdepth{<level>}` but (with the code as here) this must then be issued twice, once for `\TOCcomputelocalnumwidths`, once for `\localtableofcontents`.
3. the bold font serves above for both chapter and section `numwidth` computations, but the code from [section 19](#) uses `\bfseries` only in local TOCs. Thus the `\TOCcomputenumwidth` will set the parameter `\TOCnumwidthC` to a value slightly larger than needed in the main TOC. Hence the section style in `\TOCcompute@numwidths` should possibly insert the `\bfseries` in the box only after testing for the optional parameter `local`.



## Part V.

# Commands for the toc line styles

Here are some statistics for this part: it contains 11 sections and 2 subsections. The name of the first section is “The `\etocsetstyle`, `\etocname` and `\etocpage` commands” and the corresponding number is “31”. The name of the last section is “**Am I also red?**” and its number is “**41**”. The name of the first subsection is “Testing the scope” and the corresponding number is “40.1”. The name of the last subsection is “**This is a (pale) red subsection for illustrative purposes**” and its number is “**40.2**”.

### Contents of Part V

- 31** The `\etocsetstyle`, `\etocname` and `\etocpage` commands (page 57)
- 32** The `\etocskipfirstprefix` and `\etociffirst` commands (page 59)
- 33** The `\etocnumber` command (page 59)
- 34** The `\etocifnumbered` switch (page 59)
- 35** The `\etocthenname`, `\etocthenumber`, and `\etocthepage` commands (page 60)
- 36** The `\etoclink` command (page 60)
- 37** The `\etocthelinkedname`, `\etocthelinkednumber`, `\etocthelinkedpage` and `\etocthelink` commands (page 60)
- 38** The `\etocsavedsectiontocline`, etc... commands. (page 61)
- 39** The `\etocsetlevel` command (page 62)
- 40** Scope of commands added to the `.toc` file (page 63)
  - 40.1 Testing the scope (p. 63)
  - 40.2 **This is a (pale) red subsection for illustrative purposes** (p. 63)
- 41** **Am I also red?** (page 63)

## 31. The `\etocsetstyle`, `\etocname` and `\etocpage` commands

Let us explain how **etoc** was used to produce the table of contents displayed at the beginning of this [Part V](#). This is a local table of contents, and we used the command `\local-tableofcontents`.

### 31. The `\etocsetstyle`, `\etocname` and `\etocpage` commands

We shall distinguish between the *line styles* and the *toc display style*. The line styles were (essentially) obtained in the following manner:<sup>36</sup>

```
\etocsetstyle{section}
{\begin{enumerate}}
{\normalsize\bfseries\rmfamily\item}
{\etocname{} (page \etocpage)}
{\end{enumerate}}

\etocsetstyle{subsection}
{\begin{enumerate}}
{\normalfont\item}
{\etocname{} (p.\~\etocpage)}
{\end{enumerate}}

\etocsetstyle{subsubsection}
{\par\nobreak\beginngroup\normalfont
\footnotesize\itshape\etocskipfirstprefix}
{\allowbreak\-, -,}
{\etocname}
{.\hfil\par\endgroup\pagebreak[3]}
```

These provisory style definitions rely on the automatic numbering generated by the `enumerate` environments but it is much better to use the further command `\etocnumber` inside the item label, which gives the real thing. The improved definitions will thus be explained later.

With this style, one would have to be imaginative to design something then for paragraph and subparagraph entries! perhaps as superscripts? Well, usually one does not need paragraphs and subparagraphs numbered and listed in the TOC, so our putative user here chose a design where no provision is made for them and added the definitive:

```
\etocsetstyle{paragraph}{}{}{}{}
\etocsetstyle{subparagraph}{}{}{}{}{}

```

This is also the situation with the default package line styles!

Each `\etocsetstyle` command has five mandatory arguments:

```
\etocsetstyle{<levelname>}{<start>}{<prefix>}{<contents>}{<finish>}
```

The initially recognized `<levelname>`'s are the sectioning levels of the standard document classes: from *part* (or *book* which is used by the memoir class) down to *subparagraph*.

The `<start>` code is executed when a toc entry of that level is encountered and the previous one was at a higher level. The `<finish>` code is executed when one again encounters a higher level toc entry. In the mean-time all entries for that level are typeset by executing first the `<prefix>` code and then the `<contents>` code.

The (robust) commands `\etocname`, `\etocnumber` and `\etocpage` are provided for use inside the `<prefix>` and `<contents>` parts of the `\etocsetstyle` specification. They represent of course, the name, number, and page number of the corresponding toc entry. If package `hyperref` is active in the document and has added hyperlinks to the TOC data, then these links are kept in the commands `\etocname`, `\etocnumber` and `\etocpage` (this last one will have a link only if `hyperref` was passed option `linktoc=all`.)

<sup>36</sup>the present document has `\renewcommand{\familydefault}{\sfdefault}` in its preamble, hence `\normalfont` switches to the sans typeface; so in the section line-style, I wrote `\rmfamily` instead.

## 32. The `\etocskipfirstprefix` and `\etociffirst` commands

The chosen subsection style made use of the command `\etocskipfirstprefix`, which instructs `etoc` to *not* use for the first item the specified *<prefix>* code.<sup>37</sup>

The command `\etociffirst{<YES CODE>}{<NO CODE>}` (new with 1.08) is a more flexible way to customize the *<prefix>* (and *<contents>*) specifications. It executes the *<YES CODE>* branch if this is the first unit at that level (inside a lower level) and the *<NO CODE>* if not. This is a robust command which survives to expansion (for example in an `enumitem` label).

The variant `\etocxiffirst` does the same, but is expandable.

## 33. The `\etocnumber` command

So far, our specifications would use the numbering generated by the `enumerate` environments, but of course we generally want the actual numbers as found in the `.toc` file. This is available via the `\etocnumber` command. To get the labels in the `enumerate` list to use it we can proceed with the syntax `label=` from the package `enumitem`:

```
\etocsetstyle{section}
{\begin{enumerate}[label=\etocnumber]}
{\normalsize\bfseries\rmfamily\item}
{\etocname{} (page \etocpage)}
{\end{enumerate}}
```

Rather than just `\etocnumber` we then used something like `\fbox{\etocnumber}`. Note that `\etocnumber` is a robust command which explains why it can be used inside the label specification without needing an added `\protect`.

## 34. The `\etocifnumbered` switch

The `\fbox` would give an unaesthetic result in the case of an unnumbered section (which ended up in the table of contents via an `\addcontentsline` command).<sup>38</sup>

The `\etocifnumbered{<A>}{<B>}` command executes *<A>* if the number exists, and *<B>* if not. So we use it in the code which was finally chosen for the section level:

```
\etocsetstyle{section}
{\begin{enumerate}[leftmargin=.75cm, label=\etocifnumbered
  {\fboxrule1pt\fcolorbox{green}{white}{\etocnumber}}}]
{\normalsize\bfseries\rmfamily\item}
{\etocname{} (page \etocpage)}
{\end{enumerate}}
```

```
\etocsetstyle{subsection}
{\begin{enumerate}[leftmargin=0cm, label=\etocnumber]}
{\normalfont \item}
```

---

<sup>37</sup>With versions earlier than 1.08 it had to be the very last token in the *<start>* code. It may now appear anywhere therein.

<sup>38</sup>as seen we use `\fcolorbox` rather than `\fbox`. Due to some redefinition made by package `xcolor`, had we used `\fbox` (and not used `hyperref`) we would have needed `\protect\fbox`.

### 37. The `\etoclinkedname`, `\etoclinkednumber`, `\etoclinkedpage` and `\etocthelink` commands

```
{\etocname{}} (p.\etocpage)}  
{\end{enumerate}}
```

If we had changed only the section level, and not the subsection level, an error on compilation would have occurred because the package style for subsections expects to start ‘in vertical mode’. An additional `\par` token in the `<contents>` part of the section level would have fixed this: `{... (page \etocpage)\par}`.

The command `\etocifnumbered` is robust; `\etocxifnumbered` (new with 1.08) has the same effect but is expandable.

## 35. The `\etocthename`, `\etocthenumber`, and `\etocthepage` commands

It is sometimes desirable to have access to the name, number and page number without the `hyperref` link data: something similar to the starred variant of the `\ref` command, when package `hyperref` is used. For example one may wish to use the unit or page number in some kind of numeric context, or change its formatting. This is provided by `\etocthename`, `\etocthenumber`, and `\etocthepage`.

These commands are not protected, so in moving argument contexts (for example in a label specification) they should be preceded by `\protect`.

## 36. The `\etoclink` command

The command `\etoclink{<linkname>}` can be used in the line style specifications in a manner analogous to the argument-less commands `\etocname`, `\etocnumber` and `\etocpage`. It creates a link (if such a link was added by `hyperref` to the `.toc` file entry) whose destination is the corresponding document unit and whose name is the given argument. Hence `\etoclink{\etocthename}` is like the original `\etocname`. Notice that if `hyperref` was not instructed to put a link in the page number (via its option `linktoc=all`) then `etoc`’s `\etocpage` contains no link either, but `\etoclink{\etocthepage}` does.

The command `\etoclink` is robust. Since `etoc 1.08j` it contains the link destination in an already expanded form, so for example can be used even after a `&` in a tabular construction, if `\etocglobaldefs` was issued.

## 37. The `\etocthelinkedname`, `\etocthelinkednumber`, `\etocthelinkedpage` and `\etocthelink` commands

These macros are fragile and contain the link destination in already expanded form. So their meaning can be stored for delayed usage. For example this is done here when doing the [examples with trees](#).

Notice though that for example `\etocthelinkednumber` is really a link only if `hyperref` was configured to let the numbered heading be linked (which is the default). Similarly for the others.

`\etocthelink` will fetch the link destination if available from name or page number and allows via `\etocthelink{foo}` to link an arbitrary text to the same destination inside the produced PDF.

All these commands are fragile. The `\etocname`, `\etocnumber`, `\etocpage` and `\etoclink` are simply their robust variants. They too contain the hyperlink destination in already expanded form since 1.08a for the first three, since 1.08j for `\etoclink`.

Obsolete example of a table of contents done as an inline `enumitem` environment: **42**. Specifying the toc display style (p. 64), and **43**. Starred variants of the `\tableofcontents` etc... commands (p. 67).

```

Obsolete example of \begingroup
\etocsetstyle {section}
  {\begin{itemize*}[itemjoin={{; }}, itemjoin*={{, and }}}
  {\global\let\TmpEtocNumber\etocthelinkednumber
  \global\let\TmpEtocName\etocthelinkedname
  \global\let\TmpEtocPage\etocthelinkedpage }
% (the above needed as \item closes a group in enumitem inline environments)
  {\item [{\bfseries\TmpEtocNumber.}]\TmpEtocName\
    (\emph{p. \TmpEtocPage})}
  {\end{itemize*}.}%
\etocsetnexttocdepth {section}%
\etocsettocstyle {a table of contents done as an inline
                  \texttt{enumitem} environment: }{}%
\etocinline\tableofcontents \ref{toc:globalcmds}
\endgroup

```

**etoc** 1.08a offers a simpler way to the same result: thanks to `\etocglobaldefs` there is no need anymore here for `\etocthelinkedname`, as `\etocname` works.

```

Example of
\begingroup\etocglobaldefs % <-- NEW mit etoc 1.08.
\etocsetstyle {section}
  {\begin{itemize*}[itemjoin={{; }}, itemjoin*={{, und }}}
  {}
  {\item [{\bfseries\etocnumber.}] \etocname\ (\emph{p. \etocpage })}
  {\end{itemize*}.}%
\etocsetnexttocdepth {section}%
\etocsettocstyle {a table of contents done as an inline
                  \texttt{enumitem} environment: }{}%
\etocinline\tableofcontents \ref{toc:globalcmds}
\endgroup

```

Example of a table of contents done as an inline `enumitem` environment: **42**. Specifying the toc display style (p. 64), und **43**. Starred variants of the `\tableofcontents` etc... commands (p. 67).

### 38. The `\etocsavedsectiontocline`, etc... commands.

Since 1.08k **etoc** preserves automatically in `\etocsavedsectiontocline`, `\etocsavedchaptertocline` etc... the meaning of the  $\LaTeX$  `\l@section`, `\l@chapter`, etc... in force at the time each `\tableofcontents` or `\localtableofcontents` macro is executed.

See [section 18](#) for usage.

## 39. The `\etocsetlevel` command

As already explained in [Part IV](#), one can inform **etoc** of a level to associate to a given sectioning command with `\etocsetlevel`. For example:

```
\etocsetlevel{cell}{0}
\etocsetlevel{molecule}{1}
\etocsetlevel{atom}{2}
\etocsetlevel{nucleus}{3}
```

In compatibility mode, it will be assumed that the macros `\l@cell`, `\l@molecule`, ..., have been defined somewhere either by the user or a class: doing only `\etocsetlevel` is not enough for the corresponding level to work out-of-the-box in compatibility mode.

However, if no table of contents is typeset in compatibility mode, then all that matters is that the various line styles have been set. If, for example section is at level 1, then there is no need to do some `\etocsetstyle{molecule}{...}{...}{...}{...}` after `\etocsetlevel{molecule}{1}` if `\etocsetstyle{section}{...}{...}{...}{...}` has already been done (and it has been done by the package itself in its definition of its own line styles).

The accepted levels run from -2 to 6 inclusive. Anything else is mapped to 6, which is a dummy level, never displayed. The package does:

```
\etocsetlevel{book}{-2}
\etocsetlevel{part}{-1}
\etocsetlevel{chapter}{0}
\etocsetlevel{section}{1}
\etocsetlevel{subsection}{2}
\etocsetlevel{subsubsection}{3}
\etocsetlevel{paragraph}{4}
\etocsetlevel{subparagraph}{5}
```

**etoc** own custom styles are activated by `\etocdefaultlines`. They are illustrated by the main table of contents of this document.

These level assignments can be modified at anytime: see [Part IV](#) for various applications of this technique. As one further example, let's mention here that the [main table of contents](#) of this document was typeset following these instructions:

```
\etocsettocdepth {subsubsection} % set the initial tocdepth
\etocdefaultlines % use the package default line styles. At this early stage in
                  % the document they had not yet been modified by \etocsetstyle
                  % commands, so \etoclines could have been used, too.
\etocmarkboth\contentsname
\etocmulticolstyle[1] % one-column display
  {\pdfbookmark[1]{Table of contents}{MAINTOC}% create a bookmark in the pdf
   \noindent\bfseries\Large
   \leaders\hrule height1pt\hfill
   \MakeUppercase{Table of Contents}}
\begingroup % use a group to limit the scope of the
  \etocsetlevel{subsection}{3} % subsection level change.
  \etocsetlevel{subsubsection}{4} % anything > tocdepth=3.
  \tableofcontents \label{toc:main}
\endgroup
```

In this way, the subsections used the style originally designed for subsubsections, the subsubsections were not printed. Without this modification, the appearance would have been very different: the package line styles were targeted to be employed in documents with many many sub-sub-sections, in a two-column layout, giving thus a more compact output that what is achieved by the default L<sup>A</sup>T<sub>E</sub>X table of contents. But here, we have few sub-sub-sections and it is more interesting to drop them and print in a visually different manner sections and subsections.

## 40. Scope of commands added to the .toc file

### 40.1. Testing the scope

Let us switch to the color red, and also add this command to the .toc file:

```
\color{red!50} % changing text color
\addtocontents{toc}{\string\color{red!50}} % and also in the .toc file
```

### 40.2. This is a (pale) red subsection for illustrative purposes

Actually, this title here was printed black, due to the way the `scrartcl` class works (it would have been red in the `article` class), but we are more interested in how it looks in the tables of contents: it does appear red in the [main table of contents](#) at the beginning of this document, and also in the [table of contents for this part](#). Both entries obey as expected the `\color{red!50}` command inserted in the .toc file.

But let us now close this subsection and start a section.

## 41. Am I also red?

The question is about how it appears in the tables of contents: the answer is that, yes it is red in the [main TOC](#), and no it is not red in the [local TOC for this part](#). The reason is that the `<finish>` code for the subsection level closed a group, as it used `\end{enumerate}`.

This illustrates the discussion from [subsection 3.1](#).

The default package line styles do not contain group opening and closing instructions: the influence of a command added to the .toc file will propagate until cancelled by another explicit such command inserted in the .toc file.

```
\normalcolor
\addtocontents{toc}{\string\normalcolor}
```

Back to black. Note that this scope problem arises in real life in a multi-lingual document, as the `babel` package writes to the .toc file the language changes occurring in the document.

## Part VI.

# Commands for the toc display style

Here are some statistics for this part: it contains 2 sections and 8 subsections. The name of the first section is “Specifying the toc display style” and the corresponding number is “42”. The name of the last section is “Starred variants of the `\tableofcontents` etc... commands” and its number is “43”. The name of the first subsection is “The command `\etocsettocstyle`” and the corresponding number is “42.1”. The name of the last subsection is “The command `\etocinline`” and its number is “42.8”.

<b>42. Specifying the toc display style</b>	<b>64</b>
42.1. The command <code>\etocsettocstyle</code> . . . . .	64
42.2. The commands <code>\etocmulticolstyle</code> , <code>\etocmulticol</code> , and <code>\etoclocalmulticol</code> . . . . .	65
42.3. The command <code>\etoclocaltocstyle</code> . . . . .	65
42.4. The commands <code>\etocruledstyle</code> , <code>\etocruled</code> and <code>\etoclocalruled</code> . .	66
42.5. The commands <code>\etocframedstyle</code> , <code>\etocframed</code> , and <code>\etoclocalframed</code>	66
42.6. Headings, titles, <code>\etocoldpar</code> , <code>\etocinnertopsep</code> . . . . .	66
42.7. The compatibility mode <code>\etocstandarddisplaystyle</code> . . . . .	67
42.8. The command <code>\etocinline</code> . . . . .	67
<b>43. Starred variants of the <code>\tableofcontents</code> etc... commands</b>	<b>67</b>

## 42. Specifying the toc display style

The *toc display* style says whether the TOC appears with multiple columns or just one, whether the title is typeset as in the article or book class, or should be centered above the entries, with rules on its sides, or if the entire TOC should be put in a frame. For example, to opt for a ruled heading and single column layout, one issues commands of the following type:

```
\etocruledstyle[1]{Title} \tableofcontents      (or \localtableofcontents)
shortcuts:      \etocruled[1]{Title} (or \etoclocalruled[1]{Title})
```

### 42.1. The command `\etocsettocstyle`

This is a command with two mandatory arguments:

```
\etocsettocstyle{\<before_toc>}{\<after_toc>}
```

The `{\<before_toc>}` part is responsible for typesetting the heading, for example it can be something like `\section*{\contentsname}`.

Generally speaking this heading should leave  $\TeX$  in “vertical mode” when the actual typesetting of the contents will start: the line styles (either from the standard classes or the package default line styles) all expect to get started in ‘vertical mode’.



## 42.2. The commands `\etocmulticolstyle`, `\etocmulticol`, and `\etoclocalmulticol`

The first argument to `\etocsettocstyle` can also contain instructions to mark the page headings. Or it could check (book class) to see if two-column mode is on, and switch to one-column style, and the `\after_toc` part would then reenact the two-column mode.

The commands to be described next `\etocmulticolstyle`, `\etocruledstyle`, and `\etocframedstyle` all call `\etocsettocstyle` as a lower-level routine, to start a multi-cols environment in `\{<before_toc>\}` and close it in `\{<after_toc>\}`.

### 42.2. The commands `\etocmulticolstyle`, `\etocmulticol`, and `\etoclocalmulticol`

This is a command with one optional and one mandatory argument:

`\etocmulticolstyle[<number_of_columns>]{<heading>}`

The `<number_of_columns>` can go from 1 to 10 (it defaults to 2; if its value is 1, naturally no multicol environment is then created). The `<heading>` will typically be some ‘vertical’ material like: `<heading> = \section*{<title>}` but one may also have horizontal material like `\fbox{Hello World}` (`\etoc` adds automatically a `\par` at the end of this “heading” argument to `\etocmulticolstyle`). Here is for example how the main table of contents of this document was configured:

```
\etocmulticolstyle{\noindent\bfseries\Large
\leaders\hrule height1pt\hfill
\MakeUppercase{Table of Contents}}
```

After `\etocmulticolstyle` all future `\tableofcontents` will use the specified style, if it does not get changed in-between. A shortcut for just one table of contents and no impact on the styles of later TOCs is:

`\etocmulticol[<number_of_columns>]{<heading>}`

And there is also `\etoclocalmulticol[<number_of_columns>]{<heading>}`.

### 42.3. The command `\etocstyle`

`\etocstyle[<kind>]{<number_of_columns>}{<title>}`  
`= \etocmulticolstyle[<number_of_columns>]{\kind*{<title>}}`

where `kind` is one of chapter, section, ... and defaults to chapter or section depending on the document class.

#### 42.3.1. The command `\etocstylewithmarks`

`\etocstylewithmarks[<kind>]{<number_of_columns>}{<title>}{<mark>}`  
`= \etocmulticolstyle[<number_of_columns>]{\kind*{<title> \markboth{\MakeUppercase{<mark>}}}`

where `kind` is one of chapter, section, ... The actual display of the marks depends on the settings of the page style. There is variant `\etocstylewithmarksnouc` which does not uppercase.

##### 42.3.1.1. Do we really want paragraph entries in the TOC?

##### 42.3.1.2. really?

#### 42.4. The commands `\etocruledstyle`, `\etocruled` and `\etoclocalruled`

The general format of `\etocruledstyle` is:

`\etocruledstyle`[*<number of columns>*]{*<title of the toc>*}

The title is horizontal material (the LR mode of *L<sup>A</sup>T<sub>E</sub>X*, a document preparation system): if it does not fit on one line it should be put in a `\parbox` of a given width. The green frame for the heading of the table of contents at the [start of the second part of this document](#) was obtained with:

```
\etocruledstyle[1]{\etocfontminusone\color{green}%
  \fboxrule1pt\fboxseplex
  \framebox[\linewidth]
    {\normalcolor\hss Contents of this part\hss}}
```

As a shortcut to set the style with `\etocruledstyle` and then issue a `\tableofcontents`, all inside a group so that future table of contents will not be affected, there is:

`\etocruled`[*<number\_of\_columns>*]{*<title>*}

And the local form will be `\etoclocalruled`.

#### 42.5. The commands `\etocframedstyle`, `\etocframed`, and `\etoclocalframed`

Same mechanism:

`\etocframedstyle`[*<number\_of\_columns>*]{*<title>*}

and the accompanying shortcut:

`\etocframed`[*<number\_of\_columns>*]{*<title>*}

The shortcut is used if one does not want to modify the style of the next TOCs (the other way is to put the whole thing inside braces or a `\begingroup... \endgroup`; there is also `\etoclocalframed` for a local table of contents).

The entire table of contents is framed. The title itself is not framed: if one wants a frame one should set it up inside the *<title>* argument to `\etocframedstyle` or `\etocframed`. The colors for the background and for the components (top, left, right, bottom) of the border are specified via suitable `\renewcommand`'s (see [subsection 46.2](#)).

A minipage is used, hence the produced table of contents isn't compatible with a page break. For allowing page breaks, use of the commands of `mdframed`<sup>39</sup> or `tcolorbox`<sup>40</sup> in the arguments of `\etocsettocstyle` is recommended.

Examples in this document are on pages [17](#), [34](#), [32](#), and [78](#).

#### 42.6. Headings, titles, `\etocoldpar`, `\etocinnertopsep`

For `\etocmulticolstyle` the mandatory *<heading>* argument can be either vertical mode material like `\section*{\emph{Table of Contents}}` or horizontal mode material like in the simple `\etocmulticolstyle{Hello World}`.

No explicit `\par` or empty line can be inserted in the mandatory argument of `\etocmulticolstyle`, but `etoc` provides `\etocoldpar` as a substitute: it does `\let\etocoldpar\par`

<sup>39</sup><http://ctan.org/pkg/mdframed>

<sup>40</sup><http://ctan.org/pkg/tcolorbox>

before the `\multicols` environment and inserts this `\etocoldpar`<sup>41</sup> at the end of the heading, then does a vertical skip of value `\etocinnertopsep`. The command `\etocoldpar` can also be used explicitly if needed in the mandatory argument to `\etocmulticolstyle` (it is not allowed to insert an empty line in this argument).

On the other hand the commands `\etocruledstyle` and `\etocframedstyle` expect an argument “in LR mode” (to use the terminology from *LaTeX, a document preparation system*). This means that multiline titles are only possible if enclosing them inside something like a `\parbox`.

An important dimension used by all three of `\etocmulticolstyle`, `\etocruledstyle` and `\etocframedstyle` is `\etocinnertopsep`. It gives the amount of separation between the heading and the start of the contents. Its default value is `2ex` and it is changed with `\renewcommand*{\etocinnertopsep}{\langle new\_value \rangle}`, not with `\setlength`.

## 42.7. The compatibility mode `\etocstandarddisplaystyle`

`etoc` will then emulate what the document class would have done regarding the global display style of the table of contents, in its absence. All customizing from inside the class should be obeyed, too.

### 42.7.1. The commands `\etocarticlestyle`, `\etocbookstyle`, ...

These are the commands used internally by `etoc` in compatibility mode depending on the document class. For example `\etocarticlestyle` instructs `etoc` to use `\section*{\contentsname}` (with marks on the page) and `\etocbookstyle` says to use `\chapter*{\contentsname}`. It can prove useful to issue `\etocarticlestyle` for a `\localtableofcontents` inside a chapter, in book class and compatibility mode for the global TOC display style.

## 42.8. The command `\etocinline`

With `\etocinline`, or its synonym `\etocnopar`, the `\tableofcontents` command and its variants do *not* first issue a `\par` to close the previous paragraph. Hence, the table of contents can be printed in an inline style; or, if used only for preparing some token list or macro, it will leave nothing in the token stream on execution.

Issue `\etocdisplay` to return to the default situation that `\tableofcontents` and variants issue a `\par` to switch to vertical mode before typesetting the TOC title and contents.

## 43. Starred variants of the `\tableofcontents` etc... commands

The `\tableofcontents`, `\localtableofcontents`, `\etocmulticol`, and all their cousins and variants have starred variants (the star must be before the other arguments). For all but the memoir class, they are like the original. For the memoir class, the original prints an

<sup>41</sup>this command `\etocoldpar` (= working `\par` in the argument to `\etocmulticolstyle`) is not related to the switch `\etocinline` whose purpose is to tell `etoc` not to do a `\par` before the table of contents.

#### 43. Starred variants of the `\tableofcontents` etc... commands

entry in the `.toc` file, as is the usage for the original `\tableofcontents` command in that class, whereas the starred variants do not, as is the habit in that class.

As soon as one starts using local table of contents one discovers that the default `memoir` thing which is to create a `chapter` entry for each TOC is not convenient. The command `\etocmemoirtotocfmt{<kind>}{<name>}` will change the format (`<kind>` is `chapter`, `section`, `subsection`... and `<name>` can be for example `\contentsname`.) The initial set-up is with `chapter` and `\contentsname`.

The format of the actual heading of the TOC should also be set appropriately (for example with `\etocstyle`), to use the identical division unit as in the first argument to `\etocmemoirtotocfmt`.

A weird situation arises when one has two successive `\localtableofcontents` (obviously this is not a truly real life situation), just after a `\part` for example. The first one creates (if the default has not been modified as indicated above) a `Chapter` heading which is written to the `.toc`. Then the second one thinks to be local to this chapter . . . and as a result it displays nothing. The fix is to define the second one to be a clone of the first one.

Independently of the situation with the `memoir` class there is generally speaking a hook macro called `\etocaftertitlehook` which is inhibited by using the starred variants of the displaying commands. Except for the `memoir` class, this hook is initially defined to do nothing.

There are also (independently of the document class) `\etocaftercontentshook`, `\etocbeforetitlehook` and `\etocaftertochook` which are initially defined to do nothing and can be used for some special effects.

## Part VII.

# Using and customizing the **etoc** own styles

Here are some statistics for this part: it contains 4 sections and 6 subsections. The name of the first section is “Summary of the main style commands” and the corresponding number is “44”. The name of the last section is “One more example of colored TOC layout” and its number is “47”. The name of the first subsection is “Setting up local styles” and the corresponding number is “44.1”. The name of the last subsection is “Customizing the toc display styles” and its number is “46.2”.

This is a table of contents for the sections and subsections in this part. It carries the label `toc:c`

<b>Summary of the main style commands</b> . . . . . <b>44, p. 69</b>	<b>The package default line styles: \etoc-</b> <b>defaultlines</b> . . . . . <b>45, p. 70</b>
Setting up local styles . . . 44.1, p. 69	<b>Customizing <b>etoc</b></b> . . . . . <b>46, p. 75</b>
Setting up toc display styles . . . . . . . . . . 44.2, p. 70	Customizing the <b>etoc</b> pre-defined line styles . . . . . 46.1, p. 75
Displaying tables of contents . . . . . . . . . . 44.3, p. 70	Customizing the toc display styles . . . . . . . 46.2, p. 76
Labels and references . . . 44.4, p. 70	<b>One more example of colored TOC</b> <b>layout</b> . . . . . <b>47, p. 77</b>

## 44. Summary of the main style commands

### 44.1. Setting up local styles

```
\etocsetstyle{<levelname>}{<start>}{<prefix>}{<contents>}{<finish>}  
\etocname, \etocnumber, \etocpage, \etocifnumbered{<A>}{<B>}  
\etocthename, \etocthenumber, \etocthepage, \etoclink{<linkname>}
```

## 44.2. Setting up toc display styles

```
\etocmulticolstyle[⟨number_of_columns⟩]{⟨heading⟩}  
\etocstyle[⟨kind⟩]{⟨number_of_columns⟩}{⟨title⟩}  
\etocstylewithmarks[⟨kind⟩]{⟨number_of_columns⟩}{⟨title⟩}{⟨mark⟩}  
\etocstylewithmarksnouc[⟨kind⟩]{⟨number_of_columns⟩}{⟨title⟩}{⟨mark⟩}  
\etocruledstyle[⟨number_of_columns⟩]{⟨title⟩}  
\etocframedstyle[⟨number_of_columns⟩]{⟨title⟩}  
\etocsettocstyle{⟨before_toc⟩}{⟨after_toc⟩}
```

## 44.3. Displaying tables of contents

```
\tableofcontents  
\localtableofcontents  
\etocmulticol[⟨number_of_columns⟩]{⟨heading⟩}  
\etoclocalmulticol[⟨number_of_columns⟩]{⟨heading⟩}  
\etocruled[⟨number_of_columns⟩]{⟨title⟩}  
\etoclocalruled[⟨number_of_columns⟩]{⟨title⟩}  
\etocframed[⟨number_of_columns⟩]{⟨title⟩}  
\etoclocalframed[⟨number_of_columns⟩]{⟨title⟩}  
and their starred variants
```

## 44.4. Labels and references

The commands (starred or not) to actually display the table of contents can be followed with optional labels or references:

```
\tableofcontents \label{toc:here}  
\tableofcontents \ref{toc:far}  
\tableofcontents \label{toc:here} \ref{toc:far}  
\localtableofcontents \label{toc:here}  
\localtableofcontents \ref{toc:far}  
\localtableofcontents \label{toc:here} \ref{toc:far}  
similarly with \etocmulticol etc . . .
```

New with 1.08e: `\localtableofcontents \ref{toc:far}` as synonym for `\tableofcontents \ref{toc:far}`.

When re-displaying another toc, only its contents are transferred: both the line styles and the toc display style are the ones currently defined, not the ones from the cloned toc.

## 45. The package default line styles: `\etocdefaultlines`

These line styles were written at an early stage in the development of the package; although the next section explains how to customize the font choicess or vertical spaces, etc. . . , used

by these line styles, most other changes would require copying them from the sources and modify them directly. Admittedly they have been written at a rather scary low- $\text{\TeX}$  level, and will not serve as a very friendly starting point.

Activating their use is done via `\etocdefaultlines`, or `\etocclines` if the line styles have not been modified with `\etocsetstyle`. Sections and sub-sections are printed in essentially the same manner, except that the leading for sub-sections is a bit smaller (with document classes lacking a `\chapter` command, the sections are printed in bold typeface; this is the case in the present document). Sub-sub-sections are printed inline, in one paragraph, with no numbers or page numbers. This style was designed and tested with documents having lots of sub-sub-sections, and should be used on a two-column layout: it provides (only in that situation with many sub-sub-sections) a more compact presentation than what is achieved by the  $\text{\LaTeX}$  default.<sup>42</sup> On the other hand, used with a one-column layout, and with few sub-sub-sections, the style is a bit more spread out vertically than the  $\text{\LaTeX}$  default, sub-sections are not visually much different from sections (especially for document classes with a `\chapter` command), so the result is less hierarchical in appearance than in the  $\text{\LaTeX}$  default.

In this document, for the [main table of contents](#), we did `\etocsetlevel{subsection}{3}` hence the sub-sections were printed with the sub-sub-section inline style.

Let us, to the contrary, typeset now this main table of contents as if the document had been done with a class having the `\chapter` command: we will print sections as chapters, and subsections as sections. We use `\etocsetlevel` for that, and also we need to change the font style of “sections” (which in truth are our subsections) to use not the bold but the medium series; we modify the `\etocfontone` command for that. Also we use dot leaders which are less spread out than in the package default.

```
\etocruledstyle[2]{\normalfont\normalsize\rmfamily\itshape
  \fbox{\parbox{.6\linewidth}{
    \leftskip 0pt plus .5fil
    \rightskip 0pt plus -.5fil
    \parfillskip 0pt plus 1fil This is the global table of
    contents on two columns, using \etoc default line styles, but with
    sections as chapters, and subsections as sections.
  }}}
\etocdefaultlines
\etocsetnexttocdepth{1}
\begingroup
\etocsetlevel{section}{0}
\etocsetlevel{subsection}{1}
\renewcommand*{\etocfontone}{\normalfont \normalsize}
\renewcommand*{\etocclineleaders}
  {\hbox{\normalfont\normalsize\hbox to 1ex {\hss.\hss}}}
\sloppy
\tableofcontents
\endgroup
```

---

<sup>42</sup>and there will never be a Part or Chapter entry alone at the bottom of a column or page (except if it has no sub-unit).

*This is the global table of contents on two columns, using **etoc** default line styles, but with sections as chapters, and subsections as sections.*

## Abstract

### 1. Foreword

### 2. License

#### Part I. Overview

### 3. Do I need to be a geek to use **etoc**?

Limitations in the use of list environments for tables of contents ..... 3.1, p. 8

### 4. Line styles and toc display style

`\etocsetstyle` for the line styles ..... 4.1, p. 9

`\etocsettocstyle` for the toc display ..... 4.2, p. 9

Compatibility mode ..... 4.3, p. 10

### 5. A first example

### 6. A second example

### 7. A Beautiful Thesis example

### 8. Linked list of the main package commands

#### Part II. Arbitrarily many TOCs, and local ones too

### 9. Labeling and reusing elsewhere

### 10. A powerful functionality of **etoc**: the re-assignment of levels with `\etocsetlevel`

### 11. The `\etocsettocdepth` and `\etocsetnexttocdepth` commands

The hyperref option *bookmarksdepth* ..... 11.1, p. 19

### 12. The command `\etocsettocdepth.toc`

The commands `\etocobeytoctocdepth` and `\etocignoretoctocdepth` ..... 12.1, p. 20

### 13. The commands `\etocdepth-tag.toc` and `\etocsettagdepth`

The commands `\etocobeydepthtags` and `\etocignoredepthtags` ..... 13.1, p. 21

### 14. The commands `\etocglobaldefs` and `\etoclocaldefs`

### 15. Not displayed empty TOCs

The `\etocchecksemtiness` command ..... 15.1, p. 21

The `\etocnotocifnotoc` command ..... 15.2, p. 22

The `\etocifwasempty` command ..... 15.3, p. 22

### 16. Adding commands to the `.toc` file

The hyperref option *hidelinks* ..... 16.1, p. 23

#### Part III. Examples

### 17. Testing the compatibility mode

### 18. Another compatibility mode

### 19. Emulating the book class



20. A framed display

21. Another TOC with a background color

22. A (crazy) inline display

#### Part IV. Surprising uses of **etoc**

23. The TOC of TOCs

24. Arbitrary “Lists Of...”, `\etoc-toccontentsline`

25. A TOC with a fancy layout

26. The TOC as a tree

27. The TOC as a molecule

28. The TOC as a TikZ mind map

29. The TOC as a table

30. A TOC self-adjusting widths for its typesetting

#### Part V. Commands for the toc line styles

31. The `\etocsetstyle`, `\etocname` and `\etocpage` commands

32. The `\etocskipfirstprefix` and `\etociffirst` commands

33. The `\etocnumber` command

34. The `\etocifnumbered` switch

35. The `\etocthename`, `\etocthenumber`, and `\etocthepage` commands

36. The `\etoclink` command

37. The `\etocthelinkedname`, `\etocthelinkednumber`, `\etocthelinkedpage` and `\etocthelink` commands

38. The `\etocsavedsectiontocline`, etc... commands.

39. The `\etocsetlevel` command

40. Scope of commands added to the .toc file

Testing the scope ..... 40.1, p. 63

This is a (pale) red subsection for illustrative purposes ..... 40.2, p. 63

41. Am I also red?

#### Part VI. Commands for the toc display style

42. Specifying the toc display style

The command `\etocsettocstyle` ..... 42.1, p. 64

The commands `\etocmulticolstyle`, `\etocmulticol`, and `\etoclocalmulticol` ..... 42.2, p. 65

The command `\etocstyle` ..... 42.3, p. 65

The commands `\etocruledstyle`, `\etocruled` and `\etoclocalruled` ..... 42.4, p. 66

The commands `\etocframedstyle`, `\etocframed`, and `\etoclocalframed` ..... 42.5, p. 66

Headings, titles, `\etocoldpar`, `\etocinnertopsep` ..... 42.6, p. 66

The compatibility mode `\etocstandarddisplaystyle` ..... 42.7, p. 67

The command `\etocinline` ..... 42.8, p. 67

45. The package default line styles: `\etocdefaultlines`

### 43. Starred variants of the `\tableofcontents` etc... commands

#### Part VII. Using and customizing the **etoc** own styles

### 44. Summary of the main style commands

Setting up local styles . . . . . 44.1, p. 69

Setting up toc display styles 44.2, p. 70

Displaying tables of contents . . . . .  
. . . . . 44.3, p. 70

Labels and references . . . . . 44.4, p. 70

### 45. The package default line styles: `\etocdefaultlines`

### 46. Customizing **etoc**

Customizing the **etoc** pre-defined line styles . . . . . 46.1, p. 75

Customizing the toc display styles . . . .  
. . . . . 46.2, p. 76

### 47. One more example of colored TOC layout

#### Part VIII. Tips

### 48. The `\etocsetlocaltop.toc` command

### 49. Hacking framed parboxes

### 50. Interverting the levels

All subsections of this document . . . . .  
. . . . . 50.1, p. 82

### 51. Displaying statistics

### 52. Using depth tags

### 53. Typesetting the TOC as a table (the old way)

#### Part IX. **etoc** and the outside world

### 54. Compatibility with other packages

Generalities . . . . . 54.1, p. 89

Compatibility with beamer . . 54.2, p. 90

Compatibility with Babel . . . 54.3, p. 90

Compatibility with hyperref 54.4, p. 90

Compatibility with multicol 54.5, p. 90

Compatibility with tocloft . 54.6, p. 90

Compatibility with the memoir class . . .  
. . . . . 54.7, p. 91

Compatibility with package tocvsec2 . .  
. . . . . 54.8, p. 91

Compatibility with package tableof . . .  
. . . . . 54.9, p. 91

Compatibility with package toctyle . .  
. . . . . 54.10, p. 91

### 55. T<sub>E</sub>Xnical matters

### 56. Errors and catastrophes

#### Part X. The code

### 57. Timestamp

### 58. Change history

### 59. Implementation

## 46. Customizing **etoc**

### 46.1. Customizing the **etoc** pre-defined line styles

We will simply list the relevant commands as defined in the package. Customizing them goes through suitable `\renewcommands`:

```
\newcommand*\etocfontminustwo{\normalfont \LARGE \bfseries}
\newcommand*\etocfontminusone{\normalfont \large \bfseries}
\newcommand*\etocfontzero{\normalfont \large \bfseries}
\newcommand*\etocfontone{\normalfont \normalsize \bfseries}
% (in classes with chapter, \etocfontone does not do \bfseries)
\newcommand*\etocfonttwo{\normalfont \normalsize}
\newcommand*\etocfontthree{\normalfont \footnotesize}

\newcommand*\etocsepminustwo{4ex plus .5ex minus .5ex}
\newcommand*\etocsepminusone{4ex plus .5ex minus .5ex}
\newcommand*\etocsepzero{2.5ex plus .4ex minus .4ex}
\newcommand*\etocsepone{1.5ex plus .3ex minus .3ex}
\newcommand*\etocseptwo{.5ex plus .1ex minus .1ex}
\newcommand*\etocsepthree{.25ex plus .05ex minus .05ex}

\newcommand*\etocminustwoleftmargin{1.5em plus 0.5fil}
\newcommand*\etocminustworightmargin{1.5em plus -0.5fil}
\newcommand*\etocminusoneleftmargin{1em}
\newcommand*\etocminusonerightmargin{1em}

\newcommand*\etocbaselinespreadminustwo{1}
\newcommand*\etocbaselinespreadminusone{1}
\newcommand*\etocbaselinespreadzero{1}
\newcommand*\etocbaselinespreadone{1}
\newcommand*\etocbaselinespreadtwo{1}
\newcommand*\etocbaselinespreadthree{.9}
\newcommand*\etoclineleaders
  {\hbox{\normalfont\normalsize\hbox to 2ex {\hss.\hss}}}
\newcommand*\etocabbrevpagename{p.~} % initial of "page"
\newcommand*\etocpartname{Part} % prior to v1.08b, was \partname
% but this didn't make sense e.g. with babel+frenchb whose \frenchpartname
% takes into account the value of the part counter.
\newcommand*\etocbookname{Book} % to be modified according to language
```

No customizing of the standard line styles is possible from within **etoc**. As already explained, when `\etocstandardlines` has been issued, the package just makes itself very discrete and acts only at the global level, and the TOC entries are (hopefully) formatted as would have happened in the absence of **etoc**.<sup>43</sup>

The `\etocstandardlines` compatibility mode will work also with sectioning commands made known to **etoc** via `\etocsetlevel`, under the condition of course that these sectioning commands are accompanied with all the relevant definitions for typesetting toc entries in the L<sup>A</sup>T<sub>E</sub>X default manner (existence of the macros `\l@something` . . .).

Using the command `\etocsetstyle`, be it in the preamble or in the body of the document, has the secondary effect of switching off the compatibility mode.

---

<sup>43</sup>with the KOMA-script classes, we noticed that `\etocstandarddisplaystyle` was apparently needed for the KOMA options `toc=left` to be active at the level of the line entries.

## 46.2. Customizing the toc display styles

Again we list the relevant macros, what they do should be legible from their names. Note that `\renewcommand`'s and not `\setlength`'s have to be used for what appear to be lengths, and that color commands are not just color specifications, they must include `\color`, and are canceled by re-defining them to do `\relax`.

```
\newcommand*\etocabovetocskip{3.5ex plus 1ex minus .2ex}
\newcommand*\etocbelowtocskip{3.5ex plus 1ex minus .2ex}
```

```
\newcommand*\etoccolumnsep{2em}
\newcommand*\etocmulticolsep{0ex}
\newcommand*\etocmulticolpretolerance{-1}
\newcommand*\etocmulticoltolerance{200}
\newcommand*\etocdefaultnbcol{2}
\newcommand*\etocinnertopsep{2ex}
\newcommand*\etoc_toprule{\hrule}
\newcommand*\etoc_toprulecolorcmd{\relax}
```

% for the framed style only:

```
\newcommand*\etocinnerleftsep{2em}
\newcommand*\etocinnerrightsep{2em}
\newcommand*\etocinnerbottomsep{3.5ex}
```

```
\newcommand*\etocleftrule{\vrule}
\newcommand*\etocrightrule{\vrule}
\newcommand*\etocbottomrule{\hrule}
\newcommand*\etocleftrulecolorcmd{\relax}
\newcommand*\etocrightrulecolorcmd{\relax}
\newcommand*\etocbottomrulecolorcmd{\relax}
```

```
\newcommand*\etocbkgcolorcmd{\relax}
```

% hooks

```
\newcommand\etocframedmphook{\relax}
```

The `\etocframedmphook` is positioned immediately after the beginning of a minipage environment where the contents of the framed TOC are typeset.

The `\...colorcmd` commands are initially set to expand to `\relax` (hence do not require package `color` or `xcolor` to be loaded). If one has modified a command such as `\etocbkgcolorcmd` to expand to a color command and wants to reset it to do nothing, one *must* use `\renewcommand{\etocbkgcolorcmd}{\relax}` and not `\let\etocbkgcolorcmd\relax`.

Regarding the dimensions of the top rule they can be specified in `ex`'s or `em`'s as in this example:

```
\renewcommand{\etoc_toprule}{\hrule height 1ex}
```

The package code is done in such a manner that it is the font size in instance at the end of typesetting the title argument to `\etocruledtoc` or `\etocframedtoc` which will be used for the meaning of the `'1ex'`. Of course also the other rule commands can have their dimensions in font relative units, but their values are decided on the basis of the font in effect just before the table of contents.

The top and bottom rules do not have to be rules and can be horizontal *leaders* (of a specified height) in the general  $\mathrm{T}_{\mathrm{E}}\mathrm{X}$  sense. However the left and right rules are not used as

(horizontal) leaders but as objects of a given specified width. Note that *only* the Plain TeX syntax for rules is accepted here.

## 47. One more example of colored TOC layout

The command `\etocframedstyle` puts the title on the top rule in a centered position. This is not very convenient for this example so we included the title as part of the `<start>` code at section level, to get it *inside* the frame.

```
\begingroup
\definecolor{subsecnum}{RGB}{13,151,225}
\definecolor{secbackground}{RGB}{0,177,235}
\definecolor{tocbackground}{RGB}{212,237,252}

\renewcommand{\etocbkgcolorcmd}{\color{tocbackground}}
\renewcommand{\etocleftrulecolorcmd}{\color{tocbackground}}
\renewcommand{\etocrightrulecolorcmd}{\color{tocbackground}}
\renewcommand{\etocbottomrulecolorcmd}{\color{tocbackground}}
\renewcommand{\etocoprulcolorcmd}{\color{tocbackground}}

\renewcommand{\etocleftrule}{\vrule width 1cm}
\renewcommand{\etocrightrule}{\vrule width .5cm}
\renewcommand{\etocbottomrule}{\hrule height 12pt}
\renewcommand{\etocoprul}{\hrule height 12pt}

\renewcommand{\etocinnertopsep}{0pt}
\renewcommand{\etocinnerbottomsep}{0pt}
\renewcommand{\etocinnerleftsep}{0pt}
\renewcommand{\etocinnerrightsep}{0pt}

\newcommand\shiftedwhiterule[2]{%
  \hbox to \linewidth{\color{white}%
    \hskip#1\leaders\vrule height1pt\hfil}\nointerlineskip\vskip#2}

\etocsetstyle{subsubsection}{\etocskipfirstprefix}
{\shiftedwhiterule{\leftskip}{6pt}}
{\sffamily\footnotesize
  \leftskip2.5cm\hangindent1cm\rightskip1cm\noindent
  \hbox to 1cm{\color{subsecnum}\etocnumber\hss}%
  \color{black}\etocname\leaders\hbox to .2cm{\hss.}\hfill
  \rlap{\hbox to 1cm{\hss\etocpage\hskip.2cm}}\par
  \nointerlineskip\vskip3pt}
{}}

\etocsetstyle{subsection}{\etocskipfirstprefix}
{\shiftedwhiterule{1.5cm}{6pt}}
{\sffamily\small
  \leftskip1.5cm\hangindent1cm\rightskip1cm\noindent
  \hbox to 1cm{\color{subsecnum}\etocnumber\hss}%
  \color{black}\etocname\leaders\hbox to .2cm{\hss.}\hfill
  \rlap{\hbox to 1cm{\hss\etocpage\hskip.2cm}}\par
  \nointerlineskip\vskip6pt}
{}}

\newcommand{\coloredstuff}[2]{%
```

#### 47. One more example of colored TOC layout

```

\leftskip0pt\rightskip0pt\parskip0pt
\fbboxsep0pt % \colorbox uses \fbboxsep also when no frame!
\noindent\colorbox{secbackground}
{\parbox{\linewidth}{%
\vskip5pt
{\noindent\color{#1}#2\par}\nointerlineskip
\vskip3pt}}%
\par\nointerlineskip}

\etocsetstyle{section}
{\coloredstuff{white}
{\hfil \hyperref[toc:b]{\bfseries\large I am a twin of
that other TOC (click me!)}\hfil}}
{\vskip3pt\sffamily\small}
{\coloredstuff{white}
{\leftskip1.5cm\rightskip.5cm\parfillskip-\rightskip
\makebox[0pt][r]{\makebox[.5cm][r]{\etocnumber\hspace{.2cm}}}}%
\etocname\hfil\makebox[.5cm][r]{\etocpage\hspace{.2cm}}}%
\vskip6pt }
{}

\etocframedstyle[1]{}
\tableofcontents \label{toc:clone} \ref{toc:globalcmds}
\endgroup

```

I am a twin of that other TOC (click me!)		
42	Specifying the toc display style	64
42.1	The command \etocsettocstyle .....	64
42.2	The commands \etocmulticolstyle, \etocmulticol, and \etoclocalmulticol .....	65
42.3	The command \etocstyle .....	65
42.4	The commands \etocruledstyle, \etocruled and \etoclocalruled .....	66
42.5	The commands \etocframedstyle, \etocframed, and \etoclocalframed .....	66
42.6	Headings, titles, \etocoldpar, \etocinnertopsep .....	66
42.7	The compatibility mode \etocstandarddisplaystyle ...	67
42.8	The command \etocinline .....	67
43	Starred variants of the \tableofcontents etc... commands	67

The TOC has been put in a [float](#) which appears on this page. The coding is a bit in-

volved<sup>44</sup> as it does not use any additional package. Also, it was written at some early stage and I have not revised it since.

A better solution would be to use some package to set up a background color possibly extending accross pages, as the framed style (which we used to get this background color) can only deal with material short enough to fit on one page.

Regarding colors, generally speaking all color commands inside **etoc** are initially defined to do nothing, and the choice to use or not colors is left to the user.

---

<sup>44</sup>and reveals the author's preference for the T<sub>E</sub>X syntax...

## Part VIII.

# Tips

Here are some statistics for this part: it contains 6 sections and 1 subsection. The name of the first section is “The `\etocsetlocaltop.toc` command” and the corresponding number is “48”. The name of the last section is “Typesetting the TOC as a table (the old way)” and its number is “53”. The name of the first subsection is “All subsections of this document” and the corresponding number is “50.1”. The name of the last subsection is “All subsections of this document” and its number is “50.1”.

<b>48. The <code>\etocsetlocaltop.toc</code> command</b>	<b>80</b>
<b>49. Hacking framed parboxes</b>	<b>81</b>
<b>50. Interverting the levels</b>	<b>82</b>
50.1. All subsections of this document . . . . .	82
<b>51. Displaying statistics</b>	<b>83</b>
<b>52. Using depth tags</b>	<b>85</b>
<b>53. Typesetting the TOC as a table (the old way)</b>	<b>87</b>

## 48. The `\etocsetlocaltop.toc` command

It is important to understand that `\localtableofcontents` works entirely from data *in the .toc file*. If the document, say with `article` class, contains starred sectioning commands, which are not accompanied by suitable `\addcontentsline`, then these units are completely transparent to `\localtableofcontents`:

- If it is issued before `\section*{Foo}`, say locally to a `\section`, then the local TOC will include not only the subsections between the `\section` and the `\section*{Foo}` but also those following, and be stopped uniquely by a later `\section` or `\part` from the document’s body.
- If the command is issued right after `\section*{Foo}` and the later was itself subsequent to a (numbered) `\subsection`, then `etoc` will think it must display a TOC local to that *subsection* level.

Before concluding that `etoc` is buggy, imagine how the global TOC of the document would look like in such circumstances: the `\localtableofcontents` does exactly the job of extracting from that global TOC various local TOCs; it works only from material in that (virtual) global TOC.

Since release 1.08k, `etoc` provides the one-argument command `\etocsetlocaltop.toc` to insert into the .toc file a kind of “ghost” of a given sectioning unit. Here is an example:



```
\part*{Extra unnumbered part}
\etocsetlocaltop.toc{part}
\localtableofcontents
```

Then in this example, `\localtableofcontents` will be stopped by the next `\part` (naturally a numbered one or a `\part*` followed by suitable `\addcontentsline`) or also by a later `\etocsetlocaltop.toc{part}`.

This is all fine but keep in mind again that `\localtableofcontents` can only display units which have been pushed to the `.toc` file, so that `\part*` must be followed by numbered sections or `\section*/\addcontentsline` pairs if `\localtableofcontents` is to display anything useful. One can not use `etoc` to typeset data not actually there in the `.toc` file!

Thus, in the situation above, one may wonder why the document does not use `\addcontentsline` after the `\part*` as it must do so for sub-sectioning units if they are to appear in the local TOC.

One may imagine some document where all Parts are non-numbered but nevertheless the user wants its `\localtableofcontents` for each one: then indeed `\etocsetlocaltop.toc` will help in that case.

There are at least two more methods to handle this situation:

- the first one is simply to use the `\addcontentsline{toc}{part}{...}` commands, and to set the `part` style to do nothing, so that the main TOC will not display them,
- the second one does not add the `\addcontentsline{toc}{part}{...}` but uses “depth tags” (section 52) and multiple `\tableofcontents` (not `\localtableofcontents`), one for each desired local TOC.

There is a variant of the first method which would set the level of `part` temporarily to 6 for the typesetting of the main TOC. But then the chapter toc line style will execute inside the main TOC only once its *start* and *finish* parts (see section 31). This is actually also a problem of the second method.

But not of the first method which uses

```
\etocsetstyle{part}{}{}{}{}
```

nor of the approach with `\etocsetlocaltop.toc`: the *start* and *finish* portions of the chapter toc line style will get executed at the appropriate times in the main, global, TOC.

Be aware however that this use of `\etocsetlocaltop.toc{part}` will also cause the main global TOC to execute (once) the *start* and *finish* portions of the part toc line style (assuming here again it is not typeset in compatibility mode).

Final remark: usage of `\etocsetlocaltop.toc` interferes with `\etoccheckemptiness` only to the extent that it modifies (as explained in this section) the selection made by `\localtableofcontents`, hence the decision whether the local TOC is empty or not.

## 49. Hacking framed parboxes

```
\etocdefaultlines
\renewcommand\etocoprul{\hrule height 2pt depth 2pt}
\etocruled{\color{green}\fboxrule2pt\fboxseplex
\fbx{\raisebox{-\fontdimen22\textfont2}
{\color{blue}\parbox{.5\linewidth}
```

## 50. Interverting the levels

```
{\normalfont This text is perfectly centered
vertically with respect to the
surrounding horizontal rules.}}}}
\ref{toc:globalcmds}
```

This text is perfectly centered vertically with respect to the surrounding horizontal rules.

<b>Specifying the toc display style</b> . . . . .	The commands <code>\etocframedstyle</code> , <code>\etocframed</code> , and <code>\etoclocalframed</code> . . . . .
42, p. 64	42.5, p. 66
The command <code>\etocsettocstyle</code> . . . . .	Headings, titles, <code>\etocoldpar</code> , <code>\etocinerttopsep</code> . . . . .
42.1, p. 64	42.6, p. 66
The commands <code>\etocmulticolstyle</code> , <code>\etocmulticol</code> , and <code>\etoclocalmulticol</code> . . . . .	The compatibility mode <code>\etocstandarddisplaystyle</code> . . . . .
42.2, p. 65	42.7, p. 67
The command <code>\etoclocstyle</code> . . . . .	The command <code>\etocinline</code> . . . . .
42.3, p. 65	42.8, p. 67
The commands <code>\etocruledstyle</code> , <code>\etocruled</code> and <code>\etoclocalruled</code> . . . . .	<b>Starred variants of the <code>\tableofcontents</code> etc... commands</b> . . . . .
42.4, p. 66	43, p. 67

## 50. Interverting the levels

Let us display and count all subsections occurring in this document (see [Part IV](#) for other uses of this technique):

```
\etocsetnexttocdepth{2}
\begingroup
\etocsetlevel{part}{3}
\etocsetlevel{section}{3}
\etocsetstyle{subsection}
{
\small\begin{enumerate}[itemsep=0pt,label=,leftmargin=0pt]}
{\normalfont\bfseries\item}
{\roman{enumi}. \mdseries\etocname{ } (\etocnumber, p.~\etocpage)}
{\end{enumerate}}
\renewcommand{\etoccolumnsep}{2.75em}
\renewcommand{\columnseprule}{1pt}
\etocmulticol[3]{\subsection{All subsections of this document}}
\endgroup
```

### 50.1. All subsections of this document

i. Limitations in the use of list environments for tables of contents (3.1, p. 8)	iv. Compatibility mode (4.3, p. 10)	<code>\cignoretocdepth</code> (12.1, p. 20)
ii. <code>\etocsetstyle</code> for the line styles (4.1, p. 9)	v. The <code>\hyperref</code> option <code>\bookmarksdepth</code> (11.1, p. 19)	vii. The commands <code>\etocobeydepthtags</code> and <code>\etocignoredepthtags</code> (13.1, p. 21)
iii. <code>\etocsettocstyle</code> for the toc display (4.2, p. 9)	vi. The commands <code>\etocobeytocdepth</code> and <code>\eto-</code>	viii. The <code>\etocchecks-</code>

emptiness command (15.1, p. 21)

**ix.** The `\etocnotocifnotoc` command (15.2, p. 22)

**x.** The `\etocifwasempty` command (15.3, p. 22)

**xi.** The `hyperref` option `hidelinks` (16.1, p. 23)

**xii.** Testing the scope (40.1, p. 63)

**xiii.** This is a (pale) red subsection for illustrative purposes (40.2, p. 63)

**xiv.** The command `\etocsettocstyle` (42.1, p. 64)

**xv.** The commands `\etocmulticolstyle`, `\etocmulticol`, and `\etoclocalmulticol` (42.2, p. 65)

**xvi.** The command `\etoc-tocstyle` (42.3, p. 65)

**xvii.** The commands `\etoc-ruledstyle`, `\etocruled` and `\etoclocalruled` (42.4, p. 66)

**xviii.** The commands `\etoc-framedstyle`, `\etocframed`, and `\etoclocalframed` (42.5, p. 66)

**xix.** Headings, titles, `\etocoldpar`, `\etocinnertop-sep` (42.6, p. 66)

**xx.** The compatibility mode `\etocstandarddisplaystyle` (42.7, p. 67)

**xxi.** The command `\etoc-inline` (42.8, p. 67)

**xxii.** Setting up local styles (44.1, p. 69)

**xxiii.** Setting up toc display styles (44.2, p. 70)

**xxiv.** Displaying tables of contents (44.3, p. 70)

**xxv.** Labels and references (44.4, p. 70)

**xxvi.** Customizing the **etoc** pre-defined line styles (46.1, p. 75)

**xxvii.** Customizing the toc display styles (46.2, p. 76)

**xxviii.** All subsections of this document (50.1, p. 82)

**xxix.** Generalities (54.1, p. 89)

**xxx.** Compatibility with `beamer` (54.2, p. 90)

**xxxi.** Compatibility with `Babel` (54.3, p. 90)

**xxxii.** Compatibility with `hyperref` (54.4, p. 90)

**xxxiii.** Compatibility with `multicol` (54.5, p. 90)

**xxxiv.** Compatibility with `tocloft` (54.6, p. 90)

**xxxv.** Compatibility with the `memoir` class (54.7, p. 91)

**xxxvi.** Compatibility with package `tocvsec2` (54.8, p. 91)

**xxxvii.** Compatibility with package `tableof` (54.9, p. 91)

**xxxviii.** Compatibility with package `tocstyle` (54.10, p. 91)

## 51. Displaying statistics

Each part of this document starts with a paragraph telling how many sections and subsections it has. Well, each one of this paragraph is a table of contents! We designed a macro `\thispartstats` to do that. It uses “storage” boxes to keep the information about the first and last section or subsection. Using boxes is the simplest manner to encapsulate the `hyperref` link for later use (whether there is one or none). However, one cannot modify then the font or the color. If such a need arises, one must switch from using boxes to using macros, and store the `hyperref` data for later use as was done in the code presented in [section 27](#). We present also this second method.

But first, the code of `\thispartstats`:

```
\newsavebox\firstnamei \newsavebox\firstnumberi
\newsavebox\lastnamei \newsavebox\lastnumberi
\newsavebox\firstnameii \newsavebox\firstnumberii
\newsavebox\lastnameii \newsavebox\lastnumberii
\newcounter{mycounti} \newcounter{mycountii}
\newcommand*{\thispartstatsauxi}{} \newcommand*{\thispartstatsauxii}{}
\newcommand*{\oldtocdepth}{}
\newcommand*{\thispartstats}{%
  \setcounter{mycounti}{0}%
  \setcounter{mycountii}{0}%
```

## 51. Displaying statistics

```

\def\thispartstatsauxi{%
    \sbox{\firstnamei}{\etocname}%
    \sbox{\firstnumberi}{\etocnumber}%
    \def\thispartstatsauxi{}}%
\def\thispartstatsauxii{%
    \sbox{\firstnameii}{\etocname}%
    \sbox{\firstnumberii}{\etocnumber}%
    \def\thispartstatsauxii{}}%
\beginingroup
\etocsetstyle{subsection} {} {}
    {\thispartstatsauxii
    \stepcounter{mycountii}%
    \sbox{\lastnameii}{\etocname}%
    \sbox{\lastnumberii}{\etocnumber}} {}%
\etocsetstyle{section} {} {}
    {\thispartstatsauxi
    \stepcounter{mycounti}%
    \sbox{\lastnamei}{\etocname}%
    \sbox{\lastnumberi}{\etocnumber}}
    {Here are some statistics for this part: it contains \arabic{mycounti}
    section\ifnum\value{mycounti}>1 s\fi{} and \arabic{mycountii}
    subsection\ifnum\value{mycountii}>1 s\fi. The name of the first section is
    \unhbox\firstnamei{} and the corresponding number is \unhbox\firstnumberi.
    The name of the last section is \unhbox\lastnamei{} and its number is
    \unhbox\lastnumberi. The name of the first subsection is \unhbox\firstnameii{}
    and the corresponding number is \unhbox\firstnumberii. The name of the last
    subsection is \unhbox\lastnameii{} and its number is \unhbox\lastnumberii.}%
\etocinline % cancels the automatic \par automatically before the TOC
\etocsettocstyle {}{}
\etocsetnexttocdepth{2}%
\localtableofcontents % to be used at the top level of a Part.
\endgroup
}

```

And now, the variant with macros rather than boxes (this variant as it stands here is for using within a section).

```

\makeatletter
\newcommand*\firstsubname {} \newcommand*\lastsubname {}
\newcommand*\firstsubnumber {} \newcommand*\lastsubnumber {}
\newcommand*\thisspecialstatsaux{}
\newcommand*\thisspecialstats{%
    \setcounter{mycounti}{0}%
    \def\thisspecialstatsaux{%
        \let\firstsubname\etocthelinkedname
        \let\firstsubnumber\etocthelinkednumber
        \def\thisspecialstatsaux{}}
    \beginingroup
    \etocsetstyle{subsection} {} {}
        {\thisspecialstatsaux
        \stepcounter{mycounti}%
        \let\lastsubname\etocthelinkedname
        \let\lastsubnumber\etocthelinkednumber }
    {Here are some statistics for this section. It contains \arabic{mycounti}
    subsections. The name of its first is \emph{\firstsubname{}} and the
    corresponding number is {\firstsubnumber}. The name of the last

```

```

        subsection is \emph{\lastsubname{}} and its number is {\lastsubnumber}.}%
\etocsettocstyle {}{}
\etocinline
\etocsetnexttocdepth {1}%
\localtableofcontents % to be used within a section
\endgroup
}
\makeatother

```

## 52. Using depth tags

We want a TOC which will have a heading for each `\part` (except the last part with the code source, which we decide not to include), and will additionally open up [Part VI](#) up to paragraphs. To achieve this we added to this source various `\etocdepthtag.toc` commands, and it remains now to set the levels for each tag using `\etocsettagdepth` (this was used earlier in this document, in [section 6](#), within a group hence it did not affect the other tables of contents).

However, the package line styles do not display paragraphs, and the standard line styles of the document class give too much vertical spacing (in this context) when displaying a Part heading in the TOC. So we cook up our own, quickly designed line styles, in the style of [section 5](#) (but with a way to put page numbers on the right which is more like the method used by L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>'s `\@dottedtocline`; and multi-line headings now leave empty the area underneath the numbers contrarily to the code from [section 5](#)). And after a few minutes for choosing lengths (now that this has been done once, it can be recycled easily) we get:

A TOC using depth tags	
<b>I Overview</b>	8
<b>II Arbitrarily many TOCs, and local ones too</b>	16
<b>III Examples</b>	24
<b>IV Surprising uses of <code>etoc</code></b>	36
<b>V Commands for the toc line styles</b>	57
<b>VI Commands for the toc display style</b>	64
<b>42 Specifying the toc display style</b> .....	64
42.1 The command <code>\etocsettocstyle</code> .....	64
42.2 The commands <code>\etocmulticolstyle</code> , <code>\etocmulticol</code> , and <code>\etoclocalmulticol</code> .....	65
42.3 The command <code>\etocstyle</code> .....	65
42.3.1 The command <code>\etocstylewithmarks</code> .....	65
42.3.1.1 Do we really want paragraph entries in the TOC?	65
42.3.1.2 really?.....	65
42.4 The commands <code>\etocruledstyle</code> , <code>\etocruled</code> and <code>\etoclocal-ruled</code> .....	66
	85

## 52. Using depth tags

42.5	The commands <code>\etocframedstyle</code> , <code>\etocframed</code> , and <code>\etoclocalframed</code> .....	66
42.6	Headings, titles, <code>\etocoldpar</code> , <code>\etocinnertopsep</code> .....	66
42.7	The compatibility mode <code>\etocstandarddisplaystyle</code> .....	67
42.7.1	The commands <code>\etocarticlestyle</code> , <code>\etocbookstyle</code> , ... ..	67
42.8	The command <code>\etocinline</code> .....	67
43	Starred variants of the <code>\tableofcontents</code> etc... commands .....	67
VII	Using and customizing the <b>etoc</b> own styles .....	69
VIII	Tips .....	80
IX	<b>etoc</b> and the outside world .....	89

---

```
\etocsetnexttocdepth {all}
\begingroup
\parindent 0pt \leftskip 0cm \rightskip .75cm \parfillskip -\rightskip
\newcommand*{\EndParWithPagenoInMargin}
{
  \nobreak\hfill
  \makebox[0.75cm][r]{\mdseries\normalsize\etocpage}%
  \par}
\renewcommand*{\etococlineleaders}
{
  \hbox{\normalfont\normalsize\hbox to .75ex {\hss.\hss}}}
\newcommand*{\EndParWithPagenoInMarginAndLeaders}
{
  \nobreak\leaders\etococlineleaders\hfill
  \makebox[0.75cm][r]{\mdseries\normalsize\etocpage}%
  \par }
\etocsetstyle {part}
{
  {}
  {\leavevmode\leftskip 1cm\relax}
  {\bfseries\large\llap{\makebox[1cm][r]{\etocnumber\ }}%
  \etocname\EndParWithPagenoInMargin\smallskip}
  {}
}
\etocsetstyle {section}
{
  {}
  {\leavevmode\leftskip 1.75cm\relax}
  {\bfseries\normalsize\llap{\makebox[.75cm][l]{\etocnumber}}%
  \etocname\EndParWithPagenoInMarginAndLeaders}
  {}
}
\etocsetstyle {subsection}
{
  {}
  {\leavevmode\leftskip 2.75cm\relax }
  {\mdseries\normalsize\llap{\makebox[1cm][l]{\etocnumber}}%
  \etocname\EndParWithPagenoInMarginAndLeaders}
  {}
}
\etocsetstyle {subsubsection}
{
  {}
  {\leavevmode\leftskip 4cm\relax }
  {\mdseries\normalsize\llap{\makebox[1.25cm][l]{\etocnumber}}%
  \etocname\EndParWithPagenoInMarginAndLeaders}
  {}
}
\etocsetstyle {paragraph}
{
  {}
  {\leavevmode\leftskip 5.5cm\relax }
```

```

{\mdseries\normalsize\llap{\makebox[1.5cm][l]{\etocnumber}}}%
\etocname\EndParWithPagenoInMarginAndLeaders}
{}
\etocsettagdepth {preamble} {none}
\etocsettagdepth {overview} {part}
%\etocsettagdepth {arbitrarily}{part}% not needed explicitly, keeps value
%\etocsettagdepth {examples} {part}
%\etocsettagdepth {surprising} {part}
%\etocsettagdepth {linestyles} {part}
\etocsettagdepth {globalcmds} {paragraph}
\etocsettagdepth {custom} {part}
%\etocsettagdepth {tips} {part}
%\etocsettagdepth {etocandworld}{part}
\etocsettagdepth {code} {none}
\renewcommand\etocdotprule {\hrule height 3pt\relax }
\renewcommand\etocdotprulecolorcmd {\color{blue}}
\renewcommand\etocaftercontentshook
{\medskip\begin{group} \color{blue}\hrule height 3pt \end{group} }
\etocruledstyle [1]{\Large\bfseries
\fbbox{\makebox[8cm]{A TOC using depth tags}}}
\sloppy
\tableofcontents
\endgroup

```

## 53. Typesetting the TOC as a table (the old way)

Due to, among other things, the fact that alignment cells create and close groups, and that by default definitions of `\etocname`, `\etocnumber`, `\etocpage` made by `etoc` are local, it was not easy to typeset a TOC as table with `etoc`, prior to release 1.08.

Not only `\etocname` etc... caused a problem, but also the basic redefinition of `\contentsline` was made by `etoc` only after the first argument to `\etocsettocstyle` had been executed, hence if this argument were to open a tabular, the `etoc` redefinition of `\contentsline` would be done in the first cell of the first row and get lost thereafter.

Thus one had to resort to the technique explained in [section 26](#) of using the execution of `\tableofcontents` as a way to store data which was then displayed later.

For the record, here is how the TOC from [section 29](#) was coded in the old days. We don't have here the problems with the positioning of `\hline`'s we face with the newer method; on the other hand we must manipulate token registers which are not familiar to most  $\text{\LaTeX}$  users (macros could be used, but would be more cumbersome, except perhaps if using the  $\varepsilon\text{-TeX}$  `\unexpanded`).

The method here is the most powerful because it filters out of the `.toc` file only the data we want (the other things are not ignored, they are executed but hopefully do not create havoc; typically they are language changing instructions, etc...), and we are less susceptible to fall potential victims of various external macros inserted in the `.toc` file by other packages.

Note: rather than `\toks` registers it would be easier here to use  $\varepsilon\text{-TeX}$  `\unexpanded` primitive. See for example [section 28](#).

### 53. Typesetting the TOC as a table (the old way)

```

\newtoks\toctabletok
\newcommand*\appendtotok[2]{% #1=toks variable, #2=macro, expands once #2
  #1\expandafter\expandafter\expandafter {\expandafter\the\expandafter #1#2}}

\newcommand*\PreparePart{%
  \toks0 \expandafter{\etocthelinkednumber}%
  \toks2 \expandafter{\etocthelinkedname}%
  \toks4 \expandafter{\etocthelinkedpage}%
  \edef\toctablepiece {\noexpand\hline
    \noexpand\strut\the\toks0 &\noexpand\bfseries\the\toks2
    &\the\toks4 \noexpand\\ \noexpand\hline}%
}
\newcommand*\PrepareSection{%
  \toks0 \expandafter{\etocthelinkednumber}%
  \toks2 \expandafter{\etocthelinkedname}%
  \toks4 \expandafter{\etocthelinkedpage}%
  \edef\toctablepiece {\the\toks0 &\the\toks2 &\the\toks4 \noexpand\\}%
}
% new version for 1.07k, 2014/03/06
\newcommand*\PrepareSubsection{%
  \toks0 \expandafter{\etocthelinkednumber}%
  \toks2 \expandafter{\expandafter\itshape\etocthelinkedname\strut}%
  \toks4 \expandafter{\expandafter\itshape\etocthelinkedpage}%
  \edef\toctablepiece{&\noexpand\makebox[1cm][c]{\the\toks0}%
    \noexpand\parbox[t]{\dimexpr6cm-\tabcolsep\relax}
    {\noexpand\sloppy\the\toks2}%
    &\the\toks4 \noexpand\\}%
}

\begin{group}
\etocsetstyle{part}{}{}{\PreparePart \appendtotok\toctabletok\toctablepiece}{}

\etocsetstyle{section}{}{}{\PrepareSection \appendtotok\toctabletok\toctablepiece}{}

\etocsetstyle{subsection}{}{}{\PrepareSubsection\appendtotok\toctabletok\toctablepiece}{}

\etocsettocstyle
  {\toctabletok{\hypersetup{hidelinks}%
    \begin{longtable}{|>{\bfseries}c|p{7cm}|r|}\hline
    \multicolumn{3}{|c|}{\Large\bfseries\strut TABLE OF CONTENTS}%
    \\ \hline \hline}}
  {\global\toctabletok\expandafter{\the\toctabletok\hline\end{longtable}}}
\etocsettocdepth {subsection}
\tableofcontents
\the\toctabletok
\end{group}

```



## Part IX.

# etoc and the outside world

## 54. Compatibility with other packages

### 54.1. Generalities

The contents of the .toc file (if it already exists) are read into memory by **etoc** once, at the time of `\begin{document}`.<sup>45</sup>

The .toc file remains available to other packages for read operations until the location of the first table of contents at which time a write stream is opened by **etoc** and from that point the file is erased until its contents are again written to the disk by  $\text{\LaTeX}$  at the end of the compilation.

Don't use `\if<condition> stuff \tableofcontents\fi`, but:

`\if<condition> stuff \expandafter\tableofcontents\fi`

Also a `\else` immediately following `\tableofcontents` or `\localtableofcontents` requires a previous `\expandafter`.

**etoc** can not really cohabit with packages modifying the `\tableofcontents` command: some sort of truce can be achieved if **etoc** is loaded last, hence is the winner.

Do not modify the `\tableofcontents` command like this:

`\let\oldtableofcontents\tableofcontents`

`\renewcommand\tableofcontents{\oldtableofcontents foo}`

as this will make the `\label/\ref` mechanism impossible.

Rather use

`\renewcommand\etocaftertochook{foo}`

and there is also `\etocaftercontentshook` which is executed a bit earlier<sup>46</sup> just before the closing part of the toc display style (and thus still within a group.)

Prepending is less of a problem (and anyhow there is also `\etocbeforetitlehook` available to the user).

Under certain circumstances **etoc** imposes its views on `\tableofcontents` at the time of `\begin{document}`. You may thus have to use `\AtBeginDocument` to delay your (necessarily ugly and non-recommendable) patches. Patching after `\begin{document}` is naturally possible but I feel almost a rebel to mention this to  $\text{\LaTeX}$  users!

**etoc** requires the .toc file to use the `\contentsline` macro. It **can not** work if there is no .toc file or if the .toc file does not contain the relevant data or if the .toc file does not use the `\contentsline` macro or if the `\contentsline` macro does not invoke the `\l@<divisionunit>` macros.

<sup>45</sup>New with 1.07m. Earlier versions read the .toc file at the time of `\usepackage{etoc}`. Thanks to Denis Bitouzé who signaled a Babel related problem, which turned out to be caused by this.

<sup>46</sup>contrarily to `\etocaftertochook`, `\etocaftercontentshook` is not executed if the `tocdepth` did not allow the printing of the TOC.

## 54. Compatibility with other packages

### 54.2. Compatibility with beamer

For the very reasons mentioned right before, **etoc** is incompatible with the beamer class. However, if beamer is used in an article mode, i.e., with the article class in conjunction with the beamerarticle package, then **etoc** should work.

### 54.3. Compatibility with Babel

One must load **etoc** *after* babel. This is in order for babel's shorthands to be active at the time when **etoc** loads the .toc file.

### 54.4. Compatibility with hyperref

Please inform the author in case of issues: **etoc** was from the start designed to be 100% compatible with package hyperref.

The macros `\etocname`, `\etocnumber`, and `\etocpage` contain the hyperref links, if present (note that the `linktoc=all` option of hyperref tells it to put a link also in the page number corresponding to a given toc entry). For example, the tables of contents of the present document are all fully linked. It doesn't matter whether **etoc** or hyperref is loaded first.

### 54.5. Compatibility with multicol

**etoc** loads the package multicol.

### 54.6. Compatibility with tocloft

Release 1.07k added compatibility with package tocloft:<sup>47</sup> steps are taken to prevent the redefinition of `\tableofcontents` done by tocloft at `\begin{document}`. As long as **etoc** is left in compatibility mode the customization done by tocloft will be obeyed, for both the line styles and the TOC title. One may still benefit from the *depth tags* management by **etoc**, from its `\localtableofcontents`, from its `\label+\ref` mechanism. One may use `\etocsetstyle` to define via **etoc** the layout for one TOC and then use rather tocloft for another one, if `\tableofcontents` follows `\etocstandardlines` and `\etocstandarddisplaystyle`. In this compatibility mode `\etocsetlevel{division unit}{6}` will render invisible the chosen division level, but exchanging levels is otherwise not possible.

One should load **etoc** *after* tocloft. A warning is issued if otherwise, because if **etoc** is loaded before it will realize that at the time of `\begin{document}` and trick tocloft into believing having been loaded with the titles option.

It is possible to customize (using package tocloft for example) throughout the document the macros `\l@section`, `\l@subsection` ... and the effect will be seen in the next table of contents typeset by **etoc** in compatibility mode.

---

<sup>47</sup><http://ctan.org/pkg/tocloft>

## 54.7. Compatibility with the memoir class

Release 1.071 has also improved the compatibility with the memoir class: its appendix level has been made known to **etoc**. It is at the same level as chapter, thus the chapter line style should possibly do a test for some user defined boolean whose activation may be added to the .toc file at the suitable location via `\addtocontents{toc}{.}`, if one needs to distinguish the two kinds of divisions.

## 54.8. Compatibility with package tocvsec2

**etoc** used to be incompatible with package tocvsec2<sup>48</sup>; it now cohabits, sort of, as it deactivates tocvsec2's modification of `\tableofcontents` (done in the style above) and also cancels its other toc-related macros, but reimplements partially their functionality with `\etocsettocdepth.toc`. By the way, at least two latex runs are necessary for new uses of this command in a document to have an effect in tables of contents.

## 54.9. Compatibility with package tableof

It is possible to use simultaneously **etoc** and **tableof**<sup>49</sup>. Release 1.08 of **etoc** requires at least version 1.4a of **tableof**. If `\etocglobaldefs` is put in the preamble, this must be after the loading of package **tableof**. **tableof** command `\nexttocwithtags` should work as expected.

**tableof** commands `\tableof`, `\tablenotof`, ... will typeset the (a priori global) table of contents according to the document class defaults, obeying the **etoc** depth tags; as explained in the **tableof** documentation they do not typeset a TOC title. They should *not* be used in case `\etocglobaldefs` was issued before, except if its scope has been terminated since then, or `\etoclocaldefs` has cancelled its influence.

## 54.10. Compatibility with package tocstyle

Unknown. (not tested a.t.t.o.w).

# 55. T<sub>E</sub>Xnical matters

The `\etocname`, `\etocnumber`, `\etocpage` commands are protected against premature expansion. They contain suitable hyperref links if package **hyperref** is loaded and active for the TOC. The commands `\etoclink` and `\etocifnumbered` are also protected against premature expansion. Also `\etociffirst` and `\etoccontentsline`.

On the other hand `\etocthenname`, `\etocthenumber`, `\etocthepage` are *not* protected against expansion. And neither are `\etocthelinkedname`, `\etocthelinkednumber`, `\etocthelinkedpage`. The latter contain hyperlink data, if present in the .toc file.

The commands such as `\etocsetstyle`, `\etocsetlevel`, `\etocsettocstyle`, `\etocmulticolstyle`, `\etocruledstyle`, `\etocframedstyle` obey L<sup>A</sup>T<sub>E</sub>X's groups. All TOCs are typeset inside groups.

<sup>48</sup><http://ctan.org/pkg/tocvsec2>

<sup>49</sup><http://ctan.org/pkg/tableof>

## 56. Errors and catastrophes

When a `\localtableofcontents` is inserted by the user in the document, a line containing an **etoc** inner command and an identification number is added to the `.toc` file. The correct local table of contents will be displayed only on the next `latex` run.

## 56. Errors and catastrophes

After using `\etocsetstyle` for one level, the remaining uncustomized levels use the **etoc** default styles (those which are activated by `\etocdefaultlines`). One has to make sure that all levels needed for the next table of contents are mutually compatible: in particular the **etoc** default line styles expect each to be started in “vertical mode”.

When using multiple `\tableofcontents` commands in a document, one should beware from adding typesetting instructions directly in the `.toc` file, as they will be executed by **etoc** for all TOCs: even for a `\localtableofcontents` it doesn’t matter if that instruction seems to concern material outside of its scope, it will get executed nevertheless. If absolutely necessary (but this should never be) these instructions should be done in such a way that they can be activated or deactivated easily from the document source, as need be.

As is usual with `toc` and `labels`, after each change, one has to run `latex` a certain number of times to let the produced document get its final appearance (at least twice).

# Part X.

## The code

<b>Timestamp</b>	<b>57, p. 93</b>
<b>Change history</b>	<b>58, p. 93</b>
<b>Implementation</b>	<b>59, p. 96</b>

### 57. Timestamp

This is the documentation as of 2017/10/23, printed from the source file with the time stamp Time-stamp: <23-10-2017 at 12:03:59 CEST>. The package version is v1.08l, of 2017/10/23.

### 58. Change history

v1.08l [2017/10/23]

Workaround an issue with Emacs/AUCTeX wrongly reporting about actually non-existent LaTeX errors, which was triggered by some strings written (indirectly) to log file by etoc under some circumstances.

v1.08k [2017/09/28]

Adds `\etocsetlocaltop.toc`. See corresponding manual section for details.

Adds `\etocsavedparttocline`, `\etocsavedchaptertocline`, `\etocsavedsectiontocline`, ... They can be used in the context of the technique explained in section "Another compatibility mode".

Formerly, etoc redefined for the duration of the TOC the memoir macro `\chapternumberline` and its likes to have same meaning as `\numberline` (of course, not when executed in compatibility mode), for the sake of extraction of `\etocnumber`.

New method detects presence of any `<foo>numberline` macro without any change to originals; they can thus be used as is when applying the approach of "Another compatibility mode" section from manual.

v1.08j [2017/09/21]

Since 1.08a-2015/03/13 `\etocname`, `\etocnumber`, `\etocpage` contain, if `hyperref` is present and configured for using hyperlinks in the TOC, the link destination in already expanded form. This means one can use them even if the style

closes a group (for example from a `&` in a tabular), if `\etocglobaldefs` was issued; also one can save their meaning for delayed usage (with for example `\LetLtxMacro` as they are robust).

But for some legacy reason `\etoclink`, contrarily to `\etocthelink`, was handled differently. Now, `\etoclink` also contains the link destination in already expanded form, and can thus be used even if the line style issues a `&`, as long as `\etocglobaldefs` is issued.

Also, bugs dating back to the early days of the package, but surfacing only under relatively rare conditions such as usage of `hyperref` with its option `"linktoc=page"` got fixed.

v1.08i [2016/09/29]

This fixes an issue dating back to 1.08e-2015/04/17: under `\etocchecksempiness` regime, some circumstances (such as adding to an already compiled document a `\localtableofcontents` before the main `\tableofcontents`) created an "Undefined control sequence `\Etoc@localtop`" error. Thanks to Denis Bitouzé for reporting the problem.

On this occasion, `\etocdoesnotcheckemptiness` has been added to unset the flag.

A rather more exotic issue was fixed: the emptiness check for local tocs could get confused if the `tocdepth` counter was varying in some specific ways from inside the toc file.

After adding to a document a `\localtableofcontents`, two LaTeX passes are needed for etoc

## 58. Change history

to get a chance to print the correct local contents. Formerly, etoc issued a Warning on the first pass; it now also induces LaTeX into announcing "There were undefined references", as this is nearer to the end of the log file and console output.

v1.08h [2016/09/25]

New functioning of `\etocsetnexttocdepth`: the tocdepth counter is modified only at the time of the table of contents, not before. This fixes an issue which arose when `\etocsetnexttocdepth` was used multiple times with no intervening table of contents. Thanks to Denis Bitouzé for reporting the problem.

The PDF documentation includes about 25 LaTeX code snippets also as file attachment annotations, additionally to their verbatim typesetting. The ordering of the documentation contents has been slightly re-organized.

A previous documentation-only update on 2016/09/09 added a new section with the (approximate) translation into etoc lingua of the book class toc style, for easy customizability.

v1.08g [2015/08/29]

Downgraded to a mere info message the etoc-issued warning (relative to `\settocdepth/\maxtocdepth`) under class memoir.

v1.08f [2015/04/28]

Minor changes to the documentation. `\etocsetlevel` more economical.

v1.08e [2015/04/17]

The command `\etocchecksemtiness` tells etoc to not print, from that point on, the headings of the local tables of contents if they have empty contents. This is mainly for class authors who might want to have their `\section` or `\chapter` automatically do a `\localtableofcontents`. Could prove also useful for batch conversions of documents. Thanks to Paul Gaborit who asked for such a feature.

The command `\etocnotocifnotoc` extends this behaviour to global TOCs: indeed why should documents with no sectioning units take this as an excuse not to use package etoc ?

The command `\etocifwasempty{yes}{no}` can be used for suitable extra action.

A `\tableofcontents\ref{foo}` now expects foo to be a label to a `_local_` TOC. The use with foo a label to a `_global_` TOC is not supported anymore as it had no utility and made the code more complex.

The syntax `\localtableofcontents\ref{foo}` is now accepted as a synonym to the earlier syntax `\tableofcontents\ref{foo}`.

v1.08d [2015/04/09]

Translation into German of the additions made to the documentation for the 1.08x series of releases.

Thanks to Christine Römer!

v1.08c [2015/03/30]

- removed a few unneeded `\long` from the code.
- removed use of `\arabic` at one location of the code, as it may get redefined by some language modules for babel or polyglossia.

v1.08b [2015/03/18]

Bug fixes:

- extra space token removed from `'\localtableofcontents'` (showed only for inline TOCs.)
- `\etocpartname` (a macro used by the package own default line styles) was defined to be `\partname`, but this is not compatible at least with babel+ french context. Now simply expands to `Part`.
- some problems fixed in the German documentation.
- [2015/03/28] some more problems fixed in the documentation. Added mention of `\etocarticlestyle` and `\etocbookstyle`.

v1.08a [2015/03/13]

`\etocname`, `\etocnumber` and `\etocpage` are now the robust variants of `\etocthelinkedname`, `\etocthelinkednumber` and `\etocthelinkedpage`. This should arguably have been done since the addition of the latter to etoc with v1.07f [2013/03/07]. The earlier robust commands `\etocname` etc... contained the hyperlink destination only in an unexpanded form.

The documentation has a brand new title page and a new section The TOC as a TikZ mind map both illustrating further uses of etoc to display tables of contents as trees in an automatic manner.

v1.08 [2015/03/10]

`\etocskipfirstprefix` may now appear anywhere in the `<start>` part of a level style.

New commands `\etociffirst`, `\etocxiffirst`, `\etocxifnumbered`, `\etocglobaldefs` and `\etoclocaldefs`.

It is now possible to issue line style specifications directly with `&` and `\\` tokens, in order to typeset a TOC as a tabular or longtable with the opening for example in the first argument of `\etocsettocstyle` and the closing in its second argument.

It is mandatory for such uses to issue `\etocglobaldefs` which tells etoc to proceed globally for certain definitions. This is also useful in the context of the inline environments of package enumitem.

On this occasion, various old parts of the code have been improved.

v1.07n [2015/03/05]

No more use of `\toks@` when `etoc` constructs `\etocthelinkedname` etc... Thus `\toks@` can be put in the line styles in order to accumulate information. Only useful if it is certain nothing else will change `\toks@` either.

In the documentation: list of main commands now in alphabetic order.

v1.07m [2015/01/23]

Reading of `.toc` file is delayed to `\begin{document}` to account for possible Babel active characters used therein. Thanks to Denis Bitouzé who reported a Babel related problem.

Improved global toc display emulation under KOMA-script classes.

New command `\etocbeforetitlehook`. New command `\etocdisplay`.

v1.07l [doc of 2014/04/29]

Added to the documentation an example of use of `\etocthelinkedname` together with an `enumitem` inline `itemize*` environment; moved main TOC to immediately after the title, and license to the first pages.

Incorporation of the translation into German done on the initiative of Christine Römer by Felix Baral-Weber, Jenny Rothkrämer-Vogt, Daniel Büttner, Claudia Dahl, Christian Otto and Christine Römer (FSU Jena). My grateful thanks to all!

v1.07l [2014/04/22]

Fixes a bug with the 1.07k compatibility layer with `tocloft` which had broken the 1.07k (sic) compatibility with `memoir` (yes, `memoir` class 1.07k testing had been done before adding the `tocloft` thing to the source code . . . ). Also, `etoc` when detecting `tocvsec2` now checks if this is under the `memoir` class, as then nothing special needs to be done to rescue `\tableofcontents`, contrarily to the situation with the native `tocvsec2`.

v1.07k [2014/03/06]

Compatibility with package `tocloft`; and improved compatibility with class `memoir`. Novel TOC example in Overview.

v1.07j [2013/12/03]

Some issues with the documentation formatting (now two-sided) have been addressed, and a novel documentation section “Typesetting the TOC as a table” has been added. Very minor code change (`\Etoc@readtoc`).

v1.07i [2013/10/21]

Changes to the `\etocmulticolstyle` and `\etocruledstyle` codes to lessen the risk of a page break after the title (in the one-column case).

v1.07h [2013/10/16]

New commands `\etocdepthtag.toc`, `\etocsettagdepth`, `\etocobeydepthtags`, `\etocignoredepthtags`.

v1.07g [2013/10/13]

New commands `\etocsettocdepth`, `\etocsettocdepth.toc`, `\etocobeytoctocdepth`, `\etocignoretoctocdepth` which emulate part of `tocvsec2` functionality ; measures to make `tocvsec2` partially compatible with `etoc`.

New commands `\etocsetnexttocdepth`, `\invisibletableofcontents`, `\invisiblelocaltableofcontents`.

Switched from `tikz-qtree` to `forest` for the first ‘toc as tree’ example.

Command names are linked to their descriptions, and many other changes in the documentation.

Removed printing of temporary message when the local toc id is not yet stabilized; indeed `\localtableofcontents` can have many uses, such as filling up some token list register and one may wish to not have anything typeset, even in an intermediate run.

All of `tex etoc.dtx`, `etex etoc.dtx`, `xetex etoc.dtx`, `latex etoc.dtx`, `pdflatex etoc.dtx` are now possible, and the extracted file `etoc.tex` allows easy customization of compilation options for the documentation (default is via `dvipdfmx` which produces the smallest file).

v1.07f [2013/03/07]

New macros `\etocthelinkedname`, `\etocthelinkednumber`, `\etocthelinkedpage`, and `\etocthelink`.

v1.07e [2013/03/01]

Improvements in the package own line styles with regards to penalties and vertical spaces.

Addition to the documentation of an example of a tree-like table of contents (uses `tikz`).

More such examples added 2013/03/03.

v1.07d [2013/02/24]

Minor code improvements and new documentation section “Another compatibility mode”.

v1.07b [2013/02/02]

Removal of the `\xspace` from the macros `\etocname`, `\etocnumber`, `\etocpage`.

Additional examples in the documentation.

## 59. Implementation

v1.07 [2013/01/29]

New commands:

`\etocthenname`, `\etocthenumber`, `\etocthe-`  
`page`, `\etoclink`,  
`\etoccontentsline`, `\etoccontentsline*`  
`\etocnopar`, `\etocaftercontentshook`

Modified command: `\etocmulticolstyle`

New documentation section “Surprising uses of etoc” which explains how to do “Lists of arbitrary things”, in addition to the tables of contents.

v1.06 [2012/12/07]

The standard macros `\@section` etc... are modified only during the calls to `\tableofcontents`; they can thus be customized as will by the user (with the help of a package like `tocloft`) and this will be taken into account by etoc for the TOCs typeset in compatibility mode.

v1.05 [2012/12/01]

`\localtableofcontents` replaces `\tableofcontents*` (for compatibility with the memoir class).

Compatibility with KOMA-script and memoir document classes.

v1.04 [2012/11/24]

A (possibly local) table of contents can be labeled:

`\tableofcontents \label{toc:1}`

and reproduced elsewhere in the document (with a possibly completely different layout):

`\tableofcontents \ref{toc:1}`

v1.02 [2012/11/18]

Initial version.

## 59. Implementation

```
1 \NeedsTeXFormat{LaTeX2e}
2 \ProvidesPackage{etoc}
3 [2017/10/23 v1.08l Completely customisable TOCs (JFB)]
4 \RequirePackage{multicol}
5 \DeclareOption*{\PackageWarning{etoc}{Option ‘\CurrentOption’ is unknown.}}
6 \ProcessOptions\relax
```

1.08 suppresses `\ifEtoc@part` and defines `\ifEtoc@skipprefix`. It also defines `\etocglobaldefs`, `\etoclocaldefs` and `\Etoc@global`.

```
7 \newtoks\Etoc@toctoks
8 \def\Etoc@par{\par}
9 \def\etocinline{\def\Etoc@par{}}
10 \let\etocnopar\etocinline
11 \def\etocdisplay{\def\Etoc@par{\par}}% 1.07m 2015/01/23
```

`\etocglobaldefs` should be used only for special things such as TOC as a table; it should be put in a group to limit its scope. If used in the preamble, it must come *after* `\tableof` if the latter is loaded too.

```
12 \let\Etoc@global\@empty % 1.08 2015/03/10
13 \def\etocglobaldefs{\let\Etoc@global\global\let\tof@global\global}
14 \def\etoclocaldefs {\let\Etoc@global\@empty\let\tof@global\@empty}
15 \newif\ifEtoc@jj % book
16 \newif\ifEtoc@j % part
17 \newif\ifEtoc@ % chapter
18 \newif\ifEtoc@i % section
19 \newif\ifEtoc@ii % subsection
20 \newif\ifEtoc@iii % subsubsection
21 \newif\ifEtoc@iv % paragraph
22 \newif\ifEtoc@v % subparagraph
23 \newif\ifEtoc@number
24 \newif\ifEtoc@hyperref
25 \newif\ifEtoc@parskip % 1.07d
26 \newif\ifEtoc@tocwithid
```



```

27 \newif\ifEtoc@standard
28 \newif\ifEtoc@skipprefix % 1.08
29 \newif\ifEtoc@isfirst    % 1.08
30 \newif\ifEtoc@localtoc
31 \newif\ifEtoc@skipthisone
32 \newif\ifEtoc@stoptoc
33 \newif\ifEtoc@notactive
34 \newif\ifEtoc@mustclosegroup
35 \newif\ifEtoc@emptytoc    % 1.08e
36 \newif\ifEtoc@checksemtiness % 1.08e. Default is 'no checks'.
37 \def\etocchecksemtiness    {\Etoc@checksemtinesstrue }
38 \def\etocdoesnotchecksemtiness {\Etoc@checksemtinessfalse }% 1.08i
39 \newif\ifEtoc@notocifnotoc % 1.08e
40 \def\etocnotocifnotoc {\Etoc@checksemtinesstrue\Etoc@notocifnotoctrue }
41 \def\etoc@{\etoc@}

```

1.07g uses a second counter; this could be avoided, but ok, let's not be that strict.

```

42 \newcounter{etoc@tocid}
43 \newcounter{etoc@tocdepth}% 1.07g

```

2015/03/08 Why do I use jj in the flags and minustwo here? and is there an entity full of forgiveness out there to save my soul even if taking into consideration my use of single and doubled @'s ? Although I desperately need to waste time I miraculously decided not to change everything today and thus the somewhat spendious set-up below is maintained.

```

44 \@ifclassloaded{memoir}{\def\Etoc@minf{-\thr@@}}{\def\Etoc@minf{-\tw@}}
45 \def\Etoc@@minustwo@@{-\tw@}
46 \let\Etoc@@minusone@@\m@ne
47 \let\Etoc@@zero@@      \z@
48 \let\Etoc@@one@@       \@ne
49 \let\Etoc@@two@@       \tw@
50 \let\Etoc@@three@@      \thr@@
51 \chardef\Etoc@@four@@ 4
52 \chardef\Etoc@@five@@ 5
53 \chardef\Etoc@@six@@ 6
54 %\let\Etoc@localtop\Etoc@@minustwo@@ % 1.08e does it at each local TOC.
55 \def\Etoc@@minustwo@{minustwo}
56 \def\Etoc@@minusone@{minusone}
57 \def\Etoc@@zero@    {zero}
58 \def\Etoc@@one@     {one}
59 \def\Etoc@@two@     {two}
60 \def\Etoc@@three@   {three}
61 \def\Etoc@@four@    {four}
62 \def\Etoc@@five@    {five}

```

1.07g.

```

63 \expandafter\def\csname Etoc@-3@@\endcsname {-\thr@@}
64 \expandafter\let\csname Etoc@-2@@\endcsname \Etoc@@minustwo@@
65 \expandafter\let\csname Etoc@-1@@\endcsname \Etoc@@minusone@@
66 \expandafter\let\csname Etoc@0@@\endcsname \Etoc@@zero@@
67 \expandafter\let\csname Etoc@1@@\endcsname \Etoc@@one@@
68 \expandafter\let\csname Etoc@2@@\endcsname \Etoc@@two@@
69 \expandafter\let\csname Etoc@3@@\endcsname \Etoc@@three@@
70 \expandafter\let\csname Etoc@4@@\endcsname \Etoc@@four@@
71 \expandafter\let\csname Etoc@5@@\endcsname \Etoc@@five@@
72 \expandafter\let\csname Etoc@6@@\endcsname \Etoc@@six@@
73 \let\Etoc@all@@ \Etoc@@five@@
74 \let\Etoc@none@@ \Etoc@minf

```

Versions earlier than 1.08 did \Etoc@newlevel in a quite embarrassing way with an \edef.

## 59. Implementation

1.08f's `\etocsetlevel` avoids multiple insertions of the same level in the `\Etoc@levellist` list.

1.08k modified `\Etoc@newlevel`: formerly a token `\l@#1` was stored in place of braced name `{#1}`. This changed is to facilitate creation of the `\etocsavedsectiontocline` etc... macros later on, at each TOC. But the stored `#1` is not expanded immediately, although it is in other parts of `\etocsetlevel`, which may cause timing problems if `#1` is a macro.

```
75 \let\Etoc@levellist\@empty
76 \def\Etoc@newlevel #1{\expandafter\def\expandafter\Etoc@levellist\expandafter
77   {\Etoc@levellist\Etoc@levellist@elt{#1}}}
78 \def\etocsetlevel#1#2{%
79   \ifundefined{Etoc@#1@@}{\Etoc@newlevel{#1}}}%
80   \ifcase#2\relax
81     \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@zero@@
82     \expandafter\let \csname Etoc@#1@\endcsname\Etoc@@zero@
83   \or
84     \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@one@@
85     \expandafter\let \csname Etoc@#1@\endcsname\Etoc@@one@
86   \or
87     \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@two@@
88     \expandafter\let \csname Etoc@#1@\endcsname\Etoc@@two@
89   \or
90     \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@three@@
91     \expandafter\let \csname Etoc@#1@\endcsname\Etoc@@three@
92   \or
93     \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@four@@
94     \expandafter\let \csname Etoc@#1@\endcsname\Etoc@@four@
95   \or
96     \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@five@@
97     \expandafter\let \csname Etoc@#1@\endcsname\Etoc@@five@
98   \or
99     \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@six@@
100   \else
101     \ifnum#2=\m@ne
102       \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@minusone@@
103       \expandafter\let \csname Etoc@#1@\endcsname\Etoc@@minusone@
104     \else
105       \ifnum#2=-\tw@
106         \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@minustwo@@
107         \expandafter\let \csname Etoc@#1@\endcsname\Etoc@@minustwo@
108       \else
109         \PackageWarning{etoc}
110           {unexpected value ‘#2’ in \string\etocsetlevel.^}%
111           Should be -2,-1, 0, 1, 2, 3, 4, 5, or 6. Set to 6 (=ignored)}%
112         \expandafter\let\csname Etoc@#1@@\endcsname\Etoc@@six@@
113       \fi
114     \fi
115   \fi
116 }
117 \etocsetlevel{book}{-2}
118 \etocsetlevel{part}{-1}
119 \etocsetlevel{chapter}{0}
120 \etocsetlevel{section}{1}
121 \etocsetlevel{subsection}{2}
122 \etocsetlevel{subsubsection}{3}
123 \etocsetlevel{paragraph}{4}
124 \etocsetlevel{subparagraph}{5}
```

The first two parameters of `\Etoc@et@hop` are exchanged in 1.08 compared to earlier versions; this is in relation to changes done in `\Etoc@etoccontentsline`.

1.08i as a fourorthree rather than as formerly a sixorfive.

```

125 \AtBeginDocument{%
126 \ifpackageloaded{parskip}{\Etoc@parskiptrue}{}%
127 \ifpackageloaded{hyperref}{\Etoc@hyperreftrue
128             \def\Etoc@et@hop#1#2#3#4#5{#2{#3}{#4}{#5}#1}%
129             \long\def\Etoc@gobblefourorthree#1#2#3#4{}}
130             {\def\Etoc@et@hop#1#2#3#4{#2{#3}{#4}#1}%
131             \long\def\Etoc@gobblefourorthree#1#2#3{}}%
132 }
```

2015/03/08 One has to be careful about the `\Etoc@end@<level>` user defined macros which may close groups. This has always been the reason for the `\global`'s here and elsewhere.

1.08 uses `\Etoc@level` rather than a `#1` which used to be anyhow always the `\Etoc@tmp`. Now `\Etoc@level` is what was called formerly `\Etoc@tmp` in `\Etoc@etoccontentsline`.

Attention that `\Etoc@setflags` must not modify `\Etoc@next`!

```

133 \def\Etoc@setflags {%
134     \ifcase \Etoc@level
135         \global\Etoc@vfalse
136         \global\Etoc@ivfalse
137         \global\Etoc@iiifalse
138         \global\Etoc@iifalse
139         \global\Etoc@ifalse
140         \global\Etoc@true
141     \or
142         \global\Etoc@vfalse
143         \global\Etoc@ivfalse
144         \global\Etoc@iiifalse
145         \global\Etoc@iifalse
146         \global\Etoc@ittrue
147     \or
148         \global\Etoc@vfalse
149         \global\Etoc@ivfalse
150         \global\Etoc@iiifalse
151         \global\Etoc@iittrue
152     \or
153         \global\Etoc@vfalse
154         \global\Etoc@ivfalse
155         \global\Etoc@iiittrue
156     \or
157         \global\Etoc@vfalse
158         \global\Etoc@ivtrue
159     \or
160         \global\Etoc@vtrue
161     \else
162         \global\Etoc@vfalse
163         \global\Etoc@ivfalse
164         \global\Etoc@iiifalse
165         \global\Etoc@iifalse
166         \global\Etoc@ifalse
167         \global\Etoc@false
168         \ifnum\Etoc@level=\m@ne
169             \global\Etoc@jtrue
170         \else
171             \global\Etoc@jfalse
172             \global\Etoc@jjtrue

```

## 59. Implementation

```

173     \fi
174     \fi
175 }

```

1.08 This was originally in `\Etoc@etoccontentsline`, intermixed with the definitions of `\Etoc@contents` and `\Etoc@prefix`. Separated here for clarity of code. One has to be careful that this may close groups. The conditionals are all global. The `\Etoc@level` has been `\global\let` to a `\chardef` variable. The final `\else` of the `\ifcase` assumes none of the `\Etoc@end@. .` routines modify it, naturally.

```

176 \def\Etoc@doends {%
177     \ifcase \Etoc@level
178         \ifEtoc@v \Etoc@end@five \fi
179         \ifEtoc@iv \Etoc@end@four \fi
180         \ifEtoc@iii\Etoc@end@three\fi
181         \ifEtoc@ii \Etoc@end@two \fi
182         \ifEtoc@i \Etoc@end@one \fi
183     \or
184         \ifEtoc@v \Etoc@end@five \fi
185         \ifEtoc@iv \Etoc@end@four \fi
186         \ifEtoc@iii\Etoc@end@three\fi
187         \ifEtoc@ii \Etoc@end@two \fi
188     \or
189         \ifEtoc@v \Etoc@end@five \fi
190         \ifEtoc@iv \Etoc@end@four \fi
191         \ifEtoc@iii\Etoc@end@three\fi
192     \or
193         \ifEtoc@v \Etoc@end@five \fi
194         \ifEtoc@iv \Etoc@end@four \fi
195     \or
196         \ifEtoc@v \Etoc@end@five \fi
197     \or
198     \else
199         \ifEtoc@v \Etoc@end@five \fi
200         \ifEtoc@iv \Etoc@end@four \fi
201         \ifEtoc@iii\Etoc@end@three\fi
202         \ifEtoc@ii \Etoc@end@two \fi
203         \ifEtoc@i \Etoc@end@one \fi
204         \ifEtoc@ \Etoc@end@zero \fi
205         \ifnum \Etoc@level =-\tw@
206             \ifEtoc@j \Etoc@end@minusone\fi
207         \fi
208     \fi
209 }

```

1.08 the `\Etoc@next` will (possibly) execute the begin code, *before* `\Etoc@contents` and `\Etoc@prefix` get defined, contrarily to what I did in earlier versions. I still reset the flags before execution of the begin codes (although the flags are not supposed to be used therein anyhow).

Also the `<begin>` code does `\Etoc@isfirsttrue`.

1.08k This was originally in `\Etoc@etoccontentsline`. Extracted for re-use by `\etocsetlocaltop.toc`. Calls `\Etoc@setflags` which must not modify `\Etoc@next`.

```

210 \def\Etoc@dobegins{%
211     \let\Etoc@next\@empty
212     \Etoc@global\Etoc@isfirstfalse

```

2015/03/08 If my naming scheme had not been so baroque, I could easily do things in a more concise way here...

```

213     \ifcase\Etoc@level
214         \ifEtoc@

```

```

215         \else
216             \def\Etoc@next{\Etoc@begin@zero}%
217         \fi
218     \or
219         \ifEtoc@i
220         \else
221             \def\Etoc@next{\Etoc@begin@one}%
222         \fi
223     \or
224         \ifEtoc@ii
225         \else
226             \def\Etoc@next{\Etoc@begin@two}%
227         \fi
228     \or
229         \ifEtoc@iii
230         \else
231             \def\Etoc@next{\Etoc@begin@three}%
232         \fi
233     \or
234         \ifEtoc@iv
235         \else
236             \def\Etoc@next{\Etoc@begin@four}%
237         \fi
238     \or
239         \ifEtoc@v
240         \else
241             \def\Etoc@next{\Etoc@begin@five}%
242         \fi
243     \else
244         \ifnum\Etoc@level=\m@ne
245             \ifEtoc@j
246             \else
247                 \def\Etoc@next{\Etoc@begin@minusone}%
248             \fi
249         \else
250             \ifEtoc@jj
251             \else
252                 \def\Etoc@next{\Etoc@begin@minustwo}%
253             \fi
254         \fi
255     \fi
256     \Etoc@setflags
257     \Etoc@next
258 }

```

Heart of **etoc** hack into execution of .toc file. The .toc file must contain (at least indirectly via expansion) `\contentsline {<unit>}... instructions`, with the standard meaning of executing `\l@unit` (the hyperref modifications are taken into account). It is the various `\l@unit`'s which are (locally) `\let to \Etoc@lxyz` and it is `\Etoc@lxyz` which does the parsing of the arguments in order to extract name, number and page number. Once they are obtained, via the execution of this indirectly hacked `\contentsline`, the code executes the `\Etoc@prefix` and `\Etoc@contents` corresponding to that unit level, according to the user `\etocsetstyle` declarations. Before that the begin part is executed when **etoc** first encounters that level coming from a higher one and the end part will be executed when hitting a more important level.

`\etocskipfirstprefix` was already part of the very first release of **etoc**. Up to 1.07n it was just an elegant `\@thirdofthree` possibly executed just before a `\@firstoftwo{\Etoc@swa{#1}}{\Etoc@swb{#1}}` line. But this made it impossible for me to execute things after the begin parts.

## 59. Implementation

Executing stuff before `begin` is now a potential problem because the user can close some group in the `begin` code (this was in fact impossible in versions <1.08 because this would have erased the definitions of `\Etoc@contents` and `\Etoc@prefix`). Thus I am not so free now to define things before execution of `begin`, if I allow people to close a group therein, except if I make them global.

The mechanism of `\etocskipfirstprefix` made it impossible to add anything at the end of the `begin` codes. Thus I changed it in 1.08 to simply work with a (global) boolean flag. This gives user the freedom to use `\etocskipfirstprefix` anywhere in the `begin` code and it made room for a delimited macro I called `\etoconlyonfirst`. But then I dropped it in favor of definition of the `\etociffirst` conditional.

Macros `\Etoc@swa` and `\Etoc@swb` which did the `\Etoc@et@hop` thing have been suppressed, their effect is incorporated to the end of `\Etoc@etoccontentsline`.

```
259 \def\etocskipfirstprefix {\global\Etoc@skipprefixtrue }
```

2015/03/08 let's record that `\Etoc@level` used to be called `\Etoc@tmp` in versions earlier than 1.08 (and `\Etoc@tmp` is used in the completely different ulterior `\Etoc@lxyz` context). The code sets it for example to be `\Etoc@section@@` which has been let by `\etocsetlevel` to `\@ne`, thus is always self-terminating in `\ifnum... \fi` contexts.

1.08i-2016/09/29 makes some changes in the organization of the conditionals. It could happen in certain special circumstances (created by that part in `\Etoc@localtableofcontents` which is executed after the test of emptiness) that `\Etoc@localtop` was still undefined. Earlier version of the code was a bit lazy and did not avoid some unneeded `\ifnum` with `\Etoc@localtop` in context where the flag `\ifEtoc@stoptoc` was `\iftrue`; this was not an issue when code was first written, `\Etoc@localtop` was always initialized back then. But the code in `\Etoc@localtableofcontents` handling local TOCs with unrecognized ids would call execution of the `.toc` file under some special context with flag `\ifEtoc@stoptoc` artificially set to true. If `\Etoc@localtop` had not been initialized an undefined control sequence error would occur. Furthermore as `\ifEtoc@notactive` was also set to true, the case of no error leads then to unneededly expand `\Etoc@setflags` code slowing down compilation.

Also 1.08i splits the macro into two, with accompanying slight refactoring. I had such problems with the behaviour of TeX booleans in 2011/12 that I got excessively cautious then, and was using an `\Etoc@next` for the `\Etoc@setflags` rather than leaving it like here deep inside conditional branches.

```
260 \def\Etoc@etoccontentsline #1{%
261   \global\expandafter\let\expandafter\Etoc@level\csname Etoc@#1@@\endcsname
262   \Etoc@skipthisonefalse
263   \ifnum\Etoc@level=\Etoc@@six@@
264     \Etoc@skipthisonetrue
265   \else
266     \ifEtoc@localtoc
267       \ifEtoc@stoptoc
268         \Etoc@skipthisonetrue
269       \else
270         \ifEtoc@notactive\Etoc@setflags
271           \Etoc@skipthisonetrue
272         \else
273           \ifnum\Etoc@level<\Etoc@localtop
274             \Etoc@skipthisonetrue
275           \global\Etoc@stoptoctrue
276         \fi
277       \fi
278     \fi
279   \fi
280 \fi
281 \let\Etoc@next\Etoc@gobblefourorthree
282 \ifEtoc@skipthisone
```

```

283 \else
284 \ifnum\c@tocdepth<\Etoc@level
285 \else
286 \ifEtoc@standard
287 \let\Etoc@next\Etoc@savedcontentsline
288 \else
289 \let\Etoc@next\Etoc@etoccontentsline@
290 \fi
291 \fi
292 \fi
293 \Etoc@next{#1}%
294 }

```

1.08 New way to handle the compatibility mode. One has to worry only for the local tables of contents (in inactive state, which still must be parsed for global level flags) but in this case `\Etoc@setflags` was done above already. Earlier method proceeded with dummy empty line styles.

2015/03/08 Ever since the first release of **etoc**, the code has to be careful that the `\Etoc@end@<level>` user defined macros may close groups. This is the reason why some assignments have to be done globally. The `\let\Etoc@next\Etoc@gobblesixorfive` above could be local, because it is used only in case the stuff below is skipped, inclusive of `\Etoc@doends` which may close groups. (code has changed since, slightly).

1.08 extracts to a separate macro `\Etoc@doends`. Earlier code did `\global\let\Etoc@next\relax` but we can now give a default `\relax` value to `\Etoc@next` with no need of a `\global` prefix.

1.08k extracts `\Etoc@doends`.

```

295 \def\Etoc@etoccontentsline@ #1{%
296 \Etoc@doends % separated since 1.08
297 \Etoc@doends % separated since 1.08k. Does \Etoc@setflags.

```

1.08:

1. separates the definition of `\Etoc@prefix` and `\Etoc@contents` from the execution of the end parts. And it does the definition with some `\csname`'s.
2. implements `\etocskipfirstprefix` with a boolean, this is less slick but more flexible than the earlier method.
3. defines the `\etociffirst` conditional.
4. has `\Etoc@level` rather than `\Etoc@tmp` and thus we can dispense with the `\ifEtoc@part` switch as we will be able to test the equality `\Etoc@level=-1` inside of `\Etoc@lxyz`.
5. adds the `\Etoc@global` prefix.

1.08i uses an `\edef` rather than `\def` with a chain of `\expandafter`.

```

298 \Etoc@global\edef\Etoc@prefix {\expandafter\noexpand
299 \csname Etoc@prefix@\csname Etoc@#1@\endcsname\endcsname }%
300 \Etoc@global\edef\Etoc@contents{\expandafter\noexpand
301 \csname Etoc@contents@\csname Etoc@#1@\endcsname\endcsname }%
302 \ifEtoc@skipprefix \Etoc@global\let\Etoc@prefix\empty\fi
303 \global\Etoc@skipprefixfalse

```

1.08 modifies the ending to not use the `\Etoc@swa` and `\Etoc@swb` things. With this version, the `\Etoc@next` here is only to gobble everything if it has to (or gobble only the **etoc** things, when in compatibility mode).

For the record we don't do `\csname l@#1\endcsname` mainly because `hyperref` patches `\contentsline`.

The location where `\Etoc@savedcontentsline` is `\let` to `\contentsline` has been moved with release 1.08 to before the title as set by `\etocsettoctyle`, thus it is now possible to open

## 59. Implementation

a tabular in this part. And with the `\Etoc@global` prefix the user may opt for global definitions of `\etocname` and the like, thus use freely `&` and other group closing things. Also, some care has been put into doing only expandable things after expansion of the *<prefix>* and *<contents>* parts of a sectioning unit style.

```
304 \Etoc@et@hop{\Etoc@prefix\Etoc@contents}{\Etoc@savedcontentsline{#1}}%
305 }
```

### START OF MAINLY OBSOLETE COMMENTS

2013/03/07: up to 1.06 **etoc** defined only `\etocname`, `\etocnumber` and `\etocpage`. The `hyperref` added data is recycled in the simplest manner, prefixing it with `\leavevmode`. The included `\Hy@tocdestname` is (was, see below 1.07f and 1.08a) left unexpanded. We have to spend some time with delimited macros to dis-entangle the `.toc` data, and reconstruct the possible `hyperref` data. If the page number is not hyperlinked, `\etocpage` does *not* add the link found possibly in the name.

1.07 adds `\etocthename`, `\etocthenumber`, `\etocthepage` which are left fragile and do not have the links data, and `\etoclink{<linkname>}` which is robust and reconstructs an arbitrarily named link.

A need (for things like building up a token list to be used in a `tikzpicture`) arose later to have some form of the link which could be saved by a simple command like one can do `\global\let\lastname\etocthename`, and avoid having to manipulate `\Hy@tocdestname`. So 1.07f adds `\etocthelinkedname`, `\etocthelinkednumber`, `\etocthelinkedpage`, `\etocthelink`: they use `\hyperlink` with an expanded `\Hy@tocdestname`.

1.07f also adds `\leavevmode` to `\etoclink` which should have been done earlier, as it was included in `\etocname` etc... attention, `\@namedef{A}{B}` and not `\@namedef{A} {B}` !! on the other hand this gives a simple way to insert a space as the first token in the parameters. For `\Etoc@again` (which appears later in the code), a `\@firstofone` construct is however the simplest of all.

### END OF MAINLY OBSOLETE COMMENTS

1.08 adds `\Etoc@global` prefix (2015/03/09).

1.08a drops the original constructions of `\etocname` etc... The obvious decision to convert `\etocname`, etc ... to be the robust versions of `\etocthelinkedname`, etc ... was delayed two years (from 1.07f 2013/03/07 to 1.08a 2015/03/13) for sentimental reason. But the original versions contained the `hyperref` destination only in unexpanded form, this forced to use `\etocthelinkedname`, etc... in cases when the thing had to be stored for delayed use. Now the user can do this directly with `\etocname`, `\etocnumber`, `\etocpage` (taking into account though that these are robust macros, thus perhaps using something like `\LetLtxMacro`.)

1.08j turns `\etoclink` into the robust version of `\etocthelink`. Indeed so far only `\etocthelink` contained `\Hy@tocdestname` expanded. But `\etoclink` did not, and thus it could not be used after a group was closed for example by a `&` from the sectioning style, even if having used `\etocglobaldefs`.

```
306 \def\Etoc@lxyz #1#2{%
307   \Etoc@global\let\etocthelink\@firstofone % fall-back, was let \@empty up to 1.08i (inc.)
308   \Etoc@global\def\etocthename      {#1}% (if linked, will be removed later)
309   \Etoc@global\def\etocthelinkedname {#1}% (probably redefined later)
310   \Etoc@getthepage #2\etoc@ % defines \etocthelinkedpage
311   \Etoc@getnb #1\relax\relax\etoc@ % sets number, name, and \etocthelink
312   \ifEtoc@number
313   \else
314     \ifnum\Etoc@level=\m@ne
315       \Etoc@getit #1\hspace\relax\etoc@ % additional job for parts
316     \fi
317   \fi
318   \Etoc@global\expandafter\let\csname etoclink \endcsname\etocthelink
319   \Etoc@global\expandafter\let\csname etocname \endcsname\etocthelinkedname
320   \Etoc@global\expandafter\let\csname etocnumber \endcsname\etocthelinkednumber
```



```

321 \Etoc@global\expandafter\let\csname etocpage \endcsname\etocthelinkedpage
322 }

```

1.08 adds the \Etoc@global's.

1.08j fixes old \ifx #1\hyperlink@start bug.

```

323 \def\Etoc@getthepage #1{%
324 \let\Etoc@next\Etoc@getthepage@nohyp
325 \ifEtoc@hyperref
326 \ifx\hyper@linkstart#1%
327 \let\Etoc@next\Etoc@getthepage@hyp
328 \fi
329 \fi
330 \Etoc@next #1%
331 }
332 \def\Etoc@getthepage@nohyp #1\etoc@ {%
333 \Etoc@global\def\etocthepage {#1}%
334 \Etoc@global\def\etocthelinkedpage {#1}%
335 }

```

1.07k corrects a bug here with an extra space before ##1 in \@namedef{etoclink } ##1. In 99.99% of use cases however, the macro gets redefined at a later stage, thus the bug did not show.

1.07n replaces use of \toks@ by macro \Etoc@tmp. Thus, the user has the (not so recommended) possibility to use \toks@ in the styles, even to store information whose scope extends beyond the execution of each \contentsline.

1.08 adds the \Etoc@global's.

1.08j does not define \etoclink<space> anymore, only \etocthelink which contains the \Hy@tocdestname already expanded.

```

336 \def\Etoc@getthepage@hyp #1#2#3#4#5\etoc@ {%
337 \Etoc@global\def\etocthepage {#4}%
338 \edef\Etoc@tmp {{#3}}%
339 \Etoc@global\expandafter\def\expandafter\etocthelinkedpage\expandafter
340 { \expandafter\hyperlink \Etoc@tmp{#4}}%
341 \Etoc@global\edef\etocthelink##1{\noexpand\hyperlink\Etoc@tmp{##1}}%
342 }

```

1.08j fixes an old buggy \ifx #1\hyper@linkstart. This could show if hyperref was present with linktoc=page option for example, because in that case #1 could be braced tokens. As this is rare condition, it had not been reported so far by my single user user-base.

```

343 \def\Etoc@getnb #1{%
344 \let\Etoc@next\Etoc@getnb@nohyp
345 \ifEtoc@hyperref
346 \ifx\hyper@linkstart#1%
347 \let\Etoc@next\Etoc@getnb@hyp
348 \fi
349 \fi
350 \Etoc@next #1%
351 }
352 \def\Etoc@getit #1{%
353 \let\Etoc@next\Etoc@getit@nohyp
354 \ifEtoc@hyperref
355 \ifx\hyper@linkstart#1%
356 \let\Etoc@next\Etoc@getit@hyp
357 \fi
358 \fi
359 \Etoc@next #1%
360 }

```

1.08 adds the \Etoc@global's.

## 59. Implementation

1.08j corrects an old `\ifx#1\numberline` into `\ifx\numberline#1`.

1.08k uses a delimited macro to identify `\numberline`, `\chapternumberline` etc... in one go, avoiding the `\let\chapternumberline\numberline` etc... which were done formerly in `\Etoc@tableofcontents`. Currently **etoc** does not yet use  $\varepsilon$ -TeX.

```
361 \begingroup\@tfor\x:=NUMBZRLINZ\do{\catcode\expandafter'\x=12 }\lccode'Z 'e
362 \lowercase{\endgroup
363 \long\def\Etoc@oneifnumberline #1NUMBZRLINZ#2#3\Etoc@next{%
364   \ifx\etoc@#20\else1\fi
365 }%
366 \def\Etoc@getnb@nohyp #1#2#3\etoc@ {%
367   \def\Etoc@getname ##1\relax\relax\etoc@ {%
368     \Etoc@global\def\etocthename {##1}%
369     \Etoc@global\def\etocthelinkedname {##1}%
370   }%
371   \if1\expandafter\Etoc@oneifnumberline\string#1NUMBZRLINZ\etoc@\Etoc@next
372     \Etoc@global\def\etocthenumber {#2}%
373     \Etoc@global\def\etocthelinkednumber {#2}%
374     \Etoc@global\Etoc@numbertrue
375     \Etoc@getname #3\etoc@
376   \else % \etocthename and \etocthelinkedname already defined
377     \Etoc@global\let\etocthenumber \@empty
378     \Etoc@global\let\etocthelinkednumber \@empty
379     \Etoc@global\Etoc@numberfalse
380   \fi
381 }%

1.07n replaces use of \toks@ by macro \Etoc@tmp. 1.08 adds the \Etoc@global's.
1.08j corrects the old \ifx#1\numberline into \ifx\numberline#1.
1.08k uses a delimited macro to identify \numberline, \chapternumberline etc... in one go.

382 \def\Etoc@getnb@hyp #1#2#3#4#5#6\etoc@ {%
383   \def\Etoc@getname ##1\relax\relax\etoc@ {%
384     \Etoc@global\def\etocthename {##1}%
385     \edef\Etoc@tmp {{#3}}%
386     \Etoc@global\expandafter\def\expandafter\etocthelinkedname\expandafter
387       {\expandafter\hyperlink \Etoc@tmp{##1}}%
388   }%
389   \def\Etoc@getnbr ##1##2##3\etoc@ {%
390     \if1\expandafter\Etoc@oneifnumberline\string##1NUMBZRLINZ\etoc@\Etoc@next
391       \Etoc@global\def\etocthenumber {##2}%
392       \edef\Etoc@tmp {{#3}}%
393       \Etoc@global\expandafter\def\expandafter\etocthelinkednumber
394         \expandafter{\expandafter\hyperlink \Etoc@tmp{##2}}%
395       \Etoc@global\Etoc@numbertrue
396       \Etoc@getname ##3\etoc@
397     \else
398       \Etoc@global\let\etocthenumber \@empty
399       \Etoc@global\let\etocthelinkednumber \@empty
400       \Etoc@global\Etoc@numberfalse
401       \Etoc@global\def\etocthename {#4}%
402       \edef\Etoc@tmp {{#3}}%
403       \Etoc@global\expandafter\def\expandafter\etocthelinkedname
404         \expandafter {\expandafter\hyperlink \Etoc@tmp{#4}}%
405     \fi
406   }%
407   \Etoc@global\edef\etocthelink ##1{\noexpand\hyperlink {#3}{##1}}%
408   \Etoc@getnbr #4\relax\relax\etoc@
409 }}
```

1.08 adds the \Etoc@global's.

```
410 \def\Etoc@getit@nohyp #1\hspace#2#3\etoc@ {%
411   \def\Etoc@getname ##1\hspace\relax\etoc@ {%
412     \Etoc@global\def\etocthename {##1}%
413     \Etoc@global\def\etocthelinkedname {##1}%
414   }%
415   \ifx\relax#2%
416   \else
417     \Etoc@global\def\etocthenumber {##1}%
418     \Etoc@global\def\etocthelinkednumber {##1}%
419     \Etoc@global\Etoc@numbertrue
420     \Etoc@getname #3\etoc@
421   \fi
422 }
```

1.07n replaces use of \toks@ by macro \Etoc@tmp. 1.08 adds the \Etoc@global's.

1.08j adds \etocthelink definition here which was missing. This bug showed in case hyperref was used with option linktoc=page, for part styles: the \etocthelink would not create a link if used therein. But the linktoc=page condition could easily however trigger the more serious \ifx bug in \Etoc@getnb also fixed at 1.08j, and this would have overshadowed this one by a real compilation error rather than a missing link in the produced PDF.

```
423 \def\Etoc@getit@hyp #1#2#3#4#5#6\etoc@ {%
424   \def\Etoc@getname ##1\hspace\relax\etoc@ {%
425     \Etoc@global\def\etocthename {##1}%
426     \Etoc@global\expandafter\def\expandafter\etocthelinkedname
427       \expandafter{\expandafter\hyperlink \Etoc@tmp{##1}}%
428   }%
429   \def\Etoc@getntr ##1\hspace##2##3\etoc@ {%
430     \ifx\relax##2%
431     \else
432       \Etoc@global\def\etocthenumber {##1}%
433       \Etoc@global\expandafter\def\expandafter\etocthelinkednumber
434         \expandafter{\expandafter\hyperlink \Etoc@tmp{##1}}%
435       \Etoc@global\Etoc@numbertrue
436       \Etoc@getname ##3\etoc@
437     \fi
438   }%
439   \edef\Etoc@tmp {{##3}}%
440   \Etoc@getntr #4\hspace\relax\etoc@
441   \Etoc@global\edef\etocthelink {##1{\noexpand\hyperlink\Etoc@tmp{##1}}%
442 }
```

1.08a of 2015/03/12 does the long-postponed thing to let \etocname etc.. be simply the robust variant of \etocthelinkedname etc...

```
443 \let\etocthename \@empty
444 \let\etocthenumber \@empty
445 \let\etocthepage \@empty
446 \let\etocthelinkedname \@empty
447 \let\etocthelinkednumber \@empty
448 \let\etocthelinkedpage \@empty
449 \let\etocthelink \@firstofone % prior to 1.08j its was \let to \@empty
450 \DeclareRobustCommand*\etocname {}
451 \DeclareRobustCommand*\etocnumber {}
452 \DeclareRobustCommand*\etocpage {}
453 \DeclareRobustCommand*\etoclink {\@firstofone}
454 \DeclareRobustCommand*\etocifnumbered {}
455 {\ifEtoc@number\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi}
```

## 59. Implementation

```
456 \expandafter\let\expandafter\etocxifnumbered\csname etocifnumbered \endcsname
1.08 defines \etociffirst
457 \DeclareRobustCommand*{\etociffirst}
458   {\ifEtoc@isfirst\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi}
459 \expandafter\let\expandafter\etocxiffirst\csname etociffirst \endcsname

1.07j modifies \Etoc@readtoc.
460 \def\Etoc@readtoc {%
461   \ifeof \Etoc@tf
462   \else
463     \read \Etoc@tf to \Etoc@buffer
464     \Etoc@toctoks=\expandafter\expandafter\expandafter
465       {\expandafter\the\expandafter\Etoc@toctoks\Etoc@buffer}%
466     \expandafter\Etoc@readtoc
467   \fi
468 }

1.07m moves the reading of the toc file At Begin Document. Needed for Babel activated characters.
469 \Etoc@toctoks {}% (superfluous, but for clarity)
470 \AtBeginDocument{\IfFileExists{\jobname.toc}
471   {\endlinechar=\m@ne
472     \makeatletter
473     \newread\Etoc@tf
474     \openin\Etoc@tf\@filef@und
475     \Etoc@readtoc
476     \global\Etoc@toctoks=\expandafter{\the\Etoc@toctoks}%
477     \closein\Etoc@tf}}
478   {\typeout{No file \jobname.toc.}}}

1.07d: parskip and \@nobreakfalse stuff moved to \Etoc@tableofcontents
479 \def\Etoc@openouttoc{%
480   \ifEtoc@hyperref
481     \ifx\hyper@last\@undefined
482       \IfFileExists{\jobname .toc}
483       {\Hy@WarningNoLine
484         {old toc file detected, not used; run LaTeX again (cheers from etoc)}}%
485       \global\Etoc@toctoks={}%
486     }
487     {}%
488   \fi
489   \fi
490   \if@filesw
491     \newwrite \tf@toc
492     \immediate \openout \tf@toc \jobname .toc\relax
493   \fi
494   \global\let\Etoc@openouttoc\empty
495 }

1.08 has a new way to handle \Etoc@standardtrue.
Also the expansion of \Etoc@levelist is moved to inside \Etoc@tableofcontents as set-up
by \etocsettocstyle. This opens up the possibility for the user to open a group in the title (for
example start a tabular) which would be closed (by the first &) sometime later. In combination with
\etocglobaldefs it will be now much easier to typeset TOC as tables with etoc!
496 \def\Etoc@toctoc{%
497   \global\Etoc@vfalse
498   \global\Etoc@ivfalse
499   \global\Etoc@iiifalse
500   \global\Etoc@iifalse
```

```

501 \global\Etoc@iffalse
502 \global\Etoc@false
503 \global\Etoc@jfalse
504 \global\Etoc@jjfalse
505 \the\Etoc@toctoks
506 \ifEtoc@notactive
507 \else
508   \ifEtoc@v   \Etoc@end@five\fi
509   \ifEtoc@iv  \Etoc@end@four\fi
510   \ifEtoc@iii \Etoc@end@three\fi
511   \ifEtoc@ii  \Etoc@end@two\fi
512   \ifEtoc@i   \Etoc@end@one\fi
513   \ifEtoc@    \Etoc@end@zero\fi
514   \ifEtoc@j   \Etoc@end@minusone\fi
515   \ifEtoc@jj  \Etoc@end@minustwo\fi
516 \fi
517 }

```

Slightly better coded in 1.08b. Had some bad experiences with TeX conditionals back in 2011/2012 when I started LaTeX programming, and the code was a remnant of extra cautious attitude. Also, now `\etoc@@startlocaltoc` is expandable if it does nothing.

2015/04/16: 1.08e removes a `\ifEtoc@notactive` test as `\ifnum` can be positive only once anyhow; although the `\ifEtoc@notactive` boolean could be faster than an `\ifnum`, it adds to it as long as the local toc id is not yet found.

I replace (1.08e) the `\Etoc@global`'s for `\Etoc@localtop` by `\global`'s. The `\relax` is needed because #2 may sometimes be a count, sometimes tokens.

```

518 \def\etoc@@startlocaltoc#1#2{%
519   \ifEtoc@localtoc
520     \ifnum #1=#2\relax
521       \ifEtoc@jj \global\let\Etoc@localtop\Etoc@@minusone@@ \fi
522       \ifEtoc@j  \global\let\Etoc@localtop\Etoc@@zero@@ \fi
523       \ifEtoc@   \global\let\Etoc@localtop\Etoc@@one@@ \fi
524       \ifEtoc@i  \global\let\Etoc@localtop\Etoc@@two@@ \fi
525       \ifEtoc@ii \global\let\Etoc@localtop\Etoc@@three@@ \fi
526       \ifEtoc@iii \global\let\Etoc@localtop\Etoc@@four@@ \fi
527       \ifEtoc@iv \global\let\Etoc@localtop\Etoc@@five@@ \fi
528       \ifEtoc@v  \global\let\Etoc@localtop\Etoc@@six@@ \fi
529       \global\Etoc@notactivefalse
530       \global\Etoc@vfalse
531       \global\Etoc@ivfalse
532       \global\Etoc@iiifalse
533       \global\Etoc@iifalse
534       \global\Etoc@ifalse
535       \global\Etoc@false
536       \global\Etoc@jfalse
537       \global\Etoc@jjfalse
538     \fi
539   \fi
540 }
541 \let\etoc@startlocaltoc@gobble

```

2015/03/16: 1.08e tests if a local table of contents turns out empty. The `\Etoc@localtop` here does not have the same meaning as when set-up by the regular `\etoc@@startlocaltoc`. It is off by 1 (in general).

After some hesitation I opted for making and obeying the test always (\*) for local tables of contents, and do it also for global tables of contents but for the latter the user must issue `\etocno-tocifnotoc` for the emptiness test to be obeyed. The emptiness tests are to be trusted only when

the compilations have stabilized.

(\*) finally the user must issue `\etoccheckemptiness` once.

`\Etoc@tocid` is the number of the toc (possibly gotten via a `\ref` following a `\tableofcontents`), or it is `\z@` if the emptiness test is from a global toc. Until the compilations stabilize, some local TOCs can get printed at wrong locations naturally and emptiness tests can not be trusted either. Releases earlier than 1.08e authorized `\tableofcontents\ref{foo}` with `foo` the label of a *global* toc. But as it was impossible (without much complications) to discriminate, until auxiliary files have stabilized, between a `\ref` to a local toc whose id is not yet in the `.toc` file and a `\ref` to a global toc whose id will in the end definitely not be in the `.toc` file, I decided that this silly possibility should not be supported anymore. This allowed to remove a few tokens from `\Etoc@tableofcontents` as well. And it also allowed improvements to `\Etoc@localtableofcontents`.

Note: (1.08i 2016/09/29) the code has to handle both local and total toc. Hence the flag `\ifEtoc@notactive` has to be set prior to it. For a global toc, the `\Etoc@tocid` was set to `\z@`, and the `\ifnum` in `\etoc@startlocaltoc` did always fail, but I now prefer to simply nullify the `\etoc@startlocaltoc`. As its default fallback is `\@gobble` I simply test here for the `localtoc` flag. The `\Etoc@tocid` will be undefined for a global toc but it is not tested anymore.

Note: the mechanism for recognizing the level when a local toc is encountered is different from the one in `\etoc@@startlocaltoc` which uses the level flags. Here we just record in `\Etoc@level` where we last stood. The initialization `\global\let\Etoc@level\Etoc@minf` is thus needed in case the `.toc` file contains an `\etoc@startlocaltoc` before any `\contentsline`.

1.08i has two `testingcontentsline` macros, one for local, one for global.

```

542 \def\Etoc@setemptytocbool {%
543   \global\Etoc@emptytoctrue
544   \global\Etoc@stoptocfalse
545   \global\let\Etoc@level\Etoc@minf
546   \begingroup
547   \ifEtoc@localtoc
548     \def\etoc@startlocaltoc##1{%
549       \ifnum##1=\Etoc@tocid\relax
550         \global\let\Etoc@localtop\Etoc@level
551         \global\Etoc@notactivefalse
552       \fi }%
553     \let\contentsline\Etoc@testingcontentslinelocal
554   \else
555     \let\contentsline\Etoc@testingcontentsline
556   \fi
557   \Etoc@storetocdepth
558   \the\Etoc@toctoks
559   \Etoc@resettocdepth
560   \endgroup
561 }
562 \DeclareRobustCommand*\etocifwasempty
563   {\ifEtoc@emptytoc\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi }
564 \expandafter
565 \let\expandafter\etocxifwasempty\csname etocifwasempty \endcsname
566 \def\Etoc@testingcontentslinelocal #1{%
567   \ifEtoc@stoptoc
568   \else
569     \ifnum\csname Etoc@#1@\endcsname=\Etoc@@six@@
570     \else
571       \global\expandafter\let\expandafter\Etoc@level\csname Etoc@#1@\endcsname

```

`\ifEtoc@notactive` set to `False` by `\etoc@startlocaltoc` with right id. If that happened, check level of hit against unit. Will decide if empty or not. Set `stoptoc` flag afterwards in all cases.

`\Etoc@localtop` will have been initialized by `\etoc@@startlocaltoc` as `\ifEtoc@notactive` flag can be false only then.

```

572   \ifEtoc@notactive
573   \else
574   \ifnum\Etoc@level>\Etoc@localtop
575   \ifnum\Etoc@level>\c@tocdepth
576   \else
577   \global\Etoc@emptytocfalse
578   \global\Etoc@stoptoctrue
579   \fi
580   \else
581   \global\Etoc@stoptoctrue
582   \fi
583   \fi
584   \fi
585   \fi
586   \Etoc@gobblefourorthree\relax
587 }
588 \def\Etoc@testingcontentsline #1{%
589   \ifEtoc@stoptoc
590   \else
591   \ifnum\csname Etoc@#1@\endcsname=\Etoc@@six@@
592   \else
593   \ifnum\csname Etoc@#1@\endcsname>\c@tocdepth
594   \else
595   \global\Etoc@emptytocfalse
596   \global\Etoc@stoptoctrue
597   \fi
598   \fi
599   \fi
600   \Etoc@gobblefourorthree\relax
601 }

```

1.07g suppresses the printing of `--unknown etoc ref: run latex again--` as sometimes one uses `\localtableofcontents` to prepare something else, and one does not want any text output even in intermediate runs.

Also 1.07g adds `\etocaftertochook` to help with `\invisiblelocaltableofcontents` (and then I did `\etocsetnexttocdepth`). 1.07h makes the hook used by `\etocsetnexttocdepth` private.

2015/03/14: 1.08b suppresses a space token from an unprotected end of line at the end after the replacement text of `\etoc@startlocaltoc` (bug dating back to 1.03 2012/11/23). I didn't see it due to the originally badly formatted source of `\Etoc@next`.

2015/03/16: 1.08e lets `\localtableofcontents` do a first scan of the `.toc` file (as stored in `\Etoc@toctoks`) to determine if the table of contents will in fact end up empty. In that case, nothing is typeset. The command `\etocaftertochook` is still executed though. Other ways were envisioned (like delimited macros) to determine this potential emptiness, but in the end I opted for execution of the `.toc` file with suitable definitions for `\contentsline` and `\etoc@startlocaltoc`. Notice though that if emptiness would result from empty line styles, this can not be detected. Emptiness means “no executed `\contentsline`”.

For this detection of emptiness, assignments (here and in `\Etoc@testingcontentsline`) are made globally, I think this is the best (just in case some portions of the `.toc` file turn out to be inside some groups — perhaps for some silly color assignments, etc... — whose boundaries do not necessarily respect unit levels).

The flag `\Etoc@tocwithid` discriminates between a `\localtableofcontents` and a `\tableofcontents \ref{foo}`; the latter could so far possibly refer to a local or also to a global table of contents but release 1.08e has deprecated the latter use as it complicated the code, for

something truly silly. Thus `\ref{foo}` must now be with `foo` a label of a local TOC. As a result `\Etoc@tocwithid` is less used now.

In the case of a `\refed-to toc` whose label was just added hence is not yet in the `.aux` file, `\Etoc@tocid` is 0. `etoc` used to issue a warning to run latex again and did no printing at all. Release 1.08e in such cases prints the heading (this may gain one compilation step). Emptiness test is not executed as it would necessarily turn out positive and can not be trusted anyhow. The TOC is declared non empty, which it probably is...

Emptiness detection for local tables of contents (either from a `\localtableofcontents` or from a `\tableofcontents\ref{localtoc}`) can be trusted only when the `.toc` file has stabilized. The emptiness status of a local TOC whose Id is not yet in the `.toc` is by necessity undecided yet (and not to be trusted really as the numbering may have changed; only when compilation runs settle is the emptiness status to be trusted). The code declares the TOC non empty, as it will be in 95% of use cases.

Dropping support for `\tableofcontents \ref{globaltoc}` means here that when a TOC id is not found in the `.toc` file we can assume it definitely has to be a local TOC needing more compilations. The emptiness status is undecided, the code declares the TOC non empty.

1.08i-2016/09/29 now does `\Etoc@localtoctrue` right at the start (the earlier code could have to handle table of contents which were actually global, via the `\label/\ref` mechanism.) It does not rely on the `\ifnum` automatically false in `\etoc@@startlocaltoc` due to the special values 0 or `\z@` for `\Etoc@tocid`, but simply leaves `\etoc@@startlocaltoc` to its default `\@gobble`. The `\Etoc@emptytocfalse` flag is upfront in case some code using `\etocifwasempty` is in user hooks. The default is to assume the TOC non-empty as its contents are actually still unknown. Under the `stoptoc` flag, the `\Etoc@etoccontentsline` is more efficient now.

The `\Etoc@notactive` flag needs to be set before calling `\Etoc@setemptytocbool`.

I hesitated with 1.08i to write something to aux file in order to let  $\LaTeX$  prompt the user for extra pass, after insertion of some new `\localtableofcontents`, but finally I prefer to only trick  $\LaTeX$  into telling about undefined references.

The `\PackageWarning` approach has the advantage that at least in Emacs/AUCTeX the C-cc-c will propose LaTeX, not View. But perhaps some automated scripts checking aux file will not like the extra line which is then removed in next pass, and could possibly do one extra unneeded compilation to check aux file remains identical. Hence the second approach. (edit 2017/10/23: good thing I documented that! I had completely forgotten that rationale, but I wonder if it is correct.)

Also the `\PackageWarning` does not trigger a visible message near the end of the log file or console output, contrarily to a

`LaTeX Warning: There were undefined references.`

followed by a

`LaTeX Warning: Label(s) may have changed. Rerun to get cross-references right.`

Method used here seems to work fine also with `latexmk`: it does not seem to induce it into making too many runs.

```
602 \def\Etoc@localtableofcontents#1{%
603   \Etoc@localtoctrue
604   \global\Etoc@emptytocfalse
605   \edef\Etoc@tocid{#1}%
606   \ifnum\Etoc@tocid<\@ne
607     %\PackageWarning{etoc}{Unknown toc reference \@secondoftwo#1. Run LaTeX again}%
608     \setbox0\hbox{\ref{Unknown toc ref \@secondoftwo#1. Rerun LaTeX}}%
609     %\if@files\immediate\write\@mainaux{\noexpand\@tempwatrue}\fi
610     \global\Etoc@stoptoctrue % do only heading
611     \Etoc@tableofcontents
612     \expandafter\Etoc@gobbletoetoc@ % skip all the rest
613   \fi
614   \global\Etoc@notactivetrue
```

1.08i's `\Etoc@setemptytocbool` now checks `localtoc` flag.



```

615 \ifEtoc@checkemptiness \Etoc@setemptytocbool \fi
616 \ifEtoc@emptytoc
617 \ifEtoc@notactive
618 %\PackageWarning{etoc}{Unknown toc ID \number\Etoc@tocid. Run LaTeX again}%
619 \setbox0\hbox{\ref{Unknown toc ID \number\Etoc@tocid. Rerun LaTeX}}%
620 %\if@files\immediate\write\@mainaux{\noexpand\@tempwstrtrue}\fi
621 \global\Etoc@emptytocfalse % assume real one will be non-empty
622 \global\Etoc@stoptoctrue % and print only heading for this pass
623 \Etoc@tableofcontents
624 \expandafter\expandafter\expandafter\Etoc@gobbletoetoc@
625 \else
626 %% \PackageWarning{etoc}{This toc is apparently empty}%
627 \fi
628 \else
629 \global\Etoc@stoptocfalse
630 \global\Etoc@notactivetrue
631 \global\let\Etoc@localtop\Etoc@minustwo@@
632 \edef\etoc@startlocaltoc##1{\noexpand\etoc@@startlocaltoc{##1}{\Etoc@tocid}}%
633 \Etoc@tableofcontents
634 \fi
635 \@gobble\etoc@
636 \endgroup\ifEtoc@mustclosegroup\endgroup\fi
637 \Etoc@tocdepthreset % 1.07h, renamed 1.08h
638 \etocaftertohook % 1.07g
639 }

```

2013/03/07: I discover a `\@namedef` trick to construct the `\Etoc@again` space delimited macro:

```
\@namedef {Etoc@again} {...stuff...}
```

Original version was (copied from analogous stuff in source2e):

```
{\def\1{\Etoc@again}\expandafter\gdef\1 {...stuff...}}
```

and in the end (now that I think about it) I simply use `\@firstofone`.

```

640 \def\Etoc@getrefno #1#2\etoc@ {#1}
641 \def\Etoc@getref #1{\ifundefined{r@#1}{0}{\expandafter\expandafter\expandafter
642 \Etoc@getrefno\csname r@#1\endcsname\relax\etoc@}}
643 \def\Etoc@ref#1{\Etoc@localtableofcontents{\Etoc@getref{#1}}}
644 \def\Etoc@label#1{\label{#1}\futurelet\Etoc@nexttoken\Etoc@t@bleofcontents}
645 \@firstofone{\def\Etoc@again} {\futurelet\Etoc@nexttoken\Etoc@t@bleofcontents}

```

1.08e adds test for emptiness. And uses another coding style for conditional branching. Easier to read. Not tested for efficiency.

Also, `\ref{foo}` expects `foo` to be a label to a *local* TOC. Earlier versions accepted a reference to a global TOC, I have removed the support from `\Etoc@tableofcontents`.

The syntax `\localtableofcontents\ref{foo}` is now accepted.

```

646 \def\Etoc@dothis #1#2\etoc@ {\fi #1}
647 \def\Etoc@t@bleofcontents{%
648 \ifx\Etoc@nexttoken\label\Etoc@dothis{\expandafter\Etoc@label\@gobble}\fi
649 \ifx\Etoc@nexttoken\sptoken\Etoc@dothis{\Etoc@again}\fi

```

`\Etoc@ref` will hand over directly to `\Etoc@localtableofcontents`. Argument will be (or rather expand to) zero if the reference is non-existent yet. There was an `\Etoc@tocwithidfalse` flag which 1.08i suppresses as it is not used afterwards (since 1.08e I think.)

```
650 \ifx\Etoc@nexttoken\ref\Etoc@dothis{\expandafter\Etoc@ref\@gobble}\fi
```

Flag to check if we were called from a `\localtableofcontents`.

```
651 \ifEtoc@tocwithid\Etoc@dothis{\Etoc@localtableofcontents{\c@etoc@tocid}}\fi
```

From now on we are handling a global TOC. Earlier, I used the trick of setting `\Etoc@tocid` to `\z@` for compatibility with expansion of `\etoc@startlocaltoc`. But since 1.08i `\etoc@startlocaltoc`

## 59. Implementation

is left to be `\@gobble`, and `\Etoc@tocid` is never tested. We don't need to set the `notactive` flag as now `\Etoc@testingcontentsline` tests first the `localtoc` flag (was already the case of `\Etoc@etoccontentsline`). I change a bit the style of conditionals here for clarity of code.

```
652 \global\Etoc@emptytocfalse\ifEtoc@checksempiness\Etoc@setemptytocbool\fi
653 \ifEtoc@emptytoc
654 \ifEtoc@notocifnotoc
655 \expandafter\expandafter\expandafter\@gobble
656 \fi
657 \fi
658 \Etoc@tableofcontents
659 \endgroup\ifEtoc@mustclosegroup\endgroup\fi
660 \Etoc@tocdepthreset % 1.07h, renamed 1.08h
661 \etocaftertochook % 1.07g
662 \@gobble\etoc@
663 }
```

1.07g for consistency `Etoc@` prefix added.

1.08c does not use `\arabic` in the `\addtocontents` since I have seen that in some circumstances (for some right to left languages with `polyglossia` or `babel`), one can not rely on `\arabic` having its default definition. As the number written here will be used later in an `\ifnum`, I should not have used it in the first place (done 2015/03/30).

1.08i moves elsewhere definition of `\etoc@startlocaltoc`. In `\Etoc@table@fcontents` case it remains to be seen if a `\ref` follows which could turn the TOC into a local one. Handled by `\Etoc@t@bleofcontents`.

```
664 \def\Etoc@table@fcontents{%
665 \refstepcounter{etoc@tocid}%
666 \Etoc@tocwithidfalse
667 \futurelet\Etoc@nexttoken\Etoc@t@bleofcontents}
668 \def\Etoc@localtable@fcontents{%
669 \refstepcounter{etoc@tocid}%
670 \addtocontents{toc}{\string\etoc@startlocaltoc{\the\c@etoc@tocid}}%
671 \Etoc@tocwithidtrue
672 \futurelet\Etoc@nexttoken\Etoc@t@bleofcontents}
```

1.07g defines `\etoc@tableofcontents` to be able to undo the evil and brutal doings of some packages with `\tableofcontents`.

1.08 patches `tableof`'s influence. Version 1.4a or later of `tableof` is needed.

```
673 \let\etocaftertitlehook \@empty
674 \let\etocaftercontentshook \@empty
```

Attention that there could be a `\ref` following, thus we don't yet know whether this is a local or global table of contents.

```
675 \def\etoc@tableofcontents{%
676 \Etoc@openouttoc
677 \Etoc@tocdepthset % new with 1.08h
678 \begingroup % closed in \Etoc@t@bleofcontents or \Etoc@localtableofcontents
```

No need for the `tableof` extra group level, even it is better to get rid of it. I don't globally cancel `\tof@begingroup` and `\tof@endgroup` to leave open for the user the (dubious) possibility of using directly `\tableof`, `\tablenotof` (rather than only the reasonable thing which is `\nexttocwithtags{}}{}`).

```
679 \let\tof@begingroup \@empty
680 \let\tof@endgroup \@empty
```

1.08i removes definition of `\etoc@startlocaltoc` from here. It is done in `\Etoc@localtableofcontents` or uses its default of `\@gobble`.

```
681 \@ifstar
682 {\def\Etoc@aftertitlehook{}}\Etoc@table@fcontents}
```

```

683      {\let\Etoc@aftertitlehook\etocaftertitlehook\Etoc@table@fcontents}}
684 \let\tableofcontents\etoc\tableofcontents
685 \newcommand*\localtableofcontents{%
686   \Etoc@openouttoc
687   \Etoc@tocdepthset % new with 1.08h
688   \begingroup % closed in \Etoc@tableofcontents or \Etoc@localtableofcontents
689     \let\tof@begingroup\@empty
690     \let\tof@endgroup \@empty
691     \@ifstar
692     {\def\Etoc@aftertitlehook{}\Etoc@localtable@fcontents}
693     {\let\Etoc@aftertitlehook\etocaftertitlehook\Etoc@localtable@fcontents}}

```

1.08 moves earlier the definitions of \Etoc@savedcontentsline as well as \Etoc@levellist. It will thus not be operative to do \etocsetlevel from inside the title now. But perhaps it will be easier to deal with tables.

Also I remove a big \@firstofone, and use rather \Etoc@gobbletoetoc@ for the gobbling case.

Also, 1.08 modifies the code to do only expandable things after \Etoc@toctoc, in order to allow opening of an alignment in the first argument and closing in the second. Earlier versions had an \Etoc@@next initially set to relax after the first \Etoc@toctoc. I was in 2012/2013 quite uneasy with the TeX conditionals hence wanted for safety to move the things for the second invocation of \Etoc@toctoc after the \fi\fi\fi. I could define an \afterfififi macro, but it should be ok as it is done here.

1.08e moved the \Etoc@par to \Etoc@tableofcontents. This was needed for the implementation of the emptiness test. It might have some impact in fringe cases as the \refstepcounter is now done before the \par is issued.

1.08e suppresses the possibility to do a \tableofcontents\ref{foo} with foo the label of a global TOC: issuing \tableofcontents alone suffices. Allowed to simplify some code.

1.08g moves \Etoc@par to before \etocbeforetitlehook, not after.

```

694 \long\def\Etoc@gobbletoetoc@ #1\etoc@ {}
695 \newcommand\etocsettocstyle[2]{%
696 \def\Etoc@tableofcontents
697 {%
698   \ifnum\c@tocdepth>\Etoc@minf
699   \else
700     \expandafter\Etoc@gobbletoetoc@
701     \fi
702     \Etoc@par
703     \etocbeforetitlehook % 1.07m
704     \Etoc@storetocdepth % 1.07g
705     \let\Etoc@savedcontentsline\contentsline % moved up here 1.08
706     \let\contentsline\Etoc@etoccontentsline
707     \ifEtoc@standard
708     \else % moved here from \Etoc@toctoc (1.08)

```

Prior to 1.08, this was done in \Etoc@toctoc

1.08k defines here \etocsavedchaptertocline, \etocsavedsectiontocline, and so on for all declared level names, again at each TOC and for the duration of it.

```

709     \def\Etoc@levellist@elt####1{%
710       \expandafter\let\csname etocsaved####1tocline\expandafter\endcsname
711       \csname l@####1\endcsname
712       \expandafter\let\csname l@####1\endcsname\Etoc@lxyz}%
713     \Etoc@levellist

```

1.08k removes \let\chapternumberline\numberline etc... done formerly here

```

714   \fi
715   #1%
716   \ifEtoc@parskip\parskip\z@skip\fi %1.07d

```

## 59. Implementation

```
717 \Etoc@aftertitlehook
718 \Etoc@toctoc
719 \etocaftercontentshook
720 #2\@nobreakfalse % 1.07d: \@nobreakfalse moved here
721 \Etoc@resettocdepth % 1.07g. moved here from before #2 by 1.08

1.08: necessary if tableof 1.4a has been used in global mode and thus was forced to modify globally \contentsline. The collateral effect is that etoc is now forced to reset globally \contentsline.

722 \ifx\Etoc@global\global
723 \ifx\tof@finish\@empty
724 \else
725 \global\let\contentsline\Etoc@savedcontentsline
726 \fi
727 \fi
728 \@gobble\etoc@
729}% end of \Etoc@tableofcontents definition by \etocsettocstyle
730 }

1.08 hacks the begin parts to update therein the \ifEtoc@isfirst flag (2015/03/09).
1.08c removes unneeded \long from the \def's inside \etoc@setstyle (2015/03/29).

731 \def\etocsetstyle{\Etoc@standardfalse\etoc@setstyle}
732 \long\def\etoc@setstyle#1#2#3#4#5{%
733 \expandafter\def
734 \csname Etoc@begin@\csname Etoc@#1@\endcsname\endcsname
735 \{#2\Etoc@global\Etoc@isfirsttrue\}%
736 \expandafter\def
737 \csname Etoc@prefix@\csname Etoc@#1@\endcsname\endcsname \{#3\}%
738 \expandafter\def
739 \csname Etoc@contents@\csname Etoc@#1@\endcsname\endcsname \{#4\}%
740 \expandafter\def
741 \csname Etoc@end@\csname Etoc@#1@\endcsname\endcsname \{#5\}%
742 }

placeholder for comments

743 \def\etocfontminustwo {\normalfont \LARGE \bfseries}
744 \def\etocfontminusone {\normalfont \large \bfseries}
745 \def\etocfontzero {\normalfont \large \bfseries}
746 \def\etocfontone {\normalfont \normalsize \bfseries}
747 \def\etocfonttwo {\normalfont \normalsize}
748 \def\etocfontthree {\normalfont \footnotesize}

placeholder for comments

749 \def\etocsepminustwo {4ex \@plus .5ex \@minus .5ex}
750 \def\etocsepminusone {4ex \@plus .5ex \@minus .5ex}
751 \def\etocsepzero {2.5ex \@plus .4ex \@minus .4ex}
752 \def\etocsepone {1.5ex \@plus .3ex \@minus .3ex}
753 \def\etocseptwo {.5ex \@plus .1ex \@minus .1ex}
754 \def\etocsepthree {.25ex \@plus .05ex \@minus .05ex}

placeholder for comments

755 \def\etocbaselinespreadminustwo {1}
756 \def\etocbaselinespreadminusone {1}
757 \def\etocbaselinespreadzero {1}
758 \def\etocbaselinespreadone {1}
759 \def\etocbaselinespreadtwo {1}
760 \def\etocbaselinespreadthree {.9}

placeholder for comments
```

```

761 \def\etocminustwoleftmargin {1.5em plus 0.5fil}
762 \def\etocminustworightmargin {1.5em plus -0.5fil}
763 \def\etocminusoneleftmargin {1em}
764 \def\etocminusonerightmargin {1em}
765 \def\etoclineleaders
766     {\hbox{\normalfont\normalsize\hb@xt@2ex {\hss.\hss}}}
767 \def\etocabbrevpagename {p.~}

```

Versions earlier than 1.08b (and since v1.05 2012/12/01) defined \etocpartname (for use by **etoc**'s own line styles) to expand to \partname. But this didn't make sense in the context for example of babel and frenchb, because \frenchpartname does things depending on the current value of the counter part. The code in recent frenchb (but not yet v2.5a when \etocpartname was introduced) constructs control sequences \ordinali, etc... If the part counter is zero, this gives \ordinal. Usually this is not defined, hence no error happens (as it is constructed via \csname), but under class memoir the bug showed up. All this to explain that I found out about this long lasting problem only on 2015/03/14. Probably a sign that **etoc**'s own line styles are rarely used...

```

768 \def\etocpartname      {Part}% modified 1.08b
769 \def\etocbookname      {Book}

```

placeholder for comments The macro \etocdefaultlines was initially called \etococlines. Now \etococlines just does \Etoc@standardfalse. Version 1.07e has rewritten entirely the stuff related to penalties and \addvspace, as this was not satisfactory in the earlier versions, which were written at a early stage in the development of the package. Actually I am not fully satisfied with these line styles.

```

770 \def\etocdefaultlines{\Etoc@standardfalse %
771 %% 'book':
772 \etoc@setstyle{@minustwo}
773   {\addpenalty\@M\etocskipfirstprefix}
774   {\addpenalty\@secpenalty}
775   {\begingroup
776     \etocfontminustwo
777     \addvspace{\etocsepminustwo}%
778     \parindent \z@
779     \leftskip \etocminustwoleftmargin
780     \rightskip \etocminustworightmargin
781     \parfillskip \@flushglue
782     \vbox{\etocifnumbered{\etocbookname\enspace\etocnumber:\quad}}{\etocname
783       \baselineskip\etocbaselinespreadminustwo\baselineskip
784       \par}%
785     \addpenalty\@M\addvspace{\etocsepminusone}%
786     \endgroup}
787   {}%
788 %% 'part':
789 \etoc@setstyle{@minusone}
790   {\addpenalty\@M\etocskipfirstprefix}
791   {\addpenalty\@secpenalty}
792   {\begingroup
793     \etocfontminusone
794     \addvspace{\etocsepminusone}%
795     \parindent \z@
796     \leftskip \etocminusoneleftmargin
797     \rightskip \etocminusonerightmargin
798     \parfillskip \@flushglue
799     \vbox{\etocifnumbered{\etocpartname\enspace\etocnumber.\quad}}{\etocname
800       \baselineskip\etocbaselinespreadminusone\baselineskip
801       \par}%
802     \addpenalty\@M\addvspace{\etocsepzero}%

```

## 59. Implementation

```

803 \endgroup}
804 {}%
805 %% 'chapter':
806 \etoc@setstyle{@zero}
807 {\addpenalty\@M\etocskipfirstprefix}
808 {\addpenalty\@itempenalty}
809 {\begingroup
810 \etocfontzero
811 \addvspace{\etocsepzero}%
812 \parindent \z@ \parfillskip \@flushglue
813 \vbox{\etocifnumbered{\etocnumber.\enspace{}}\etocname
814 \baselineskip\etocbaselinespreadzero\baselineskip
815 \par}%
816 \endgroup}
817 {\addpenalty{-\@highpenalty}\addvspace{\etocsepmminusone}}%
818 %% 'section':
819 \etoc@setstyle{@one}
820 {\addpenalty\@M\etocskipfirstprefix}
821 {\addpenalty\@itempenalty}
822 {\begingroup
823 \etocfontone
824 \addvspace{\etocsepone}%
825 \parindent \z@ \parfillskip \z@
826 \setbox\z@\vbox{\parfillskip\@flushglue
827 \etocname\par
828 \setbox\tw@lastbox
829 \global\setbox\@ne\hbox{\unhbox\tw@ } }%
830 \dimen\z@=\wd\@ne
831 \setbox\z@=\etoclineleaders
832 \advance\dimen\z@\wd\z@
833 \etocifnumbered
834 {\setbox\tw@\hbox{\etocnumber, \etocabbrevpagename\etocpage}}
835 {\setbox\tw@\hbox{\etocabbrevpagename\etocpage}}%
836 \advance\dimen\z@\wd\tw@
837 \ifdim\dimen\z@ < \linewidth
838 \vbox{\etocname~%
839 \leaders\box\z@\hfil\box\tw@
840 \baselineskip\etocbaselinespreadone\baselineskip
841 \par}% 1.08b adds the % here
842 \else
843 \vbox{\etocname~%
844 \leaders\copy\z@\hfil\break
845 \hbox{ }\leaders\box\z@\hfil\box\tw@
846 \baselineskip\etocbaselinespreadone\baselineskip
847 \par}%
848 \fi
849 \endgroup}
850 {\addpenalty\@secpenalty\addvspace{\etocsepzero}}%
851 %% 'subsection':
852 \etoc@setstyle{@two}
853 {\addpenalty\@medpenalty\etocskipfirstprefix}
854 {\addpenalty\@itempenalty}
855 {\begingroup
856 \etocfonttwo
857 \addvspace{\etocseptwo}%
858 \parindent \z@ \parfillskip \z@
859 \setbox\z@\vbox{\parfillskip\@flushglue

```



## 59. Implementation

1.08b revisits this old code written perhaps at a time I didn't know about `\expandafter` !

```
909 \newcommand\etocmulticolstyle[2][\etocdefaultnbcol]{%
910 \etocsettocstyle
911   {\let\etocoldpar\par
912    \addvspace{\etocabovetocskip}%
913    \ifnum #1>\@ne
914      \expandafter\@firstoftwo
915    \else \expandafter\@secondoftwo
916    \fi
917    {\multicolpretolerance\etocmulticolpretolerance
918     \multicoltolerance\etocmulticoltolerance
919     \setlength{\columnsep}{\etoccolumnsep}%
920     \setlength{\multicolsep}{\etocmulticolsep}%
921     \begin{multicols}{#1}[#2\etocoldpar\addvspace{\etocinnertopsep}]}
922    {#2\ifvmode\else\begin{group}\interlinepenalty\@M\parskip\z@skip
923     \@@par\end{group}
924     \fi
925     \nobreak\addvspace{\etocinnertopsep}%
926     \pretolerance\etocmulticolpretolerance
927     \tolerance\etocmulticoltolerance}%
928   }%
929   {\ifnum #1>\@ne
930     \expandafter\@firstofone
931   \else \expandafter\@gobble
932   \fi
933   {\end{multicols}}}%
934   \addvspace{\etocbelowtocskip}}%
935 }
```

placeholder for comments

```
936 \def\etocinnerbottomsep{3.5ex}
937 \def\etocinnerleftsep{2em}
938 \def\etocinnerrightsep{2em}
939 \def\etoctoprule{\hrule}
940 \def\etocleftrule{\vrule}
941 \def\etocrightrule{\vrule}
942 \def\etocbottomrule{\hrule}
943 \def\etoctoprulecolorcmd{\relax}
944 \def\etocbottomrulecolorcmd{\relax}
945 \def\etocleftrulecolorcmd{\relax}
946 \def\etocrightrulecolorcmd{\relax}
```

1.07i moves the `\nobreak` to before the `\vskip\etocinnertopsep` (especially important for the single column case).

```
947 \def\etoc@ruledheading #1{%
948   \hb@xt@\linewidth{\color@begin{group}
949     \hss #1\hss\hskip-\linewidth
950     \etoctoprulecolorcmd\leaders\etoctoprule\hss
951     \phantom{#1}%
952     \leaders\etoctoprule\hss\color@end{group}}%
953     \nointerlineskip\nobreak\vskip\etocinnertopsep}
954 \newcommand*\etocruledstyle[2][\etocdefaultnbcol]{%
955 \etocsettocstyle
956   {\addvspace{\etocabovetocskip}%
957    \ifnum #1>\@ne
958      \expandafter\@firstoftwo
959    \else \expandafter\@secondoftwo
960    \fi
```



```

961      {\multicolpretolerance\etocmulticolpretolerance
962       \multicoltolerance\etocmulticoltolerance
963       \setlength{\columnsep}{\etoccolumnsep}%
964       \setlength{\multicolsep}{\etocmulticolsep}%
965       \begin{multicols}{#1}[\etoc@ruledheading{#2}]}
966       {\etoc@ruledheading{#2}%
967        \pretolerance\etocmulticolpretolerance
968        \tolerance\etocmulticoltolerance}}
969 {\ifnum #1>\@ne\expandafter\@firstofone
970   \else \expandafter\@gobble
971   \fi
972   {\end{multicols}}}%
973   \addvspace{\etocbelowtocskip}}}}

```

1.07k defines `\Etoc@relax` and `\etocbkgcolorcmd` as `\long`: the user manual says to use `\renewcommand\etocbkgcolorcmd`, and an `\ifx` test is used in the framed style. It was thus a bug to have non-long definitions before. Or I could have said in the user manual to use `\renewcommand*`, or in `\etocframedstyle` I should test for the two, or I should pause to try to remember about this code and think about it.

```

974 \def\etocframedmphook{\relax}
975 \long\def\etocbkgcolorcmd{\relax}
976 \long\def\Etoc@relax{\relax}
placeholder for comments
977 \newbox\etoc@framed@titlebox
978 \newbox\etoc@framed@contentsbox
979 \newcommand*\etocframedstyle[2][\etocdefaultnbcol]{%
980 \etocsettocstyle{%
981   \addvspace{\etocabovetocskip}%
982   \sbox\z@{#2}%
983   \dimen\z@=\dp\z@
984   \ifdim\wd\z@<\linewidth \dp\z@=\z@ \else \dimen\z@=\z@ \fi
985   \setbox\etoc@framed@titlebox=\hb@xt@\linewidth{\color@begingroup
986     \hss
987     \ifx\etocbkgcolorcmd\Etoc@relax
988     \else
989       \sbox\tw@{\color{white}%
990        \vrule\@width\wd\z@\@height\ht\z@\@depth\dimen\z@}%
991       \ifdim\wd\z@<\linewidth \dp\tw@=\z@ \fi
992       \box\tw@
993       \hskip-\wd\z@
994     \fi
995     \copy\z@
996     \hss
997     \hskip-\linewidth
998     \etoctoprulecolorcmd\leaders\etocoprul\hss
999     \hskip\wd\z@
1000    \etocoprulcolorcmd\leaders\etocoprul\hss\color@endgroup}%
1001   \setbox\z@\hbox{\etocleftrule\etocrightrule}%
1002   \dimen\tw@\linewidth\advance\dimen\tw@-\wd\z@
1003   \advance\dimen\tw@-\etocinnerleftsep
1004   \advance\dimen\tw@-\etocinnerrightsep
1005   \setbox\etoc@framed@contentsbox=\vbox\bgroup
1006     \hsize\dimen\tw@
1007     \kern\dimen\z@
1008     \vskip\etocinnertopsep
1009     \hbox\bgroup
1010     \begin{minipage}{\hsize}%

```

## 59. Implementation

```

1011      \etocframedmphook
1012      \ifnum #1>\@ne
1013          \expandafter\@firstoftwo
1014      \else \expandafter\@secondoftwo
1015      \fi
1016      {\multicolpretolerance\etocmulticolpretolerance
1017      \multicoltolerance\etocmulticoltolerance
1018      \setlength{\columnsep}{\etoccolumnsep}%
1019      \setlength{\multicolsep}{\etocmulticolsep}%
1020      \begin{multicols}{#1}}
1021      {\pretolerance\etocmulticolpretolerance
1022      \tolerance\etocmulticoltolerance}}
1023  {\ifnum #1>\@ne\expandafter\@firstofone
1024   \else \expandafter\@gobble
1025   \fi
1026   {\end{multicols}\unskip }%
1027  \end{minipage}%
1028  \egroup
1029  \vskip\etocinnerbottomsep
1030  \egroup
1031  \vbox{\hsize\linewidth
1032        \ifx\etocbkgcolorcmd\Etoc@relax
1033        \else
1034            \kern\ht\etoc@framed@titlebox
1035            \kern\dp\etoc@framed@titlebox
1036            \hb@xt@\linewidth{\color@begingroup
1037            \etocleftrulecolorcmd\etocleftrule
1038            \etocbkgcolorcmd
1039            \leaders\vrule
1040                \@height\ht\etoc@framed@contentsbox
1041                \@depth\dp\etoc@framed@contentsbox
1042            \hss
1043            \etocrightrulecolorcmd\etocrightrule
1044            \color@endgroup}\nointerlineskip
1045            \vskip-\dp\etoc@framed@contentsbox
1046            \vskip-\ht\etoc@framed@contentsbox
1047            \vskip-\dp\etoc@framed@titlebox
1048            \vskip-\ht\etoc@framed@titlebox
1049        \fi
1050        \box\etoc@framed@titlebox\nointerlineskip
1051        \hb@xt@\linewidth{\color@begingroup
1052        {\etocleftrulecolorcmd\etocleftrule}%
1053        \hss\box\etoc@framed@contentsbox\hss
1054        \etocrightrulecolorcmd\etocrightrule\color@endgroup}
1055        \nointerlineskip
1056        \vskip\ht\etoc@framed@contentsbox
1057        \vskip\dp\etoc@framed@contentsbox
1058        \hb@xt@\linewidth{\color@begingroup\etocbottomrulecolorcmd
1059        \leaders\etocbottomrule\hss\color@endgroup}}
1060        \addvspace{\etocbelowtocskip}}}

```

placeholder for comments

```

1061 \newcommand\etoc@multicoltoc[2][\etocdefaultnbcol]{%
1062     \etocmulticolstyle[#1]{#2}%
1063     \tableofcontents}
1064 \newcommand\etoc@multicoltoctoc[2][\etocdefaultnbcol]{%
1065     \etocmulticolstyle[#1]{#2}%
1066     \tableofcontents*}

```

```

1067 \newcommand\etoc@local@multicoltoc[2][\etocdefaultnbcol]{%
1068     \etocmulticolstyle[#1]{#2}%
1069     \localtableofcontents}
1070 \newcommand\etoc@local@multicoltoi[2][\etocdefaultnbcol]{%
1071     \etocmulticolstyle[#1]{#2}%
1072     \localtableofcontents*}

placeholder for comments
1073 \newcommand*\etoc@ruledtoc[2][\etocdefaultnbcol]{%
1074     \etocruledstyle[#1]{#2}%
1075     \tableofcontents}
1076 \newcommand*\etoc@ruledtoi[2][\etocdefaultnbcol]{%
1077     \etocruledstyle[#1]{#2}%
1078     \tableofcontents*}
1079 \newcommand*\etoc@local@ruledtoc[2][\etocdefaultnbcol]{%
1080     \etocruledstyle[#1]{#2}%
1081     \localtableofcontents}
1082 \newcommand*\etoc@local@ruledtoi[2][\etocdefaultnbcol]{%
1083     \etocruledstyle[#1]{#2}%
1084     \localtableofcontents*}

placeholder for comments
1085 \newcommand*\etoc@framedtoc[2][\etocdefaultnbcol]{%
1086     \etocframedstyle[#1]{#2}%
1087     \tableofcontents}
1088 \newcommand*\etoc@framedtoi[2][\etocdefaultnbcol]{%
1089     \etocframedstyle[#1]{#2}%
1090     \tableofcontents*}
1091 \newcommand*\etoc@local@framedtoc[2][\etocdefaultnbcol]{%
1092     \etocframedstyle[#1]{#2}%
1093     \localtableofcontents}
1094 \newcommand*\etoc@local@framedtoi[2][\etocdefaultnbcol]{%
1095     \etocframedstyle[#1]{#2}%
1096     \localtableofcontents*}

placeholder for comments
1097 \def\etocmulticol{\begingroup
1098     \Etoc@mustclosegrouptrue
1099     \@ifstar
1100     {\etoc@multicoltoi}
1101     {\etoc@multicoltoc}}
1102 \def\etocruled{\begingroup
1103     \Etoc@mustclosegrouptrue
1104     \@ifstar
1105     {\etoc@ruledtoi}
1106     {\etoc@ruledtoc}}
1107 \def\etocframed{\begingroup
1108     \Etoc@mustclosegrouptrue
1109     \@ifstar
1110     {\etoc@framedtoi}
1111     {\etoc@framedtoc}}
1112 \def\etoclocalmulticol{\begingroup
1113     \Etoc@mustclosegrouptrue
1114     \@ifstar
1115     {\etoc@local@multicoltoi}
1116     {\etoc@local@multicoltoc}}
1117 \def\etoclocalruled{\begingroup
1118     \Etoc@mustclosegrouptrue
1119     \@ifstar

```

## 59. Implementation

```

1120 {\etoc@local@ruledtoci}
1121 {\etoc@local@ruledtoc}}
1122 \def\etoclocalframed{\begingroup
1123   \Etoc@mustclosegrouptrue
1124   \@ifstar
1125   {\etoc@local@framedtoci}
1126   {\etoc@local@framedtoc}}
  placeholder for comments
1127 \def\etocarticlestyle{%
1128   \etocsettocstyle
1129   {\section *{\contentsname
1130     \@mkboth {\MakeUppercase \contentsname}
1131              {\MakeUppercase \contentsname}}}
1132   {}}
1133 \def\etocarticlestylenomarks{%
1134   \etocsettocstyle
1135   {\section *{\contentsname}}
1136   {}}
  placeholder for comments
1137 \def\etocbookstyle{%
1138   \etocsettocstyle
1139   {\if@twocolumn \@restonecoltrue \onecolumn \else \@restonecolfalse \fi
1140     \chapter *{\contentsname
1141       \@mkboth {\MakeUppercase \contentsname}
1142                {\MakeUppercase \contentsname}}}
1143   {\if@restonecol \twocolumn \fi}}
1144 \def\etocbookstylenomarks{%
1145   \etocsettocstyle
1146   {\if@twocolumn \@restonecoltrue \onecolumn \else \@restonecolfalse \fi
1147     \chapter *{\contentsname}}
1148   {\if@restonecol \twocolumn \fi}}
1149 \let\etocreportstyle\etocbookstyle
1150 \let\etocreportstylenomarks\etocbookstylenomarks
1151 \def\etocmemoirtocformat #1#2{%
1152   \def\Etoc@addsuitablecontentsline{\addcontentsline {toc}{#1}{#2}}%
1153   \renewcommand*\etocaftertitlehook{%
1154     \ifmem@em@starred@listof
1155     \else
1156       \phantomsection
1157       \aftergroup\Etoc@addsuitablecontentsline
1158     \fi}%
1159 }
1160 \def\etocmemoirstyle{%
1161   \etocsettocstyle
1162   {\ensureonecol \par \begingroup \@nameuse {@tocmaketitle}
1163     \Etoc@aftertitlehook\let\Etoc@aftertitlehook\relax
1164     \parskip \cftparskip \@nameuse {cfttocbeforelisthook}}
1165   {\@nameuse {cfttocafterlisthook}\endgroup\restorefromonecol}}
  placeholder for comments
1166 \def\etocscartclstyle{%
1167   \etocsettocstyle
1168   {\let\if@dynlist\if@tocleft
1169     \def\@currentx{toc}% added 1.07m 2015/01/23
1170     \iftocfeature {toc}{onecolumn}
1171     {\iftocfeature {toc}{leveldown}
1172     {}

```

```

1173         {\if@twocolumn \aftergroup \twocolumn \onecolumn \fi }}
1174     {}%
1175     1.07k: next line to do as within current scrartcl 2013/12/19 v3.12 KOMA-Script article class.
1176     \iftocfeature {toc}{numberline}{\def \nonumberline {\numberline {}{}}}%
1177     \tocbasic@listhead {\listoftocname}%
1178     \begingroup \expandafter \expandafter \expandafter
1179     \endgroup \expandafter
1180     \ifx
1181     \csname microtypesetup\endcsname \relax
1182     \else
1183     \iftocfeature {toc}{noprotrusion}{}
1184     {\microtypesetup {protrusion=false}%
1185     \PackageInfo {tocbasic}%
1186     {character protrusion at toc deactivated}}%
1187     \fi
1188     \setlength {\parskip }{\z@ }%
1189     \setlength {\parindent }{\z@ }%
1190     \setlength {\parfillskip }{\z@ \@plus 1fil}%
1191     \csname tocbasic@@before@hook\endcsname
1192     \csname tb@toc@before@hook\endcsname
1193     {\csname tb@toc@after@hook\endcsname
1194     \csname tocbasic@@after@hook\endcsname}}
1195 \let\etocscrbookstyle\etocscrartclstyle
1196 \def\etocstandarddisplaystyle{\etocarticlestyle}
1197 \newcommand*{\etocmarkboth}[1]{%
1198     \mkboth{\MakeUppercase{#1}}{\MakeUppercase{#1}}}
1199 \newcommand*{\etocmarkbothnouc}[1]{\mkboth{#1}{#1}}
1200 \newcommand\etocstyle[3][section]{\etocmulticolstyle[#2]%
1201     {\csname #1\endcsname *{#3}}}
1202 \newcommand\etocstylewithmarks[4][section]{\etocmulticolstyle[#2]%
1203     {\csname #1\endcsname *{#3\etocmarkboth{#4}}}}
1204 \newcommand\etocstylewithmarksnouc[4][section]{\etocmulticolstyle[#2]%
1205     {\csname #1\endcsname *{#3\etocmarkbothnouc{#4}}}}
1206 \def\etoc@redefetocstyle#1{%
1207     \renewcommand\etocstylewithmarks[4][#1]
1208     {\etocmulticolstyle[##2]%
1209     {\csname ##1\endcsname *{##3\etocmarkboth{##4}}}}
1210     \renewcommand\etocstylewithmarksnouc[4][#1]
1211     {\etocmulticolstyle[##2]%
1212     {\csname ##1\endcsname *{##3\etocmarkbothnouc{##4}}}}
1213     \renewcommand\etocstyle[3][#1]{%
1214     \etocmulticolstyle[##2]{\csname ##1\endcsname *{##3}}}
1215 \ifclassloaded{scrartcl}
1216     {\renewcommand*{\etocstandarddisplaystyle{\etocscrartclstyle}}{}}
1217 \ifclassloaded{book}
1218     {\renewcommand*{\etocfontone{\normalfont\normalsize}
1219     \renewcommand*{\etocstandarddisplaystyle{\etocbookstyle}
1220     \etoc@redefetocstyle{chapter}}{}}
1221 \ifclassloaded{report}
1222     {\renewcommand*{\etocfontone{\normalfont\normalsize}
1223     \renewcommand*{\etocstandarddisplaystyle{\etocreportstyle}
1224     \etoc@redefetocstyle{chapter}}{}}
1225 \ifclassloaded{scrbook}

```

## 59. Implementation

```

1226 {\renewcommand*\etocfontone{\normalfont\normalsize}
1227 \renewcommand*\etocstandarddisplaystyle{\etocscrbookstyle}
1228 \Etoc@redefetocstyle{chapter}}{}
1229 \@ifclassloaded{scrreprt}
1230 {\renewcommand*\etocfontone{\normalfont\normalsize}
1231 \renewcommand*\etocstandarddisplaystyle{\etocscrreprtstyle}
1232 \Etoc@redefetocstyle{chapter}}{}

```

1.07k (2014/03/06) adds the appendix to the list of known levels if class memoir is detected.

```

1233 \@ifclassloaded{memoir}
1234 {\etocsetlevel{appendix}{0}%
1235 \renewcommand*\etocfontone{\normalfont\normalsize}
1236 \etocmemoirtotocfmt{chapter}{\contentsname}%
1237 \renewcommand*\etocstandarddisplaystyle{\etocmemoirstyle}
1238 \Etoc@redefetocstyle{chapter}}{}

```

1.07k (2014/03/06) adds the compatibility with the tocloft package; in compatibility mode etoc will obey the tocloft customisation for the division headings as well as for the toc title.

1.07l (2014/04/02) fixes the bug from the tocloft compatibility layer which was memoir incompatible: memoir has its version of tocloft which doesn't have the \ifcftnctoc boolean.

```

1239 \def\etocloftstyle {%
1240 \etocsettocstyle
1241 {\@cfttocstart\par\beginingroup
1242 \parindent\z@ \parskip\cftparskip \@cftmaketoctitle
1243 \if@cfttocbibind\@cftdobibtoc\fi }%
1244 {\endgroup\@cfttocfinish }%
1245 }
1246 \@ifclassloaded{memoir}{}
1247 {\@ifpackageloaded{tocloft}
1248 {\ifcftnctoc\else
1249 \renewcommand*\etocstandarddisplaystyle{\etocloftstyle}
1250 \AtBeginDocument{\let\tableofcontents\etoc\tableofcontents}
1251 \fi }
1252 {\AtBeginDocument
1253 {\@ifpackageloaded{tocloft}
1254 {\ifcftnctoc\else
1255 \PackageWarning{etoc}
1256 {Package 'tocloft' was loaded after 'etoc'!^^}%
1257 **** to prevent it from overwriting \protect\tableofcontents, it will be tricked^^}%
1258 **** into believing to have been loaded with option 'titles'. For better^^}%
1259 **** compatibility, please load 'tocloft' before 'etoc'}%
1260 \AtEndDocument{\PackageWarning{etoc}
1261 {Please load 'tocloft' /before/ 'etoc'!}}\fi
1262 \@cftnctoctrue }%
1263 {}%
1264 }%
1265 }%
1266 }

```

placeholder for comments

```

1267 \def\Etoc@addtocontents #1#2%
1268 {\ifEtoc@hyperref
1269 \addtocontents {toc}{\protect\contentsline
1270 {#1}{#2}%
1271 {\thepage }{\@currentHref }}%
1272 \else
1273 \addtocontents {toc}{\protect\contentsline
1274 {#1}{#2}{\thepage }}%

```

```

1275     \fi}
1276 \def\Etoc@addcontentsline@ #1#2#3%
1277     {\@namedef{toclevel@#1}{#3}%
1278     \addcontentsline {toc}{#1}{#2}}
1279 \DeclareRobustCommand*\etoc@toccontentsline{
1280     {\@ifstar{\Etoc@addcontentsline@}{\Etoc@addtocontents}}

```

All the following added for version 1.07g [2013/10/13]. Motivated by a demand about tocvsec2. But it is impossible to make tocvsec2 compatible, if not re-doing all ‘toc’ macros in tocvsec2 (v1.3, 2011/08/07). I do not retain the stack idea, which does not convince me as useful. The empty line in the .toc file is put there by tocvsec2 (see its code line 143).

I take this opportunity to add to the start of the toc file \let\etoc@startlocaltoc\@gobble in case etoc is not detected, I should possibly have done that long ago (like I did in tableof).

In 1.07h I decide that I should not have used \Etoc@ but \etoc@ for \Etoc@settocdepth. So I now use lowercase and add a temporary line for transition in the very improbable situation that someone actually did already use the just released 1.07g.

In 1.07i the deactivation of \etoc@settocdepth done at the start of the .toc file (for the case this file is used after having removed etoc from the document, or imported in another document) is done via letting it to \@gobble rather than \count@ (which would not be ok, as what follows would be undefined too). The reason for the \count@ also used originally in \etocignoretoctocdepth is that when I first worked on 1.07g I possibly had more than one token after \etoc@settocdepth, but I changed that later.

```

1281 \def\Etoc@storetocdepth {\global\c@etoc@tocdepth\c@tocdepth }
1282 \def\Etoc@resettocdepth {\global\c@tocdepth\c@etoc@tocdepth }
1283 \def\etocobeytoctocdepth {\def\etoc@settocdepth
1284     {\afterassignment\Etoc@@nottodeep \global\c@tocdepth}}
1285 \def\Etoc@@nottodeep {\ifnum\c@tocdepth>\c@etoc@tocdepth
1286     \global\c@tocdepth\c@etoc@tocdepth\fi }
1287 \def\etocignoretoctocdepth {\let\etoc@settocdepth\@gobble }
1288 \def\etocsettocdepth {\futurelet\Etoc@nexttoken\Etoc@set@tocdepth }
1289 \def\Etoc@set@tocdepth {\ifx\Etoc@nexttoken\bgroup
1290     \expandafter\Etoc@set@tocdepth@
1291     \else\expandafter\Etoc@set@toctocdepth
1292     \fi }
1293 \def\Etoc@set@tocdepth@ #1{\@ifundefined {Etoc@#1@@}
1294     {\PackageWarning{etoc}
1295     {Unknown sectioning unit #1, \protect\etocsettocdepth\space ignored}}
1296     {\global\c@tocdepth\csname Etoc@#1@@\endcsname}%
1297 }
1298 \def\Etoc@set@toctocdepth #1#{\Etoc@set@toctocdepth@ }
1299 \def\Etoc@set@toctocdepth@ #1{%
1300     \@ifundefined{Etoc@#1@@}%
1301     {\PackageWarning{etoc}
1302     {Unknown sectioning depth #1, \protect\etocsettocdepth.toc ignored}}%
1303     {\addtocontents {toc}
1304     {\protect\etoc@settocdepth\expandafter\protect\csname Etoc@#1@@\endcsname}}%
1305 }

```

1.07h adds the depth tags, and an even more general mechanism could be added at some point: we could insert ‘action’ tags and have the user map them to arbitrary commands. For the time being we just provide \etocdepthtag.toc {name} and \etocsettagdepth {tag\_name}{level} where level is numeric or alphabetical. The numeric level -3 is now again not accepted by \etocsetlevel, it can be used with \etocsettagdepth (which accepts equivalently none).

```

1306 \def\etocdepthtag #1#{\Etoc@depthtag } % \etocdepthtag.toc {name}
1307 \def\Etoc@depthtag #1{\addtocontents {toc}{\protect\etoc@depthtag {#1}}}
1308 \def\etocignoredepthtags {\let\etoc@depthtag \@gobble }

```

## 59. Implementation

```

1309 \def\etocobeydepthtags {\let\etoc@depthtag \Etoc@depthtag@ }
1310 \def\Etoc@depthtag@ #1{\@ifundefined{Etoc@depthof@#1}%
1311     {}% ignore in silence if tag has no associated depth
1312     {\afterassignment\Etoc@@nottodeep
1313         \global\c@tocdepth\cname Etoc@depthof@#1\endcsname}%
1314 }
1315 \def\etocsettagdepth #1#2{\@ifundefined{Etoc@#2@@}%
1316     {\PackageWarning{etoc}
1317         {Unknown sectioning depth #2, \protect\etocsettagdepth\space ignored}}%
1318     {\@namedef{Etoc@depthof@#1}{\@nameuse{Etoc@#2@@}}}%
1319 }

```

We must cancel all tocvsec2 toc-related actions. But a check must be done for the memoir class, as its tocvsec2 emulation does not have the incompatible things etoc needs to revert. This extra check added with etoc 1.071. Release 1.08g modifies the info message under class memoir (and downgrades it from warning to info).

```

1320 \def\Etoc@tocvsec@err #1{\PackageError {etoc}
1321     {the command \protect#1\space is incompatible with etoc}
1322     {use \protect\etocsettocdepth.toc as replacement (see etoc manual)}}%
1323 }%
1324 \AtBeginDocument {%
1325     \@ifclassloaded{memoir}
1326     {\PackageInfo {etoc}
1327         {Regarding memoir class command \protect\settocdepth, consider^^J
1328         \protect\etocsettocdepth.toc as a drop-in replacement with
1329         more capabilities^^J (see etoc manual). Similarly
1330         \protect\etocsettocdepth\space and
1331         \protect\etocsetnexttocdepth^^J should replace
1332         use of memoir's \protect\maxtocdepth\space command.}}
1333     {\@ifpackageloaded {tocvsec2}
1334         {\def\maxtocdepth #1{\Etoc@tocvsec@err \maxtocdepth }%
1335         \def\settocdepth #1{\Etoc@tocvsec@err \settocdepth }%
1336         \def\resettocdepth {\@ifstar {\Etoc@tocvsec@err \resettocdepth }%
1337             {\Etoc@tocvsec@err \resettocdepth }%
1338             }%
1339         \def\save@tocdepth #1#2#3{}% % if etoc is added to file previously
1340         \let\reset@tocdepth\relax % using tocvsec2
1341         \let\remax@tocdepth\relax
1342         \let\tableofcontents\etoc\tableofcontents
1343         \PackageWarning {etoc}
1344         {package tocvsec2 detected and its modification of^^J
1345         \protect\tableofcontents\space reverted. Use
1346         \protect\etocsettocdepth.toc as a replacement^^J
1347         for the tocvsec2 toc-related commands}}}%
1348     }%
1349 }%

```

1.07g. Initial version of \etocsetnexttocdepth.

1.08h: avoid after-effect of consecutive uses of \etocsetnexttocdepth with no intervening TOC, due to the fact that the command did an immediate modification of the tocdepth counter. This is now delayed until the actual TOC typesetting.

```

1350 \def\invisibletableofcontents {\etocsetnexttocdepth {-3}\tableofcontents }%
1351 \def\invisiblelocaltableofcontents
1352     {\etocsetnexttocdepth {-3}\localtableofcontents }%
1353 \def\etocsetnexttocdepth #1{%
1354     \@ifundefined{Etoc@#1@@}
1355     {\PackageWarning{etoc}
1356         {Unknown sectioning unit #1, \protect\etocsetnexttocdepth\space ignored}}

```



```

1357     {\def\Etoc@tocdepthset
1358       {\edef\Etoc@tocdepthreset {%
1359         \global\c@tocdepth\the\c@tocdepth\space
1360         \global\let\noexpand\Etoc@tocdepthreset\noexpand\@empty }%
1361       \global\c@tocdepth\csname Etoc@#1@\endcsname
1362       \global\let\Etoc@tocdepthset\@empty}}%
1363 }%

```

1.08k. Serves to make local TOCs even after headings which were not added to the .toc file; but of course the local TOCs can only display sub-headings which made their way into the .toc file...

```

1364 \def\etocsetlocaltop #1#{\Etoc@set@localtop}%
1365 \def\Etoc@set@localtop #1{%
1366   \@ifundefined{Etoc@#1@@}%
1367   {\PackageWarning{etoc}
1368     {Unknown sectioning depth #1, \protect\etocsetlocaltop.toc ignored}}%
1369   {\addtocontents {toc}
1370     {\protect\etoc@setlocaltop\expandafter\protect\csname Etoc@#1@@\endcsname}}%
1371 }%
1372 \def\etoc@setlocaltop #1{%
1373   \global\let\Etoc@level #1%
1374   \Etoc@skipthisonefalse
1375   \ifnum\Etoc@level=\Etoc@@six@@
1376     \Etoc@skipthisonetrue
1377   \else
1378     \ifEtoc@localtoc
1379       \ifEtoc@stoptoc\Etoc@skipthisonetrue
1380     \else
1381       \ifEtoc@notactive
1382         \Etoc@setflags
1383         \Etoc@skipthisonetrue
1384       \else
1385         \ifnum\Etoc@level<\Etoc@localtop
1386           \Etoc@skipthisonetrue
1387           \global\Etoc@stoptoctrue
1388         \fi
1389       \fi
1390     \fi
1391   \fi
1392   \let\Etoc@next\@gobble
1393   \ifEtoc@skipthisone
1394   \else
1395     \ifnum\c@tocdepth<\Etoc@level
1396     \else
1397       \ifEtoc@standard
1398       \else
1399         \let\Etoc@next\@firstofone
1400       \fi
1401     \fi
1402   \fi
1403   \Etoc@next{\Etoc@doends\Etoc@dobegins}%
1404 }%
1405 }%

```

1.07h. In order for the .toc file to be usable without error even without etoc. In 1.07i use of \@gobble for \etoc@settocdepth.

```

1406 \addtocontents {toc}{\protect\@ifundefined{etocstyle}%
1407   {\let\protect\etoc@startlocaltoc\protect\@gobble
1408   \let\protect\etoc@settocdepth\protect\@gobble

```

## 59. Implementation

```
1409          \let\protect\etoc@depthtag\protect\@gobble
1410          \let\protect\etoc@setlocaltop\protect\@gobble}{}}%
    Initializations.
1411 \def\etocstandardlines {\Etoc@standardtrue}
1412 \def\etococlines      {\Etoc@standardfalse}
1413 \etocdefaultlines
1414 \etocstandardlines
1415 \etocstandarddisplaystyle
1416 \etocobeytoctocdepth
1417 \etocobeydepthtags
1418 \let\Etoc@tocdepthreset\@empty % new 1.08h functioning of \etocsetnexttocdepth
1419 \let\Etoc@tocdepthset  \@empty %
1420 \let\etocaftertochook  \@empty % public for end user.
1421 \let\etocbeforetitlehook \@empty % new with 1.07m
1422 \endinput
```