

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 its filename in the margin after the verbatim rendering of the corresponding code. Clicking on this filename will on supporting viewers trigger some dialog for extracting the file, or perhaps even will extract it automatically and open it in some text editor. This is much better than copy-paste which often loses indentation. At 1.09h the formerly used icon in the text body has been replaced by the filename itself, placed in the page margin.

In some PDF viewers side panels may provide a convenient clickable list of all these **attachments annotations**.

Throughout the documentation the command names displayed **with this colour** are doubly hyperlinked: the left half of the name links to the user documentation, the second half links to the source code. You can try it out now: `\localtableofcontents`. But read this first: if you get lost inside the source code, clicking on control sequences displayed **with this colour** brings you back to the part of the user manual discussing that specific command.

TABLE OF CONTENTS

etoc

Abstract	p. 6
Foreword	1, p. 6
License	2, p. 7

Part I. Overview

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

Documentation (2023/01/15) generated from the source file with timestamp "Time-stamp: <15-01-2023 at 11:56:19 CET>".

Part II. Arbitrarily many TOCs, and local ones too

The <code>\tableofcontents</code> , <code>\localtableofcontents</code> and <code>\localtableofcontentswith-relativedepts</code> commands	9, p. 16
Labeling and reusing elsewhere	10, p. 17
A powerful functionality of etoc : the re-assignment of levels with <code>\etocset-level</code>	11, p. 18
The <code>\etocsettocdepth</code> and <code>\etocsetnexttocdepth</code> commands	12, p. 19
The hyperref option <code>bookmarksdepth</code> .	
The command <code>\etocsettocdepth.toc</code>	13, p. 20
The commands <code>\etocobeytocdepth</code> and <code>\etocignoretocdepth</code> .	
The commands <code>\etocdepthtag.toc</code> and <code>\etocsettagdepth</code>	14, p. 21
The commands <code>\etocobeydepthtags</code> and <code>\etocignoredepthtags</code> .	
The commands <code>\etocglobaldefs</code> and <code>\etoclocaldefs</code>	15, p. 22
Not displayed empty TOCs	16, p. 23
The <code>\etoccheckemptiness</code> command – The <code>\etocnotocifnoc</code> command – The <code>\etocifwasempty</code> command.	
Adding commands to the <code>.toc</code> file	17, p. 24
The hyperref option <code>hidelinks</code> .	

Part III. Examples

Testing the compatibility mode	18, p. 26
Another compatibility mode	19, p. 27
Emulating the book class	20, p. 29
A framed display	21, p. 33
Another TOC with a background color	22, p. 34
A (crazy) inline display	23, p. 36

Part IV. Surprising uses of **etoc**

The TOC of TOCs	24, p. 39
Arbitrary “Lists Of...”, <code>\etococcontentsline</code>	25, p. 41
The TOC as a tree	26, p. 42
The TOC as a molecule	27, p. 45

CONTENTS

The TOC as a TikZ mind map	28, p. 47
The TOC as a (long) table	29, p. 50
A TOC self-adjusting widths for its typesetting	30, p. 57

Part V. Commands for the toc line styles

The <code>\etocsetstyle</code> , <code>\etocname</code> and <code>\etocpage</code> commands	31, p. 59
The <code>\etocskipfirstprefix</code> and <code>\etociffirst</code> commands	32, p. 61
The <code>\etocnumber</code> command	33, p. 61
The <code>\etocifnumbered</code> switch	34, p. 61
The <code>\etocthenname</code> , <code>\etocthenumber</code> , and <code>\etocthepage</code> commands	35, p. 62
The <code>\etoclink</code> command	36, p. 62
The <code>\etocthelinkedname</code> , <code>\etocthelinkednumber</code> , <code>\etocthelinkedpage</code> and <code>\etoc-thelink</code> commands	37, p. 63
DEPRECATED <code>\etocsavedsectiontocline</code> , etc... commands.	38, p. 64
The <code>\etocsetlevel</code> command	39, p. 64
Scope of commands added to the <code>.toc</code> file	40, p. 65
Testing the scope – This is a (pale) red subsection for illustrative purposes.	
Am I also red?	41, p. 66

Part VI. Commands for the toc display style

Specifying the toc display style	42, p. 67
The command <code>\etocsettocstyle</code> – The commands <code>\etocmulticolstyle</code> , <code>\etocmulticol</code> , and <code>\etoc-localmulticol</code> – The command <code>\etoclocalmulticol</code> – The command <code>\etoclocalruled</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 and hooks	43, p. 71

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

Summary of the main style commands	44, p. 72
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. 73
Customizing <code>etoc</code>	46, p. 77
Customizing the <code>etoc</code> pre-defined line styles – Customizing the toc display styles.	

One more example of colored TOC layout	47, p. 79
---	------------------

Part VIII. Tips

The <code>\etoclocaltop</code> command	48, p. 82
The <code>\etocsetlocaltop.toc</code> command	49, p. 83
Hacking framed parboxes	50, p. 85
Interverting the levels	51, p. 85
All subsections of this document.	
Displaying statistics	52, p. 86
Using depth tags	53, p. 88
Typesetting the TOC as a table (the old way)	54, p. 90

Part IX. `etoc` and the outside world

Constraints on the <code>.toc</code> file constitution	55, p. 93
Compatibility with document classes	56, p. 94
Compatibility with the KOMA-script classes – Compatibility with the memoir class – Compatibility with beamer.	
Compatibility with other packages	57, p. 95
Compatibility with babel – Compatibility with hyperref – Compatibility with multicol – Compatibility with tocloft – Compatibility with tocvsec2 – Compatibility with tableof – Compatibility with tocstyle.	
T_EXnical matters	58, p. 97

Part X. The code

Timestamp	59, p. 98
Change history	60, p. 98
Implementation	61, p. 102

etoc

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 times¹ 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

Throughout the documentation the command names displayed **with this colour** are doubly hyperlinked: the left half of the name links to the user documentation, the second half links to the source code. You can try it out now: `\localtableofcontents`. But read this first: if you get lost inside the source code, clicking on control sequences displayed **with this colour** brings you back to the part of the user manual discussing that specific command.

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^AT_EX 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

¹If `\etockeeporiginaltableofcontents` is issued in the preamble after package loading, then `\tableofcontents` keeps the non-**etoc** meaning and one must use `\etoc\tableofcontents` in its place to get the **etoc** features. This was added to fix a compatibility issue with `listings`’s `\lstlistoflistings`, as it needs the `\tableofcontents` macro to keep its original meaning.

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: 1.1b
% License: LPPL 1.3c
% Copyright (C) 2012-2023 Jean-Francois Burnol <jfbu at free dot fr>
%
% 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 Author of this Work is:
% Jean-Francois Burnol <jfbu at free dot fr>
%
% This Work consists of the main source file etoc.dtx and the derived
% files etoc.sty, etoc.tex, etoc.pdf, etoc.dvi, README.md.
%
% Running etex (or latex or pdflatex) on etoc.dtx extracts etoc.sty,
% etoc.tex and README.md. See README.md for further instructions.
```

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`, 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 53](#), 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 20](#).

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 42 visible tables of contents (and a few invisible ones) and uses only one `.toc` file!² So one should think twice before adding manually extra commands to the `.toc` file (see [section 17](#)).

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 environments: 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

²and 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.

4.2. `\etocsettocstyle` for the toc display

their scope limited in this way inside a group (L^AT_EX’s environments create scope limiting groups).

Therefore the built-in “line styles” proposed by `etoc` as an example (and which are illustrated³ 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 51.1](#) (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.⁴

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`, `\etocmulticolstyle`, `\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`.

³with a twist, subsections having been downgraded to the subsubsection style...

⁴under 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”, **etoc** defaults to a compatibility mode which uses the defaults from the document class.⁵ This can always be re-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` sets the line styles to custom ones 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 (but see footnote⁵); 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 19](#).

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...`,⁶ the local TOC will exist only through its clones elsewhere in the document.

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

3. Do I need to be a geek to use etoc?	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 <code>\etocsetstyle</code> for the line styles	9
4.2 <code>\etocsettocstyle</code> for the toc display	9

⁵for the “toc display style” **etoc** checks if it knows the class, and then uses emulation code which was added manually to its source, and if not it defaults to the `article` class layout. No automated way to recover the global toc display for arbitrary document classes is implemented. But **etoc** will detect if `tocloft` has customized the TOC title.

⁶this 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.

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

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 53](#) 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^AT_EX2e'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
```

Depending on your PDF viewer, clicking on the margin filename may allow to extract it. [etocsnippet-01.tex](#)

6. A second example

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 14](#) and [section 53](#). 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	59
32. THE <code>\ETOCSKIPFIRSTPREFIX</code> AND <code>\ETOCIFFIRST</code> COMMANDS	61
33. THE <code>\ETOCNUMBER</code> COMMAND.....	61
34. THE <code>\ETOCIFNUMBERED</code> SWITCH	61
35. THE <code>\ETOCTHENAME</code> , <code>\ETOCTHENUMBER</code> , AND <code>\ETOCTHEPAGE</code> COMMANDS.....	62
36. THE <code>\ETOC LINK</code> COMMAND.....	62
37. THE <code>\ETOCTHELINKEDNAME</code> , <code>\ETOCTHELINKEDNUMBER</code> , <code>\ETOCTHELINKEDPAGE</code> AND <code>\ETOCTHELINK</code> COMMANDS	63
38. DEPRECATED <code>\ETOC SAVED SECTION TO CLINE</code> , ETC... COMMANDS.	64
39. THE <code>\ETOCSETLEVEL</code> COMMAND.....	64
40. SCOPE OF COMMANDS ADDED TO THE <code>.TOC</code> FILE.....	65
<i>Testing the scope (40.1, p. 65). This is a (pale) red subsection for illustrative purposes (40.2, p. 66).</i>	
41. AM I ALSO RED?	66

PART VI – COMMANDS FOR THE TOC DISPLAY STYLE

42. SPECIFYING THE TOC DISPLAY STYLE	67
<i>The command <code>\etocsettocstyle</code> (42.1, p. 67). The commands <code>\etocmulticolstyle</code>, <code>\etocmulticol</code>, and <code>\etoclocalmulticol</code> (42.2, p. 68). The command <code>\etoclocalstyle</code> (42.3, p. 68). The commands <code>\etocruledstyle</code>, <code>\etocruled</code> and <code>\etoclocalruled</code> (42.4, p. 69). The commands <code>\etocframedstyle</code>, <code>\etocframed</code>, and <code>\etoclocalframed</code> (42.5, p. 69). Headings, titles, <code>\etocoldpar</code>, <code>\etocinnertopsep</code> (42.6, p. 70). The compatibility mode <code>\etocstandarddisplaystyle</code> (42.7, p. 70). The command <code>\etocinline</code> (42.8, p. 70).</i>	

43. STARRED VARIANTS AND HOOKS.....	71
-------------------------------------	----

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

```

Depending on your PDF viewer, clicking on the margin filename may allow to extract it. [etocsnipet-02.tex](#)

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

7. A Beautiful Thesis example

	etoc?	8
3.1	<i>Limitations in the use of list environments for tables of contents</i>	8
Chapter 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
Chapter 5	A first example	10
Chapter 6	A second example	12
Chapter 7	A Beautiful Thesis example	13
Chapter 8	Linked list of the main package commands	15

```
\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}
\endgroup
```

[etocsnippet-03.tex](#) Depending on your PDF viewer, clicking on the margin filename may allow to extract it.

8. Linked list of the main package commands

<code>\tableofcontents</code>	<code>\etocsettocstyle</code>	}	<code>\etocname</code>
<code>(optional \ref or \label)</code>	<code>\etocsetstyle</code>		<code>\etocnumber</code>
<code>\localtableofcontents</code>	<code>\etocsetnexttocdepth</code>		<code>\etocpage</code>

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

The above is not an exhaustive list of all the package user commands. And for legacy arbitrary reasons some of the more obscure commands are included here.

Part II.

Arbitrarily many TOCs, and local ones too

Here are some statistics for this part: it contains 9 sections and 7 subsections. The name of the first section is “The `\tableofcontents`, `\localtableofcontents` and `\localtableofcontentswithrelativedepts` commands” and the corresponding number is “9”. The name of the last section is “Adding commands to the `.toc` file” and its number is “17”. The name of the first subsection is “The hyperref option `bookmarksdepth`” and the corresponding number is “12.1”. The name of the last subsection is “The hyperref option `hidelinks`” and its number is “17.1”.

9. The <code>\tableofcontents</code>, <code>\localtableofcontents</code> and <code>\localtableofcontentswithrelativedepts</code> commands	16
10. Labeling and reusing elsewhere	17
11. A powerful functionality of <code>etoc</code>: the re-assignment of levels with <code>\etocsetlevel</code>	18
12. The <code>\etocsettocdepth</code> and <code>\etocsetnexttocdepth</code> commands	19
12.1. The hyperref option <code>bookmarksdepth</code>	20
13. The command <code>\etocsettocdepth.toc</code>	20
13.1. The commands <code>\etocobeytoctocdepth</code> and <code>\etocignoretoctocdepth</code> .	21
14. The commands <code>\etocdepthtag.toc</code> and <code>\etocsettagdepth</code>	21
14.1. The commands <code>\etocobeydepthtags</code> and <code>\etocignoredepthtags</code>	22
15. The commands <code>\etocglobaldefs</code> and <code>\etoclocaldefs</code>	22
16. Not displayed empty TOCs	23
16.1. The <code>\etoccheckemptiness</code> command	23
16.2. The <code>\etocnotocifnotoc</code> command	23
16.3. The <code>\etocifwasempty</code> command	24
17. Adding commands to the <code>.toc</code> file	24
17.1. The hyperref option <code>hidelinks</code>	25

9. The `\tableofcontents`, `\localtableofcontents` and `\localtableofcontentswithrelativedepts` commands

`\tableofcontents` can be used arbitrarily many times in the document. Styling either globally the TOC or its individual entries is customizable at any time in the document.

`\etoc` is a synonym to `etoc`'s `\tableofcontents`. See footnote¹ on page 6.

`\localtableofcontents` will print local tables of contents: *i.e.* all sections and subunits inside a given chapter, or all subsections and lower inside a given subsection, etc... (see also `\etocsetnexttocdepth`).^{7,8}

`\localtableofcontentswithrelativedepth{<number>}` can be used to override the document or current `tocdepth` setting (see section 11 for a discussion of `tocdepth`) to become relative to where the local TOC originates. For example, assuming the default numeric level assignments to standard sectioning units

```
\section{This is a section}
\localtableofcontentswithrelativedepth{+2}
```

will create a local table of contents taking into account the subsections and subsubsections inside this section, independently of what is the value of the `tocdepth` counter at this position in the document. If the numeric argument had been 3, the local TOC would have displayed also paragraphs. If the section had been a chapter, and again for a relative `tocdepth` of 2, the taken into account levels would have been sections and subsections.⁹

The macro `\localtableofcontentswithrelativedepth` is **new with 1.09**.¹⁰

10. Labeling and reusing elsewhere

`etoc` allows the labeling of a TOC with (for example) `\label{toc:A}` and will redisplay 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 the following page.

We used this:

```
\begin{figure}[ht!]
  \centering
  \begingroup
  \etocstandardlines
  \renewcommand{\etocbkgcolorcmd}{\color{green!5}}
  \renewcommand{\etocbelowtocskip}{0pt\relax}
  \fboxsep1ex
  \etocframedstyle [1]{\fbox{\makebox[.5\linewidth]{\etocfontminusone
    \hyperref[toc:c]{I am from far away}}}}
```

⁷As is explained in section 10 the syntax allows to create somewhere a local table of contents and to display it at some other location either before or after its origin.

⁸As is explained in section 11 `etoc` allows at anytime to locally redefine the numeric levels associated to named ones, which brings great flexibility to achieve special effects, all done using only a single auxiliary file, the standard `.toc` file.

⁹The situation may be more complex, as the `.toc` file itself may well contain `\setcounter{tocdepth}{...}` commands. This is not recommended practice with `etoc`. See the discussion of `\etocsettocdepth.toc` and `\etocdepthtag.toc` for more on this.

¹⁰Thanks to Tony ROBERTS for feature request.

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

I am from far away

44. Summary of the main style commands	72
44.1. Setting up local styles	72
44.2. Setting up toc display styles	72
44.3. Displaying tables of contents	73
44.4. Labels and references	73
45. The package default line styles: <code>\etocdefaultlines</code>	73
46. Customizing etoc	77
46.1. Customizing the etoc pre-defined line styles	77
46.2. Customizing the toc display styles	78
47. One more example of colored TOC layout	79

```
\tableofcontents \label{toc:d} \ref{toc:c}
\endgroup
\end{figure}
```

[etocsnippet-04.tex](#)

Depending on your PDF viewer, clicking on the margin filename may allow to extract it. In the above example, not only did we use `\ref{toc:c}` to print here the distant (local) table of contents which has been labeled `toc:c` but we added a (possibly confusing) `\label{toc:d}`. This is done for the down-to-earth reason of being able to use, as we did in the previous paragraph, `\vpageref{toc:d}`. But if one wants to clone again the original local table of contents, one must reference its original label: `\tableofcontents \ref{toc:c}`.¹¹ This original local table of contents is to be found on page 72.

11. 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^AT_EX `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 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

¹¹Why does this author always give complicated examples rather than down-to-earth ones?

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)¹²

12. 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.¹³ 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>}`.

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. The package defines `\invisibletableofcontents` essentially as

```
\etocsetnexttocdepth{none}\tableofcontents
```

¹²and one should naturally not print this TOC of paragraphs in compatibility mode, which would insist on inserting a gigantic left margin.

¹³In the standard classes (at least), it also influences the `\listoftables` and `\listoffigures`, via `\@dottedtocline`.

13. The command `\etocsettocdepth.toc`

The simplest organization is probably 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`.

It is possible to use `\etocsettocdepth` inside the first argument of `\etocsettocstyle` (possibly in conjunction with checking the `\etoclocaltop` value, *which however will be up-to-date there only if `\etoccheckemptiness` was executed*). There is no worry then about possible impact on hyperref bookmarks later on, because `etoc` always resets the `tocdepth` counter after typesetting a TOC to the value it had before it.

Since 1.09 the macro `\etocsetnexttocdepth` works also if located in first argument of `\etocsettocstyle`, but there is no reason to use it there as `\etocsettocdepth` has no durable effect on the `tocdepth` counter if executed there.

Check `\localtableofcontentswithrelativedepth` for a simpler way to control the depth of local tables of contents. This has the advantage of working reliably whether or not the `\etoccheckemptiness` is used.

12.1. 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.

13. The command `\etocsettocdepth.toc` and `\etocimmediatesettocdepth.toc`

This command `\etocsettocdepth.toc` implements some functionality of `tocvsec2`¹⁴, 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.

¹⁴ I thank Denis Brouzé for drawing my attention to the incompatibility of this package with `etoc`.

14. The commands `\etocdepthtag.toc` and `\etocsettagdepth`

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

Also `\etocimmediatesettagdepth.toc` is provided. For explanations, refer to the discussion of `\etocimmediatedepthtag.toc` in the next section.

13.1. The commands `\etocobeytoctocdepth` and `\etocignoretoctocdepth`

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

14. The commands `\etocdepthtag.toc`, `\etocimmediatedepthtag.toc` and `\etocsettagdepth`

Release 1.07h has a command `\etocdepthtag.toc` which allows to control dynamically the which contents end up included in the displayed TOCs (this documentation also described formerly a way using `\etocsettagdepth.toc` with some dummy level name, which got then set via `\etocsetlevel` according to what was locally needed, but it was too hacky and I am not sure if it was understandable).

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 (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`.

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 53](#).

When using `\etocdepthtag.toc` in combination with L^AT_EX's `\include`, data may not end up in the `.toc` file in the correct order. For example in this situation:

```
\clearpage % or anything ending up causing its presence here right before
            % the \etocdepthtag.toc
```

15. The commands `\etocglobaldefs` and `\etoclocaldefs`

```
\etocdepthtag.toc{sometag}  
\include{some file containing sections}
```

The tag will end up in the `.toc` file *after* all section headings from the included file. The cause is that \LaTeX inserts immediately in the main auxiliary file a command to input the auxiliary file of the included file (which in turn, contains instructions to add data to the `.toc` file). But `\etocdepthtag.toc` does not internally use such immediateness, as it uses the same interface as `\section` and alike commands when they want to write extra data to the `.toc` file.

So¹⁵ 1.09f adds `\etocimmediatedepthtag.toc` which will force the tag to be written immediately to the `.toc` file (well, rather immediately to the `.aux` file, so before the inclusion of the auxiliary file of the included file).

One should not use this variant systematically. For example if your document looks like:

```
\clearpage  
  
\section{bbbb}  
  
Some text  
  
\etocdepthtag.toc{sometag}  
\etocimmediatedepthtag.toc{someimmediatetag}  
\section{cccc}  
  
Some text  
  
\end{document}
```

then the `someimmediatetag` will end up being inserted in `.toc` file *before* the `bbbb` section. This is because \LaTeX 's `\section` uses a *delayed* write, not an *immediate* one. And `\etocdepthtag.toc` wisely uses a *delayed* write.

As it seems very hard programmatically to identify automatically if the *immediate* variant of `\etocdepthtag.toc` should be used, the package provides two separate commands and it is up to user to make the correct choice.

14.1. 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.

15. The commands `\etocglobaldefs` and `\etoclocaldefs`

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

¹⁵Thanks to Norman RAMSEY who reported this problem, together with a fix, in July...2016. Sorry for long delay before updating `etoc` six years later...

By default `etoc`'s definitions of `\etocname` etc... are local. This causes problems in certain contexts such as TOC as tables (section 29, section 54) 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.

16. Not displayed empty TOCs

16.1. 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.¹⁶ This can be useful to authors of \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 `\etocaftertochook` command; and the test for emptiness itself executes everything else found in the `.toc` file. See section 17 in this context.

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

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.

16.2. 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`

¹⁶Thanks to Paul Gaborit who asked for such a feature.

17. Adding commands to the .toc file

{etoc} nor `\tableofcontents?` ; -). Well, there is always the case of batch conversions of documents having or not sectioning units.

16.3. 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`).

17. 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`, `\etocignoretoctocdepth` and `\etocobeytoctocdepth` are the way to go; or their variants `\etocdepthtag.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.

17.1. 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 “18”. The name of the last section is “A (crazy) inline display” and its number is “23”.

18. Testing the compatibility mode	26
19. Another compatibility mode	27
20. Emulating the book class	29
21. A framed display	33
22. Another TOC with a background color	34
23. A (crazy) inline display	36

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](#).

18. 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.¹⁷ 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

42. Specifying the toc display style	67
42.1. The command <code>\etocsettocstyle</code>	67
42.2. The commands <code>\etocmulticolstyle</code> , <code>\etocmulticol</code> , and <code>\etoclocalmulticol</code>	68
42.3. The command <code>\etoclocstyle</code>	68
42.4. The commands <code>\etocruledstyle</code> , <code>\etocruled</code> and <code>\etoclocalruled</code> . . .	69
42.5. The commands <code>\etocframedstyle</code> , <code>\etocframed</code> , and <code>\etoclocalframed</code> .	69

¹⁷the present document uses the `scartcl` 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>	70
42.7. The compatibility mode <code>\etocstandarddisplaystyle</code>	70
42.8. The command <code>\etocinline</code>	70
43. Starred variants and hooks	71

19. 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.

One only needs to use the L^AT_EX standard `\l@chapter`, `\l@section`, etc... commands *Modified at 1.1a* inside the TOC line style definitions via `\etocsetstyle`, re-constituting their arguments using `\etocname`, `\etocnumber`, `\etocpage` as one wishes.

The aliases `\etocsavedsectiontocline`, `\etocsavedchaptertocline` etc... are set to have the exact same meanings as the L^AT_EX `\l@chapter`, `\l@section`, etc... commands each time a TOC is typeset. They are DEPRECATED as the latter L^AT_EX commands are not modified (nor used) by `etoc` since release 1.1a.

You should use directly `\l@chapter`, etc... within suitable `\makeatletter`/`\makeatother`. Here is an example.

```
\makeatletter
\newcommand{\MyStandardTOC}{%
  \begingroup
  % for the book or article classes:
  %\etocsetstyle{part}{}{}
  % {\l@part{\etocnumber\hspace{1em}\etocname}{\etocpage}}{}%
  % for the scrbook or scrartcl classes:
  \etocsetstyle{part}{}{}
  {\l@part{\numberline{\etocnumber}\etocname}{\etocpage}}{}%
  % following is identical in book/article/scrbook/scrartcl classes:
  \etocsetstyle{chapter}{}{} %% only for book and scrbook
  {\l@chapter{\numberline{\etocnumber}\etocname}{\etocpage}}{}%
  \etocsetstyle{section}{}{}
  {\l@section{\numberline{\etocnumber}\etocname}{\etocpage}}{}%
```

Contents

```
\etocsetstyle{subsection}{}{}
  {\l@subsection{\numberline{\etocnumber}\etocname}{\etocpage}}{}%
% etc... if further sectioning units are used
\etocstandarddisplaystyle % this is for the title, page-marks, etc...
\tableofcontents
\endgroup}
\makeatother
```

[etocsnippet-05.tex](#) Depending on your PDF viewer, clicking on the margin filename may allow to extract it.

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.¹⁸ 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`.¹⁹ 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

3. Do I need to be a geek to use <code>etoc</code>?	8
3.1. Limitations in the use of list environments for tables of contents	8
4. Line styles and toc display style	9
<i>\etocsetstyle for the line styles</i> (4.1) – <i>\etocsettocstyle for the toc display</i> (4.2) – <i>Compatibility mode</i> (4.3).	
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

This example only has sections and subsections (assumed to be numbered), and the code used was:

¹⁸and 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.

¹⁹For 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.

```

\makeatletter
\newcommand*{\MyQuasiStandardTOC}[1]{%
  \begingroup
  \etocsetstyle{section}
  {}
  {\ifnum\etocthenumber=4
    \etocsetstyle{subsection}
    {\par\nopagebreak\begingroup
     \leftskip1.5em \rightskip\@tocrmarg
     \parfillskip \@flushglue
     \parindent 0pt
     \normalfont\normalsize\rmfamily\itshape
     \etocskipfirstprefix}
    {\allowbreak\-,-,}
    {\etocname\ \textup{(\etocnumber)}}}
    {\.\par\endgroup}}%
  \else
    \etocsetstyle{subsection}
    {}%
    {}%
    {\l@section{\numberline{\etocnumber}\etocname}{\etocpage}}%
    {}%
  \fi
  }% prefix for section sets the style dynamically for subsections!
  {\l@section{\numberline{{\color{cyan}\etocthenumber}}\etocname}{\etocpage}}%
  {}%
  \etocstandarddisplaystyle
  \etocsetnexttocdepth {2}%
  \tableofcontents #1
  \endgroup
}
\makeatother

```

Depending on your PDF viewer, clicking on the margin filename may allow to extract it. [etocsnippet-06.tex](#)

The page heading (on the page where this TOC appears) may have been modified as is expected from usage of `\etocstandarddisplaystyle` in the code. 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. We had to examine the `scrartcl` sources to determine what to use for `\leftskip` and `\rightskip` for our customized section entry (the one with number 4).

20. 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 57.4](#)) 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 19](#).

20. Emulating the book class

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 BITOUZÉ 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 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 \TOCrighthmargin\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}%

```

20. Emulating the book class

```
\par }%
}
{}
{\advance\leftskip-#2\relax}%
}
\makeatother
```

[etocsnippet-07.tex](#) Depending on your PDF viewer, clicking on the margin filename may allow to extract it. 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.

```
\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

```

Depending on your PDF viewer, clicking on the margin filename may allow to extract it. As [etocsnippet-08.tex](#) 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.

21. 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 following page).

```

\etocdefaultlines
\begingroup
\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{\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
\fbxrule1pt
\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}
\endgroup

```

Depending on your PDF viewer, clicking on the margin filename may allow to extract it. [etocsnippet-09.tex](#)

22. 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	The commands <code>\etocframedstyle</code> , <code>\etocframed</code> , and <code>\etoclocalframed</code>
. 42, p. 67 42.5, p. 69
The command <code>\etocsettocstyle</code>	Headings, titles, <code>\etocoldpar</code> , <code>\etocinnertopsep</code> 42.6, p. 70
. 42.1, p. 67	The compatibility mode <code>\etocstandarddisplaystyle</code> 42.7, p. 70
The commands <code>\etocmulticolstyle</code> , <code>\etocmulticol</code> , and <code>\etoclocalmulticol</code> 42.2, p. 68	The command <code>\etocinline</code> 42.8, p. 70
The command <code>\etoclocmulticol</code> 42.2, p. 68	
The command <code>\etoclocmulticol</code> 42.2, p. 68	
The command <code>\etoclocmulticol</code> 42.2, p. 68	
The command <code>\etoclocmulticol</code> 42.2, p. 68	
The commands <code>\etocruledstyle</code> , <code>\etocruled</code> and <code>\etoclocalruled</code> 42.4, p. 69	
	Starred variants and hooks 43, p. 71

22. 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 36.

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](#).

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

\renewcommand{\etocbkgcolorcmd}{\color{tocbackground}}
\renewcommand{\etocleftulecolorcmd}{\color{tocbackground}}
\renewcommand{\etocrightrulecolorcmd}{\color{tocbackground}}
\renewcommand{\etocbottomrulecolorcmd}{\color{tocbackground}}
\renewcommand{\etocoprerulecolorcmd}{\color{tocbackground}}

\renewcommand{\etocleftule}{\vrule width 3cm}
\renewcommand{\etocrightrule}{\vrule width 1cm}
\renewcommand{\etocbottomrule}{\hrule height 12pt}
\renewcommand{\etocoprerule}{\hrule height 12pt}

\renewcommand{\etocinnertopsep}{0pt}
\renewcommand{\etocinnerbottomsep}{0pt}
```

22. Another TOC with a background color

```
\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
  \fboxsep0pt % \colorbox uses \fboxsep 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})}
```

23. A (crazy) inline display

`\end{figure}`

[etocsnippet-10.tex](#) Depending on your PDF viewer, clicking on the margin filename may allow to extract it.

Contents of Part One		
3	Do I need to be a geek to use etoc ?	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	<code>\etocsetstyle</code> for the line styles	9
4.2	<code>\etocsettocstyle</code> 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](#))

23. 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. . **etoc**: *Abstract, Foreword, License. Overview: Do I need to be a geek to use **etoc**?*²⁰, *Line styles and toc display style*²¹, *A first example, A second example, A Beautiful Thesis example, Linked list of the main package commands. Arbitrarily many TOCs, and local ones too: The `\tableofcontents`, `\localtableofcontents` and `\localtableofcontentswithrelativedepths` commands, Labeling and reusing elsewhere, A powerful functionality of **etoc**: the re-assignment of levels with `\etocsetlevel`, The `\etocsettocdepth` and `\etocsetnexttocdepth` commands²², The command `\etocsettocdepth.toc`²³, The commands `\etocdepthtag.toc` and `\etocsettagdepth`²⁴, The commands `\etocglobaldefs` and `\etoclocaldefs`, Not*

²⁰Limitations in the use of list environments for tables of contents.

²¹`\etocsetstyle` for the line styles; `\etocsettocstyle` for the toc display; Compatibility mode.

²²The hyperref option `bookmarksdepth`.

²³The commands `\etocobeytocdepth` and `\etocignoretocdepth`.

²⁴The commands `\etocobeydepthtags` and `\etocignoredepthtags`.

displayed empty TOCs²⁵, Adding commands to the .toc file²⁶. **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...”, \etoccontentsline, The TOC as a tree, The TOC as a molecule, The TOC as a TikZ mind map, The TOC as a (long) 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 \etocthenname, \etocthenumber, and \etocthepage commands, The \etoclink command, The \etocthelinkedname, \etocthelinkednumber, \etocthelinkedpage and \etocthelink commands, DEPRECATED \etocsavedsectiontocline, etc... commands., The \etocsetlevel command, Scope of commands added to the .toc file²⁷, Am I also red?. **Commands for the toc display style:** Specifying the toc display style²⁸, Starred variants and hooks. **Using and customizing the etoc own styles:** Summary of the main style commands²⁹, The package default line styles: \etocdefaultlines, Customizing etoc³⁰, One more example of colored TOC layout. **Tips:** The \etoclocaltop command, The \etocsetlocaltop.toc command, Hacking framed parboxes, Interverting the levels³¹, Displaying statistics, Using depth tags, Typesetting the TOC as a table (the old way). **etoc and the outside world:** Constraints on the .toc file constitution, Compatibility with document classes³², Compatibility with other packages³³, T_EXnical matters. **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}\footnotesize\upshape\etocskipfirstprefix}
  {; }
```

²⁵The \etoccheckemptiness command; The \etocnotocifnotoc command; The \etocifwasempty command.

²⁶The hyperref option *hidelinks*.

²⁷Testing the scope; **This is a (pale) red subsection for illustrative purposes.**

²⁸The command \etocsettocstyle; The commands \etocmulticolstyle, \etocmulticol, and \etoclocalmulticol; The command \etocstocstyle; The commands \etocruledstyle, \etocruled and \etoclocalruled; The commands \etocframedstyle, \etocframed, and \etoclocalframed; Headings, titles, \etocoldpar, \etocinnertopsep; The compatibility mode \etocstandarddisplaystyle; The command \etocinline.

²⁹Setting up local styles; Setting up toc display styles; Displaying tables of contents; Labels and references.

³⁰Customizing the etoc pre-defined line styles; Customizing the toc display styles.

³¹All subsections of this document.

³²Compatibility with the KOMA-script classes; Compatibility with the memoir class; Compatibility with beamer.

³³Compatibility with babel; Compatibility with hyperref; Compatibility with multicol; Compatibility with tocloft; Compatibility with tocsec2; Compatibility with tableof; Compatibility with tocstyle.

23. A (crazy) inline display

```
{\etocname}
{\end{lrbox}\footnote{\unhbox\forsubsection}}
\etocsetstyle{subsubsection}
{ (\itshape\etocskipfirstprefix
{, }
{\etocname}
{\/\upshape})}
\etocsettocstyle{Here is the inline table of contents. }{.\par}
\tableofcontents \label{toc:crazyinline}
\endgroup
```

[etocsnippet-11.tex](#) Depending on your PDF viewer, clicking on the margin filename may allow to extract it.

Part IV.

Surprising uses of **etoc**

Here are some statistics for this part: it contains 7 sections and 0 subsection. The name of the first section is “The TOC of TOCs” and the corresponding number is “24”. The name of the last section is “A TOC self-adjusting widths for its typesetting” and its number is “30”.

24. The TOC of TOCs	39
25. Arbitrary “Lists Of...”, <code>\etoccontentsline</code>	41
26. The TOC as a tree	42
27. The TOC as a molecule	45
28. The TOC as a TikZ mind map	47
29. The TOC as a (long) table	50
30. A TOC self-adjusting widths for its typesetting	57

24. The TOC of TOCs

Here is the numbered and linked list of all tables of contents which are displayed within this document:³⁴ [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#), [9](#), [10](#), [11](#), [12](#), [13](#), [14](#), [15](#), [16](#), [17](#), [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](#). 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 after the title). So, rather than including `\refstepcounter{visibletoc}` inside `\etocaftertitlehook`, we just put there `\stepcounter{visibletoc}` followed by the command `\etoccontentsline{visibletoc}{\thevisibletoc}`. This **etoc** command

³⁴The 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...

24. The TOC of TOCs

`\etoccontentsline{<level_name>}{<name>}` has the same effect as:

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

but its usefulness is to circumvent³⁵ 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').³⁶

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}
\begingroup
  \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
\endgroup

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

[etocsnippet-12.tex](#) Depending on your PDF viewer, clicking on the margin filename may allow to extract it.

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 51](#) gives another use of the shuffling of levels.

³⁵using `\addtocontents` rather than `\addcontentsline`

³⁶The 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`.

25. Arbitrary “Lists Of...”, `\etoc` commands and `\etocimmediatetoc`

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
```

26. The TOC as a tree

Depending on your PDF viewer, clicking on the margin filename may allow to extract it. [etocsnippet-13.t](#)

A related command `\etocimmediatetoccontentsline` (and its starred version) is also provided. For discussion and the meaning of “immediate”, refer to the analogous case of `\etocimmediatedepthtag.toc`.

26. The TOC as a tree

Using `tikz` and the package `forest` 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 LAMPOR’s book has no mention whatsoever of token lists, and L^AT_EX 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.³⁷ 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.

At 1.1a the commands `\etocthelinkedname`, etc. . . , are always providing an hyperlink, so it is not true that `\etocname`, etc. . . , are always simply their robust variants.

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.³⁸ This is the later job of a `forest` environment which will be given the contents of `\treetok`.

The resulting tree has been put in a `float`, which appears on the facing page. Here is the code used for its production:

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

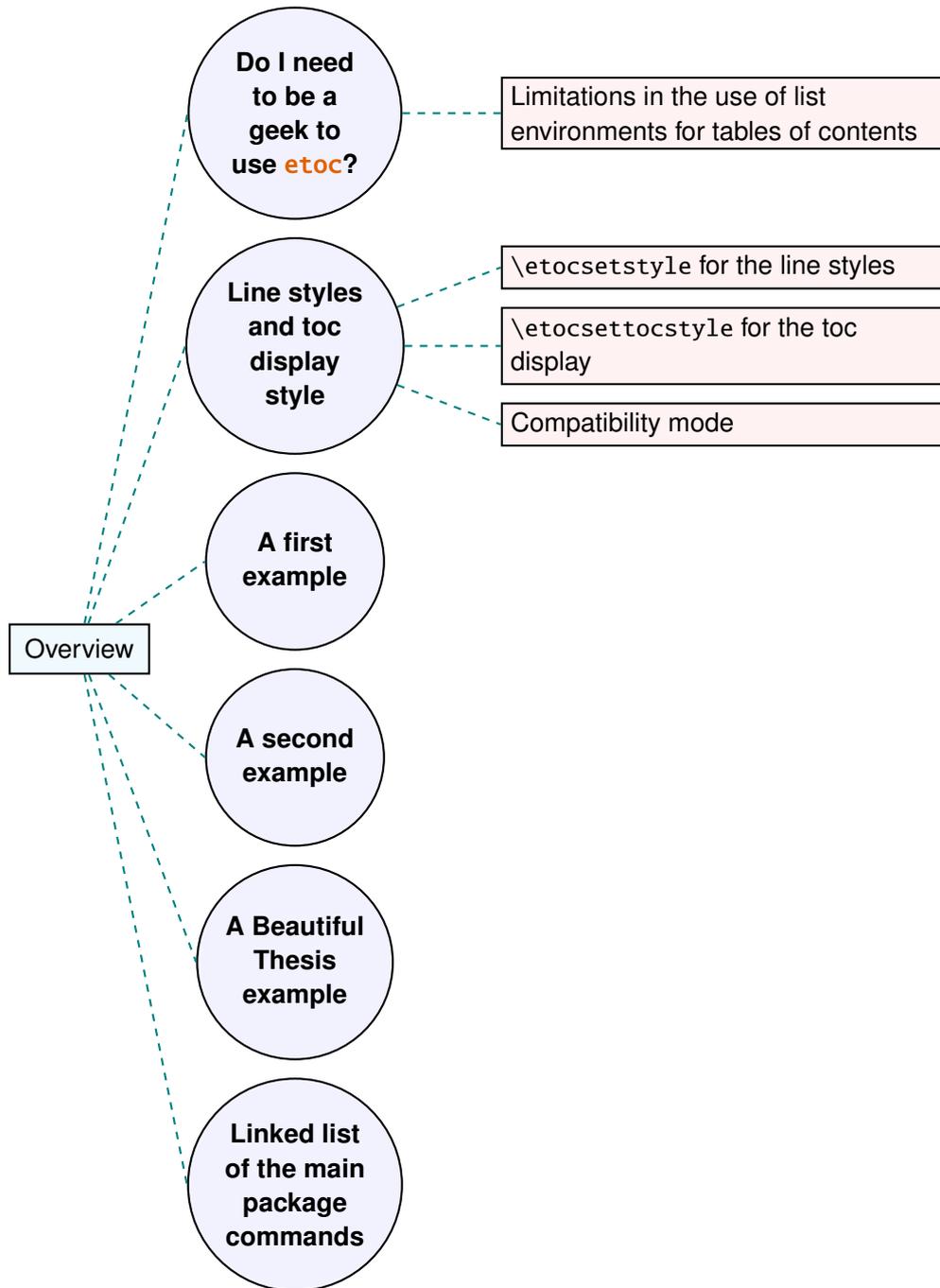
\newcommand*\appendtotok[2]{% #1=toks variable, #2=macro, expands once #2
  #1\expandafter\expandafter\expandafter
  {\expandafter\the\expandafter #1#2}}

\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}}}%
}
```

³⁷This 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).

³⁸There 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`.



}

```

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

```

\etocsetstyle{subsection}
  
```

26. The TOC as a tree

```
{\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}
```

[etocsnippet-14.tex](#) Depending on your PDF viewer, clicking on the margin filename may allow to extract it. Why `\hypersetup{hidelinks}`? as explained in [subsection 17.1](#), 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 52](#)) 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 54](#)); 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 T_EX (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}

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

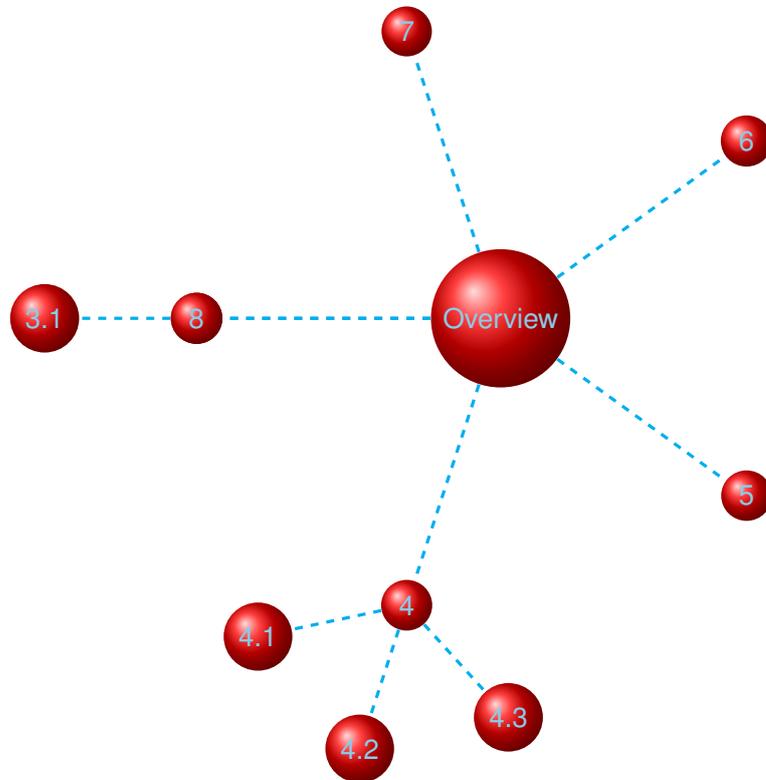
\etocsettocstyle
  {\treetok{\node {\hyperref[part:overview]{Overview}}}}
  {\global\appendtotok\treetok{ ;}}

\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,circle,text=SkyBlue},
     edge from parent path={[dashed,very thick,color=cyan]
       (\tikzparentnode) -(\tikzchildnode)}]
    \the\treetok
  \end{tikzpicture}
```

27. The TOC as a molecule

`\end{figure}`

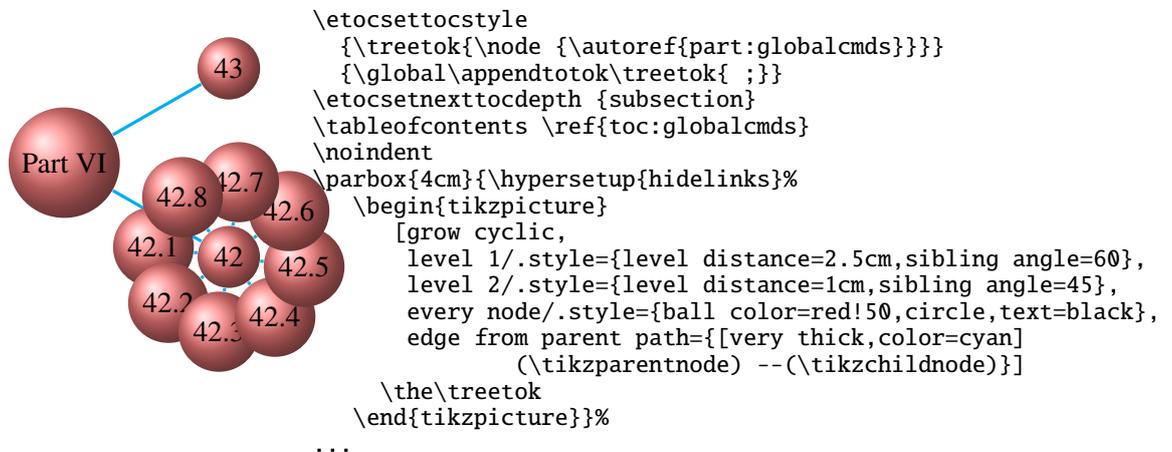
[etocsnippet-15.tex](#) Depending on your PDF viewer, clicking on the margin filename may allow to extract it.



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 above.

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 ε -TeX primitive `\unexpanded` will simplify the code.³⁹

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 following 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 }%
}

\newcounter{partco}

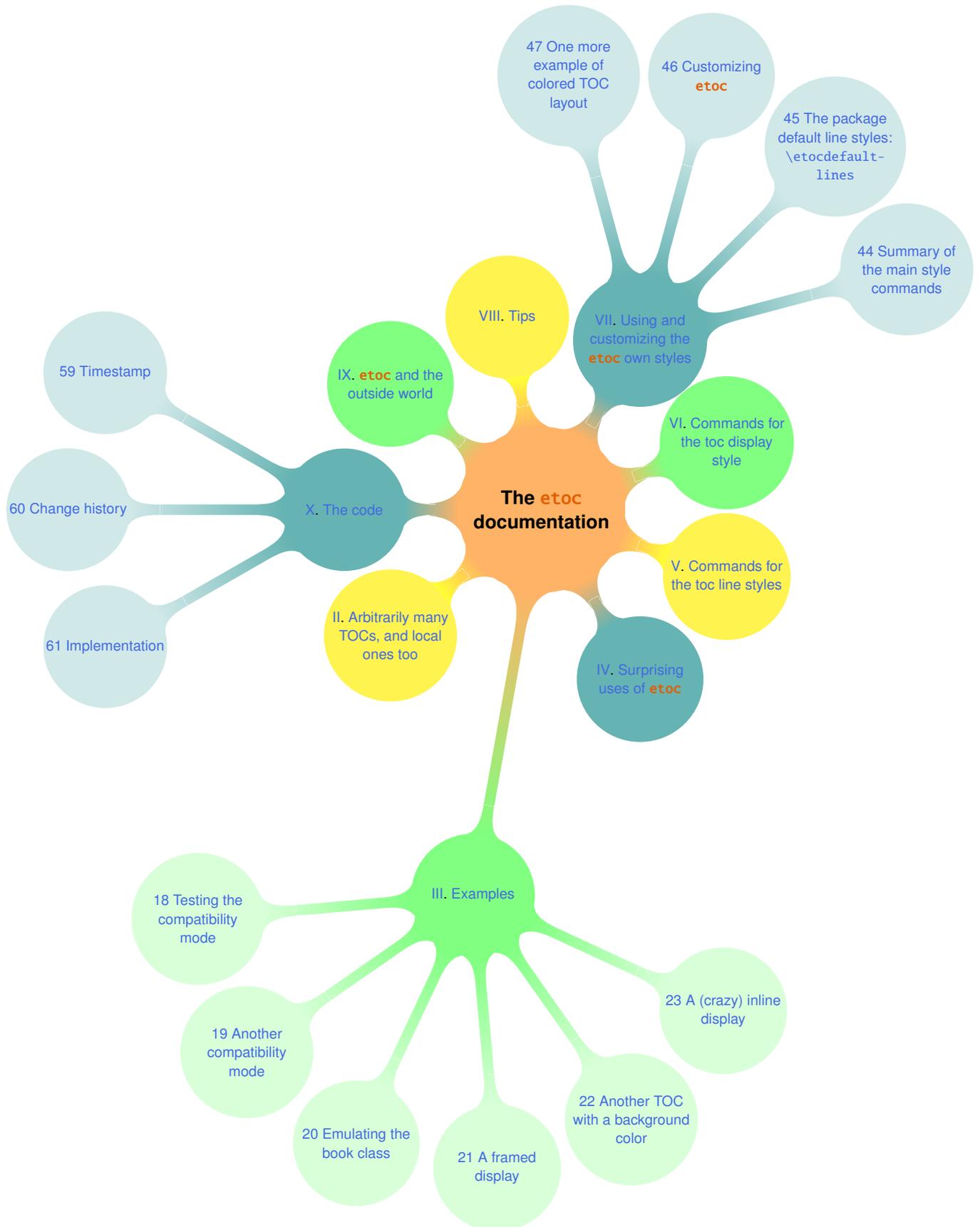
% 1,2,3,4,5,... -> 1,2,3,1,2,3,1,2,3
\def\pseudomodthree #1{\numexpr #1 + 3 - 3*((#1+1)/3)\relax}

\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}%

```

³⁹The “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



28. The TOC as a TikZ mind map

```

% 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

\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!]

```

29. The TOC as a (long) table

```
\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
```

[etocsnippet-16.tex](#) Depending on your PDF viewer, clicking on the margin filename may allow to extract it. 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 demo 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 (long) 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 `<start>`, `<prefix>`, `<contents>` or `<finish>` specification with a sole `\hline`, *i.e.* one not preceded by a `\` (it is however possible to put `\` at the end of `<prefix>` and the `\hline` at the start of `<contents>`).
3. as is explained next, it is recommended to put the `\` at the start of the `<prefix>` or `<contents>` 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 (yes this is only *one* table), as is possible relatively simply now with 1.08. The code follows.

TABLE OF CONTENTS

etoc

	Abstract	6
1	Foreword	6
2	License	7

I. Overview

3	Do I need to be a geek to use etoc ?	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 <code>\etocsetstyle</code> for the line styles	9
	4.2 <code>\etocsettocstyle</code> for the toc display	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. Arbitrarily many TOCs, and local ones too

9	The <code>\tableofcontents</code> , <code>\localtableofcontents</code> and <code>\localtableofcontentswithrelativedepths</code> commands	16
10	Labeling and reusing elsewhere	17
11	A powerful functionality of etoc : the re-assignment of levels with <code>\etocsetlevel</code>	18
12	The <code>\etocsettocdepth</code> and <code>\etocsetnexttocdepth</code> commands	19
	12.1 <i>The hyperref option</i> <code>bookmarksdepth</code>	20
13	The command <code>\etocsettocdepth.toc</code>	20
	13.1 <i>The commands</i> <code>\etocobeytocdepth</code> and <code>\etocignoretocdepth</code>	21

29. The TOC as a (long) table

14	The commands <code>\etocdepthtag.toc</code> and <code>\etocsettagdepth</code>	21
	14.1 <i>The commands <code>\etocobeydepthtags</code> and <code>\etocignoredepthtags</code></i>	22
15	The commands <code>\etocglobaldefs</code> and <code>\etoclocaldefs</code>	22
16	Not displayed empty TOCs	23
	16.1 <i>The <code>\etoccheckemptiness</code> command</i>	23
	16.2 <i>The <code>\etocnotocifnotoc</code> command</i>	23
	16.3 <i>The <code>\etocifwasempty</code> command</i>	24
17	Adding commands to the <code>.toc</code> file	24
	17.1 <i>The <code>hyperref</code> option <code>hidelinks</code></i>	25

III. Examples

18	Testing the compatibility mode	26
19	Another compatibility mode	27
20	Emulating the book class	29
21	A framed display	33
22	Another TOC with a background color	34
23	A (crazy) inline display	36

IV. Surprising uses of `etoc`

24	The TOC of TOCs	39
25	Arbitrary “Lists Of...”, <code>\etococcontentsline</code>	41
26	The TOC as a tree	42
27	The TOC as a molecule	45
28	The TOC as a TikZ mind map	47
29	The TOC as a (long) table	50
30	A TOC self-adjusting widths for its typesetting	57

V. Commands for the toc line styles

31	The <code>\etocsetstyle</code> , <code>\etocname</code> and <code>\etocpage</code> commands	59
32	The <code>\etocskipfirstprefix</code> and <code>\etociffirst</code> commands	61
33	The <code>\etocnumber</code> command	61
34	The <code>\etocifnumbered</code> switch	61
35	The <code>\etocthename</code> , <code>\etocthenumber</code> , and <code>\etocthepage</code> commands	62

36	The <code>\etoclink</code> command	62
37	The <code>\etocthelinkedname</code> , <code>\etocthelinkednumber</code> , <code>\etocthelinkedpage</code> and <code>\etocthelink</code> commands	63
38	DEPRECATED <code>\etocsavedsectionto-</code> <code>cline</code> , etc... commands.	64
39	The <code>\etocsetlevel</code> command	64
40	Scope of commands added to the <code>.toc</code> file	65
	40.1 <i>Testing the scope</i>	65
	40.2 <i>This is a (pale) red subsection for illustrative purposes</i>	66
41	Am I also red?	66

VI. Commands for the toc display style

42	Specifying the toc display style	67
	42.1 <i>The command <code>\etocsettocstyle</code></i>	67
	42.2 <i>The commands <code>\etocmulticolstyle</code>, <code>\etocmulticol</code>, and <code>\etoclocalmulticol</code></i>	68
	42.3 <i>The command <code>\etoclocalmulticol</code></i>	68
	42.4 <i>The commands <code>\etocruledstyle</code>, <code>\etocruled</code> and <code>\etoclocalruled</code></i>	69
	42.5 <i>The commands <code>\etocframedstyle</code>, <code>\etocframed</code>, and <code>\etoclocalframed</code></i>	69
	42.6 <i>Headings, titles, <code>\etocoldpar</code>, <code>\etocinnertopsep</code></i>	70
	42.7 <i>The compatibility mode <code>\etocstandarddisplaystyle</code></i>	70
	42.8 <i>The command <code>\etocinline</code></i>	70
43	Starred variants and hooks	71

VII. Using and customizing the `etoc` own styles

44	Summary of the main style commands	72
	44.1 <i>Setting up local styles</i>	72
	44.2 <i>Setting up toc display styles</i>	72
	44.3 <i>Displaying tables of contents</i>	73
	44.4 <i>Labels and references</i>	73
45	The package default line styles: <code>\etocdefaultlines</code>	73
46	Customizing <code>etoc</code>	77
	46.1 <i>Customizing the <code>etoc</code> pre-defined line styles</i>	77

29. The TOC as a (long) table

	46.2 <i>Customizing the toc display styles</i>	78
47	One more example of colored TOC layout	79

VIII. Tips

48	The <code>\etoclocaltop</code> command	82
49	The <code>\etocsetlocaltop.toc</code> command	83
50	Hacking framed parboxes	85
51	Interverting the levels	85
	51.1 <i>All subsections of this document</i>	86
52	Displaying statistics	86
53	Using depth tags	88
54	Typesetting the TOC as a table (the old way)	90

IX. `etoc` and the outside world

55	Constraints on the <code>.toc</code> file constitution	93
56	Compatibility with document classes	94
	56.1 <i>Compatibility with the KOMA-script classes</i>	94
	56.2 <i>Compatibility with the memoir class</i>	94
	56.3 <i>Compatibility with beamer</i>	95
57	Compatibility with other packages	95
	57.1 <i>Compatibility with babel</i>	95
	57.2 <i>Compatibility with hyperref</i>	95
	57.3 <i>Compatibility with multicol</i>	96
	57.4 <i>Compatibility with tocloft</i>	96
	57.5 <i>Compatibility with tocvsec2</i>	96
	57.6 <i>Compatibility with tableof</i>	96
	57.7 <i>Compatibility with tocstyle</i>	96
58	TeXnical matters	97

X. The code

59	Timestamp	98
60	Change history	98
61	Implementation	102

```

\begingroup
\etocglobaldefs % necessary for \etocname etc... to survive &
\makeatletter
% hack into longtable \hline to avoid annoying (here) stray lines at top
\def\LT@hline{%
  \ifx\@let@token\hline
    \global\let\@gtempa\@gobble
    \global\let\@gtempb\@firstofone %% ADDED
    \gdef\LT@sep{\penalty-\@medpenalty\vskip\doublerulesep}%
  \fi
}

```

```

\else
  \global\let\@gtempa\@empty
  \global\let\@gtempb\@gobble      %% ADDED
  \gdef\LT@sep{\penalty-\@lowpenalty\vskip-\arrayrulewidth}%
\fi
\ifnum0='{\fi}%
\multispan\LT@cols
  \unskip\leaders\hrule\@height\arrayrulewidth\hfill\cr
\@gtempb{%
  %% ADDED
\noalign{\LT@sep}%
\multispan\LT@cols
  \unskip\leaders\hrule\@height\arrayrulewidth\hfill\cr
\noalign{\penalty\@M}%
}%
  %% ADDED
\@gtempa}
\makeatother

% observe the locations of the \
\etocsetstyle{part}
  {}
  {}
  {\hline\multicolumn{3}{c}{\bfseries\vrule height6ex depth3ex width0pt
  \etocifnumbered{\etocnumber. }}\etocname}}
  {}

\etocsetstyle{section}
  {}
  {\etociffirst{\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

```

Depending on your PDF viewer, clicking on the margin filename may allow to extract it. [etocsnippet-17.tex](#)
Examining the code above the reader will wonder why the `\` are always given first in `<prefix+contents>` 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

29. The TOC as a (long) table

`\` 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 \TeX constructs alignments any definition or assignment done after an `\` 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 \TeX to start a cell and thus makes it impossible then to insert an `\hline`.

The safest way is thus to start with an `\` 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).

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 see [section 54](#).

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

Section	number	page
Do I need to be a geek to use <code>etoc</code> ?	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\code}{\\hline\code}}
  {\etocname & \etocnumber & \etocpage }
  {}

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

\etocglobaldefs % MANDATORY !!
\etocsetnexttocdepth{1}

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

```

Depending on your PDF viewer, clicking on the margin filename may allow to extract it. [etocsnippet-18.tex](#)

30. A TOC self-adjusting widths for its typesetting

This is a continuation of [section 20](#). 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}%
}%

```

30. A TOC self-adjusting widths for its typesetting

```
\newcommand*\TOCcomputelocalnumwidths [1][0.5em]{%  
  \TOCcompute@numwidths {local}{#1}%  
}%  
\makeatother
```

[etocsnippet-19.tex](#) Depending on your PDF viewer, clicking on the margin filename may allow to extract it. 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.
2. if you want to adjust the `tocdepth` recall from [subsection 12.1](#) 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 20](#) 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 59)
- 32** The `\etocskipfirstprefix` and `\etociffirst` commands (page 61)
- 33** The `\etocnumber` command (page 61)
- 34** The `\etocifnumbered` switch (page 61)
- 35** The `\etocthenname`, `\etocthenumber`, and `\etocthepage` commands (page 62)
- 36** The `\etoclink` command (page 62)
- 37** The `\etocthelinkedname`, `\etocthelinkednumber`, `\etocthelinkedpage` and `\etocthelink` commands (page 63)
- 38** DEPRECATED `\etocsavedsectiontocline`, etc... commands. (page 64)
- 39** The `\etocsetlevel` command (page 64)
- 40** Scope of commands added to the `.toc` file (page 65)
 - 40.1 Testing the scope (p. 65)
 - 40.2 This is a (pale) red subsection for illustrative purposes (p. 66)
- 41** Am I also red? (page 66)

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 `\localtableofcontents`.

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:⁴⁰

```
\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\begin{group}\normalfont
\footnotesize\itshape\etocskipfirstprefix}
{\allowbreak\-, \,}
{\etocname}
{.\hfil\par\end{group}\pagebreak[3]}
```

[etocsnippet-20.tex](#) Depending on your PDF viewer, clicking on the margin filename may allow to extract it.

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 meantime 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

⁴⁰the 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.

one will have a link only if `hyperref` was passed either option `linktoc=all` or option `linktoc=page`.)⁴¹ In accordance with the `hyperref` native behavior, no link gets incorporated into `\etocpage` if the page number is empty.

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.⁴²

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).⁴³

The `\etocifnumbered{<A>}{}` command executes `<A>` if the number exists, and `` 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
```

⁴¹As expected, in case of `linktoc=page`, only `\etocpage` is an hyperlink, not `\etocname` nor `\etocnumber`.

See `\etoclink` on how to create hyperlinks with the entry target.

⁴²With versions earlier than 1.08 it had to be the very last token in the `<start>` code. It may now appear anywhere therein.

⁴³as 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`.

36. The `\etoclink` command

```
    {{{\fboxrule1pt\fcolorbox{green}{white}{\etocnumber}}}}}
{\normalsize\bfseries\rmfamily\item}
{\etocname{} (page \etocpage)}
{\end{enumerate}}

\etocsetstyle{subsection}
{\begin{enumerate}[leftmargin=0cm, label=\etocnumber]}
{\normalfont \item}
{\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{<text>}` can be used in the line style specifications in a manner analogous to `\etocname`, `\etocnumber` and `\etocpage`. It creates a link (if `hyperref` is present⁴⁴) whose target is the corresponding document unit and whose name is the given `<text>` mandatory argument.

Hence `\etoclink{\etocthename}` is under default conditions of `hyperref` like the original `\etocname`, because the latter is already hyperlinked. Under `linktoc=page` context `\etoclink{\etocthename}` adds the hyperlink which is missing from `\etocname`. Similarly under the default `hyperref` condition (i.e. `linktoc=section`) `\etocpage` is not an hyperlink, but one can use `\etoclink{\etocthepage}`.

The command `\etoclink` is robust.

⁴⁴Prior to 1.1a, no such link was added if the `.toc` file entry was encountered with `hyperref`'s option `linktoc` set to none.

37. The `\etoc` commands, `\etoc` commands, `\etoc` commands and `\etoc` commands

The meanings of these commands can be stored for delayed usage. For example this is done in the [examples with trees](#).

There has been a **breaking change** at 1.1a. Here is the behavior *prior* to this release:

- `\etoc` commands and `\etoc` commands were hyperlinks only if `hyperref` was configured via `linktoc=all` or `linktoc=section` (the default),
- `\etoc` commands was an hyperlink only if `hyperref` was configured via `linktoc=all` or `linktoc=page` and the page number was not empty.

This behavior was coherent with the commands `\etoc`, `\etoc`, and `\etoc` being the robust variants of `\etoc`, `\etoc`, and `\etoc`.

At 1.1a it was decided that the commands should match their denominations.⁴⁵ So they are now *always* hyperlinks independently of `linktoc` `hyperref` option (`\etoc` has no hyperlink if the page number is empty, to match `hyperref` behavior):

- `\etoc` commands and `\etoc` commands and `\etoc` commands are always (in presence of `hyperref`) hyperlinks (for `\etoc` the page number must not be empty).

A further command is provided: `\etoc`, which wraps⁴⁶ an hyperlink around its argument: `\etoc{foo}` hyperlinks an arbitrary text `foo` to the target sectioning unit in the document. The command `\etoc` is its robust variant.

Obsolete example of a table of contents done as an inline `enumitem` environment:

42. Specifying the toc display style (p. 67), and **43.** Starred variants and hooks (p. 71).

Obsolete example of `\begin`

```

\etocsetstyle {section}
  {\begin{itemize*}[itemjoin={{; }}, itemjoin*={{, and }}]}
  {\global\let\TmpEtocNumber\etoclinkednumber
   \global\let\TmpEtocName\etoclinkedname
   \global\let\TmpEtocPage\etoclinkedpage }
% (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

```

⁴⁵To tell the whole truth, the author in refactoring the code completely at 1.1a was tricked by the names and forgot to read the old documentation so the new behavior was implemented and it was decided to keep the change.

⁴⁶Prior to 1.1a, there was a link added only if `hyperref` option `linktoc` was not none.

39. The `\etocsetlevel` command

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*={{, and }}}
    {}
    {\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.* 67), and **43**. Starred variants and hooks (*p.* 71).

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

Prior to 1.1a, **etoc** modified (locally) at each `\tableofcontents` or `\localtableofcontents` encountered the meaning of the L^AT_EX internal commands `\l@part`, `\l@chapter`, `\l@section`, etc... It stored (since 1.08k) their original meanings (at the time of the encountered TOC), into `\etocsavedchaptertocline`, `\etocsavedsectiontocline`, etc... commands, which allowed to do some simple changes via **etoc** of the rendering via the technique explained in [section 19](#).

This is all obsolete at 1.1a because **etoc** does not modify anymore (even locally) `\l@part`, `\l@chapter`, `\l@section`, etc... so they can be used directly in the context of the techniques of [section 19](#).

Thus `\etocsavedchaptertocline`, `\etocsavedsectiontocline`, etc... raise some warning that they will be removed at some later release and that you should stop using them.

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 commands `\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
```

Depending on your PDF viewer, clicking on the margin filename may allow to extract [etocsnippet-21.tex](#) it. 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^AT_EX 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
```

41. Am I also red?

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 and hooks” 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”.

42. Specifying the toc display style	67
42.1. The command <code>\etocsettocstyle</code>	67
42.2. The commands <code>\etocmulticolstyle</code> , <code>\etocmulticol</code> , and <code>\etoclocalmulticol</code>	68
42.3. The command <code>\etocstyle</code>	68
42.4. The commands <code>\etocruledstyle</code> , <code>\etocruled</code> and <code>\etoclocalruled</code> . .	69
42.5. The commands <code>\etocframedstyle</code> , <code>\etocframed</code> , and <code>\etoclocalframed</code>	69
42.6. Headings, titles, <code>\etocoldpar</code> , <code>\etocinnertopsep</code>	70
42.7. The compatibility mode <code>\etocstandarddisplaystyle</code>	70
42.8. The command <code>\etocinline</code>	70
43. Starred variants and hooks	71

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’.

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.

42. Specifying the toc display style

T_EX hacker note: We have seen in the previous chapter some examples of using `\begin{enumerate}` and `\end{enumerate}` within TOC line styles. There is a real difficulty with doing however something like this:

```
\etocsettocstyle{\begin{enumerate}}{\end{enumerate}}
\etocsetstyle{section}{{\item \sectionname}}{}}
```

Indeed for example on first compilation of a document, the `.toc` file is empty and there will be the typical “Something’s wrong—perhaps a missing `\item`”. Using `\etoc-checksempiness` does not help, because as is explained there, on first encountering a new TOC added at end of a document, `etoc` considers it non-empty, hence querying `\etoc-ifwaseempty` serves nothing. No, one must do something like this:

```
\makeatletter
\etocsettocstyle
  {\begin{enumerate}}
  {\if@newlist\item Please run \LaTeX\ again\fi
  \end{enumerate}}
\makeatother
\etocsetstyle{section}{{\item \sectionname}}{}}
```

The commands to be described next `\etocmulticolstyle`, `\etocruledstyle`, and `\etocframedstyle` all call `\etocsettocstyle` as a lower-level routine, to start a `multicols` 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 `multicols` 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 *LaTeX*, 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\fboxsep1ex
  \framebox[\linewidth]
    {\normalcolor\hss Contents of this part\hss}}
```

As a shortcut to set the style with `\etocruledstyle` and then issue a `\table-of-contents`, 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](#)).

42. Specifying the toc display style

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` or `tcolorbox` in the arguments of `\etocsettocstyle` is recommended.

Examples in this document are on pages 18, 36, 34, and 81.

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` before the `multicols` environment and inserts this `\etocoldpar`⁴⁷ 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}{<new_value>}`, 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

⁴⁷this 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.

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 and hooks

The `\tableofcontents`, `\localtableofcontents`, `\etocmulticol`, and all their cousins have starred variants (the star must be before the other arguments). The non-starred variants execute the `\etocaftertitlehook`, whose default definition is to do nothing. The starred variants do not execute this hook.

For example, imagine you are using book class and want `\localtableofcontents` to use a section-like title, but unnumbered. Assuming the main `\tableofcontents` comes first in the document, you can insert this after it:

```
\etocarticlestyle
\renewcommand{\etocaftertitlehook}{\addcontentsline{toc}{section}{\contentsname}}
```

This configures the way `\localtableofcontents` will behave (or `\tableofcontents`) from now on in the document.

The first line tells essentially to use `\section*{\contentsname}`, and the second line says to insert the title in the `.toc` file itself (thus to be displayed by the main table of contents). Notice that `hyperref` package will then automatically create suitable anchor and one should *not* use explicitly `\phantomsection` here (it would let the anchor be located below not above the title).

With this set-up issuing `\localtableofcontents*` will ignore the `\etocaftertitlehook` hence not send the local toc title to the `.toc` file. This mimicks the `memoir` class behavior, and can also be used with it. For more on `memoir` class with `etoc`, see [subsection 56.2](#).

There are further hook macros: `\etocaftercontentshook`, `\etocbeforetitlehook` and `\etocaftertochook` which are initially defined to do nothing and can be used for some special effects. They are executed whether or not the table of contents command was starred.⁴⁸

⁴⁸Hackers can test within these hooks if the context is starred or not via an `\ifx\Etoc@aftertitlehook\empty` conditional. Perhaps a future release will provide `\etociftocisstarred` wrapper.

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`

Summary of the main style commands	44, p. 72	The package default line styles: <code>\etocdefaultlines</code>	45, p. 73
Setting up local styles	44.1, p. 72	Customizing etoc	46, p. 77
Setting up toc display styles	44.2, p. 72	Customizing the etoc pre-defined line styles	46.1, p. 77
Displaying tables of contents	44.3, p. 73	Customizing the toc display styles	46.2, p. 78
Labels and references	44.4, p. 73	One more example of colored TOC layout	47, p. 79

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>}
\etocstylewithmarksnoc[<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 . . .

```

`\localtableofcontents \ref{toc:far}` acts the same as `\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- \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

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

than what is achieved by the \LaTeX default.⁴⁹ 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 \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 \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*{\etoclineleaders}
  {\hbox{\normalfont\normalsize\hbox to 1ex {\hss.\hss}}}
\slippy
\tableofcontents
\endgroup
```

[etocsnippet-22.tex](#) Depending on your PDF viewer, clicking on the margin filename may allow to extract it.

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

etoc

Part I. Overview

Abstract

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

1. Foreword

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

2. License

⁴⁹and 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).

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. The `\tableofcontents`, `\localtableofcontents` and `\localtableofcontentswithrelativedepths` commands

10. Labeling and reusing elsewhere

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

12. The `\etocsettocdepth` and `\etocsetnexttocdepth` commands

The hyperref option `bookmarksdepth` 12.1, p. 20

13. The command `\etocsettocdepth.toc`

The commands `\etocobeytoctocdepth` and `\etocignoretoctocdepth` 13.1, p. 21

14. The commands `\etocdepthtag.toc` and `\etocsettagdepth`

The commands `\etocobeydepthtags` and `\etocignoredepthtags` 14.1, p. 22

15. The commands `\etocglobaldefs` and `\etoclocaldefs`

16. Not displayed empty TOCs

The `\etoccheckemptiness` command 16.1, p. 23

The `\etocnotocifnotoc` command 16.2, p. 23

The `\etocifwasempty` command 16.3, p. 24

17. Adding commands to the `.toc` file

The hyperref option `hidelinks` 17.1, p. 25

Part III. Examples

18. Testing the compatibility mode

19. Another compatibility mode

20. Emulating the book class

21. A framed display

22. Another TOC with a background

23. A (crazy) inline display

Part IV. Surprising uses of `etoc`

24. The TOC of TOCs

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

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

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 (long) 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. DEPRECATED `\etocsavedsectiontocline`, etc... commands.

39. The `\etocsetlevel` command

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

Testing the scope 40.1, p. 65

This is a (pale) red subsection for illustrative purposes 40.2, p. 66

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. 67

The commands `\etocmulticolstyle`, `\etocmulticol`, and `\etoclocalmulticol` 42.2, p. 68

The command `\etoclocaltocstyle` 42.3, p. 68

The commands `\etocruledstyle`, `\etocruled` and `\etoclocalruled` 42.4, p. 69

The commands `\etocframedstyle`, `\etocframed`, and `\etoclocalframed` 42.5, p. 69

Headings, titles, `\etocoldpar`, `\etocinerttopsep` 42.6, p. 70

The compatibility mode `\etocstandarddisplaystyle` 42.7, p. 70

The command `\etocinline` 42.8, p. 70

43. Starred variants and hooks

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

44. Summary of the main style commands

Setting up local styles 44.1, p. 72

Setting up toc display styles 44.2, p. 72

Displaying tables of contents 44.3, p. 73

Labels and references 44.4, p. 73

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

46. Customizing **etoc**

Customizing the **etoc** pre-defined line styles 46.1, p. 77

Customizing the toc display styles 46.2, p. 78

47. One more example of colored TOC layout

Part VIII. Tips

48. The `\etoclocaltop` command

49. The `\etocsetlocaltop.toc` command

50. Hacking framed parboxes

51. Interverting the levels

All subsections of this document 51.1, p. 86

52. Displaying statistics

53. Using depth tags

54. Typesetting the TOC as a table (the old way)

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

55. Constraints on the `.toc` file constitution

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}
```

56. Compatibility with document classes

Compatibility with the KOMA-script classes 56.1, p. 94

Compatibility with the memoir class ... 56.2, p. 94

Compatibility with beamer .. 56.3, p. 95

57. Compatibility with other packages

Compatibility with babel 57.1, p. 95

Compatibility with hyperref .. 57.2, p. 95

Compatibility with multicol .. 57.3, p. 96

Compatibility with tocloft ... 57.4, p. 96

Compatibility with tocvsec2 . 57.5, p. 96

Compatibility with tableof ... 57.6, p. 96

Compatibility with tocstyle .. 57.7, p. 96

58. T_EXnical matters

Part X. The code

59. Timestamp

60. Change history

61. Implementation

46. Customizing *etoc*

```
\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*\etocseptthree{.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*\etocclineleaders
  {\hbox{\normalfont\normalsize\hbox to 2ex {\hss.\hss}}}
\newcommand*\etocabbrevpagenam{p.~} % initial of "page"
\newcommand*\etocpartname{Part} % prior to 1.08b, 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*.⁵⁰

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^AT_EX 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.

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}
```

⁵⁰with 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.

47. One more example of colored TOC layout

```
\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*\etoctoprule{\hrule}
\newcommand*\etocbottomrulecolorcmd{\relax}

% for the framed style only:
\newcommand*\etocinnerleftsep{2em}
\newcommand*\etocinnerrightsep{2em}
\newcommand*\etocinnerbottomsep{3.5ex}

\newcommand*\etoclefttrule{\vrule}
\newcommand*\etocrighttrule{\vrule}
\newcommand*\etocbottomrule{\hrule}
\newcommand*\etoclefttrulecolorcmd{\relax}
\newcommand*\etocrighttrulecolorcmd{\relax}
\newcommand*\etocbottomrulecolorcmd{\relax}

\newcommand*\etocbkgcolorcmd{\relax}

% hooks
\newcommand\etocframedmhook{\relax}
```

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

The `\dotscolorcmd` 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{\etocbottomrule}{\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 `\etocruled` or `\etocframed` 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 \TeX 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

47. One more example of colored TOC layout

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{\etoctoprulecolorcmd}{\color{tocbackground}}

\renewcommand{\etocleftrule}{\vrule width 1cm}
\renewcommand{\etocrightrule}{\vrule width .5cm}
\renewcommand{\etocbottomrule}{\hrule height 12pt}
\renewcommand{\etoctoprule}{\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]{%
  \leftskip0pt\rightskip0pt\parskip0pt
  \fboxsep0pt % \colorbox uses \fboxsep 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\hfill\makebox[.5cm][r]{\etocpage\hspace{.2cm}}}%
  \vskip6pt }
{}}

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

```

Depending on your PDF viewer, clicking on the margin filename may allow to extract it. [etocsnipet-23.tex](#)

I am a twin of that other TOC (click me!)		
42	Specifying the toc display style	67
42.1	The command <code>\etocsettocstyle</code>	67
42.2	The commands <code>\etocmulticolstyle</code> , <code>\etocmulticol</code> , and <code>\etoclocalmulticol</code>	68
42.3	The command <code>\etoclocalmulticol</code>	68
42.4	The commands <code>\etocruledstyle</code> , <code>\etocruled</code> and <code>\etoclocalruled</code>	69
42.5	The commands <code>\etocframedstyle</code> , <code>\etocframed</code> , and <code>\etoclocalframed</code>	69
42.6	Headings, titles, <code>\etocoldpar</code> , <code>\etocinnertopsep</code>	70
42.7	The compatibility mode <code>\etocstandarddisplaystyle</code> ...	70
42.8	The command <code>\etocinline</code>	70
43	Starred variants and hooks	71

The TOC has been put in a `float` which appears on the current page. The coding is a bit involved⁵¹ 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 across 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.

⁵¹and reveals the author's preference for the TeX syntax...

Part VIII.

Tips

Here are some statistics for this part: it contains 7 sections and 1 subsection. The name of the first section is “The `\etoclocaltop` 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 “54”. The name of the first subsection is “All subsections of this document” and the corresponding number is “51.1”. The name of the last subsection is “All subsections of this document” and its number is “51.1”.

48. The <code>\etoclocaltop</code> command	82
49. The <code>\etocsetlocaltop.toc</code> command	83
50. Hacking framed parboxes	85
51. Interverting the levels	85
51.1. All subsections of this document	86
52. Displaying statistics	86
53. Using depth tags	88
54. Typesetting the TOC as a table (the old way)	90

48. The `\etoclocaltop` command

Within either the TOC style (`\etocsettocstyle`) or the local title styles (`\etocsetstyle`), the control sequence `\etoclocaltop` is made equivalent for the duration of `\localtableofcontents` to a numeric (self-delimiting) denotation of the current top level.

Thus: it will in numeric contexts (`\ifnum`, `\ifcase`, ...) represent zero for a local TOC corresponding to chapter, or one if in a section, or two if in a subsection, etc..., assuming of course here that the default levels are obeyed (see [section 11](#)).

`\etoclocaltop` from inside the TOC heading (first argument of `\etocsettocstyle`) has the correct value *only under `\etoccheckemptiness` regime*. Special circumstances correspond to some special values:

`-3` (`-\thr@@`)

signals that `etoc` considers the local TOC to be “unknown”; this happens at the last local TOC, for the first \LaTeX run after adding a new `\localtableofcontents` to the document. In doubt, `etoc` assumes the TOC will prove non empty, hence it prints (independently of whether the check for emptiness was activated or not) the heading as specified by `\etocsettocstyle`. Thus, check if `\etoclocaltop` gives `-3` as a `<number>` to detect that situation from within the first argument of `\etocsettocstyle`, if desired.

-1000 (`-\@m`)

is in case of a `\localtableofcontents` being considered “known” (although it may still refer to the data in the `.toc` file from the previous run) but without the check for emptiness having been executed.

-10000 (`-\@M`)

is the value when accessed from the title of a global TOCs.

When executed from within a local table of contents **line styles** (`\etocsetstyle`), `\etoclocaltop` always will hold the correct value, whether or not the emptiness check was executed.

For a global table of contents however, it will always keep the value `-3`.

Attention! `\etoclocaltop` is only to be queried; modifying it is strictly forbidden and falls under the most rigorous prosecution allowed by federal law.

49. The `\etocsetlocaltop.toc` and `\etocimmediatesetlocaltop.toc` commands

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 `\localtableofcontents` 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 it will stop only at encountering 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 the *subsection*.

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
```

So with no `\part` heading inserted into the table of contents via an `\addcontentsline`, still `\localtableofcontents` will know it is local to a part. In this example the local contents will be delimited by the next numbered `\part`, or `\part*` with `\addcontentsline`, or also by a later, second, `\etocsetlocaltop.toc{part}`.

As a (counter)-example consider this document:

```
\documentclass{article}
```

49. The `\etocsetlocaltop.toc` command

```
\usepackage{etoc}
\begin{document}
\tableofcontents

\part*{A}
\etocsetlocaltop.toc{part}
\localtableofcontents

\section{I}

\section{II}

\part*{B}

\section{III}

\part*{C}

\section{IV}
\end{document}
```

It uses only `\part*`. Thanks to the `\etocsetlocaltop.toc` the `\localtableofcontents` knows it should report only sections. But the other `\part*` are invisible to it as nothing is recorded in the `.toc` file. So the local table of contents in this example will list *all* sections not only I and II. To fix this one may e.g. insert another `\etocsetlocaltop.toc{part}`, this time after `\part*{B}` (or make this a numbered part, or use `\addcontentsline` for it).

The above document amended with added `\etocsetlocaltop.toc{part}` after each unnumbered part will thus have its main TOC without any Part heading, but each `\part` can show a correct `\localtableofcontents`. The simpler approach would be to use `\addcontentsline` with each unnumbered `\part` so that it ends up in the `.toc` file, but **etoc** is keen on allowing the most diverse point of views.

It should be stressed that the various `\etocsetlocaltop.toc{<sect. unit>}` do impact the global `\tableofcontents`: they really act like actual sectioning units, except for not inducing any typesetting. In usual document classes, this would appear to mean that they are completely transparent to the global `\tableofcontents`. Not the case with **etoc**, which adds a virtual assembly of levels: the `.toc` data originating in `\etocsetlocaltop.toc{<sect. unit>}` will trigger the execution of the `{<finish>}` parts of the line styles of finer sectioning units encountered before (either in the global `\tableofcontents` or in an active `\localtableofcontents`); and it triggers the `{<start>}` parts of the line styles of finer units encountered after it (again in the global `\tableofcontents`, but also in any `\localtableofcontents` which is already activated at a coarser lever).

Depending on how the toc line styles are configured this may translate into some visual effect; for example with the **etoc** own line styles the `{<start>}` and `{<finish>}` mostly insert penalties or vertical spaces.

It is a matter of debate if this is good design; a variant serving purely to influence boundaries of local table of contents with no collateral effects could be provided. And the name of the macro was perhaps not so well chosen as it suggests it acts as would such an hypothetical variant. In absence of feature requests we leave the matter standing for now.⁵²

⁵²This documentation was updated and this paragraph added for the 1.09f release.

Usage of `\etocsetlocaltop.toc` interacts with `\etoccheckemptiness` in the expected way: it modifies (as explained above) the selection made by `\localtableofcontents`, hence the decision whether this local TOC will end up empty or not.

There is also `\etocimmediatesetlocaltop.toc`. This may be useful in some very special circumstances involving `\include`. For related discussion see the documentation of `\etocimmediatedepthtag.toc`.

50. Hacking framed parboxes

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

Specifying the toc display style	The commands <code>\etocframedstyle</code> , <code>\etocframed</code> , and <code>\etoclocalframed</code>
. 42, p. 67 42.5, p. 69
The command <code>\etocsettocstyle</code>	Headings, titles, <code>\etocoldpar</code> , <code>\etocinnertopsep</code>
. 42.1, p. 67	42.6, p. 70
The commands <code>\etocmulticolstyle</code> , <code>\etocmulticol</code> , and <code>\etoclocalmulticol</code>	The compatibility mode <code>\etocstandarddisplaystyle</code>
. 42.2, p. 68	42.7, p. 70
The command <code>\etoclocstyle</code>	The command <code>\etocinline</code>
. 42.3, p. 68 42.8, p. 70
The commands <code>\etocruledstyle</code> , <code>\etocruled</code> and <code>\etoclocalruled</code>	Starred variants and hooks
. 42.4, p. 69 43, p. 71

51. 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}}
```

52. Displaying statistics

```
\renewcommand{\etoccolumnsep}{2.75em}  
\renewcommand{\columnseprule}{1pt}  
\etocmulticol[3]{\subsection{All subsections of this document}}  
\endgroup
```

51.1. All subsections of this document

- | | | |
|---|--|--|
| <p>i. Limitations in the use of list environments for tables of contents (3.1, p. 8)</p> <p>ii. <code>\etocsetstyle</code> for the line styles (4.1, p. 9)</p> <p>iii. <code>\etocsettocstyle</code> for the toc display (4.2, p. 9)</p> <p>iv. Compatibility mode (4.3, p. 10)</p> <p>v. The <code>hyperref</code> option <code>bookmarksdepth</code> (12.1, p. 20)</p> <p>vi. The commands <code>\etocobeytoctocdepth</code> and <code>\etocignoretocdepth</code> (13.1, p. 21)</p> <p>vii. The commands <code>\etocobeydepthtags</code> and <code>\etocignoredepthtags</code> (14.1, p. 22)</p> <p>viii. The <code>\etocchecksemptiness</code> command (16.1, p. 23)</p> <p>ix. The <code>\etocnotocifnotoc</code> command (16.2, p. 23)</p> <p>x. The <code>\etocifwasempty</code> command (16.3, p. 24)</p> <p>xi. The <code>hyperref</code> option <code>hidelinks</code> (17.1, p. 25)</p> <p>xii. Testing the scope (40.1, p. 65)</p> | <p>xiii. This is a (pale) red subsection for illustrative purposes (40.2, p. 66)</p> <p>xiv. The command <code>\etocsettocstyle</code> (42.1, p. 67)</p> <p>xv. The commands <code>\etocmulticolstyle</code>, <code>\etocmulticol</code>, and <code>\etoclocalmulticol</code> (42.2, p. 68)</p> <p>xvi. The command <code>\etocsettocstyle</code> (42.3, p. 68)</p> <p>xvii. The commands <code>\etocruledstyle</code>, <code>\etocruled</code> and <code>\etoclocalruled</code> (42.4, p. 69)</p> <p>xviii. The commands <code>\etocframedstyle</code>, <code>\etocframed</code>, and <code>\etoclocalframed</code> (42.5, p. 69)</p> <p>xix. Headings, titles, <code>\etocoldpar</code>, <code>\etocinnertopsep</code> (42.6, p. 70)</p> <p>xx. The compatibility mode <code>\etocstandarddisplaystyle</code> (42.7, p. 70)</p> <p>xxi. The command <code>\etocinline</code> (42.8, p. 70)</p> <p>xxii. Setting up local styles (44.1, p. 72)</p> <p>xxiii. Setting up toc display styles (44.2, p. 72)</p> <p>xxiv. Displaying tables of contents (44.3, p. 73)</p> | <p>xxv. Labels and references (44.4, p. 73)</p> <p>xxvi. Customizing the <code>etoc</code> pre-defined line styles (46.1, p. 77)</p> <p>xxvii. Customizing the toc display styles (46.2, p. 78)</p> <p>xxviii. All subsections of this document (51.1, p. 86)</p> <p>xxix. Compatibility with the KOMA-script classes (56.1, p. 94)</p> <p>xxx. Compatibility with the memoir class (56.2, p. 94)</p> <p>xxxi. Compatibility with beamer (56.3, p. 95)</p> <p>xxxii. Compatibility with babel (57.1, p. 95)</p> <p>xxxiii. Compatibility with <code>hyperref</code> (57.2, p. 95)</p> <p>xxxiv. Compatibility with <code>multicol</code> (57.3, p. 96)</p> <p>xxxv. Compatibility with <code>tocloft</code> (57.4, p. 96)</p> <p>xxxvi. Compatibility with <code>tocvsec2</code> (57.5, p. 96)</p> <p>xxxvii. Compatibility with <code>tableof</code> (57.6, p. 96)</p> <p>xxxviii. Compatibility with <code>tocstyle</code> (57.7, p. 96)</p> |
|---|--|--|

52. 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}%
  \def\thispartstatsauxi{%
    \sbox{\firstnamei}{\footnotesize\etocname}%
    \sbox{\firstnumberi}{\footnotesize\etocnumber}%
    \def\thispartstatsauxi{}}%
  \def\thispartstatsauxii{%
    \sbox{\firstnameii}{\footnotesize\etocname}%
    \sbox{\firstnumberii}{\footnotesize\etocnumber}%
    \def\thispartstatsauxii{}}%
  \begingroup
  \etocsetstyle{subsection} {} {}
  {\thispartstatsauxii
   \stepcounter{mycountii}%
   \sbox{\lastnameii}{\footnotesize\etocname}%
   \sbox{\lastnumberii}{\footnotesize\etocnumber}} {}%
  \etocsetstyle{section} {} {}
  {\thispartstatsauxi
   \stepcounter{mycounti}%
   \sbox{\lastnamei}{\footnotesize\etocname}%
   \sbox{\lastnumberi}{\footnotesize\etocnumber}}
  {{\footnotesize\itshape
  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.\par}}%
  \etocinline % cancels the automatic \par automatically before the TOC
  \etocsettocstyle {}{}
  \etocsetnexttocdepth{2}%
  \localtableofcontents % to be used at the top level of a Part.
  \endgroup
}

```

Depending on your PDF viewer, clicking on the margin filename may allow to extract it. [etocsnippet-24.tex](#)
 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 {}

```

53. Using depth tags

```
\newcommand*\firstsubnumber {} \newcommand*\lastsubnumber {}
\newcommand*\thisspecialstatsaux{}
\newcommand*{\thisspecialstats}{%
  \setcounter{mycounti}{0}%
  \def\thisspecialstatsaux{%
    \let\firstsubname\etocthelinkedname
    \let\firstsubnumber\etocthelinkednumber
    \def\thisspecialstatsaux{}}
\begingroup
\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
```

[etocsnippet-25.tex](#) Depending on your PDF viewer, clicking on the margin filename may allow to extract it.

53. 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^AT_EX₂_ε'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

I Overview	8
II Arbitrarily many TOCs, and local ones too	16
III Examples	26
IV Surprising uses of <code>etoc</code>	39

V	Commands for the toc line styles	59
VI	Commands for the toc display style	67
42	Specifying the toc display style	67
42.1	The command <code>\etocsettocstyle</code>	67
42.2	The commands <code>\etocmulticolstyle</code> , <code>\etocmulticol</code> , and <code>\etoclocalmulticol</code>	68
42.3	The command <code>\etoclocalmulticol</code>	68
42.3.1	The command <code>\etoclocalmulticol</code>	69
42.3.1.1	Do we really want paragraph entries in the TOC?	69
42.3.1.2	really?	69
42.4	The commands <code>\etocruledstyle</code> , <code>\etocruled</code> and <code>\etoclocalruled</code>	69
42.5	The commands <code>\etocframedstyle</code> , <code>\etocframed</code> , and <code>\etoclocalframed</code>	69
42.6	Headings, titles, <code>\etocoldpar</code> , <code>\etocinnertopsep</code>	70
42.7	The compatibility mode <code>\etocstandarddisplaystyle</code>	70
42.7.1	The commands <code>\etocarticlestyle</code> , <code>\etocbookstyle</code> ,	70
42.8	The command <code>\etocinline</code>	70
43	Starred variants and hooks	71
VII	Using and customizing the <code>etoc</code> own styles	72
VIII	Tips	82
IX	<code>etoc</code> and the outside world	93

```

\etocsetnexttocdepth {all}
\begingroup
\parindent 0pt \leftskip 0cm \rightskip .75cm \parfillskip -\rightskip
\newcommand*{\EndParWithPagenoInMargin}
  {\nobreak\hfill
   \makebox[0.75cm][r]{\mdseries\normalsize\etocpage}%
   \par}
\renewcommand*{\etoclineleaders}
  {\hbox{\normalfont\normalsize\hbox to .75ex {\hss.\hss}}}
\newcommand*{\EndParWithPagenoInMarginAndLeaders}
  {\nobreak\leaders\etoclineleaders\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}

```

54. Typesetting the TOC as a table (the old way)

```
\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\etoctoprule {\hrule height 3pt\relax }
\renewcommand\etoctoprulecolorcmd {\color{blue}}
\renewcommand\etocaftercontentshook
{\medskip\begingroup \color{blue}\hrule height 3pt \endgroup }
\etocruledstyle [1]{\Large\bfseries
\fbbox{\makebox[8cm]{A TOC using depth tags}}}
\sloppy
\tableofcontents
\endgroup
```

[etocsnippet-26.tex](#) Depending on your PDF viewer, clicking on the margin filename may allow to extract it.

54. 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.

54. Typesetting the TOC as a table (the old way)

For the record, here is how the TOC from [section 29](#) was coded in the old days.⁵³ 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 \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](#).

```

\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\\}%
}

\begingroup
\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

```

⁵³At release 1.09f the design of the TOC from [section 29](#) was modified; the code here reproduces the former looks.

54. Typesetting the TOC as a table (the old way)

```

\multicolumn{3}{|c|}{\Large\bfseries\strut TABLE OF CONTENTS}%
\\ \hline \hline}
{\global\toctabletok\expandafter{\the\toctabletok\hline\end{longtable}}}
\etocsettocdepth {subsection}
\tableofcontents
\the\toctabletok
\endgroup
```

[etocsnippet-27.tex](#) Depending on your PDF viewer, clicking on the margin filename may allow to extract it.

Part IX.

etoc and the outside world

55. Constraints on the .toc file constitution

The contents of the .toc file (if it already exists) are read into memory by **etoc** once, at the time of `\begin{document}`.⁵⁴

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 \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, redefine `\etocaftertochook`

```
\renewcommand\etocaftertochook{foo}
```

and there is also `\etocaftercontentshook` which is executed a bit earlier⁵⁵ 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 \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.

⁵⁴Versions earlier than 1.07m 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.

⁵⁵contrarily to `\etocaftertochook`, `\etocaftercontentshook` is not executed if the `tocdepth` did not allow the printing of the TOC.

56. Compatibility with document classes

etoc has mainly been tested with the `article` and `book` standard classes. Some compatibility layer with the `KOMA-script` and `memoir` classes was added at 1.05 of 2012/12/01. Such compatibility layer regards what this documentation calls the global display style: when **etoc** gets activated because the user employed some `\etocsetstyle{section}` (etc), i.e. it is not used purely in compatibility mode (which happens I guess for `\localtableofcontents` functionality as it does not require reading the user manual), **etoc** tries in absence of usage by the user of `\etocsettocstyle` to create itself a “TOC display style” emulating the default behaviour of the document class. There does not seem to be an easy way to extract this in an automated manner dynamically, so it is basically some manual work which the author initiated in 2012 and which got sporadically updated since.

56.1. Compatibility with the `KOMA-script` classes

Not really tested... well, tested by this document with its dozens of **etoc** TOCs and which uses `scrartcl`! The package code contains

```
\@ifclassloaded{scrartcl}
  {\renewcommand*\etocstandarddisplaystyle{\etocscrartclstyle}}}
```

with `\etocscrartclstyle` trying to emulate the global display style of the `\tableofcontents` within the class `scrartcl`. Thus **etoc** is ready for basic usage in compatibility mode (last updated at 1.09f).

56.2. 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.

The `memoir` mechanism relative to `\tableofcontents` versus `\tableofcontents *` is obeyed automatically, and applies with `\localtableofcontents` too.

But, attention, this means that `\localtableofcontents` should preferentially be used with the `*`. Else the `memoir` mechanism will insert a specific line in the `.toc` file, using by default the chapter sectioning level. But **etoc** has already written to the `.toc` file an instruction that the user wants a local table of contents. As a result, the fake chapter will let **etoc** consider that the `\localtableofcontents` has empty contents, except if it used in a `\part` or `\book`.

This may even cause successive document builds to never stabilize if `\etoccheckemptiness` was issued in the preamble: as the local TOC is empty, on next run the extra entry in `.toc` is absent, so the local TOC is not empty and is typeset, but then `.toc` changes, so the `re-runfilecheck` mechanism instructs to compile once more, and we end in a never stabilizing conundrum.

Thus we see that the fact that `\localtableofcontents` by default use like the global `\tableofcontents` a chapter style is inconvenient. One can use the `memoir` macros to modify this: according to `memoir` documentation, `\toheadstart` and `\printtoctitle`

seem to be the most relevant. If you do that you will probably want the added line in `.toc` file to use `{section}` not `{chapter}`. `etoc` provides to this effect `\etocmemoirtotocfmt{<kind>}{<name>}` whose first argument `<kind>` is `chapter`, `section`, `subsection`, etc... and the second argument stands for the title and thus `\contentsname` is appropriate. Hence

```
\etocmemoirtotocfmt{section}{\contentsname}
```

seems the natural choice in this context as an addition to the customizing the `memoir` toc related macros to achieve a section like rendering (for more details on the latter, please refer to `memoir` manual). This is appropriate for local TOCs in chapters. Adapt to finer sectioning levels.

Another approach is to overrule the `etoc` default under `memoir` like this

```
\etocarticlestyle % means to use \section*{\contentsname} (and marks)
\renewcommand{\etocaftertitlehook}{\addcontentsline{toc}{section}{\contentsname}}
```

exactly as one would do with `book` class for example. This emulates the default under `memoir` that table of contents write a line to the `.toc` file representing themselves, except that `chapter` has been replaced by `section`. So we can use `\localtableofcontents` after a `\chapter`. But if you want one after a `\section`, you should modify `\etocaftertitlehook` to assign subsection level to the `\localtableofcontents` own styling.

56.3. Compatibility with *beamer*

For the reasons mentioned already regarding the constraints on the `.toc` file constitution, `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.

57. Compatibility with other packages

57.1. 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.

57.2. 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.

57. Compatibility with other packages

57.3. Compatibility with **multicol**

etoc loads the package **multicol**.

57.4. Compatibility with **tocloft**

Release 1.07k added compatibility with package **tocloft**: 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 modify midway in the document the macros `\l@section`, `\l@subsection` ... but the effect will be seen only in table of contents typeset by **etoc** in compatibility mode (and of course after those customizations). It will have no effect on true **etoc** TOCs.

57.5. Compatibility with **tocvsec2**

etoc used to be incompatible with package **tocvsec2**; it now cohabits, sort of, as it deactivates **tocvsec2**'s modification of `\tableofcontents` 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.

57.6. Compatibility with **tableof**

It is possible to use simultaneously **etoc** and **tableof**. 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.

57.7. Compatibility with **tocstyle**

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

58. T_EXnical matters

The `\etocname`, `\etocnumber`, `\etocpage` commands are protected against premature expansion. They are hyperlinks if package `hyperref` is loaded and depending on its option `linktoc` value; under the default `linktoc=section`, only name and number are hyperlinked, not the page number.

On the other hand `\etocthename`, `\etocthenumber`, `\etocthepage` are *not* protected against expansion. And neither are `\etocthelinkedname`, `\etocthelinkednumber`, `\etocthelinkedpage`. They were modified at 1.1a and now are always hyperlinks (except for the latter if the page number is empty), if `hyperref` is present, independently of `linktoc` status.

The commands `\etoclink` and `\etocifnumbered` are also protected against premature expansion. Also `\etociffirst` and `\etococcontentsline`.

Commands such as `\etocsetstyle`, `\etocsetlevel`, `\etocsettocstyle`, `\etocmulticolstyle`, `\etocruledstyle`, `\etocframedstyle` obey L^AT_EX's groups. All TOCs are typeset inside groups.

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 on first subsequent compilation. The correct local table of contents will be displayed only on the second compilation.

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 to add extra commands to the `.toc` file, make it in such a way that they can be activated or deactivated easily from the document source, e.g. via some booleans.

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

Timestamp	59, p. 98
Change history	60, p. 98
Implementation	61, p. 102

59. Timestamp

This is the documentation as of 2023/01/15, printed from the source file with the time stamp `Time-stamp: <15-01-2023 at 11:56:19 CET>`. The package version is 1.1b, of 2023/01/15.

60. Change history

1.1b [2023/01/15]

Documentation fix, 1.1a forgot to mention the following change: `\etocthelinkedname`, `\etocthelinkednumber`, `\etocthelinkedpage` are now always hyperlinks independently of `linktoc` status.

1.1a [2023/01/14]

This version brings no new functionality, despite the number bump. It implements a complete rewrite of old legacy core internals. Formerly, `etoc` waited for `hyperref` (if present) to have added hyperlinks via its patch to LaTeX's `\contentsline`. `etoc` examined the arguments of `\@section` and other commands to extract hyperlinking information, if any. With this release `etoc` decides earlier according to `hyperref` `linktoc` status whether section names and page numbers should be hyperlinked, and add links itself via `\hyperlink`. `etoc` is thus now immune to the details of how `hyperref` patches the `\contentsline` command, which is not executed anymore. Overall, the code is greatly simplified.

`\etoclink` now wraps its argument in an `\hyperlink` even if `hyperref` is configured via `linktoc=none`. Formerly no `\hyperlink` was added then.

Deprecation of `\etocsavedsectiontocline` and similarly named commands. They are not needed as `\@section` et al. are with this release left unmodified during the table of contents typesetting.

L^AT_EX kernel from 2020/10/01 or later is required.

1.09i [2022/11/21]

Fix bug showing when a document uses both `\etoccheckemptiness` and `\etocsetlocal-top.toc`: the start and finish parts of some levels were executed possibly causing extra printed output.

More hyperlinking in the implementation part of the documentation.

1.09h [2022/11/20]

Documentation improvements. In particular, attached code snippets are now visible via their filenames in the page margins. Also, command names are doubly hyperlinked: first half links to the devoted part of the user manual, second half links to the implementation part.

1.09g [2022/11/17]

Compatibility hotfix with recent `hyperref` 7.00u of 2022-11-13. Thanks to Denis Bitouzé for signaling the breakage to the author.

1.09f [2022/08/30]

No more shipping of a German translation of the documentation, as it was last updated in April 2015.

(`etoc.pdf`) User level commands `\hyperlink` from their code source definitions to their descriptions in the documentation part. Macros used

- in the code source hyperlink to where they first got defined there.
- Wrap the `\etocpartname` (from `etoc`'s package provided `toc` line style) together with the part number in a potential common hyperlink.
- Try to sync the emulation of the global display style with KOMA-script v3.37 (in particular regarding the `noparskipfake` KOMA `toc` feature).
- Improve documentation of some aspects under memoir class.
- Remove the `\nonumberline` token, even though empty, from the meaning of `\etocthename` (KOMA-script classes).
- Add `\etocimmediatedepthtag.toc` to work around problems related to `\include` (see user doc). Thanks to Norman Ramsey who reported the problem and proposed a work-around in July 2016. Apologies for the somewhat longish delay in incorporating it...
- Also add `\etocimmediatesettocdepth.toc`.
- Also add `\etocimmediatetoccontentsline` and its starred variant.
- Also add `\etocimmediatesetlocaltop.toc`.
- Fix an obscure bug (see source code comments) in the `\etocsetlocaltop.toc` mechanism.
- 1.09e [2021/09/23]
 Needed (if `etoc` is used without `hyperref`) updates to internal macros to prepare for the upcoming LaTeX November 2021 change to `\contentsline`.
 Related updates to the user macro `\etoccontentsline`.
- 1.09d [2021/07/13]
 Some minor synching with `tableof` 1.4c.
 Add `\etockeeporiginaltableofcontents` to provide a work-around to a compatibility issue with `listings`'s `\lstlistoflistings`, which abuses `\tableofcontents` for doing something unrelated to the actual contents. Thanks to Denis Bitouzé for report.
 Usage: `\usepackage{etoc}\etockeeporiginaltableofcontents`, then however you must employ `\etoc\tableofcontents`, not `\tableofcontents`.
- 1.09c [2020/05/15]
 Syncs with KOMA-script deprecation of `\iftocfeature`.
- 1.09a, 1.09b [2019/11/17]
 Sync with memoir v3.7i which has a better location of the TOC `hyperref` anchor. The `\etoc-aftertitlehook` can now freely be used also with memoir class (formerly its usage in case of memoir class was preempted by `etoc` itself). For more details refer to the section "Compatibility with the memoir class".
- 1.09 [2019/03/09]
 New features: `\etoclocaltop`, `\localtableofcontentswithrelativedepth`. Thanks to Tony Roberts for feature request.
 Note to hackers: internal control sequence `\Etoc@localtop` is gone.
`etoc` now requires e-TeX (`\numexpr`, `\unless`).
- 1.08p [2018/07/04]
 Fixed bug surfacing in case of `linktoc=page` option of `hyperref`. Thanks to Denis Bitouzé for report (cf. <https://github.com/ho-tex/hyperref/issues/65>, <https://github.com/dbitouze/yathesis/issues/61>).
- 1.08o [2018/06/15]
 Fixed bug showing up if an unnumbered TOC entry starts with a brace, and document uses `hyperref`. Caused by a typo in a macro name at previous release.
- 1.08n [2018/02/23]
 Refactoring of core macros detecting `\numberline` and its variants.
- 1.08m [2018/02/07]
 Fix to 1.08k's introduced incompatibility with KOMA-script and `tocbasic`'s `\nonumberline`.
- 1.08l [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.
- 1.08k [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.
- 1.08j [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.

1.08i [2016/09/29]

This fixes an issue dating back to 1.08e-2015/04/17: under `\etocchecksemtiness` 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` 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.

1.08h [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.

1.08g [2015/08/29]

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

1.08f [2015/04/28]

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

1.08e [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}`.

1.08d [2015/04/09]

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

Thanks to Christine Römer!

1.08c [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`.

1.08b [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`.
- 1.08a [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 1.07f [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.
- 1.08 [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.
- 1.07n [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.
- 1.07m [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`.
- 1.07l [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!
- 1.07l [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`.
- 1.07k [2014/03/06]
Compatibility with package `tocloft`; and improved compatibility with class `memoir`. Novel TOC example in Overview.
- 1.07j [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`).
- 1.07i [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).
- 1.07h [2013/10/16]
New commands `\etocdepthtag.toc`, `\etocsettagdepth`, `\etocobeydepthtags`, `\etocignoredepthtags`.
- 1.07g [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).

1.07f [2013/03/07]

New macros `\etoclinkedname`, `\etoclinkednumber`, `\etoclinkedpage`, and `\etoclink`.

1.07e [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.

1.07d [2013/02/24]

Minor code improvements and new documentation section “Another compatibility mode”.

1.07b [2013/02/02]

Removal of the `\xspace` from the macros `\etocname`, `\etocnumber`, `\etocpage`.

Additional examples in the documentation.

1.07 [2013/01/29]

New commands:

`\etocthename`, `\etocthenumber`, `\etocthepage`, `\etoclink`,

`\etococcontentsline`, `\etococcontentsline*`

`\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.

1.06 [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.

1.05 [2012/12/01]

`\localtableofcontents` replaces `\tableofcontents*` (for compatibility with the memoir class).

Compatibility with KOMA-script and memoir document classes.

1.04 [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}`

1.02 [2012/11/18]

Initial version.

61. Implementation

The “syntax highlighting” was added at release 1.09f of 2022/08/30.

Control sequences are mostly hyperlinks. When a user level command gets defined it hyperlinks to the user documentation with bold face and **using this colour**. Further instances if they occur will use **this colour** to link to their place of first definition. The latter applies also to non-user level macros. And such non-user level macros, at the location of their first definitions will have their names displayed using bold face and **this colour** and there is no hyperlink then. Comments belonging to the code source have been configured to be rendered in their own colour, and **non commented-out and non-control sequences tokens use this colour**. Other tokens use the fall-back normal colour.

Further, when a package macro is mentioned in code comments it also hyperlinks to the location of its first definition using **this colour** (user level commands do not link back to the user manual, but they link to their place of first macro definition, which is the one hyperlinking to the relevant user manual section). Sometimes however the location of first definition is disappointing as it may basically be provisory. Or the macro is in fact re-defined during execution of other macros. Besides, some macros are defined using `\csname . . . \endcsname` constructs (typically things defined with `part`, `chapter`, `section` etc... in their names) and thus can not be detected by the syntax highlighting automatization.

Apology: the code comments served mainly as a record for the author’s benefit of the historical evolution of the package and rarely as a description of what the macros do. At 1.1a I have removed almost all code comments which had accumulated as in a palimpsest. As a result, very few comments actually remain.

etoc, when not left in compatibility mode, hijacks the `\contentsline` expansion so as to not execute `\l@chapter`, `\l@section` etc..., but rather to parse the data and extract from it the *name*, *number*, and *page number*. The \LaTeX `.toc` data is *not structured*, but contains already typesetting mark-up. The **etoc** maneuvers to disentangle *name* and *number* are somewhat fragile as they expect the `.toc` file to contain the `\contentsline` arguments to be arranged in a certain manner. Of course **etoc** can be easily broken if changes happen to how data is stored there. Things would have been much easier for **etoc** in 2012 if the `\contentsline` arguments had considered the section titles (aka *name* for **etoc**) and their numbers (which are not numbers in the sense of things with which \TeX can compute, in general) separately, each providing an argument to `\contentsline`. But some mix is prepared, which may depend on the document class also, and besides usually handles `\part` levels very differently. Fortunately upstream changes happen rarely.

The other core part of **etoc** present from day one of the package is that it creates a tree-like structure of the sectioning levels present in the `.toc` file. But this is purely virtual, and handled via a notion of “level” and \TeX conditionals. It could be fun to implement officially such a tree (where the children of a sectioning title are the sectioning levels at a greater depth such as subsections versus a section). Let us recall that \LaTeX provides zero means to know from a subsection for example, what is the title of the section containing it, or chapter, or part. To do this one has to create a really structured document which neither core \LaTeX nor the main document classes do. This remark was given for document body, but it also applies to the `.toc` data. But **etoc** adds at least some kind of follow-up to the successive encountered sectioning titles, and is thus able, to “on-the-fly” add some kind of structure and follow the chaining of levels. Ultimately this is why the `\etocsetstyle` offers `{<start>}` and `{<finish>}` parts in additions to `{<contents>}` (which I divided into a `{<prefix>}` and a `{<contents>}`). At some point one could imagine that a really *structured* document (in opposition to what core \LaTeX from thirty years ago up to nowadays realizes) would store in the `.toc` data directly a tree structure, where each node would have attributes name, number, page number, completely separated from any typesetting. Once this exists then basically **etoc** disappears. In brief, once **etoc** ideas will have permeated the society, it will disappear as its was born only to palliate the absence of real structure in the `.toc` file (which is sort of inherited from the absence of real structure in a \LaTeX document body).

1.1a implements a radical change to all internals for compatibility with (atow future) `hyperref`. In order to facilitate this overhaul, we require \LaTeX 2020/10/01 for the fourth argument to `\contentsline` lines in the `.toc` file to be always present.

```

1 \NeedsTeXFormat{LaTeX2e}[2020/10/01]
2 \ProvidesPackage{etoc}
3 [2023/01/15 v1.1b Completely customisable TOCs (JFB)]
4 \RequirePackage{multicol}
5 \DeclareOption*{\PackageWarning{etoc}{Option ‘\CurrentOption’ is unknown.}}
6 \ProcessOptions\relax
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}}
```

`\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
13 \def\etocglobaldefs{\let\Etoc@global\global\let\tof@global\global}
14 \def\etoclocaldefs {\let\Etoc@global\@empty\let\tof@global\@empty}
```

This paragraph and the next were added very late in the history of the package, at 1.09h. Prior to 1.1a, etoc aliased \l@chapter, \l@section, etc... (at the time of TOC typesetting only) to its own \Etoc@lxyz in order to leave time to hyperref to add its marks-up. Thus the disentangling was more complex than it is now. With 1.1a, etoc leaves \l@chapter etc... unmodified and inserts itself hyperlinks via \hyperlink.

```

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
26 \newif\ifEtoc@tocwithid
27 \newif\ifEtoc@standard
28 \newif\ifEtoc@skipprefix
29 \newif\ifEtoc@isfirst
30 \newif\ifEtoc@localtoc
31 \newif\ifEtoc@skipthisone
32 \newif\ifEtoc@stoptoc
33 \newif\ifEtoc@notactive
34 \newif\ifEtoc@mustclosegroup
35 \newif\ifEtoc@emptytoc
36 \newif\ifEtoc@checkemptiness
37 \def\etoccheckemptiness {\Etoc@checkemptinesstrue }
38 \def\etocdoesnotcheckemptiness {\Etoc@checkemptinessfalse }
39 \newif\ifEtoc@notocifnotoc
40 \def\etocnotocifnotoc {\Etoc@checkemptinesstrue\Etoc@notocifnotoctrue }
41 \def\etoc@{\etoc@}
42 \newif\ifEtoc@bracedname
43 \newcounter{etoc@tocid}
44 \newcounter{etoc@tocdepth}
45 \@ifclassloaded{memoir}{\def\Etoc@minf{-\thr@@}}{\def\Etoc@minf{-\tw@}}
46 \def\Etoc@@minustwo@@{-\tw@}
47 \let\Etoc@@minusone@@\m@ne
48 \let\Etoc@@zero@@ \z@
49 \let\Etoc@@one@@ \@ne
50 \let\Etoc@@two@@ \tw@
51 \let\Etoc@@three@@ \thr@@
52 \chardef\Etoc@@four@@ 4
53 \chardef\Etoc@@five@@ 5
54 \chardef\Etoc@@six@@ 6
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}
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

```

At 1.1a, the `\etocsave` section to `toc` line and cousins are deprecated and when they will be removed, `\Etoc@levelist` will be too as it now serves only for their definitions.

```

75 \let\Etoc@levelist\@empty
76 \def\Etoc@newlevel #1{\expandafter\def\expandafter\Etoc@levelist\expandafter
77   {\Etoc@levelist\Etoc@levelist@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}

```

1.09g adapts to `hyperref` depending on whether the latter is at 7.00u or earlier. Indeed internal

changes to `hyperref` at 7.00u broke `etoc`. Thanks to Denis Brrouzé for reporting the issue.

But 1.1a radically simplifies matters at `etoc` core, and with it the breakage induced from `hyperref` 7.00u would not have happened.

```
125 \AtBeginDocument{%
126 \@ifpackageloaded{parskip}{\Etoc@parskiptrue}{}%
127 \@ifpackageloaded{hyperref}{\Etoc@hyperreftrue}{}%
128 }
```

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.

Attention that `\Etoc@setflags` must not modify `\Etoc@next!`

```
129 \def\Etoc@setflags {%
130   \ifcase \Etoc@level
131     \global\Etoc@vfalse
132     \global\Etoc@ivfalse
133     \global\Etoc@iiifalse
134     \global\Etoc@iifalse
135     \global\Etoc@ifalse
136     \global\Etoc@true
137   \or
138     \global\Etoc@vfalse
139     \global\Etoc@ivfalse
140     \global\Etoc@iiifalse
141     \global\Etoc@iifalse
142     \global\Etoc@itru
143   \or
144     \global\Etoc@vfalse
145     \global\Etoc@ivfalse
146     \global\Etoc@iiifalse
147     \global\Etoc@iitru
148   \or
149     \global\Etoc@vfalse
150     \global\Etoc@ivfalse
151     \global\Etoc@iiiitru
152   \or
153     \global\Etoc@vfalse
154     \global\Etoc@ivtrue
155   \or
156     \global\Etoc@vtrue
157   \else
158     \global\Etoc@vfalse
159     \global\Etoc@ivfalse
160     \global\Etoc@iiifalse
161     \global\Etoc@iifalse
162     \global\Etoc@ifalse
163     \global\Etoc@false
164     \ifnum\Etoc@level=\m@ne
165       \global\Etoc@jtrue
166     \else
167       \global\Etoc@jfalse
168       \global\Etoc@jjtrue
169     \fi
170   \fi
171 }
```

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.

```

172 \def\Etoc@doends {%
173   \ifcase \Etoc@level
174     \ifEtoc@v \Etoc@end@five \fi
175     \ifEtoc@iv \Etoc@end@four \fi
176     \ifEtoc@iii \Etoc@end@three \fi
177     \ifEtoc@ii \Etoc@end@two \fi
178     \ifEtoc@i \Etoc@end@one \fi
179   \or
180     \ifEtoc@v \Etoc@end@five \fi
181     \ifEtoc@iv \Etoc@end@four \fi
182     \ifEtoc@iii \Etoc@end@three \fi
183     \ifEtoc@ii \Etoc@end@two \fi
184   \or
185     \ifEtoc@v \Etoc@end@five \fi
186     \ifEtoc@iv \Etoc@end@four \fi
187     \ifEtoc@iii \Etoc@end@three \fi
188   \or
189     \ifEtoc@v \Etoc@end@five \fi
190     \ifEtoc@iv \Etoc@end@four \fi
191   \or
192     \ifEtoc@v \Etoc@end@five \fi
193   \or
194   \else
195     \ifEtoc@v \Etoc@end@five \fi
196     \ifEtoc@iv \Etoc@end@four \fi
197     \ifEtoc@iii \Etoc@end@three \fi
198     \ifEtoc@ii \Etoc@end@two \fi
199     \ifEtoc@i \Etoc@end@one \fi
200     \ifEtoc@ \Etoc@end@zero \fi
201     \ifnum \Etoc@level <-\tw@
202       \ifEtoc@j \Etoc@end@minusone \fi
203   \fi
204 \fi
205 }

```

Calls `\Etoc@setflags` which must not modify `\Etoc@next`.

```

206 \def\Etoc@dobegins{%
207   \let\Etoc@next \@empty
208   \Etoc@global\Etoc@isfirstfalse
209   \ifcase\Etoc@level
210     \ifEtoc@
211     \else
212       \def\Etoc@next{\Etoc@begin@zero}%
213     \fi
214   \or
215     \ifEtoc@i
216     \else
217       \def\Etoc@next{\Etoc@begin@one}%
218     \fi
219   \or
220     \ifEtoc@ii
221     \else
222       \def\Etoc@next{\Etoc@begin@two}%
223     \fi
224   \or
225     \ifEtoc@iii

```

```

226     \else
227     \def\Etoc@next{\Etoc@begin@three}%
228     \fi
229   \or
230   \ifEtoc@iv
231   \else
232     \def\Etoc@next{\Etoc@begin@four}%
233     \fi
234   \or
235   \ifEtoc@v
236   \else
237     \def\Etoc@next{\Etoc@begin@five}%
238     \fi
239   \else
240     \ifnum\Etoc@level=\m@ne
241       \ifEtoc@j
242       \else
243         \def\Etoc@next{\Etoc@begin@minusone}%
244         \fi
245       \else
246         \ifEtoc@jj
247         \else
248           \def\Etoc@next{\Etoc@begin@minustwo}%
249           \fi
250       \fi
251     \fi
252     \Etoc@setflags
253     \Etoc@next
254 }
255 \def\etocskipfirstprefix {\global\Etoc@skipprefixtrue }
    Heart of etoc hack into execution of .toc file.
256 \def\Etoc@etoccontentsline #1{%
257   \global\expandafter\let\expandafter\Etoc@level\csname Etoc@#1@\endcsname
258   \Etoc@skipthisonefalse
259   \ifnum\Etoc@level=\Etoc@@six@@
260     \Etoc@skipthisonetrue
261   \else
262     \ifEtoc@localtoc
263     \ifEtoc@stoptoc
264       \Etoc@skipthisonetrue
265     \else
266       \ifEtoc@notactive\Etoc@setflags
267       \Etoc@skipthisonetrue
268     \else
269       \unless\ifnum\Etoc@level>\etoclocaltop
270       \Etoc@skipthisonetrue
271       \global\Etoc@stoptoctrue
272     \fi
273   \fi
274   \fi
275   \fi
276   \fi
277   \let\Etoc@next\@gobblefour
278   \ifEtoc@skipthisone
279   \else
280     \unless\ifnum\c@tocdepth<\Etoc@level
281     \ifEtoc@standard

```

```

282     \let\Etoc@next\Etoc@savedcontentsline
283     \else
284     \let\Etoc@next\Etoc@etoccontentsline@
285     \fi
286     \fi
287     \fi
288     \Etoc@next{#1}%
289 }

```

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

1.1a implements a radical change to all internals for compatibility with (atow future) **hyperref**. Formerly this macro fetched only the first argument. It now also fetches all four (the fourth argument of `\contentsline` is always present since L^AT_EX 2020/10/01). The `\Etoc@lxyz` used to receive only two arguments like `\l@chapter`, `\l@section`, etc... which were formerly `\let` to it. It now also receives the fourth argument of `\contentsline`.

```

290 \def\Etoc@etoccontentsline@ #1#2#3#4{%
291     \Etoc@doends
292     \Etoc@dobegins
293     \Etoc@global\edef\Etoc@prefix {\expandafter\noexpand
294         \csname Etoc@prefix@\csname Etoc@#1@\endcsname\endcsname }%
295     \Etoc@global\edef\Etoc@contents{\expandafter\noexpand
296         \csname Etoc@contents@\csname Etoc@#1@\endcsname\endcsname }%
297     \ifEtoc@skipprefix \Etoc@global\let\Etoc@prefix\@empty\fi
298     \global\Etoc@skipprefixfalse

```

1.1a made core rewrites for **hyperref** compatibility, and simplified the logic. The code does not execute the original `\contentsline` in a custom wrapper, which needed to have aliased all `\l@section` etc... to `\Etoc@lxyz`. Now the **etoc** custom `\contentsline` passes over directly to `\Etoc@lxyz` which will do itself the job of adding hyperlinking according to the status of **hyperref**'s `\Hy@linktoc`.

This is a breaking change if a user hacked `\contentsline` to do some specific pre-processing of the data, as this extra will now be ignored. The kind of hack one can think of is perhaps to pre-process the section title to turn it into uppercase, this kind of things, but why do such things when one is using **etoc** which precisely provides a general interface for such customization? Besides as the L^AT_EX legacy set-up already mixes up in various ways name and number in the second argument of `\contentsline`, doing such hacks in a non-breaking way was not easy, and could have broken **etoc** easily anyhow.

The major hacker was **hyperref**... Indeed in 2012 when I started work on **etoc**, it was not clear to me how **hyperref** would end up using the fourth argument of `\contentsline` and I did not want to spend too much time tracing **hyperref** code. So I simply let **hyperref** do its stuff, and added specific post-processing branches to unravel it. It looks quite dumb in retrospect (at this time the `.toc` file lines with `\contentsline` had either three or four arguments which contributed for the design decisions back then).

All **hyperref** specific branches are now gone, replaced by extra code added depending on the status of the `\ifEtoc@hyperref` boolean. We also check the `\Hy@linktoc \chardef` status and (imitating **hyperref** do not hyperlink the page number argument if it turns out empty. This maintains backwards-compatibility with earlier releases of **etoc**.

```

299     \Etoc@lxyz{#2}{#3}{#4}%
300     \Etoc@prefix\Etoc@contents
301 }

```

1.1a makes a core rewrite for compatibility with **hyperref**. In particular `\Etoc@lxyz` now has a third argument which was the fourth argument of `\contentsline`. All various “@hyp” code branches have been removed. We only check the `\ifEtoc@hyperref` boolean here.

A **breaking change** is made at 1.1a: `\etoclink` will always create an hyperlink, even in case of `hyperref` being (possibly locally) configured to obey `linktoc=none`. Formerly, in such case, `\etoclink` added no hyperlink because `etoc` identified the hyperlink target from the `hyperref` hacked arguments of `\l@section` et al, rather than picking it from the fourth argument of `\contentsline`.

Another **breaking change** (documented only at 1.1b): all three of `\etocthelinkedname`, `\etocthelinkednumber`, and `\etocthelinkedpage` are always hyperlinks (for the latter, only if page number is not empty to match `hyperref` ways). Formerly they obeyed the `linktoc` status, somewhat counterintuitively, but this meant that `\etocname` etc... were their robust variants, which meant one could store easily for later usage (see the documentation examples with “treetoks”) their precise meaning. The breaking change happened in part because I was fooled myself by the macro names, and refactored the code in two steps separated by months so in second step I forgot I had only provisory code. And I decided finally to keep the breaking change.

```

302 \def\Etoc@lxyz #1#2#3{%
303   \ifEtoc@hyperref
304     \Etoc@global\def\etocthelink##1{\hyperlink{#3}{##1}}%
305   \else
306     \Etoc@global\let\etocthelink\@firstofone
307   \fi
308   \Etoc@global\def\etocthepage {#2}%
309   \ifEtoc@hyperref
310     \ifx\etocthepage\ltx@empty
311       \Etoc@global\let\etocthelinkedpage\etocthepage
312     \else
313       \Etoc@global\def\etocthelinkedpage{\hyperlink {#3}{#2}}%
314     \fi
315   \else
316     \Etoc@global\let\etocthelinkedpage\etocthepage
317   \fi

```

Define `\etocthename` and `\etocthelinkedname`, but this will perhaps be redone later if it is found out that the entry was numbered.

```

318   \Etoc@global\def\etocthename{#1}%
319   \ifEtoc@hyperref
320     \Etoc@global\def\etocthelinkedname{\hyperlink{#3}{#1}}%
321   \else
322     \Etoc@global\let\etocthelinkedname\etocthename
323   \fi

```

Now we check if the entry was numbered and then disentangle the number from the name.

```

324   \futurelet\Etoc@getnb@token
325   \Etoc@@getnb #1\relax\relax\etoc@

```

We may believe the entry was not numbered due it being a `\part` entry which has a specific format. So we need to do some additional job in such cases.

```

326   \ifEtoc@number
327   \else
328     \ifnum\Etoc@level=\m@ne
329     \futurelet\Etoc@getit@token
330     \Etoc@@getit #1\hspace\relax\etoc@
331   \fi
332   \fi

```

If there was a number the `\etocthename` and `\etocthenumber` got redefined. We know need to give proper definition to `\etocthelinkedname` and `\etocthelinkednumber`.

```

333   \ifEtoc@number
334   \ifEtoc@hyperref

```

```

335     \def\Etoc@tmp##1##2{\Etoc@global\def##2{\hyperlink{#3}{##1}}}%
336     \expandafter\Etoc@tmp\expandafter{\etocthename}\etocthelinkedname
337     \expandafter\Etoc@tmp\expandafter{\etocthenumber}\etocthelinkednumber
338     \else
339     \Etoc@global\let\etocthelinkedname\etocthename
340     \Etoc@global\let\etocthelinkednumber\etocthenumber
341     \fi
342     \else
343     \Etoc@global\let\etocthenumber\@empty
344     \Etoc@global\let\etocthelinkednumber\@empty
345     \fi
346     \Etoc@global\expandafter\let\csname etoclink \endcsname\etocthelink
347     \Etoc@global\expandafter\let\csname etocname \endcsname\etocthename
348     \Etoc@global\expandafter\let\csname etocnumber \endcsname\etocthenumber
349     \Etoc@global\expandafter\let\csname etocpage \endcsname\etocthepage
350     \ifEtoc@hyperref
351     \Etoc@lxyz@hyper
352     \fi
353 }

```

In presence of `hyperref`, `etoc` 1.1a adds the hyperlinks itself. For this it imitates the `hyperref` own logic and tests `\Hy@linktoc` and if the page number is empty. This adds a dependency that `\Hy@linktoc` should exist and have the expected interpretation.

Matters of tagging will have to wait for `LATEX` itself to show me what it does in `\l@section` etc... so that I can imitate.

```

354 \def\Etoc@lxyz@hyper{%
355     \ifcase\Hy@linktoc
356     % none: nothing to do
357     \or % section (aka name for etoc): link name and number
358     \Etoc@global\expandafter\let\csname etocname \endcsname\etocthelinkedname
359     \Etoc@global\expandafter\let\csname etocnumber \endcsname\etocthelinkednumber
360     \or % page. Do as hyperref: add a link only if page number isn't empty.
361     \ifx\etocthepage\ltx@empty
362     \else
363     \Etoc@global\expandafter\let\csname etocpage \endcsname\etocthelinkedpage
364     \fi
365     \else % all
366     \Etoc@global\expandafter\let\csname etocname \endcsname\etocthelinkedname
367     \Etoc@global\expandafter\let\csname etocnumber \endcsname\etocthelinkednumber
368     \ifx\etocthepage\ltx@empty
369     \else
370     \Etoc@global\expandafter\let\csname etocpage \endcsname\etocthelinkedpage
371     \fi
372     \fi
373 }
374 \def\Etoc@@getnb {%
375     \Etoc@bracednamefalse
376     \let\Etoc@next\Etoc@getnb
377     \ifx\Etoc@getnb@token@sptoken\let\Etoc@next\Etoc@getnb@nohyp@nonbr\fi
378     \ifx\Etoc@getnb@token\bgroup \let\Etoc@next\Etoc@getnb@checkifbraced\fi
379     \Etoc@next
380 }
381 \def\Etoc@tworelax{\relax\relax}
382 \def\Etoc@getnb@checkifbraced #1#2\etoc@ {%
383     \def\Etoc@tmp{#2}%
384     \ifx\Etoc@tmp\Etoc@tworelax
385     \expandafter\expandafter\expandafter\Etoc@getnb@nameisbraced

```

```

386     \expandafter\@firstofone
387     \else
388     \expandafter\Etoc@getnb
389     \fi {#1}#2\etoc@
390 }
391 \def\Etoc@getnb@nameisbraced {%
392     \Etoc@bracednametrue
393     \futurelet\Etoc@getnb@token\Etoc@getnb
394 }

```

The @nohyp in the macro names is a legacy of pre-1.1a code.

```

395 \def\Etoc@getnb {%
396     \ifx\Etoc@getnb@token\bgroup
397     \expandafter\Etoc@getnb@nohyp@nonbr
398     \else
399     \expandafter\Etoc@getnb@nohyp
400     \fi
401 }

```

1.08n tries to handle reasonably the \nonumberline of *KOMA-script*. If it expands to \numberline {}, **etoc** will thus consider the line numbered with an empty number.

At 1.09f complications to get rid of a *KOMA-script* \nonumberline even though its meaning is \empty then, but let's get rid of it in case the tokens are stored by the user section styles for delayed use.

```

402 \def\Etoc@getnb@nohyp #1{%
403     \in@{#1}{\numberline\chapternumberline\partnumberline\booknumberline}%
404     \ifin@
405     \let\Etoc@next\Etoc@getnb@nohyp@nmbrd
406     \else
407     \let\Etoc@next\Etoc@getnb@nohyp@nonbr
408     \in@{#1}{\nonumberline}%
409     \ifin@
410         \ifx\nonumberline\empty
411             \let\Etoc@next\Etoc@getnb@nohyp@nonumberline
412         \else
413             \def\Etoc@next{\expandafter\Etoc@getnb@nohyp@nmbrd}%
414         \fi
415     \fi
416     \fi
417     \Etoc@next #1%
418 }
419 \def\Etoc@getnb@nohyp@nmbrd #1#2#3\relax\relax\etoc@ {%
420     \Etoc@global\Etoc@numbertrue
421     \Etoc@global\def\etocthenumber {#2}%
422     \Etoc@global\def\etocthelinkednumber {#2}%
423     \ifEtoc@bracedname
424         \Etoc@global\def\etocthename{#{3}}%
425     \else
426         \Etoc@global\def\etocthename{#3}%
427     \fi
428     \Etoc@global\let\etocthelinkedname \etocthename
429 }

```

\etocthename and \etocthelinkedname already defined in \Etoc@lxyz, no need to update.

```

430 \def\Etoc@getnb@nohyp@nonbr #1\etoc@ {%
431     \Etoc@global\Etoc@numberfalse
432     \Etoc@global\let\etocthenumber \@empty
433     \Etoc@global\let\etocthelinkednumber \@empty

```

434 }

Special KOMA branch: #1 starts with `\nonumberline` which has here same meaning as `\empty`. It is in the current provisory definition of `\etocthename`. We get rid of this token, in case usage is delayed, to avoid improbable problems.

```
435 \def\Etoc@getnb@nohyp@nonumberline #1\relax\relax\etoc@ {%
436   \Etoc@global\Etoc@numberfalse
437   \Etoc@global\let\etocthenumber \@empty
438   \Etoc@global\let\etocthelinkednumber \@empty
439   \ifEtoc@bracedname
440     \Etoc@global\expandafter\def\expandafter\etocthename
441                                   \expandafter{\expandafter{#1}}%
442   \else
443     \Etoc@global\expandafter\def\expandafter\etocthename\expandafter{#1}%
444   \fi
445   \Etoc@global\let\etocthelinkedname \etocthename
446 }
```

1.1a maps directly to `\Etoc@getit@nohyp`, and the `@nohyp` will perhaps be removed from its name at some later stage.

```
447 \def\Etoc@@getit{%
448   \ifEtoc@bracedname
449     \expandafter\expandafter\expandafter\Etoc@getit@nameisbraced
450     \expandafter\@firstofone
451   \else
452     \expandafter\Etoc@getit@nohyp
453   \fi
454 }
455 \def\Etoc@getit@nameisbraced {\futurelet\Etoc@getit@token\Etoc@getit@nohyp }
456 \def\Etoc@getit@nohyp #1\hspace#2#3\etoc@ {%
457   \ifx\relax#2%
458   \else
459     \Etoc@global\def\etocthenumber {#1}%
460     \Etoc@global\def\etocthelinkednumber {#1}%
461     \Etoc@global\Etoc@numbertrue
462     \Etoc@getit@nohyp@getname #3\etoc@
463   \fi
464 }
465 \def\Etoc@getit@nohyp@getname #1\hspace\relax\etoc@ {%
466   \ifEtoc@bracedname
467     \Etoc@global\def\etocthename {{#1}}%
468   \else
469     \Etoc@global\def\etocthename {#1}%
470   \fi
471   \Etoc@global\let\etocthelinkedname\etocthename
472 }
```

1.08a of 2015/03/12 does the long-postponed thing to let `\etocname` etc.. be simply the robust variant of `\etocthelinkedname` etc...

```
473 \let\etocthename \@empty
474 \let\etocthenumber \@empty
475 \let\etocthepage \@empty
476 \let\etocthelinkedname \@empty
477 \let\etocthelinkednumber \@empty
478 \let\etocthelinkedpage \@empty
479 \let\etocthelink \@firstofone
480 \DeclareRobustCommand*\etocname {}
481 \DeclareRobustCommand*\etocnumber {}
```

```

482 \DeclareRobustCommand*\etocpage {}
483 \DeclareRobustCommand*\etoclink {\@firstofone}
484 \DeclareRobustCommand*\etocifnumbered
485   {\ifEtoc@number\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi}
486 \expandafter\let\expandafter\etocifnumbered\csname etocifnumbered \endcsname
487 \DeclareRobustCommand*\etociffirst
488   {\ifEtoc@isfirst\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi}
489 \expandafter\let\expandafter\etociffirst\csname etociffirst \endcsname
490 \def\Etoc@readtoc {%
491   \ifeof \Etoc@tf
492   \else
493     \read \Etoc@tf to \Etoc@buffer
494     \Etoc@toctoks=\expandafter\expandafter\expandafter
495       {\expandafter\the\expandafter\Etoc@toctoks\Etoc@buffer}%
496     \expandafter\Etoc@readtoc
497   \fi
498 }

1.07m moves the reading of the toc file At Begin Document. Needed for Babel activated characters.
499 \Etoc@toctoks {}% (superfluous, but for clarity)
500 \AtBeginDocument{\IfFileExists{\jobname.toc}
501   {\endlinechar=\m@ne
502     \makeatletter
503     \newread\Etoc@tf
504     \openin\Etoc@tf\@filef@und
505     \Etoc@readtoc
506     \global\Etoc@toctoks=\expandafter{\the\Etoc@toctoks}%
507     \closein\Etoc@tf}}
508   {\typeout{No file \jobname.toc.}}}
509 \def\Etoc@openouttoc{%
510   \ifEtoc@hyperref
511   \ifx\hyper@last\@undefined
512     \IfFileExists{\jobname .toc}
513     {\Hy@WarningNoLine
514       {old toc file detected, not used; run LaTeX again (cheers from etoc)}}%
515     \global\Etoc@toctoks={}%
516   }
517   {}%
518   \fi
519   \fi
520   \if@filesw
521     \newwrite \tf@toc
522     \immediate \openout \tf@toc \jobname .toc\relax
523   \fi
524   \global\let\Etoc@openouttoc\empty
525 }
526 \def\Etoc@toctoc{%
527   \global\Etoc@vfalse
528   \global\Etoc@ivfalse
529   \global\Etoc@iiifalse
530   \global\Etoc@iifalse
531   \global\Etoc@ifalse
532   \global\Etoc@false
533   \global\Etoc@jfalse
534   \global\Etoc@jjfalse
535   \the\Etoc@toctoks
536   \ifEtoc@notactive
537   \else

```

```

538 \ifEtoc@v \Etoc@end@five\fi
539 \ifEtoc@iv \Etoc@end@four\fi
540 \ifEtoc@iii \Etoc@end@three\fi
541 \ifEtoc@ii \Etoc@end@two\fi
542 \ifEtoc@i \Etoc@end@one\fi
543 \ifEtoc@ \Etoc@end@zero\fi
544 \ifEtoc@j \Etoc@end@minusone\fi
545 \ifEtoc@jj \Etoc@end@minustwo\fi
546 \fi
547 }

```

Memo: `\etoclocaltop` has only meaningful meaning when the local toc has its "active" flag set. Except that I used a "notactive" flag to torture myself, so: has the "notactive" flag off.

Also attention 1.09 now sets the `\c@tocdepth` there in relation to implementation of `\localtableofcontentswithrelativedepth`. This is needed for line styles in compatibility mode as the code from standard classes checks the tocdepth counter.

```

548 \def\etoc@startlocaltoc#1#2{%
549   \ifEtoc@localtoc
550     \ifnum #1=#2\relax
551       \ifEtoc@jj \global\let\etoclocaltop\Etoc@@minustwo@@\fi
552       \ifEtoc@j \global\let\etoclocaltop\Etoc@@minusone@@\fi
553       \ifEtoc@ \global\let\etoclocaltop\Etoc@@zero@@ \fi
554       \ifEtoc@i \global\let\etoclocaltop\Etoc@@one@@ \fi
555       \ifEtoc@ii \global\let\etoclocaltop\Etoc@@two@@ \fi
556       \ifEtoc@iii \global\let\etoclocaltop\Etoc@@three@@ \fi
557       \ifEtoc@iv \global\let\etoclocaltop\Etoc@@four@@ \fi
558       \ifEtoc@v \global\let\etoclocaltop\Etoc@@five@@ \fi
559       \global\c@tocdepth\Etoc@localtocdepth
560       \global\Etoc@notactivefalse
561       \global\Etoc@vfalse
562       \global\Etoc@ivfalse
563       \global\Etoc@iiifalse
564       \global\Etoc@iifalse
565       \global\Etoc@ifalse
566       \global\Etoc@false
567       \global\Etoc@jfalse
568       \global\Etoc@jjfalse
569     \fi
570   \fi
571 }
572 \let\etoc@startlocaltoc\@gobble

```

`\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.

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 `\ifEtoc@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`.

```

573 \def\Etoc@setemptytocbool {%
574   \global\Etoc@emptytoctrue

```

```

575 \global\Etoc@stoptocfalse
576 \global\let\Etoc@level\Etoc@minf
577 \begingroup
578 \ifEtoc@localtoc
579 \def\etoc@startlocaltoc##1{%
580 \ifnum##1=\Etoc@tocid\relax
581 \global\let\etoclocaltop\Etoc@level
582 \global\Etoc@notactivefalse
583 \fi }%
584 \let\contentsline\Etoc@testingcontentslinelocal
585 \else
586 \let\contentsline\Etoc@testingcontentsline
587 \fi
588 \Etoc@storetocdepth

```

1.09i fixes a bug arising when document has used `\etocsetlocaltop.toc`.

```

589 \let\Etoc@setlocaltop@doends@dobegins\@empty
590 \the\Etoc@toctoks
591 \Etoc@restoretocdepth
592 \endgroup
593 }
594 \DeclareRobustCommand*\etocifwasempty
595 {\ifEtoc@emptytoc\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi }
596 \expandafter
597 \let\expandafter\etocxifwasempty\csname etocifwasempty \endcsname
598 \def\Etoc@testingcontentslinelocal #1{%
599 \ifEtoc@stoptoc
600 \else
601 \ifnum\csname Etoc@#1@\endcsname=\Etoc@six@@
602 \else
603 \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 `\ifEtoc@stoptoc` flag afterwards in all cases. `\Etoc@localtop` will have been initialized by `\etoc@startlocaltoc` as `\ifEtoc@notactive` flag can be false only then.

```

604 \ifEtoc@notactive
605 \else
606 \ifnum\Etoc@level>\etoclocaltop
607 \unless\ifnum\Etoc@level>\Etoc@localtocdepth
608 \global\Etoc@emptytocfalse
609 \global\Etoc@stoptoctrue
610 \fi
611 \else
612 \global\Etoc@stoptoctrue
613 \fi
614 \fi
615 \fi
616 \fi
617 \@gobblefour\relax
618 }
619 \def\Etoc@testingcontentsline #1{%
620 \ifEtoc@stoptoc
621 \else
622 \ifnum\csname Etoc@#1@\endcsname=\Etoc@six@@
623 \else
624 \unless\ifnum\csname Etoc@#1@\endcsname>\c@tocdepth
625 \global\Etoc@emptytocfalse

```

```

626     \global\Etoc@stoptoctrue
627     \fi
628     \fi
629     \fi
630     \@gobblefour\relax
631 }

```

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 `\ifEtoc@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 `\ifEtoc@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` 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 `\Etoc@stoptoctrue` flag, the `\Etoc@etoccontentsline` is more efficient now.

The `\ifEtoc@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-c-C-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.

```
632 \def\Etoc@localtableofcontents#1{%
633   \gdef\etoclocaltop{-\@m}%
634   \Etoc@localtoctrue
635   \global\Etoc@emptytocfalse
636   \edef\Etoc@tocid{#1}%
637   \ifnum\Etoc@tocid<\@ne
638     \setbox0\hbox{\ref{Unknown toc ref \@secondoftwo#1. Rerun LaTeX}}%
```

Do only heading, skip all the rest.

```
639   \global\Etoc@stoptoctrue
640   \gdef\etoclocaltop{-\thr@}%
641   \Etoc@tableofcontents
642   \expandafter\Etoc@gobbletoetoc@
643   \fi
644   \global\Etoc@notactivetrue
645   \ifEtoc@checkemptiness
646     \Etoc@setemptytocbool
647   \fi
648   \ifEtoc@emptytoc
649     \ifEtoc@notactive
650       \setbox0\hbox{\ref{Unknown toc ID \number\Etoc@tocid. Rerun LaTeX}}%
```

Assume real one will be non-empty and print only heading for this pass.

```
651   \global\Etoc@emptytocfalse
652   \global\Etoc@stoptoctrue
653   \gdef\etoclocaltop{-\thr@}%
654   \Etoc@tableofcontents
655   \expandafter\expandafter\expandafter\Etoc@gobbletoetoc@
656   \fi
657   \else
658     \global\Etoc@stoptocfalse
659     \global\Etoc@notactivetrue
```

We can end up here either if the emptiness check was done and turned negative (then `\etoclocaltop` has the correct level for usage in first argument of `\etocsettocstyle`), or if the emptiness check was not done. For the latter case `\etoclocaltop` has setting `-\@m`.

```
660   \edef\etoc@startlocaltoc##1%
661     {\noexpand\etoc@@startlocaltoc{##1}{\Etoc@tocid}}%
662   \Etoc@tableofcontents
663   \fi
664   \@gobble\etoc@
665   \endgroup\ifEtoc@mustclosegroup\endgroup\fi
666   \Etoc@tocdepthreset
667   \etocaftertochook
```

1.09 hesitates about putting this before of after the `\etocaftertochook`. The information of the used `tocdepth` is lost if putting it first. The usage of `\etocaftertochook` may be broken if unexpected tokens follows.

```
668   \let\Etoc@localtocdepth\c@tocdepth
669 }% \Etoc@localtableofcontents
```

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`.

```
670 \def\Etoc@getrefno #1#2\etoc@ {#1}
671 \def\Etoc@getref #1{\@ifundefined{r@#1}{0}{\expandafter\expandafter\expandafter
672   \Etoc@getrefno\csgname r@#1\endcsgname\relax\etoc@}}
673 \def\Etoc@ref#1{\Etoc@localtableofcontents{\Etoc@getref{#1}}}
674 \def\Etoc@label#1{\label{#1}\futurelet\Etoc@nexttoken\Etoc@t@bleofcontents}
675 \@firstofone{\def\Etoc@again} {\futurelet\Etoc@nexttoken\Etoc@t@bleofcontents}
```

`\ref{foo}` expects `foo` to be a label to a *local* TOC.

The syntax `\localtableofcontents\ref{foo}` is supported.

```
676 \def\Etoc@dothis #1#2\etoc@ {\fi #1}
677 \def\Etoc@t@bleofcontents{%
678   \gdef\etoclocaltop{-\@M}%
679   \ifx\Etoc@nexttoken\label\Etoc@dothis{\expandafter\Etoc@label\@gobble}\fi
680   \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.

```
681   \ifx\Etoc@nexttoken\ref\Etoc@dothis{\expandafter\Etoc@ref\@gobble}\fi
```

Flag to check if we were called from a `\localtableofcontents`.

```
682   \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` is left to be `\@gobble`, and `\Etoc@tocid` is never tested. We don't need to set the `\ifEtoc@toc@notactive` flag as now `\Etoc@testingcontentsline` tests first the `\ifEtoc@localtoc` flag (was already the case of `\Etoc@etoccontentsline`). I change a bit the style of conditionals here for clarity of code.

```
683   \global\Etoc@emptytocfalse\ifEtoc@checkemptiness\Etoc@setemptytocbool\fi
684   \ifEtoc@emptytoc
685     \ifEtoc@notocifnotoc
686     \expandafter\expandafter\expandafter\@gobble
687     \fi
688   \fi
689   \Etoc@tableofcontents
690   \endgroup\ifEtoc@mustclosegroup\endgroup\fi
691   \Etoc@tocdepthreset
692   \etocaftertochook
693   \@gobble\etoc@
694   }% \Etoc@t@bleofcontents
```

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).

```
695 \def\Etoc@table@fcontents{%
696   \refstepcounter{etoc@tocid}%
697   \Etoc@tocwithidfalse
698   \futurelet\Etoc@nexttoken\Etoc@t@bleofcontents}
699 \def\Etoc@localtable@fcontents{%
700   \refstepcounter{etoc@tocid}%
701   \addtocontents{toc}{\string\etoc@startlocaltoc{\the\c@etoc@tocid}}%
702   \Etoc@tocwithidtrue
```

```

703 \futurelet\Etoc@nexttoken\Etoc@t@bleofcontents}
704 \let\etocaftertitlehook \@empty
705 \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.

The `\Etoc@tocdepthset` is for `\etocsetnexttocdepth` mechanism.

```

706 \def\etoctableofcontents{%
707   \Etoc@openouttoc
708   \Etoc@tocdepthset
709   \begingroup

```

This group will be closed in `\Etoc@t@bleofcontents` or `\Etoc@localtableofcontents`.

Prior to its release 1.4c, `tableof` added a group pair via `\tof@begin` and `\tof@finish`. This was removed at 1.4c. So no need to do anything now here about silencing `\tof@begingroup` and `\tof@endgroup`: they are inserted only in the `tableof` private copy of the `.toc` file which is used by its own table of contents typesetting command.

1.09b uses a `\def` in non-starred variant for allowing tricks to recognize later on if we are in a starred or non-starred case, whatever the user definition of `\etocaftertitlehook` may be.

```

710   \ifstar
711   {\let\Etoc@aftertitlehook\@empty\Etoc@table@fcontents}
712   {\def\Etoc@aftertitlehook{\etocaftertitlehook}\Etoc@table@fcontents}%
713 }% \etoctableofcontents

```

See at end of this file for the explanation of this 1.09d added saving of original meaning at package loading time.

```

714 \let\etocoriginaltableofcontents\tableofcontents
715 \let\tableofcontents\etoctableofcontents
716 \newcommand*\localtableofcontents{%
717   \Etoc@openouttoc
718   \Etoc@tocdepthset
719   \begingroup

```

This group closed in `\Etoc@t@bleofcontents` or `\Etoc@localtableofcontents`. Same comment relative to `tableof`. No need to do anything here.

```

720   \ifstar
721   {\let\Etoc@aftertitlehook\@empty\Etoc@localtable@fcontents}
722   {\def\Etoc@aftertitlehook{\etocaftertitlehook}\Etoc@localtable@fcontents}%
723 }% \localtableofcontents

```

1.09 adds `\localtableofcontentswithrelativedepth`. The motivation is that it is impossible to set in a completely satisfactory way the `tocdepth` from the first argument of `\etocsettocstyle` because the emptiness check does not execute that argument. If one does not care about emptiness check, one can of course set the `tocdepth` from the first argument of `\etocsettocstyle`. One can even use `\etocsetnexttocdepth` for that since 1.09 but anyway direct usage of `\etocsettocdepth` (or `\setcounter{tocdepth}{...}`) is easier as its effect when executed there is automatically limited to duration of the TOC.

```

724 \newcommand*\localtableofcontentswithrelativedepth[1]{%
725   \edef\Etoc@localtocdepth
726     {\numexpr\noexpand\etoclocaltop+\the\numexpr#1\relax\relax}%
727   \localtableofcontents
728 }% \localtableofcontentswithrelativedepth
729 \let\Etoc@localtocdepth\c@tocdepth

```

The code does only expandable things after `\Etoc@toctoc`, in order to allow opening of an alignment in the first argument and closing in the second.

```

730 \long\def\Etoc@gobbletoetoc@ #1\etoc@ {}
731 \newcommand\etocsettocstyle[2]{%

```

```

732 \def\Etoc@tableofcontents
733 {%
734   \ifnum\c@tocdepth>\Etoc@minf
735   \else
736     \expandafter\Etoc@gobbletoetoc@
737   \fi
738   \Etoc@par
739   \etocbeforetitlehook
740   \Etoc@storetocdepth
741   \let\Etoc@savedcontentsline\contentsline
742   \let\contentsline\Etoc@etoccontentsline
743   \ifEtoc@standard
744   \else

```

1.1a's `\Etoc@lxyz` now fetches 3 not 2 arguments and the `\l@section` etc... are not `\let` to it anymore, as they used to be here formerly. For backwards compatibility the `\etocsavedchapter-tocline` etc... are still defined, but raise a deprecation warning.

```

745   \def\Etoc@levellist@elt####1{%
746     \expandafter\def\csname etocsaved####1tocline\endcsname
747       {\PackageWarning{etoc}{\expandafter\string\csname
748         etocsaved####1tocline\endcsname\space is deprecated
749         and to be\MessageBreak soon removed from etoc.
750         Use directly \expandafter\string\csname l@####1\endcsname\MessageBreak }}%
751     \expandafter\let\csname etocsaved####1tocline\expandafter\endcsname
752       \csname l@####1\endcsname
753     \csname l@####1\endcsname
754   }%
755 }%
756 \Etoc@levellist
757 \fi

```

1.09 makes `\etocsetnexttocdepth` usable in #1 (but this is not 100% compatible with the emptiness check).

1.09 makes an `\etoclocaltop` usable in #1 if under checksemtiness regime.

```

758 #1%
759 \Etoc@tocdepthset
760 \ifEtoc@parskip\parskip\z@skip\fi
761 \Etoc@aftertitlehook

```

1.09 has replaced former `\Etoc@localtop` (*minus one*) by `\etoclocaltop`. Under `checksemtinesstrue` regime its value is already known, but it will be obtained again from the toc file execution. As it is used only if TOC is active, resetting it here this way is decorative and could be removed.

```

762 \gdef\etoclocaltop{-\thr@}%
763 \Etoc@toctoc
764 \etocaftercontentshook
765 #2\@nobreakfalse
766 \Etoc@restoretocdepth

```

`\contentsline` was set to `\Etoc@etoccontentsline` by a non-global `\let`, and it will recover its normal value from exiting a scope limiting group. But `tableof` (since 1.4a) under `\etocglobaldefstrue` does a global redefinition of `\contentsline`. Its `\tof@finish` then does a global restore of `\contentsline`, but it will be to the `etoc` set value. `\tof@finish` is active only if either the table of contents was typeset using `\tableof`, `\tablenotof`, `\tableoftaggedcontents`, or `\nextocwithtags` was used. If not active it is either undefined (no package `tableof`) or `\@empty`. Prior to `tableof` 1.4c, the `\tof@finish` closed a group and could be undefined as well, but not if `\etocglobaldefstrue`.

If rather than `\@empty` the `\tof@finish` fall-back was `\relax` we could use here `\@ifundefined` to check in one go (matters of speaking because expansion of `\@ifundefined` is not in

"one-go"). Maybe I should update `tableof`, but for time being I will simply add an extra test. All this is probably lots of time on irrelevant issue.

```

767 \ifx\Etoc@global\global
768 \@ifundefined{tof@finish}
769 {}
770 {\ifx\tof@finish\@empty
771 \else
772 \global\let\contentsline\Etoc@savedcontentsline
773 \fi
774 }%
775 \fi
776 \@gobble\etoc@
777 }% end of \Etoc@tableofcontents definition by \etocsettocstyle
778 }
779 \def\etocsetstyle{\Etoc@standardfalse\etoc@setstyle}
780 \long\def\etocsetstyle#1#2#3#4#5{%
781 \expandafter\def
782 \csname Etoc@begin@\csname Etoc@#1@\endcsname\endcsname
783 {#2\Etoc@global\Etoc@isfirsttrue}%
784 \expandafter\def
785 \csname Etoc@prefix@\csname Etoc@#1@\endcsname\endcsname {#3}%
786 \expandafter\def
787 \csname Etoc@contents@\csname Etoc@#1@\endcsname\endcsname {#4}%
788 \expandafter\def
789 \csname Etoc@end@\csname Etoc@#1@\endcsname\endcsname {#5}%
790 }
791 \def\etocfontminustwo {\normalfont \LARGE \bfseries}
792 \def\etocfontminusone {\normalfont \large \bfseries}
793 \def\etocfontzero {\normalfont \large \bfseries}
794 \def\etocfontone {\normalfont \normalsize \bfseries}
795 \def\etocfonttwo {\normalfont \normalsize}
796 \def\etocfontthree {\normalfont \footnotesize}

placeholder for comments
797 \def\etocsepminustwo {4ex \@plus .5ex \@minus .5ex}
798 \def\etocsepminusone {4ex \@plus .5ex \@minus .5ex}
799 \def\etocsepzero {2.5ex \@plus .4ex \@minus .4ex}
800 \def\etocsepone {1.5ex \@plus .3ex \@minus .3ex}
801 \def\etocseptwo {.5ex \@plus .1ex \@minus .1ex}
802 \def\etocseptthree {.25ex \@plus .05ex \@minus .05ex}

placeholder for comments
803 \def\etocbaselinespreadminustwo {1}
804 \def\etocbaselinespreadminusone {1}
805 \def\etocbaselinespreadzero {1}
806 \def\etocbaselinespreadone {1}
807 \def\etocbaselinespreadtwo {1}
808 \def\etocbaselinespreadthree {.9}

placeholder for comments
809 \def\etocminustwoleftmargin {1.5em plus 0.5fil}
810 \def\etocminustworightmargin {1.5em plus -0.5fil}
811 \def\etocminusoneleftmargin {1em}
812 \def\etocminusonerightmargin {1em}
813 \def\etocclineleaders
814 {\hbox{\normalfont\normalsize\hb@xt@2ex {\hss.\hss}}}
815 \def\etocabbrevpagename {p.~}

```

Versions earlier than 1.08b (and since 1.05 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 `babel-french`, because `\frenchpartname` does things depending on the current value of the counter `part`. The code in recent `babel-french` (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...

```
816 \def\etocpartname      {Part}
817 \def\etocbookname     {Book}
```

The macro `\etocdefaultlines` was initially called `\etococlines`. Now `\etococlines` just does `\Etoc@standardfalse`.

Version 1.09f wraps `\etocbookname`, respectively `\etocpartname`, in the book, resp. part, line styles inside a (potential) hyperlink together with the number.

```
818 \def\etocdefaultlines{\Etoc@standardfalse
819 %% 'book':
820 \etoc@setstyle{@minustwo}
821   {\addpenalty\@M\etocskipfirstprefix}
822   {\addpenalty\@secpenalty}
823   {\begingroup
824     \etocfontminustwo
825     \addvspace{\etocsepminustwo}%
826     \parindent \z@
827     \leftskip  \etocminustwoleftmargin
828     \rightskip \etocminustworightmargin
829     \parfillskip \@flushglue
830     \vbox{\etocifnumbered{\etoclink{\etocbookname\enspace\etocthenumber:\quad}}{}}%
831           \etocname
832           \baselineskip\etocbaselinespreadminustwo\baselineskip
833     \par}%
834   \addpenalty\@M\addvspace{\etocsepminusone}%
835   \endgroup}
836 {}%
837 %% 'part':
838 \etoc@setstyle{@minusone}
839   {\addpenalty\@M\etocskipfirstprefix}
840   {\addpenalty\@secpenalty}
841   {\begingroup
842     \etocfontminusone
843     \addvspace{\etocsepminusone}%
844     \parindent \z@
845     \leftskip  \etocminusoneleftmargin
846     \rightskip \etocminusonerightmargin
847     \parfillskip \@flushglue
848     \vbox{\etocifnumbered{\etoclink{\etocpartname\enspace\etocthenumber.\quad}}{}}%
849           \etocname
850           \baselineskip\etocbaselinespreadminusone\baselineskip
851     \par}%
852   \addpenalty\@M\addvspace{\etocsepzero}%
853   \endgroup}
854 {}%
855 %% 'chapter':
856 \etoc@setstyle{@zero}
857   {\addpenalty\@M\etocskipfirstprefix}
```

```

858 {\addpenalty\@itempenalty}
859 {\begingroup
860 \etocfontzero
861 \addvspace{\etocsepzero}%
862 \parindent \z@ \parfillskip \@flushglue
863 \vbox{\etocifnumbered{\etocnumber.\enspace}{}}\etocname
864 \baselineskip\etocbaselinespreadzero\baselineskip
865 \par}%
866 \endgroup}
867 {\addpenalty{-\@highpenalty}\addvspace{\etocsepminusone}}%
868 %% 'section':
869 \etoc@setstyle{@one}
870 {\addpenalty\@M\etocskipfirstprefix}
871 {\addpenalty\@itempenalty}
872 {\begingroup
873 \etocfontone
874 \addvspace{\etocsepone}%
875 \parindent \z@ \parfillskip \z@
876 \setbox\z@\vbox{\parfillskip\@flushglue
877 \etocname\par
878 \setbox\tw@\lastbox
879 \global\setbox\@ne\hbox{\unhbox\tw@ } }%
880 \dimen\z@=\wd\@ne
881 \setbox\z@=\etocclineleaders
882 \advance\dimen\z@\wd\z@
883 \etocifnumbered
884 {\setbox\tw@\hbox{\etocnumber, \etocabbrevpagename\etocpage}}
885 {\setbox\tw@\hbox{\etocabbrevpagename\etocpage}}%
886 \advance\dimen\z@\wd\tw@
887 \ifdim\dimen\z@ < \linewidth
888 \vbox{\etocname~%
889 \leaders\box\z@\hfil\box\tw@
890 \baselineskip\etocbaselinespreadone\baselineskip
891 \par}%
892 \else
893 \vbox{\etocname~%
894 \leaders\copy\z@\hfil\break
895 \hbox{ }\leaders\box\z@\hfil\box\tw@
896 \baselineskip\etocbaselinespreadone\baselineskip
897 \par}%
898 \fi
899 \endgroup}
900 {\addpenalty\@secpenalty\addvspace{\etocsepzero}}%
901 %% 'subsection':
902 \etoc@setstyle{@two}
903 {\addpenalty\@medpenalty\etocskipfirstprefix}
904 {\addpenalty\@itempenalty}
905 {\begingroup
906 \etocfonttwo
907 \addvspace{\etocseptwo}%
908 \parindent \z@ \parfillskip \z@
909 \setbox\z@\vbox{\parfillskip\@flushglue
910 \etocname\par\setbox\tw@\lastbox
911 \global\setbox\@ne\hbox{\unhbox\tw@ } }%
912 \dimen\z@=\wd\@ne
913 \setbox\z@=\etocclineleaders
914 \advance\dimen\z@\wd\z@

```



```

967   {\multicolpretolerance\etocmulticolpretolerance
968   \multicoltolerance\etocmulticoltolerance
969   \setlength{\columnsep}{\etoccolumnsep}%
970   \setlength{\multicolsep}{\etocmulticolsep}%
971   \begin{multicols}{#1}[#2\etocoldpar\addvspace{\etocinnertopsep}]}
972   {#2\ifvmode\else\begin{group}\interlinepenalty\@M\parskip\z@skip
973     \@@par\endgroup
974     \fi
975     \nobreak\addvspace{\etocinnertopsep}%
976     \pretolerance\etocmulticolpretolerance
977     \tolerance\etocmulticoltolerance}%
978   }%
979   {\ifnum #1>\@ne
980     \expandafter\@firstofone
981     \else \expandafter\@gobble
982     \fi
983   {\end{multicols}}}%
984   \addvspace{\etocbelowtocskip}}}%
985 }

placeholder for comments

986 \def\etocinnerbottomsep{3.5ex}
987 \def\etocinnerleftsep{2em}
988 \def\etocinnerrightsep{2em}
989 \def\etoctoprule{\hrule}
990 \def\etocleftrule{\vrule}
991 \def\etocrightrule{\vrule}
992 \def\etocbottomrule{\hrule}
993 \def\etoctoprulecolorcmd{\relax}
994 \def\etocbottomrulecolorcmd{\relax}
995 \def\etocleftrulecolorcmd{\relax}
996 \def\etocrightrulecolorcmd{\relax}

placeholder

997 \def\etoc@ruledheading #1{%
998   \hb@xt@\linewidth{\color@begingroup
999     \hss #1\hss\hskip-\linewidth
1000     \etoctoprulecolorcmd\leaders\etoctoprule\hss
1001     \phantom{#1}}%
1002     \leaders\etoctoprule\hss\color@endgroup}%
1003     \nointerlineskip\nobreak\vskip\etocinnertopsep}
1004 \newcommand*\etocruledstyle[2][\etocdefaultnbc1]{%
1005 \etocsettocstyle
1006   {\addvspace{\etocabovetocskip}}%
1007   \ifnum #1>\@ne
1008     \expandafter\@firstoftwo
1009     \else \expandafter\@secondoftwo
1010     \fi
1011     {\multicolpretolerance\etocmulticolpretolerance
1012     \multicoltolerance\etocmulticoltolerance
1013     \setlength{\columnsep}{\etoccolumnsep}%
1014     \setlength{\multicolsep}{\etocmulticolsep}%
1015     \begin{multicols}{#1}[\etoc@ruledheading{#2}]}
1016     {\etoc@ruledheading{#2}}%
1017     \pretolerance\etocmulticolpretolerance
1018     \tolerance\etocmulticoltolerance}}
1019   {\ifnum #1>\@ne\expandafter\@firstofone
1020     \else \expandafter\@gobble

```

```

1021 \fi
1022 {\end{multicols}}%
1023 \addvspace{\etocbelowtocskip}}
placeholder
1024 \def\etocframedmphook{\relax}
1025 \long\def\etocbkgcolorcmd{\relax}
1026 \long\def\Etoc@relax{\relax}
placeholder for comments
1027 \newbox\etoc@framed@titlebox
1028 \newbox\etoc@framed@contentsbox
1029 \newcommand*\etocframedstyle[2][\etocdefaultnbcoll]{%
1030 \etocsettocstyle{%
1031 \addvspace{\etocabovetocskip}}%
1032 \sbox\z@{#2}%
1033 \dimen\z@\dp\z@
1034 \ifdim\wd\z@<\linewidth \dp\z@\z@ \else \dimen\z@\z@ \fi
1035 \setbox\etoc@framed@titlebox=\hb@xt@\linewidth{\color@begingroup
1036 \hss
1037 \ifx\etocbkgcolorcmd\Etoc@relax
1038 \else
1039 \sbox\tw@{\color{white}%
1040 \vrule\width\wd\z@\@height\ht\z@\@depth\dimen\z@}%
1041 \ifdim\wd\z@<\linewidth \dp\tw@\z@\fi
1042 \box\tw@
1043 \hskip-\wd\z@
1044 \fi
1045 \copy\z@
1046 \hss
1047 \hskip-\linewidth
1048 \etocoprulecolorcmd\leaders\etocoprule\hss
1049 \hskip\wd\z@
1050 \etocoprulecolorcmd\leaders\etocoprule\hss\color@endgroup}%
1051 \setbox\z@\hbox{\etocleftrule\etocrightrule}%
1052 \dimen\tw@\linewidth\advance\dimen\tw@-\wd\z@
1053 \advance\dimen\tw@-\etocinnerleftsep
1054 \advance\dimen\tw@-\etocinnerrightsep
1055 \setbox\etoc@framed@contentsbox=\vbox\bgroup
1056 \hsize\dimen\tw@
1057 \kern\dimen\z@
1058 \vskip\etocinnertopsep
1059 \hbox\bgroup
1060 \begin{minipage}{\hsize}%
1061 \etocframedmphook
1062 \ifnum #1>\@ne
1063 \expandafter\@firstoftwo
1064 \else \expandafter\@secondoftwo
1065 \fi
1066 {\multicolpretolerance\etocmulticolpretolerance
1067 \multicoltolerance\etocmulticoltolerance
1068 \setlength{\columnsep}{\etoccolumnsep}%
1069 \setlength{\multicolsep}{\etocmulticolsep}%
1070 \begin{multicols}{#1}}
1071 {\pretolerance\etocmulticolpretolerance
1072 \tolerance\etocmulticoltolerance}}
1073 {\ifnum #1>\@ne\expandafter\@firstofone
1074 \else \expandafter\@gobble

```

```

1075     \fi
1076     {\end{multicols}\unskip }%
1077 \end{minipage}%
1078 \egroup
1079 \vskip\etocinnerbottomsep
1080 \egroup
1081 \vbox{\hsize\linewidth
1082     \ifx\etocbkgcolorcmd\Etoc@relax
1083     \else
1084         \kern\ht\etoc@framed@titlebox
1085         \kern\dp\etoc@framed@titlebox
1086         \hb@xt@\linewidth{\color@begingroup
1087             \etocleftcolorcmd\etocleftcolor
1088             \etocbkgcolorcmd
1089             \leaders\vrule
1090                 \@height\ht\etoc@framed@contentsbox
1091                 \@depth\dp\etoc@framed@contentsbox
1092             \hss
1093             \etocrightrulecolorcmd\etocrightrule
1094             \color@endgroup}\nointerlineskip
1095             \vskip-\dp\etoc@framed@contentsbox
1096             \vskip-\ht\etoc@framed@contentsbox
1097             \vskip-\dp\etoc@framed@titlebox
1098             \vskip-\ht\etoc@framed@titlebox
1099         \fi
1100         \box\etoc@framed@titlebox\nointerlineskip
1101         \hb@xt@\linewidth{\color@begingroup
1102             {\etocleftcolorcmd\etocleftcolor}%
1103             \hss\box\etoc@framed@contentsbox\hss
1104             \etocrightrulecolorcmd\etocrightrule\color@endgroup}
1105         \nointerlineskip
1106         \vskip\ht\etoc@framed@contentsbox
1107         \vskip\dp\etoc@framed@contentsbox
1108         \hb@xt@\linewidth{\color@begingroup\etocbottomrulecolorcmd
1109             \leaders\etocbottomrule\hss\color@endgroup}}
1110         \addvspace{\etocbelowtocskip}}}}
placeholder for comments
1111 \newcommand\etoc@multicoltoc[2][\etocdefaultnbcoll]{%
1112     \etocmulticolstyle[#1]{#2}%
1113     \tableofcontents}
1114 \newcommand\etoc@multicoltoCi[2][\etocdefaultnbcoll]{%
1115     \etocmulticolstyle[#1]{#2}%
1116     \tableofcontents*}
1117 \newcommand\etoc@local@multicoltoc[2][\etocdefaultnbcoll]{%
1118     \etocmulticolstyle[#1]{#2}%
1119     \localtableofcontents}
1120 \newcommand\etoc@local@multicoltoCi[2][\etocdefaultnbcoll]{%
1121     \etocmulticolstyle[#1]{#2}%
1122     \localtableofcontents*}
placeholder for comments
1123 \newcommand*\etoc@ruledtoc[2][\etocdefaultnbcoll]{%
1124     \etocruledstyle[#1]{#2}%
1125     \tableofcontents}
1126 \newcommand*\etoc@ruledtoCi[2][\etocdefaultnbcoll]{%
1127     \etocruledstyle[#1]{#2}%
1128     \tableofcontents*}

```

```

1129 \newcommand*\etoc@local@ruledtoc[2][\etocdefaultnbcol]{%
1130   \etocruledstyle[#1]{#2}%
1131   \localtableofcontents}
1132 \newcommand*\etoc@local@ruledtoci[2][\etocdefaultnbcol]{%
1133   \etocruledstyle[#1]{#2}%
1134   \localtableofcontents*}
  placeholder for comments
1135 \newcommand*\etoc@framedtoc[2][\etocdefaultnbcol]{%
1136   \etocframedstyle[#1]{#2}%
1137   \tableofcontents}
1138 \newcommand*\etoc@framedtoci[2][\etocdefaultnbcol]{%
1139   \etocframedstyle[#1]{#2}%
1140   \tableofcontents*}
1141 \newcommand*\etoc@local@framedtoc[2][\etocdefaultnbcol]{%
1142   \etocframedstyle[#1]{#2}%
1143   \localtableofcontents}
1144 \newcommand*\etoc@local@framedtoci[2][\etocdefaultnbcol]{%
1145   \etocframedstyle[#1]{#2}%
1146   \localtableofcontents*}
  placeholder for comments
1147 \def\etocmulticol{\begingroup
1148   \Etoc@mustclosegrouptrue
1149   \@ifstar
1150   {\etoc@multicoltoctoci}
1151   {\etoc@multicoltoctoc}}
1152 \def\etocruled{\begingroup
1153   \Etoc@mustclosegrouptrue
1154   \@ifstar
1155   {\etoc@ruledtoci}
1156   {\etoc@ruledtoc}}
1157 \def\etocframed{\begingroup
1158   \Etoc@mustclosegrouptrue
1159   \@ifstar
1160   {\etoc@framedtoci}
1161   {\etoc@framedtoc}}
1162 \def\etoclocalmulticol{\begingroup
1163   \Etoc@mustclosegrouptrue
1164   \@ifstar
1165   {\etoc@local@multicoltoctoci}
1166   {\etoc@local@multicoltoctoc}}
1167 \def\etoclocalruled{\begingroup
1168   \Etoc@mustclosegrouptrue
1169   \@ifstar
1170   {\etoc@local@ruledtoci}
1171   {\etoc@local@ruledtoc}}
1172 \def\etoclocalframed{\begingroup
1173   \Etoc@mustclosegrouptrue
1174   \@ifstar
1175   {\etoc@local@framedtoci}
1176   {\etoc@local@framedtoc}}
  placeholder for comments
1177 \def\etocarticlestyle{%
1178   \etocsettocstyle
1179   {\section *{\contentsname
1180     \@mkboth {\MakeUppercase \contentsname}
1181     {\MakeUppercase \contentsname}}}}

```

```

1182     {}
1183 \def\etocarticlestylenomarks{%
1184     \etocsettocstyle
1185     {\section *{\contentsname}}
1186     {}
    placeholder for comments
1187 \def\etocbookstyle{%
1188     \etocsettocstyle
1189     {\if@twocolumn \@restonecoltrue \onecolumn \else \@restonecolfalse \fi
1190     \chapter *{\contentsname
1191         \@mkboth {\MakeUppercase \contentsname}
1192                 {\MakeUppercase \contentsname}}}
1193     {\if@restonecol \twocolumn \fi}}
1194 \def\etocbookstylenomarks{%
1195     \etocsettocstyle
1196     {\if@twocolumn \@restonecoltrue \onecolumn \else \@restonecolfalse \fi
1197     \chapter *{\contentsname}}
1198     {\if@restonecol \twocolumn \fi}}
1199 \let\etocreportstyle\etocbookstyle
1200 \let\etocreportstylenomarks\etocbookstylenomarks

```

v3.7i of memoir has moved the `\phantomsection` to a better location, before typesetting the title and we follow suit at 1.09a, and less rashly at 1.09b. Formerly `etoc` used `\etocaftertitlehook` to mimic the memoir code but as its name indicate, it is supposedly executed after the title... and this also had the defect of making `\etocaftertitlehook` not anymore a user command. Thus we here use some refactoring of the `\Etoc@aftertitlehook` internal mechanism to help recognize if we are in the starred case or not.

`\phantomsection` is always defined by memoir, empty if hyperref absent.

```

1201 \def\etocmemoirtocformat #1#2{%
1202     \def\Etoc@addsuitablecontentsline{\addcontentsline {toc}{#1}{#2}}%
1203 }
1204 \def\etocmemoirstyle{%
1205     \etocsettocstyle
1206     {\ensureonecol \par \begingroup \phantomsection
1207     \ifx\Etoc@aftertitlehook\@empty
1208     \else
    non-starred variant
1209         \ifmem@em@starred@listof
1210         \else
1211             \Etoc@addsuitablecontentsline
1212         \fi
1213     \fi
1214     \@nameuse {@tocmaketitle} %<< space token here from memoir code
1215     \ifx\Etoc@aftertitlehook\@empty
1216     \else

```

Execute `etoc` hook before the `\cfttocbeforelisthook` and keep distinction between starred and non-starred contexts for other hooks. Notice that the memoir class way of implementing `\tableofcontents` leaves no way for code executed by the TOC code to know if it is executed in starred or non-starred context.

```

1217     \Etoc@aftertitlehook \let \Etoc@aftertitlehook \relax
1218     \fi
1219     \parskip \cftparskip \@nameuse {cfttocbeforelisthook}}
1220     {\@nameuse {cfttocafterlisthook}\endgroup\restorefromonecol}}

```

1.09c (2020/05/15) does an update as KOMA-script has deprecated `\iftocfeature`. Thanks to Bilel Omrani for report. I did not check if cloning of KOMA code required some further updates.

Finally 1.09f added some such updates.

```

1221 \ifdefined\Iftocfeature
1222   \def\etoc@Iftocfeature{\Iftocfeature}%
1223 \else
1224   \def\etoc@Iftocfeature{\iftocfeature}%
1225 \fi
1226 \def\etocscartclstyle{%
1227   \etocsettocstyle
1228     {\let\ifdynlist\if@tocleft
1229     \def@currentxt{toc}}%

```

1.09f adds this `\list@fname` definition hunk although `\listoftocname` is presumably always defined. Only to match closely KOMA-script code.

```

1230   \@ifundefined{listoftocname}%
1231     {\def\list@fname{\listofname~toc}}%
1232     {\expandafter\let\expandafter\list@fname\csname listoftocname\endcsname}%
1233   \etoc@Iftocfeature {toc}{onecolumn}
1234     {\etoc@Iftocfeature {toc}{leveldown}
1235     {}
1236     {\if@twocolumn \aftergroup \twocolumn \onecolumn \fi }}
1237     {}%
1238   \etoc@Iftocfeature {toc}{numberline}{\def \nonumberline {\numberline {}}}%
1239   \expandafter\tocbasic@listhead\expandafter {\list@fname}%
1240   \begingroup \expandafter \expandafter \expandafter
1241   \endgroup \expandafter
1242   \ifx
1243     \csname microtypesetup\endcsname \relax
1244   \else
1245     \etoc@Iftocfeature {toc}{noprotrusion}{}
1246     {\microtypesetup {protrusion=false}%
1247     \PackageInfo {tocbasic}%
1248     {character protrusion at toc deactivated}}%
1249   \fi

```

1.09f adds this hunk with the `noparskipfake` test to match current `scartcl` (testing done with `scrbook 2022/07/04 v3.37`).

```

1250   \etoc@Iftocfeature{toc}{noparskipfake}{}%
1251     \ifvmode \@tempskipa\lastskip \vskip-\lastskip
1252     \addtolength{\@tempskipa}{\parskip}\vskip\@tempskipa\fi
1253   }%
1254   \setlength {\parskip }{\z@ }%
1255   \setlength {\parindent }{\z@ }%
1256   \setlength {\parfillskip }{\z@ \@plus 1fil}%
1257   \csname tocbasic@@before@hook\endcsname
1258   \csname tb@toc@before@hook\endcsname}
1259   {}%

```

At 1.09f I considered adding this `\BeforeClosingMainAux` hunk to the second argument of `\etocsettocstyle`-emulation of KOMA-script. But:

- there seems to be no interface to `\tocbasic@end@toc@file`,
- it defaults to issuing a `\par`, but we want `etoc` to still be able to produce other TOCs, possibly inline, and they should not be influenced by it and I don't want at this stage to add an interface to enable/disable and have to document it,
- the whole thing appears to me to be ill-conceived in so far as it sort of implies the `\tableofcontents` is used only once, as each instance will again add this `\tocbasic@end@toc@file` to end of toc file, which may thus end up being executed multiple times.

So rather than putting the thing in the .toc file, we will execute it here. This way it will not impact other TOCs typeset via etoc design facilities in the document.

```

1260 %           \BeforeClosingMainAux
1261 %           {\addtocontents
1262 %             {toc}{\string\providecommand\string\tocbasic@end@toc@file{}}%
1263 %             \string\tocbasic@end@toc@file}%
1264 %           }%
1265           \providecommand\tocbasic@end@toc@file{\tocbasic@end@toc@file
1266           \csname tb@toc@after@hook\endcsname
1267           \csname tocbasic@@after@hook\endcsname}}
1268 \let\etocscrbookstyle\etocscrtclstyle
1269 \let\etocscrreprtstyle\etocscrtclstyle

placeholder for comments
1270 \def\etocstandarddisplaystyle{\etocarticlestyle}
1271 \newcommand*{\etocmarkboth[1]}{%
1272   \@mkboth{\MakeUppercase{#1}}{\MakeUppercase{#1}}}
1273 \newcommand*{\etocmarkbothnouc[1]}{\@mkboth{#1}{#1}}
1274 \newcommand\etocstyle[3][section]{\etocmulticolstyle[#2]}%
1275   {\csname #1\endcsname *{#3}}
1276 \newcommand\etocstylewithmarks[4][section]{\etocmulticolstyle[#2]}%
1277   {\csname #1\endcsname *{#3}\etocmarkboth{#4}}
1278 \newcommand\etocstylewithmarksnouc[4][section]{\etocmulticolstyle[#2]}%
1279   {\csname #1\endcsname *{#3}\etocmarkbothnouc{#4}}

placeholder for comments
1280 \def\Etoc@redefetocstyle#1{%
1281   \renewcommand\etocstylewithmarks[4][#1]
1282   {\etocmulticolstyle[##2]}%
1283   {\csname ##1\endcsname *{##3}\etocmarkboth{##4}}}
1284   \renewcommand\etocstylewithmarksnouc[4][#1]
1285   {\etocmulticolstyle[##2]}%
1286   {\csname ##1\endcsname *{##3}\etocmarkbothnouc{##4}}}
1287   \renewcommand\etocstyle[3][#1]{%
1288     \etocmulticolstyle[##2]{\csname ##1\endcsname *{##3}}}
1289 \@ifclassloaded{scrartcl}
1290   {\renewcommand*{\etocstandarddisplaystyle{\etocscrtclstyle}}{}}
1291 \@ifclassloaded{book}
1292   {\renewcommand*{\etocfontone{\normalfont\normalsize}
1293     \renewcommand*{\etocstandarddisplaystyle{\etocbookstyle}
1294     \Etoc@redefetocstyle{chapter}}{}}
1295 \@ifclassloaded{report}
1296   {\renewcommand*{\etocfontone{\normalfont\normalsize}
1297     \renewcommand*{\etocstandarddisplaystyle{\etocreportstyle}
1298     \Etoc@redefetocstyle{chapter}}{}}
1299 \@ifclassloaded{scrbook}
1300   {\renewcommand*{\etocfontone{\normalfont\normalsize}
1301     \renewcommand*{\etocstandarddisplaystyle{\etocscrbookstyle}
1302     \Etoc@redefetocstyle{chapter}}{}}
1303 \@ifclassloaded{scrreprt}
1304   {\renewcommand*{\etocfontone{\normalfont\normalsize}
1305     \renewcommand*{\etocstandarddisplaystyle{\etocscrreprtstyle}
1306     \Etoc@redefetocstyle{chapter}}{}}
1307 \@ifclassloaded{memoir}
1308   {\etocsetlevel{appendix}{0}}%
1309   \renewcommand*{\etocfontone{\normalfont\normalsize}
1310   \etocmemoirtocformat{chapter}{\contentsname}}%
1311   \renewcommand*{\etocstandarddisplaystyle{\etocmemoirstyle}

```

```

1312     \Etoc@redefetocstyle{chapter}}{}
1313 \def\etocloftstyle {%
1314     \etocsettocstyle
1315     {\@cfttocstart\par\begingroup
1316     \parindent\z@ \parskip\cftparskip \@cftmaketoctitle
1317     \ifcfttocbibind\@cftdobibtoc\fi }%
1318     {\endgroup\@cfttocfinish }%
1319 }
1320 \@ifclassloaded{memoir}{}
1321 {\@ifpackageloaded{tocloft}
1322     {\ifcftnctoc\else
1323     \renewcommand*\etocstandarddisplaystyle{\etocloftstyle}
1324     \AtBeginDocument{\let\tableofcontents\etoc\tableofcontents}
1325     \fi }
1326     {\AtBeginDocument
1327     {\@ifpackageloaded{tocloft}
1328     {\ifcftnctoc\else
1329     \PackageWarning {etoc}
1330     {Package 'tocloft' was loaded after 'etoc'!^^}%
1331     **** to prevent it from overwriting \protect\tableofcontents, it will be tricked^^}%
1332     **** into believing to have been loaded with option 'titles'. For better^^}%
1333     **** compatibility, please load 'tocloft' before 'etoc'}%
1334     \AtEndDocument{\PackageWarning{etoc}
1335     {Please load 'tocloft' /before/ 'etoc'!}}\fi
1336     \@cftnctoctrue }%
1337     {}%
1338     }%
1339     }%
1340 }

```

L^AT_EX 2021 fall release lets `\contentsline` always grab four arguments, so with 1.09e 2021/09/23 we make sure our `\addtocontents` will always provide `\contentsline` with four arguments. This extra `{}` is done without checking LaTeX's version by laziness, as an impact on documents compiled with former LaTeX could be visible only with very special contexts that only the author himself would ever consider.

Let's also add `\protected@file@percent` at 1.09e although this is a priori of no relevance as **etoc** reads the toc file with `\endlinechar=-1` regime.

When using `\addcontentsline` nothing needs to be done as both things are handled by L^AT_EX upstream.

```

1341 \def\Etoc@addtocontents #1#2%
1342     {\ifEtoc@hyperref
1343     \addtocontents {toc}{\protect\contentsline
1344     {#1}{#2}%
1345     {\thepage }{\@currentHref }%
1346     \ifdefined\protected@file@percent\protected@file@percent\fi
1347     }%
1348     \else
1349     \addtocontents {toc}{\protect\contentsline
1350     {#1}{#2}{\thepage }{}}%
1351     \ifdefined\protected@file@percent\protected@file@percent\fi
1352     }%
1353     \fi}
1354 \def\Etoc@addcontentsline@ #1#2#3{%
1355     \@namedef{toclevel@#1}{#3}\addcontentsline {toc}{#1}{#2}%
1356 }
1357 \DeclareRobustCommand*\etoc\contentsline{
1358     {\@ifstar{\Etoc@addcontentsline@}{\Etoc@addtocontents}}

```

```

1359 \def\Etoc@addtocontents@immediately#1#2{%
1360     \begingroup
1361     \let\Etoc@originalwrite\write
1362     \def\write{\immediate\Etoc@originalwrite}%
1363     \Etoc@addtocontents{#1}{#2}%
1364     \endgroup
1365 }
1366 \def\Etoc@addcontentsline@@immediately#1#2#3{%
1367     \begingroup
1368     \let\Etoc@originalwrite\write
1369     \def\write{\immediate\Etoc@originalwrite}%
1370     \Etoc@addcontentsline@{#1}{#2}{#3}%
1371     \endgroup
1372 }
1373 \DeclareRobustCommand*\etocimmediatetoccontentsline}
1374     {\@ifstar{\Etoc@addcontentsline@@immediately}{\Etoc@addtocontents@immediately}}
placeholder
1375 \def\Etoc@storetocdepth   {\global\c@etoc@tocdepth\c@tocdepth }
1376 \def\Etoc@restoretocdepth {\global\c@tocdepth\c@etoc@tocdepth }
1377 \def\etocobeytoctocdepth {\def\etoc@settocdepth
1378     {\afterassignment\Etoc@@nottoodeep \global\c@tocdepth}}
1379 \def\Etoc@@nottoodeep {\ifnum\c@tocdepth>\c@etoc@tocdepth
1380     \global\c@tocdepth\c@etoc@tocdepth\fi }
1381 \def\etocignoretoctocdepth {\let\etoc@settocdepth@gobble }
1382 \def\etocsettocdepth     {\futurelet\Etoc@nexttoken\Etoc@set@tocdepth }
1383 \def\Etoc@set@tocdepth   {\ifx\Etoc@nexttoken\bgroup
1384     \expandafter\Etoc@set@tocdepth@
1385     \else\expandafter\Etoc@set@toctocdepth
1386     \fi }
1387 \def\Etoc@set@tocdepth@ #1{\@ifundefined {Etoc@#1@@}
1388     {\PackageWarning{etoc}
1389     {Unknown sectioning unit #1, \protect\etocsettocdepth\space ignored}}
1390     {\global\c@tocdepth\csname Etoc@#1@@\endcsname}%
1391 }
1392 \def\Etoc@set@toctocdepth #1#{\Etoc@set@toctocdepth@ }
1393 \def\Etoc@set@toctocdepth@ #1{%
1394     \@ifundefined{Etoc@#1@@}%
1395     {\PackageWarning{etoc}
1396     {Unknown sectioning depth #1, \protect\etocsettocdepth.toc ignored}}%
1397     {\addtocontents {toc}
1398     {\protect\etoc@settocdepth\expandafter\protect\csname Etoc@#1@@\endcsname}}%
1399 }
placeholder
1400 \def\etocimmediatesettocdepth #1#{\Etoc@set@toctocdepth@immediately}
1401 \def\Etoc@set@toctocdepth@immediately #1{%
1402     \@ifundefined{Etoc@#1@@}%
1403     {\PackageWarning{etoc}
1404     {Unknown sectioning depth #1, \protect\etocimmediatesettocdepth.toc ig-
1405     nored}}%
1406     {\begingroup
1407     \let\Etoc@originalwrite\write
1408     \def\write{\immediate\Etoc@originalwrite}%
1409     \addtocontents {toc}
1410     {\protect\etoc@settocdepth\expandafter\protect
1411     \csname Etoc@#1@@\endcsname}%

```

```

1412 }%
1413 }
placeholder
1414 \def\etocdepthtag #1#\Etoc@depthtag }
1415 \def\Etoc@depthtag #1{\addtocontents {toc}{\protect\etoc@depthtag {#1}}}
1.09f adds \etocimmediatedepthtag.toc. This can serve in some circumstances, see user docu-
mentation. Apologies for long delay to Norman Ramsey who reported problem and his fix in July...
2016!
1416 \def\etocimmediatedepthtag #1#\Etoc@depthtag@immediately }
1417 \def\Etoc@depthtag@immediately #1{%
1418 \begingroup
1419 \let\Etoc@originalwrite\write
1420 \def\write{\immediate\Etoc@originalwrite}%
1421 \addtocontents {toc}{\protect\etoc@depthtag {#1}}%
1422 \endgroup
1423 }
1424 \def\etocignoredepthtags {\let\etoc@depthtag \@gobble }
1425 \def\etocobeydepthtags {\let\etoc@depthtag \Etoc@depthtag@ }
1426 \def\Etoc@depthtag@ #1{\@ifundefined{Etoc@depthof@#1}%
1427 {}% ignore in silence if tag has no associated depth
1428 {\afterassignment\Etoc@@nottodeep
1429 \global\c@tocdepth\c@name Etoc@depthof@#1\endcsname}%
1430 }
1431 \def\etocsettagdepth #1#2{\@ifundefined{Etoc@#2@@}%
1432 {\PackageWarning{etoc}
1433 {Unknown sectioning depth #2, \protect\etocsettagdepth\space ignored}}%
1434 {\namedef{Etoc@depthof@#1}{\@nameuse{Etoc@#2@@}}}%
1435 }
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.
1436 \def\Etoc@tocvsec@err #1{\PackageError {etoc}
1437 {the command \protect#1\space is incompatible with etoc}
1438 {use \protect\etocsettocdepth.toc as replacement (see etoc manual)}}%
1439 }%
1440 \AtBeginDocument {%
1441 \@ifclassloaded{memoir}
1442 {\PackageInfo {etoc}
1443 {Regarding memoir class command \protect\settocdepth, consider^^J
1444 \protect\etocsettocdepth.toc as a drop-in replacement with
1445 more capabilities^^J (see etoc manual). Similarly
1446 \protect\etocsettocdepth\space and
1447 \protect\etocsetnexttocdepth^^J should replace
1448 use of memoir's \protect\maxtocdepth\space command.}}
1449 {\@ifpackageloaded {tocvsec2}
1450 {\def\maxtocdepth #1{\Etoc@tocvsec@err \maxtocdepth }%
1451 \def\settocdepth #1{\Etoc@tocvsec@err \settocdepth }%
1452 \def\resettocdepth {\@ifstar {\Etoc@tocvsec@err \resettocdepth }%
1453 {\Etoc@tocvsec@err \resettocdepth }%
1454 }%
If etoc is added to a LATEX document using already tocvsec2.
1455 \def\save@tocdepth #1#2#3{%
1456 \let\reset@tocdepth\relax
1457 \let\remax@tocdepth\relax
1458 \let\tableofcontents\etoc\tableofcontents
1459 \PackageWarning {etoc}

```

```

1460     {package tocvsec2 detected and its modification of^^]
1461     \protect\tableofcontents\space reverted. Use
1462     \protect\etocsettocdepth.toc as a replacement^^]
1463     for the tocvsec2 toc-related commands}}}%
1464   }%
1465 }%

placeholder

1466 \def\invisibletableofcontents {\etocsetnexttocdepth {-3}\tableofcontents }%
1467 \def\invisiblelocaltableofcontents
1468     {\etocsetnexttocdepth {-3}\localtableofcontents }%
1469 \def\etocsetnexttocdepth #1{%
1470   \@ifundefined{Etoc@#1@@}
1471   {\PackageWarning{etoc}
1472     {Unknown sectioning unit #1, \protect\etocsetnexttocdepth\space ignored}}
1473   {\Etoc@setnexttocdepth{\csname Etoc@#1@@\endcsname}}}%
1474 }%
1475 \def\Etoc@setnexttocdepth#1{%
1476   \def\Etoc@tocdepthset{%
1477     \Etoc@tocdepthreset
1478     \edef\Etoc@tocdepthreset {%
1479       \global\c@tocdepth\the\c@tocdepth\space
1480       \global\let\noexpand\Etoc@tocdepthreset\noexpand\@empty
1481     }%
1482     \global\c@tocdepth#1%
1483     \global\let\Etoc@tocdepthset\@empty
1484   }%
1485 }%
1486 \let\Etoc@tocdepthreset\@empty
1487 \let\Etoc@tocdepthset \@empty
1488 \def\etocsetlocaltop #1#{\Etoc@set@localtop}%
1489 \def\Etoc@set@localtop #1{%
1490   \@ifundefined{Etoc@#1@@}%
1491   {\PackageWarning{etoc}
1492     {Unknown sectioning depth #1, \protect\etocsetlocaltop.toc ignored}}%
1493   {\addtocontents {toc}
1494     {\protect\etoc@setlocaltop\expandafter\protect\csname Etoc@#1@@\endcsname}}}%
1495 }%

placeholder

1496 \def\etocimmediatesetlocaltop #1#{\Etoc@set@localtop@immediately}%
1497 \def\Etoc@set@localtop@immediately #1{%
1498   \@ifundefined{Etoc@#1@@}%
1499   {\PackageWarning{etoc}
1500     {Unknown sectioning depth #1, \protect\etocimmediatesetlocaltop.toc ig-
1501     nored}}}%
1502   {\begingroup
1503     \let\Etoc@originalwrite\write
1504     \def\write{\immediate\Etoc@originalwrite}%
1505     \addtocontents {toc}
1506       {\protect\etoc@setlocaltop\expandafter\protect
1507         \csname Etoc@#1@@\endcsname}%
1507   \endgroup
1508 }%
1509 }%

```

1.09i would like to rename this to `\Etoc@setlocaltop`, for consistency with internal macros, but too late it is already in user `.toc` files.

```

1510 \def\etoc@setlocaltop #1{%
1511   \global\let\Etoc@level #1%
1512   \Etoc@skipthisonefalse
1513   \ifnum\Etoc@level=\Etoc@@six@@
1514     \Etoc@skipthisonetrue
1515   \else
1516     \ifEtoc@localtoc
1517       \ifEtoc@stoptoc\Etoc@skipthisonetrue
1518     \else
1519       \ifEtoc@notactive
1520         \Etoc@setflags
1521         \Etoc@skipthisonetrue
1522     \else

```

Attention, 1.09 has `\etoclocaltop` which is former `\Etoc@localtop` *minus one*. Hence the shape of the conditional. Memo: it makes sense only for code executed under "active" status of the TOC.

```

1523     \unless\ifnum\Etoc@level>\etoclocaltop
1524     \Etoc@skipthisonetrue
1525     \global\Etoc@stoptoctrue
1526   \fi
1527 \fi
1528 \fi
1529 \fi
1530 \fi
1531 \let\Etoc@next\@empty
1532 \ifEtoc@skipthisone
1533 \else
1534   \ifnum\c@tocdepth<\Etoc@level
1535   \else
1536     \ifEtoc@standard
1537   \else
1538     \let\Etoc@next\Etoc@setlocaltop@doends@dobegins
1539   \fi
1540 \fi
1541 \fi

```

At 1.09f the `\global\Etoc@skipprefixfalse` is added. This fixes a weird problem that an `\etocskipfirstprefix` from this level (which a priori only serves to delimit the local contents) impacted (in the global TOC for example) the rendering of the first finer subsequent heading. Not resetting the boolean was an oversight, and it shows in context of the package built-in level styles as they do use `\etocskipfirstprefix`. But this also stresses that the name of the macro is a bit of a misnomer, yes it serves to delimit local table of contents, but really it is implemented as a ghost of a sectioning unit which does have an impact (on the global TOC or local TOCs from encompassing levels), as it triggers when encountered the `{\finish}` portions of previous finer levels (and the `{\finish}` code of its own level will be executed sooner or later), and the `{\start}` code of subsequent finer levels (as well as its own `{\start}` code at least once, depending on how levels are nested).

1.09i fixes a bad behaviour of `etoc` in `\etocchecksempiness` regime when `\etocsetlocaltop.toc` has been made use of in the document. Indeed the execution of `\Etoc@toctoks` for the emptiness check thus triggered execution of `{\start}` and `{\finish}` parts of the line styles. So we abstract an `\Etoc@setlocaltop@doends@dobegins` which will be made a no-op in emptiness-checking contexts. Also, the code lines above have been edited now that `\Etoc@next` here is not followed by tokens within a brace pair.

```

1542   \Etoc@next
1543 }%
1544 \def\Etoc@setlocaltop@doends@dobegins
1545   {\Etoc@doends\Etoc@dobegins\global\Etoc@skipprefixfalse}%

```

```

1546 \addtocontents {toc}{\protect\@ifundefined{etocstyle}%
1547     {\let\protect\etoc@startlocaltoc\protect\@gobble
1548     \let\protect\etoc@settocdepth\protect\@gobble
1549     \let\protect\etoc@depthtag\protect\@gobble
1550     \let\protect\etoc@setlocaltop\protect\@gobble}}}%

```

Initializations.

```

1551 \def\etocstandardlines {\Etoc@standardtrue}
1552 \def\etococlines      {\Etoc@standardfalse}
1553 \etocdefaultlines
1554 \etocstandardlines
1555 \etocstandarddisplaystyle
1556 \etocbeytocdepth
1557 \etocbeydepthtags
1558 \let\etocaftertoohook \@empty
1559 \let\etocbeforetitlehook \@empty

```

listings abuses `\tableofcontents` for its `\lstlistoflistings`. It doesn't seem worth to let my version of `\tableofcontents` have to check for this special circumstance. So at 1.09d, simply add this (and induced annoying updates to documentation at at least two locations):

```

1560 \def\etockeeporiginaltableofcontents
1561     {\let\tableofcontents\etocoriginaltableofcontents}%
1562 \endinput

```