The etoc package

Jean-François Burnol

jfbu (at) free (dot) fr
Package version:
v1.08f (2015/04/28)

I. Overview
- 7 Linked list of the main package commands
- 6 A second example
- 5 A first example
- 4 Line styles and toc display style
- 3 Do I need to be a geek to use etoc?
- 41 Implementation
- 40 Change history
- 39 Timestamp

II. Arbitrarily many TOCs, and local ones too
- 14 Two Examples
- 13 Adding commands to the .toc file
- 12 The commands \etoc-depth-tag.toc and \etocset-tagdepth
- The command \etocset-tocdepth.toc
- The \etocset-tocdepth and \etoc-setnext-tocdepth commands
- etoc: the re-assignment of levels with \etocset-level ... using elsewhere

III. Surprising uses of etoc
- 16 Arbitrary "Lists Of...", \etoctoc-con-tentsline
- 17 A TOC with a fancy layout
- 18 Another compatibility mode
- 19 The TOC as a tree
- 20 The TOC as a molecule
- 21 The TOC as a TikZ mind map
- 22 The TOC as a table
- 23 Not displayed empty TOCs
- 15 The TOC of TOCs

IV. Commands for the toc line styles
- 24 The \etocset-style command
- 25 The \etocset-level command
- 26 Scope of commands added to the .toc file
- 27 Am I also red?

V. Commands for the toc display style
- 28 Specifying the toc display
- 29 Starred variants of the \tableof-contents etc... comm
- 30 Table of contents for this part
- 31 Summary of the main style comm
- 32 The package default line styles: \etocde-fault-lines

VI. Using and customizing etoc
- 33 Customizing etoc
- 34 One more example of colored TOC layout

VII. Tips
- 35 ... and tricks

VIII. etoc and the outside world
- 36 Compatibility with other packages
- 37 TeXnical matters
- 38 Errors and catastrophes

IX. The code

TABLE OF CONTENTS

Part IV. Commands for the toc line styles

Part V. Commands for the toc display style

Part VI. Using and customizing etoc

Part VII.   Tips

Part VIII.   etoc and the outside world

Part IX.   The code

Abstract

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

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

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

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

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

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

1. Foreword

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

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

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

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

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

Starting with release `1.07k` it is possible to use **etoc** concurrently with package `tocloft`.

2. License

```
% Package: etoc
% Version: 1.08f (2015/04/28)
% License: LPPL 1.3c or later
% Copyright (C) 2012-2015 Jean-Francois Burnol <jfbu at free dot fr>
% Copyright (C) 2014-2015 Christine Roemer <Christine_Roemer at t-online dot de>
% and collaborators for the translation into German of the documentation
%
%     This Work may be distributed and/or modified under the
%     conditions of the LaTeX Project Public License, either
%     version 1.3c of this license or (at your option) any later
%     version. This version of this license is in
%         http://www.latex-project.org/lppl/lppl-1-3c.txt
%     and the latest version of this license is in
%         http://www.latex-project.org/lppl.txt
%     and version 1.3 or later is part of all distributions of
%     LaTeX version 2005/12/01 or later.
%
% The Authors of this Work are:
% Jean-Francois Burnol <jfbu at free dot fr>
%     for the source code and English documentation, and
% Christine Roemer <Christine_Roemer at t-online dot de> and collaborators
%     for the translation into German of the documentation.
%
```

```
% This Work consists of the main source file etoc.dtx and the derived
% files etoc.sty, etoc.ins, etoc.tex, etoc-DE.tex, etoc.pdf, etoc-DE.pdf,
% etoc.dvi, etoc-DE.dvi, README.md.
%
% Running etex (or latex or pdflatex) on etoc.dtx extracts etoc.sty,
% etoc.ins, etoc.tex, etoc-DE.tex, and README.md. See README.md for
% further instructions.
```

Part I.
Overview

Here are some statistics for this part: it contains 5 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 "7". 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`,[1] 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 IV. 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 subsection 35.5, allows to mimick very well, if so desired, the standard looks.

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 III is devoted to this, and some more is to be found in Part VII.

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 40 visible tables of contents (and a few invisible ones) and uses only one `.toc` file![2] So

---

[1] http://ctan.org/pkg/enumitem

> one should think twice before adding manually extra commands to the `.toc` file (see section 13).

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 their scope limited in this way inside a group (LATEX's environments create groups).

Therefore the built-in "line styles" proposed by **etoc** as an example (and which are illustrated[3] 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 IV and the subsection 35.3 (which is a TOC!) have their line styles expressed in terms of enumerate 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

---

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

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

[4]under the `memoir` class, **etoc** knows `appendix` as a sectioning name.

has two arguments, the first one is executed before the TOC contents (typically it will print "Contents" and define suitable marks for the page headings) and the second is executed after the TOC contents.

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

These commands must be followed either by \tableofcontents or \localtableof-contents.

4.3. Compatibility mode

Both for the "line styles" and the "toc display style", it is possible to switch into a compatibility mode which uses the defaults from the document class.[5] This is activated by:

\etocstandardlines   % 'line entries' as without \usepackage{etoc}
\etocstandarddisplaystyle % 'toc display' as without \usepackage{etoc}

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

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

The command \etocdefaultlines resets the line styles as pre-defined internally by **etoc** and described in section 32.

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

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.

---

[5]for the "toc display style" **etoc** checks if it knows the class, and if not defaults to the article class layout. It will also check if tocloft has customized the TOC title.

[6]this is a shortcut for setting temporarily the tocdepth to -3, which has the effect to tell **etoc** not to print the TOC, and not even the heading.

We could use 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.

My first **etoc**: TOC of Part I (Overview)

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 subsection 35.5 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 LaTeX2e's \@dottedtocline macro does.

```
\begingroup\parindent 0pt \parfillskip 0pt \leftskip 0cm \rightskip 1cm
\etocsetstyle {section}
             {}
             {\leavevmode\leftskip 0cm\relax}
             {\bfseries\normalsize\makebox[.5cm][l]{\etocnumber.}%
              \etocname\nobreak\hfill\nobreak
              \rlap{\makebox[1cm]{\mdseries\etocpage}}\par}
             {}
\etocsetstyle {subsection}
             {}
             {\leavevmode\leftskip .5cm\relax }
             {\mdseries\normalsize\makebox[1cm][l]{\etocnumber}%
              \etocname\nobreak\hfill\nobreak
              \rlap{\makebox[1cm]{\etocpage}}\par}
             {}
\etocsetstyle {subsubsection}
             {}
             {\leavevmode\leftskip 1.5cm\relax }
             {\mdseries\normalsize\makebox[1cm][l]{\etocnumber}%
              \etocname\nobreak\hfill\nobreak
              \rlap{\makebox[1cm]{\etocpage}}\par}
```

```
              {}
\etocruledstyle[1]{\bfseries \Large My first \etoc: TOC of
  \autoref{part:overview} (\nameref{part:overview})}
\tableofcontents \ref {toc:overview}
\endgroup
```

6. A second example

This second example displays only the contents from Part IV and Part V. This selection is done via the technique of *depth tags*, described in section 12 and subsection 35.5. Its layout is a bit like the one of the main document TOC, although the line styles are coded very differently.

<div align="center">

CONTENTS

</div>

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}
\etocsettagdepth {arbitrarily}{none}
\etocsettagdepth {surprising} {none}
\etocsettagdepth {linestyles} {subsection}
\etocsettagdepth {globalcmds} {subsection}
\etocsettagdepth {custom}     {none}
\etocsettagdepth {tips}       {none}
\etocsettagdepth {etocandworld}{none}
\etocsettagdepth {code}       {none}

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

7. Linked list of the main package commands

\etocaftercontentshook   \etocframed   \etoclocalmulticol
\etocaftertitlehook   \etociffirst   \etoclocalruled
\etocaftertochook   \etocifnumbered   \etocmulticolstyle
\etocbeforetitlehook   \etocignoredepthtags   \etocmulticol
\etocdefaultlines   \etocignoretoctocdepth \etocname
\etocdepthtag.toc   \etocinline   \etocnumber
\etocdisplay   \etoclink   \etocobeydepthtags
\etocframedstyle   \etoclocalframed   \etocobeytoctocdepth

7. Linked list of the main package commands

```
\etocpage              \etocsettocdepth       \etocthelink
\etocruledstyle        \etocsettocstyle       \etocthename
\etocruled             \etocskipfirstprefix   \etocthenumber
\etocsetlevel          \etocstandarddisplaystyle \etocthepage
\etocsetnexttocdepth   \etocstandardlines     \etoctoccontentsline
\etocsetstyle          \etocthelinkedname     \etoctoclines
\etocsettagdepth       \etocthelinkednumber
\etocsettocdepth.toc   \etocthelinkedpage
```

Part II.
Arbitrarily many TOCs, and local ones too

Here are some statistics for this part: it contains 7 sections and 6 subsections. The name of the first section is "Labeling and reusing elsewhere" and the corresponding number is "8". The name of the last section is "Two Examples" and its number is "14". The name of the first subsection is "The hyperref option *bookmarksdepth*" and the corresponding number is "10.1". The name of the last subsection is "A TOC with a background color" and its number is "14.2".

8. Labeling and reusing elsewhere

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

   etoc allows the labeling of a TOC with (for example) \label{toc:A} and will re-display it elsewhere when told \tableofcontents\ref{toc:A}. The actual layout (title inclusive) used for the cloned TOC will be decided locally. The line styles and toc display style (including the title) will be the current ones and the current value of the tocdepth counter is obeyed. As an example the table of contents of Part VI 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}}}}
  \tableofcontents \label{toc:d} \ref{toc:c}
  \endgroup
\end{figure}
```

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

9. 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 LaTeX 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 III.

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

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

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

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

10. 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: 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 con-

---

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

tents. Thus, \invisibletableofcontents is essentially \etocsetnexttocdepth{none} \tableofcontents.

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

10.1. The hyperref option *bookmarksdepth*

When modifying the counter tocdepth for the purposes of multiple uses of \tableof-contents 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 \etocset-nexttocdepth has been added to the package, we have used it systematically and there was no need for bookmarksdepth=3 anymore.

11. The command \etocsettocdepth.toc

This command \etocsettocdepth.toc implements some functionality of tocvsec2[8], a package which however was incompatible with **etoc** (it can still be used for its secnumdepth-related commands, but its toc-related activities will get canceled by **etoc**) and more-or-less designed for a single table of contents.

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

The command \etocsettocdepth.toc, like \etocsettocdepth, accepts both numeric and named arguments. In the case of a named argument, the actual numeric value to be used is not yet decided at the time the .toc file is created; it will be the value currently specified for the named level at the time each table of contents (not having done \etocignoretoctocdepth) is typeset.

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

This ⟨gets executed in ALL tables of contents.⟩

---

[8] http://ctan.org/pkg/tocvsec2. I thank D.B. for drawing my attention to the incompatibility of this package with **etoc**.

11.1. The commands \etocobeytoctocdepth and \etocignoretoctocdepth

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

12. The commands \etocdepthtag.toc and \etocsettagdepth

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

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

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

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

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

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 subsection 35.5.

12.1. 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.

13. 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.

13.1. 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).

14. Two Examples

14.1. A Beautiful Thesis example

Here is another 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.

```
\begingroup % 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
```

My Beautiful Thesis

Chapter 3   Do I need to be a geek to use etoc?   6

14.2. A TOC with a background color

Let us now try out some more sophisticated line styles. The display will use the \etoc-framedstyle 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 the current page.

\tableofcontents \ref{toc:overview} (*cf.* subsection 14.2 and this other toc)

The actual design is not pre-built in **etoc**; it uses its '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. This design (with other colors) is in use also for this other toc, and the reader is referred to its subsection for the coding used.

Part III.
Surprising uses of etoc

Here are some statistics for this part: it contains 9 sections and 4 subsections. The name of the first section is "The TOC of TOCs" and the corresponding number is "15". The name of the last section is "Not displayed empty TOCs" and its number is "23". The name of the first subsection is "The commands \etocglobaldefs and \etoclocaldefs" and the corresponding number is "22.1". The name of the last subsection is "The \etocifwasempty command" and its number is "23.3".

15. The TOC of TOCs

Here is the numbered and linked list of all tables of contents which are displayed within this document:[9] 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 19, 18, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40. 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.

---

[9]The TOCs put in floats may change the order: the numbers are listed in the order the TOCs are typeset in the document; but the numbering itself is from the order of the TOCs in the *source* of this document...

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 \etoctoccontentsline{visibletoc}{\thevisibletoc}. This **etoc** command \etoctoccontentsline{⟨*level_name*⟩}{⟨*name*⟩} has the same effect as:

$$\text{\textbackslash addcontentsline\{toc\}\{⟨\textit{level\_name}⟩\}\{⟨\textit{name}⟩\}}$$

but its usefulness is to circumvent[10] 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').[11]

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}\etoctoccontentsline{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)
```

---

[10] using \addtocontents rather than \addcontentsline

[11] The package provides a starred variant \etoctoccontentsline*, 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.

```
    \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
```

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

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 subsection 35.2 gives another use of the shuffling of levels.

16. Arbitrary "Lists Of...", `\etoctoccontentsline`

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 exercice number will be recoverable via \etocname: v--here--v
\newcommand*{\exercisetotoc}{\etoctoccontentsline{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}
   {\etoctoccontentsline{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 `\etoctoccontentsline{..}{..}` 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 ⟨*n*⟩, one can use (here with ⟨*n*⟩=2) the starred variant `\etoctoccontentsline*` which has an additional argument:

```
\newcommand{\exercisetotoc}{\etoctoccontentsline*{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\etoctoclineleaders\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
```

17. A TOC with a fancy layout

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

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

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

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

Another technique for doing this kind of things is explained in .

TABLE OF CONTENTS

PART I

<div style="text-align:center">

PART II
Government.

</div>

18. Another compatibility mode

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

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

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

Here is how to proceed. One puts in the preamble:

```
\makeatletter
\newcommand{\MyStandardTOC}{%
  \begingroup
```

```
  \let\savedpartline\l@part
  \let\savedchapterline\l@chapter  %% remove if article/scrartcl class
  \let\savedsectionline\l@section
  \let\savedsubsectionline\l@subsection
  % and so on if \subsubsection, etc... is used
  %
  % for the book or article classes:
  \etocsetstyle{part}{}{}
    {\savedpartline{\etocnumber\hspace{1em}\etocname}{\etocpage}}{}%
  % for the scrbook or scrartcl classes:
  \etocsetstyle{part}{}{}
    {\savedpartline{\numberline{\etocnumber}\etocname}{\etocpage}}{}%
  % identical in book/article/scrbook/scrartcl classes:
  \etocsetstyle{chapter}{}{}   %%% only for book and scrbook
    {\savedchapterline{\numberline{\etocnumber}\etocname}{\etocpage}}{}%
  \etocsetstyle{section}{}{}
    {\savedsectionline{\numberline{\etocnumber}\etocname}{\etocpage}}{}%
  \etocsetstyle{subsection}{}{}
    {\savedsubsectionline{\numberline{\etocnumber}\etocname}{\etocpage}}{}%
  % etc... if further sectioning units are used
  % (see the text for what to do with the memoir class)
  \etocstandarddisplaystyle % this is for the title, page-marks, etc...
  \tableofcontents
  \endgroup}
\makeatother
```

Of course 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.[12] 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`.[13] 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 \etoc-thename 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

---

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

[13]For the `memoir` class, one needs a bit more: each of the command \booknumberline, \partnumber-line and \chapternumberline will have to be saved with a \let, and, one then specifies:

```
        \etocsetstyle{chapter}{}{}{\savedchapterline{\savedchapternumberline
                          {\etocnumber}\etocname}{\etocpage}}{}
```

(and analogously for `part`, respectively `book`).

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

```
\etocsetstyle{section}{}
  {\ifnum\etocthenumber=4
      \etocsetstyle{subsection}
        {\par\nopagebreak\begingroup
         \leftskip1.5em \rightskip\@tocrmarg \parfillskip\@flushglue
         \parindent 0pt \normalfont\normalsize\rmfamily\itshape
         % \columnsep1em
         % \begin{minipage}{\dimexpr\linewidth-\leftskip-\rightskip\relax}%
         % \begin{multicols}{2}%
         \etocskipfirstprefix}
        {\allowbreak\,--\,}
        {\etocname\ \textup{(\etocnumber)}}
        {.\par\endgroup}%
        % {.\par\end{multicols}\end{minipage}\par\endgroup}%
   \else
     \etocsetstyle{subsection}
       {}{}
       {\savedsubsectionline{\numberline{\etocnumber}\etocname}{\etocpage}}
       {}%
   \fi}
  {\savedsectionline{\numberline{{\color{cyan}\etocthenumber}}\etocname}{\etocpage}}
  {}%
```

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

19. The TOC as a tree

Using tikz[14] and the package forest[15] we shall display the table of contents of this part
as a tree. The technique is to use the **etoc** modified command \tableofcontents not
for typesetting, but to prepare a macro, or rather here a *token list* variable, with all the
instructions to be executed later. LESLIE LAMPORT's book has no mention whatsoever of

---

[14]http://ctan.org/pkg/pgf
[15]http://ctan.org/pkg/forest

token lists, and LATEX 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.[16] The commands \etocname, etc..., are robust, it is easier to work with \etocthelinkednumber, \etocthelinkedname, and \etocthe-linkedpage which contain the same information in an easier accessible form.

For this forest tree we have designed very special **etoc** styles for sections and subsections. They use a token list register called \treetok and a macro \appendtotok whose rôle is to append to a given token list variable the contents of a macro given as second argument. All this will happen in reaction to a \tableofcontents command, but *nothing* has yet been printed in the process.[17] 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 following page. Here is the code used for its production:

```
\newtoks\treetok
\newtoks\tmptok

\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}}}%
}

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

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

---

[16]This manual up to the version of May 14, 2013 used package tikz-qtree 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).

[17]There is always a \par, which here is not a problem, but can be suppressed if need be via the command \etocinline or its synonym \etocnopar.

Do I need to be a geek to use etoc?

Limitations in the use of list environments for tables of contents

Line styles and toc display style

\etocsetstyle for the line styles

\etocsettocstyle for the toc display

Compatibility mode

Overview

A first example

A second example

Linked list of the main package commands
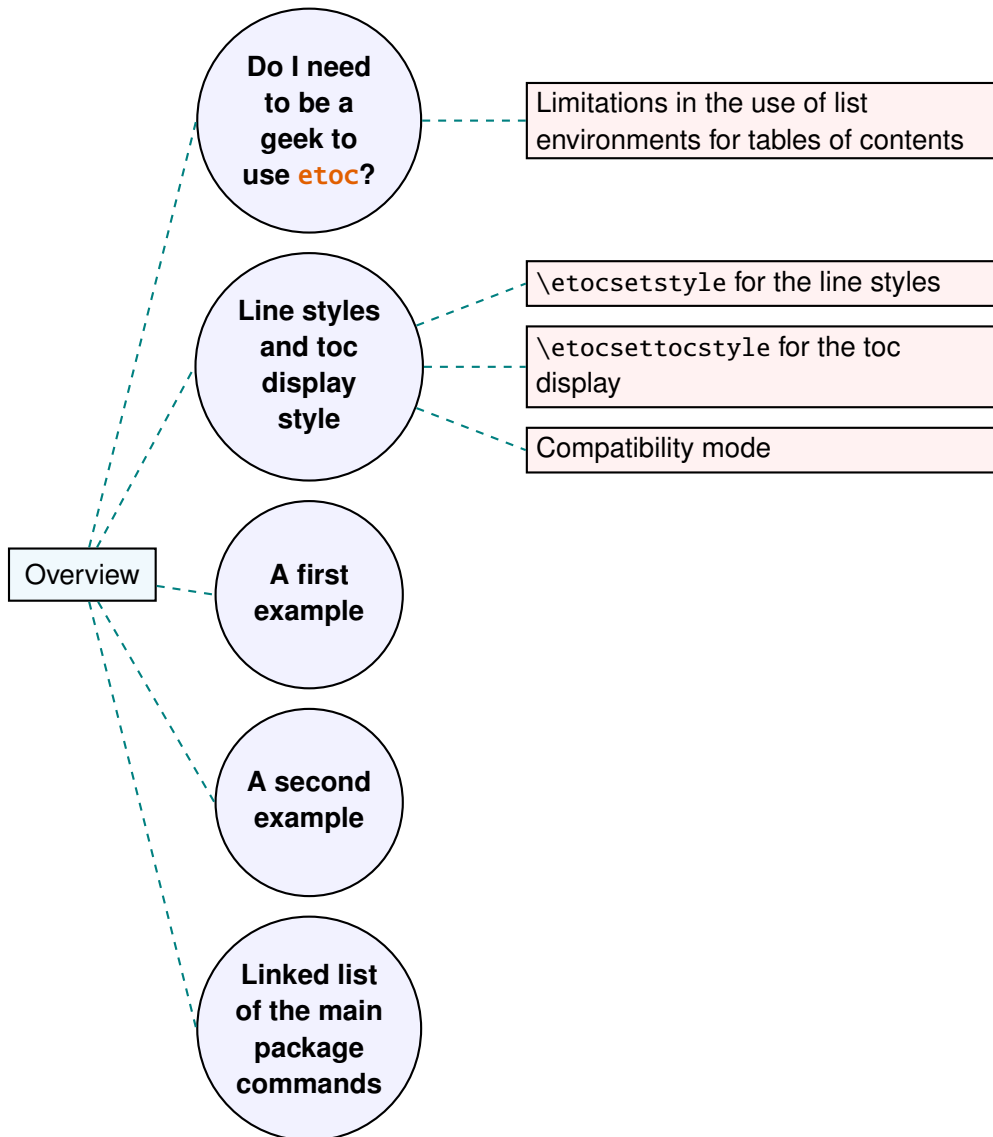
A table of contents displayed as a forest tree.

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

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

\begin{figure}[th!]\centering
    \etocsetnexttocdepth{subsection}
    \tableofcontents \label{toc:forest}\ref{toc:overview}
```

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

Why `\hypersetup{hidelinks}`? as explained in subsection 13.1, I prefer the links in TOCs not to be colorized, 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 subsection 35.4) 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 subsection 35.6); 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.

20. The TOC as a molecule

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

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

```
% \newtoks\treetok % already done earlier
\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{ ;}}

% moved to the inside of the figure environment for correct \pageref{toc:molecule}
% \etocsetnexttocdepth{subsection}
% \tableofcontents \label{toc:molecule} \ref{toc:overview}
```

The \tableofcontents command appears just above the tikzpicture environment in a figure float (for the label of the table of contents to certainly refer to the same page as the one where the picture will actually be printed). We thus get the table of contents as a "molecule", which appears on the next page. Here is the code actually producing it:

```
\begin{figure}[th!]\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}
\end{figure}
```

This TikZ TOC is fully hyperlinked, like the previous Forest TOC.
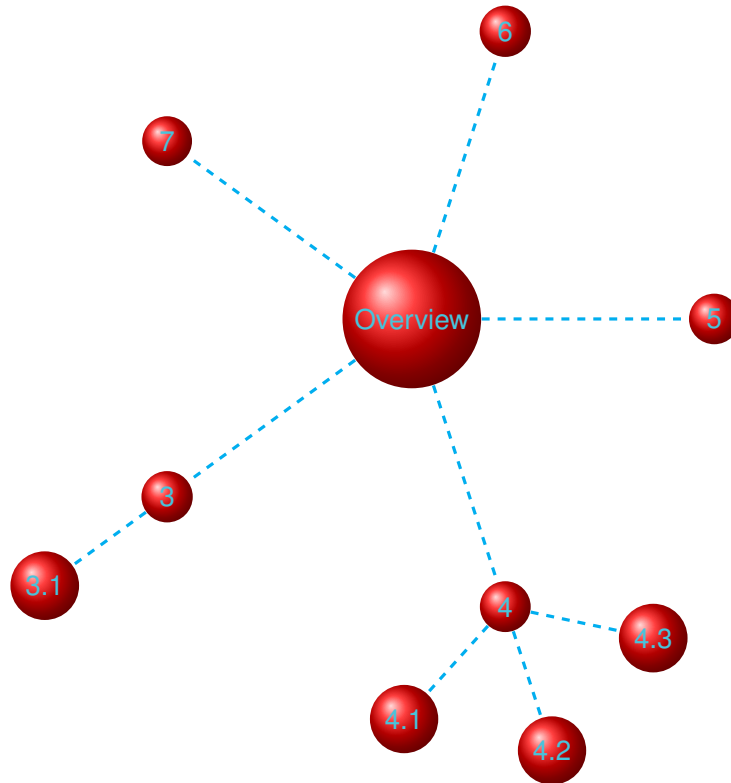
A table of contents as a "molecule."
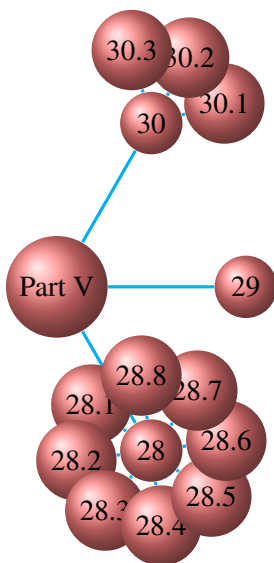
On the side, the (fully hyperlinked) table of contents of Part V.

```
\etocsettocstyle
  {\treetok{\node {\autoref{part:globalcmds}}}}
  {\global\appendtotok\treetok{ ;}}
\etocsetnexttocdepth {subsection}
\tableofcontents \ref{toc:globalcmds}
\noindent
\parbox{4cm}{\hypersetup{hidelinks}%
   \begin{tikzpicture}
      [grow cyclic,
       level 1/.style={level distance=2.5cm,sibling angle=60},
       level 2/.style={level distance=1cm,sibling angle=45},
       every node/.style={ball color=red!50,circle,text=black},
       edge from parent path={[very thick,color=cyan]
                (\tikzparentnode) --(\tikzchildnode)}]
      \the\treetok
   \end{tikzpicture}}%
...
```

21. **The TOC as a TikZ mind map**

This is in the same spirit as the "molecule" example. The use of the ε-TEX primitive \un-expanded will simplify the code.[18]

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 facing page.

```
\begingroup
\newtoks\treetok
\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}}

\renewcommand*{\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}%
```

---

[18]The "molecule" example was added to this documentation on 2013/03/03. The "mindmap" example was motivated on 2015/03/11 by http://tex.stackexchange.com/a/232584/4686. Further help was then obtained via http://tex.stackexchange.com/q/232816/4686 and this led to the title page which is a further example.

The **etoc documentation**

- 34 One more example of colored TOC layout
- 33 Customizing **etoc**
- 32 The package default line styles: \etocdefault-lines
- 31 Summary of the main style commands
- VI. Using and customizing **etoc**
- VII. Tips
- VIII. **etoc** and the outside world
- 39 Timestamp
- 40 Change history
- 41 Implementation
- IX. The code
- V. Commands for the toc display style
- I. Overview
- IV. Commands for the toc line styles
- II. Arbitrarily many TOCs, and local ones too
- 23 Not displayed empty TOCs
- 22 The TOC as a table
- 15 The TOC of TOCs
- III. Surprising uses of **etoc**
- 21 The TOC as a TikZ mind map
- 16 Arbitrary "Lists Of...", \etoctoc-contentsline
- 17 A TOC with a fancy layout
- 20 The TOC as a molecule
- 18 Another compatibility mode
- 19 The TOC as a tree

```
% 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 {surprising} {section}
\etocsettagdepth {linestyles} {part}
\etocsettagdepth {globalcmds} {part}
\etocsettagdepth {custom}      {section}
\etocsettagdepth {tips}        {part}
\etocsettagdepth {etocandworld}{part}
\etocsettagdepth {code}        {section}

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

\begin{figure}[thp!]
\tableofcontents\label{toc:mindmap}%
```

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

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

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

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

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

for an elaboration of this.

22. The TOC as a table

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

1. it is mandatory to issue \etocglobaldefs for **etoc**'s definitions to have global scope,

2. it is impossible to start one of the ⟨*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 that at the end with the idea to close the *current* row; and when the TOC is a partial one (a \localtableofcontents) this is (in almost all situations) mandatory.

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

CONTENTS		
	Abstract	4
1	Foreword	5
2	License	5
I	**Overview**	6
3	Do I need to be a geek to use **etoc**?	6
	3.1 *Limitations in the use of list environments for tables of contents*	7
4	Line styles and toc display style	7
	4.1 `\etocsetstyle` *for the line styles*	7
	4.2 `\etocsettocstyle` *for the toc display*	7
	4.3 *Compatibility mode*	8
5	A first example	8
6	A second example	10
7	Linked list of the main package commands	11
II	**Arbitrarily many TOCs, and local ones too**	13
8	Labeling and reusing elsewhere	13
9	A powerful functionality of **etoc**: the reassignment of levels with `\etocsetlevel`	14
10	The `\etocsettocdepth` and `\etocsetnext-tocdepth` commands	15
	10.1 *The hyperref option* bookmarksdepth	16
11	The command `\etocsettocdepth.toc`	16
	11.1 *The commands* `\etocobeytoc-tocdepth` *and* `\etocignoretoc-tocdepth`	17
12	The commands `\etocdepthtag.toc` and `\etocsettagdepth`	17
	12.1 *The commands* `\etocobeydepth-tags` *and* `\etocignoredepthtags`	17
13	Adding commands to the `.toc` file	18
	13.1 *The hyperref option* hidelinks	18
14	Two Examples	19
	14.1 *A Beautiful Thesis example*	19
	14.2 *A TOC with a background color*	20
III	**Surprising uses of etoc**	22
15	The TOC of TOCs	22
16	Arbitrary "Lists Of...", `\etoctoccon-tentsline`	24
17	A TOC with a fancy layout	25
18	Another compatibility mode	26
19	The TOC as a tree	28

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

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

```
    {}

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

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

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

\etocsetnexttocdepth {subsection}

\tableofcontents
\endgroup
```

Examining the code above the reader will wonder why the \\ 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 \\ 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).

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

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

For the hardliner's old way of obtaining the exact same result, see .

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 **etoc**?	3	6
Line styles and toc display style	4	7
A first example	5	8
A second example	6	10
Linked list of the main package commands	7	11

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

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

\etocglobaldefs % MANDATORY !!
\etocsetnexttocdepth{1}

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

22.1. 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.

By default **etoc**'s definitions of \etocname etc... are local. This causes problems in certain contexts such as seen here with TOC as tables, but also with enumitem *inline* variants of its standard environments, because the command \item then closes a group.

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.

23. Not displayed empty TOCs

23.1. The \etocchecksemptiness command

The user needs to issue \etocchecksemptiness to tell **etoc** to check whether local tables of contents are empty and in case of emptiness to print nothing at all.[19] 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 13 in this context.

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.

23.2. The \etocnotocifnotoc command

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

23.3. 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 \etocchecksemptiness 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).

---

[19]Thanks to Paul Gaborit who asked for such a feature.

Part IV.
Commands for the toc line styles

Here are some statistics for this part: it contains 4 sections and 9 subsections. The name of the first section is "The \etocsetstyle command" and the corresponding number is "24". The name of the last section is "Am I also red?" and its number is "27". The name of the first subsection is "The \etocname and \etocpage commands" and the corresponding number is "24.1". The name of the last subsection is "This is a (pale) red subsection for illustrative purposes" and its number is "26.2".

24. The \etocsetstyle command

24.1. The \etocname and \etocpage commands

Let us explain how **etoc** was used to produce the table of contents displayed at the beginning of this second part. This is a local table of contents, and we used the command \localtableofcontents.

We shall distinguish between the *line styles* and the *toc display style*. The line styles were (essentially) obtained in the following manner:[20]

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

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

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

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

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

Each `\etocsetstyle` command has five mandatory arguments:

$$\texttt{\textbackslash etocsetstyle}\{\langle\textit{levelname}\rangle\}\{\langle\textit{start}\rangle\}\{\langle\textit{prefix}\rangle\}\{\langle\textit{contents}\rangle\}\{\langle\textit{finish}\rangle\}$$

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

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

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

---

[20] the present document has \renewcommand{\familydefault}{\sfdefault} in its preamble, hence \normalfont switches to the sans typeface; so in the section line-style, I wrote \rmfamily instead.

24.2. The **\etocskipfirstprefix** and **\etociffirst** commands

The chosen subsubsection style made use of the command \etocskipfirstprefix, which instructs **etoc** to *not* use for the first item the specified ⟨*prefix*⟩ code.[21]

   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.

24.3. 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.

24.4. 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).[22]

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

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

\etocsetstyle{subsection}
{\begin{enumerate}[leftmargin=0cm, label=\etocnumber]}
{\normalfont \item}
{\etocname{} (p.~\etocpage)}
{\end{enumerate}}
```

---

[21]With versions earlier than 1.08 it had to be the very last token in the ⟨*start*⟩ code. It may now appear anywhere therein.

[22]as seen we use \fcolorbox rather than \fbox. Due to some redefinition made by package xcolor, had we used \fbox (and not used hyperref) we would have needed \protect\fbox.

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.

24.5. 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.

24.6. The \etoclink command

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

The command \etoclink is robust.

24.7. The \etocthelinkedname, \etocthelinkednumber, \etocthelinkedpage and \etocthelink commands

This is for advanced uses by advanced users. The \etocthename macro has been mentioned before; using it in instructions such as \global\let\lastone\etocthename in an **etoc** line style will define the macro \lastone to expand to the last name seen at the corresponding level. But no facilities was previously available to do the same with the link data.

The package provides since 1.07f 2013/03/07 \etocthelinkedname to do the similar thing, with the link data included. It was used in this documentation when doing the examples with trees.

Also provided with the similar goal: \etocthelinkednumber, \etocthelinkedpage (which contains a link only if hyperref added one to the page number) and \etocthelink{⟨*linkname*⟩} which allows to make a link with an arbitrary name.

All these commands are fragile. In fact starting with 1.08a, \etocname is exactly the robust variant of \etocthelinkedname.

Obsolete example of a table of contents done as an inline enumitem environment: **28.** Specifying the toc display style (*p. 51*); **29.** Starred variants of the \tableofcontents etc... commands (*p. 55*), and **30.** Table of contents for this part (*p. 55*).

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

etoc 1.08a offers a simpler way to the same result. And there is no need anymore here for \etocthelinkedname, \etocname works.

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

Example of a table of contents done as an inline enumitem environment: **28.** Specifying the toc display style (*p. 51*); **29.** Starred variants of the \tableofcontents etc... commands (*p. 55*), und **30.** Table of contents for this part (*p. 55*).

25. The \etocsetlevel command

As already explained in Part III, one can inform etoc of a level to associate to a given sectioning command with \etocsetlevel. For example:

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

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

However, if no table of contents is typeset in compatibility mode, then all that matters is that the various line styles have been set. If, for example section is at level 1, then there is no need to do some \etocsetstyle{molecule}{..}{..}{..}{..} after

`\etocsetlevel{molecule}{1}` if `\etocsetstyle{section}{..}{..}{..}{..}` has already been done (and it has been done by the package itself in its definition of its own line styles).

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

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

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

These level assignments can be modified at anytime: see Part III 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 \etoctoclines could have been used, too.
\etocmarkboth\contentsname
\etocmulticolstyle[1]                    % one-column display
   {\pdfbookmark[1]{Table of contents}{MAINTOC}% create a bookmark in the pdf
    \noindent\bfseries\Large
    \leaders\hrule height1pt\hfill
    \MakeUppercase{Table of Contents}}
\begingroup                              % use a group to limit the scope of the
  \etocsetlevel{subsection}{3}           %   subsection level change.
  \etocsetlevel{subsubsection}{4}        % anything > tocdepth=3.
  \tableofcontents \label{toc:main}
\endgroup
```

In this way, the subsections used the style originally designed for subsubsections, the subsubsections were not printed. Without this modification, the appearance would have been very different: the package line styles were targeted to be employed in documents with many many sub-sub-sections, in a two-column layout, giving thus a more compact output that what is achieved by the default LaTeX 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.

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

26.1. 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
```

26.2. 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.

27. 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 V.
Commands for the toc display style

Here are some statistics for this part: it contains 3 sections and 11 subsections. The name of the first section is "Specifying the toc display style" and the corresponding number is "28". The name of the last section is "Table of contents for this part" and its number is "30". The name of the first subsection is "The command `\etocsettocstyle`" and the corresponding number is "28.1". The name of the last subsection is "A (crazy) inline display" and its number is "30.3".

28. 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})
```

28.1. The command `\etocsettocstyle`

This is a command with two mandatory arguments:

$$\texttt{\textcolor{blue}{\etocsettocstyle}}\{\langle\textit{before\_toc}\rangle\}\{\langle\textit{after\_toc}\rangle\}$$

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.

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

28.2. 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*⟩}.

28.3. The command \etoctocstyle

\etoctocstyle[⟨*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.

28.3.1. The command \etoctocstylewithmarks

\etoctocstylewithmarks[⟨*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 \etoctocstylewithmarksnouc which does not uppercase.

28.3.1.1. Do we really want paragraph entries in the TOC?

28.3.1.2. really?

28.4. 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 \tableofcontents, all inside a group so that future table of contents will not be affected, there is:

\etocruled[⟨*number_of_columns*⟩]{⟨*title*⟩}

And the local form will be \etoclocalruled.

28.5. 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 33.2).

A minipage is used, hence the produced table of contents isn't compatible with a page break. For allowing page breaks, use of the commands of mdframed[23] or tcolorbox[24] in the arguments of \etocsettocstyle is recommended.

Examples in this document are on pages 14, 20, 57, and 68.

---

[23] http://ctan.org/pkg/mdframed
[24] http://ctan.org/pkg/tcolorbox

28.6. 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`[25] at the end of the heading, then does a vertical skip of value `\etocinnertopsep`. The command `\etocoldpar` can also be used explicitely 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`.

28.7. 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.

28.7.1. 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.

28.8. The command `\etocinline`

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

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

---

[25]this command `\etocoldpar` (= working `\par` in the argument to `\etocmulticolstyle`) is not related to the switch `\etocinline` whose purpose is to tell **etoc** not to do a `\par` before the table of contents.

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

The `\tableofcontents`, `\localtableofcontents`, `\etocmulticol`, and all their cousins and variants have starred variants (the star must be before the other arguments). For all but the `memoir` class, they are like the original. For the `memoir` class, the original prints an entry in the `.toc` file, as is the usage for the original `\tableofcontents` command in that class, whereas the starred variants do not, as is the habit in that class.

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

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

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

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

There is `\etocaftercontentshook`, similarly defined to do nothing. And `\etocbeforetitlehook` is also initially defined to do nothing. They can be used for some special effects.

30. Table of contents for this part

30.1. Testing the compatibility mode

As a third example we now print the local table of contents for this part. First we will test the compatibility mode.[26] The original was invisibly defined with a label at the beginning of this Part V.

```
\KOMAoptions{toc=left}
\etocstandarddisplaystyle % necessary for the display to obey toc=left
\etocstandardlines
\tableofcontents \ref{toc:globalcmds}
```

Contents

---

[26]the present document uses the `scrartcl` class, and we check here that the **etoc** compatibility mode does respect the customizing done via the class commands.

30.2. A framed display

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

```
\etocdefaultlines
\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{\etoctoprulecolorcmd}{\color{red!25}}
\renewcommand{\etocbottomrule}{\hrule height 1pt}
\renewcommand{\etocbottomrulecolorcmd}{\color{red!25}}
% set up the left and right rules
\renewcommand{\etocleftrule}{\vrule width 5pt}
\renewcommand{\etocrightrule}{\vrule width 5pt}
\renewcommand{\etocleftrulecolorcmd}{\color{red!25}}
\renewcommand{\etocrightrulecolorcmd}{\color{red!25}}
% use \fcolorbox to set up a colored frame for the title
\fboxrule1pt
\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`

> *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`.*

30.3. A (crazy) inline display

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

Here is the inline table of contents. *Abstract, Foreword, License*. **Overview:** *Do I need to be a geek to use* `etoc`*?*[27], *Line styles and toc display style*[28], *A first example, A second example, Linked list of the main package commands*. **Arbitrarily many TOCs, and local ones too:** *Labeling and reusing elsewhere, A powerful functionality of* `etoc`*: the re-assignment of levels with* `\etocsetlevel`*, The* `\etocsettocdepth` *and* `\etocset-nexttocdepth` *commands*[29], *The command* `\etocsettocdepth.toc`[30], *The commands* `\etocdepthtag.toc` *and* `\etocsettagdepth`[31], *Adding commands to the* `.toc` *file*[32], *Two Examples*[33]. **Surprising uses of** `etoc`**:** *The TOC of TOCs, Arbitrary "Lists Of...",* `\etoc-toccontentsline`*, A TOC with a fancy layout, Another compatibility mode, The TOC as*

---

[27] Limitations in the use of list environments for tables of contents.

[28] `\etocsetstyle` for the line styles; `\etocsettocstyle` for the toc display; Compatibility mode.

[29] The hyperref option *bookmarksdepth*.

[30] The commands `\etocobeytoctocdepth` and `\etocignoretoctocdepth`.

[31] The commands `\etocobeydepthtags` and `\etocignoredepthtags`.

[32] The hyperref option *hidelinks*.

[33] A Beautiful Thesis example; A TOC with a background color.

a tree, The TOC as a molecule, The TOC as a TikZ mind map, The TOC as a table[34], Not displayed empty TOCs[35]. **Commands for the toc line styles:** *The* `\etocsetstyle` *command[36], The* `\etocsetlevel` *command, Scope of commands added to the* `.toc` *file[37], Am I also red?.* **Commands for the toc display style:** *Specifying the toc display style[38], Starred variants of the* `\tableofcontents` *etc... commands, Table of contents for this part[39].* **Using and customizing** **etoc:** *Summary of the main style commands[40], The package default line styles:* `\etocdefaultlines`*, Customizing* **etoc**[41]*, One more example of colored TOC layout.* **Tips:** *... and tricks[42].* **etoc** **and the outside world:** *Compatibility with other packages, T_EXnical matters, Errors and catastrophes.* **The code:** *Timestamp, Change history, Implementation.*

The code used:

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

---

[34]The commands `\etocglobaldefs` and `\etoclocaldefs`.

[35]The `\etocchecksemptiness` command; The `\etocnotocifnotoc` command; The `\etocifwasempty` command.

[36]The `\etocname` and `\etocpage` commands; The `\etocskipfirstprefix` and `\etociffirst` commands; The `\etocnumber` command; The `\etocifnumbered` switch; The `\etocthename`, `\etocthenumber`, and `\etocthepage` commands; The `\etoclink` command; The `\etocthelinkedname`, `\etocthelinkednumber`, `\etocthelinkedpage` and `\etocthelink` commands.

[37]Testing the scope; This is a (pale) red subsection for illustrative purposes.

[38]The command `\etocsettocstyle`; The commands `\etocmulticolstyle`, `\etocmulticol`, and `\etoclocalmulticol`; The command `\etoctocstyle`; The commands `\etocruledstyle`, `\etocruled` and `\etoclocalruled`; The commands `\etocframedstyle`, `\etocframed`, and `\etoclocalframed`; Headings, titles, `\etocoldpar`, `\etocinnertopsep`; The compatibility mode `\etocstandarddisplaystyle`; The command `\etocinline`.

[39]Testing the compatibility mode; A framed display; A (crazy) inline display.

[40]Setting up local styles; Setting up toc display styles; Displaying tables of contents; Labels and references.

[41]Customizing the **etoc** pre-defined line styles; Customizing the toc display styles.

[42]Hacking framed parboxes; Interverting the levels; All subsections of this document; Displaying statistics; Using depth tags; Typesetting the TOC as a table (the old way).

Part VI.
Using and customizing `etoc`

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 "31". The name of the last section is "One more example of colored TOC layout" and its number is "34". The name of the first subsection is "Setting up local styles" and the corresponding number is "31.1". The name of the last subsection is "Customizing the toc display styles" and its number is "33.2".

31. Summary of the main style commands

31.1. Setting up local styles

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

31.2. Setting up toc display styles

```
\etocmulticolstyle[⟨number_of_columns⟩]{⟨heading⟩}
\etoctocstyle[⟨kind⟩]{⟨number_of_columns⟩}{⟨title⟩}
\etoctocstylewithmarks[⟨kind⟩]{⟨number_of_columns⟩}{⟨title⟩}{⟨mark⟩}
\etoctocstylewithmarksnouc[⟨kind⟩]{⟨number_of_columns⟩}{⟨title⟩}{⟨mark⟩}
\etocruledstyle[⟨number_of_columns⟩]{⟨title⟩}
\etocframedstyle[⟨number_of_columns⟩]{⟨title⟩}
\etocsettocstyle{⟨before_toc⟩}{⟨after_toc⟩}
```

31.3. 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
```

31.4. Labels and references

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

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

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

The commands for local tables of contents do *not* react to a `\ref` following them.

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.

32. 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 `\etoctoclines` if the line styles have not been modified with `\etocsetstyle`. Sections and sub-sections are printed in essentially the same manner, except that the leading for sub-sections is a bit smaller (with document classes lacking a `\chapter` command, the sections are printed in bold typeface; this is the case in the present document). Sub-sub-sections are printed inline, in one paragraph, with no numbers or page numbers. This style was designed and tested with documents having lots of sub-sub-sections, and should be used on a two-column layout: it

provides (only in that situation with many sub-sub-sections) a more compact presentation than what is achieved by the LATEX default.[43] 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*{\etoctoclineleaders}
        {\hbox{\normalfont\normalsize\hbox to 1ex {\hss.\hss}}}
\sloppy
\tableofcontents
\endgroup
```

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

Abstract

Part I.   Overview

1. Foreword

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

2. License

---

[43] and there will never be a Part or Chapter entry alone at the bottom of a column or page (except if it has no sub-unit).

4. Line styles and toc display style

5. A first example

6. A second example

7. Linked list of the main package commands

Part II. Arbitrarily many TOCs, and local ones too

8. Labeling and reusing elsewhere

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

10. The \etocsettocdepth and \etocsetnexttocdepth commands

11. The command \etocsettocdepth.toc

12. The commands \etocdepthtag.toc and \etocsettagdepth

13. Adding commands to the .toc file

14. Two Examples

Part III. Surprising uses of etoc

15. The TOC of TOCs

16. Arbitrary "Lists Of...", \etoctoccontentsline

17. A TOC with a fancy layout

18. Another compatibility mode

19. The TOC as a tree

20. The TOC as a molecule

21. The TOC as a TikZ mind map

22. The TOC as a table

23. Not displayed empty TOCs

Part IV. Commands for the toc line styles

24. The \etocsetstyle command

25. The \etocsetlevel command

26. Scope of commands added to the .toc file

27. Am I also red?

Part V.   Commands for the toc display style

28. Specifying the toc display style

29. Starred variants of the \table-ofcontents etc... commands

30. Table of contents for this part

Part VI.   Using and customizing etoc

31. Summary of the main style commands

32. The package default line styles: \etocdefaultlines

33. Customizing etoc

34. One more example of colored TOC layout

33. Customizing etoc

33.1. Customizing the etoc pre-defined line styles

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

```
\newcommand*\etocfontminustwo{\normalfont \LARGE \bfseries}
\newcommand*\etocfontminusone{\normalfont \large \bfseries}
\newcommand*\etocfontzero{\normalfont \large \bfseries}
\newcommand*\etocfontone{\normalfont \normalsize \bfseries}
\newcommand*\etocfonttwo{\normalfont \normalsize}
\newcommand*\etocfontthree{\normalfont \footnotesize}

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

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

\newcommand*\etocbaselinespreadminustwo{1}
\newcommand*\etocbaselinespreadminusone{1}
\newcommand*\etocbaselinespreadzero{1}
\newcommand*\etocbaselinespreadone{1}
\newcommand*\etocbaselinespreadtwo{1}
\newcommand*\etocbaselinespreadthree{.9}
\newcommand*\etoctoclineleaders
```

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

No customizing of the standard line styles is possible from within **etoc**. As already explained, when \etocstandardlines has been issued, the package just makes itself very discrete and acts only at the global level, and the TOC entries are (hopefully) formatted as would have happened in the absence of **etoc**.[44]

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 LaTeX 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.

33.2. Customizing the toc display styles

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

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

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

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

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

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

---

[44]with the KOMA-script classes, we noticed that \etocstandarddisplaystyle was apparently needed for the KOMA options toc=left to be active at the level of the line entries.

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

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

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

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

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

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

The top and bottom rules do not have to be rules and can be horizontal *leaders* (of a specified height) in the general 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.

34. One more example of colored TOC layout

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

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

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

The TOC has been put in a [float](#) which appears on the following page. The coding is a bit involved[45] as it does not use any additional package. Also, it was written at some early stage and I have not revised it since.

A better solution would be to use some package to set up a background color possibly extending accross pages, as the framed style (which we used to get this background color)

---

[45] and reveals the author's preference for the T_EX syntax...

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.

Part VII.
Tips

Here are some statistics for this part: it contains 1 section and 6 subsections. The name of the first section is "... and tricks" and the corresponding number is "35". The name of the last section is "... and tricks" and its number is "35". The name of the first subsection is "Hacking framed parboxes" and the corresponding number is "35.1". The name of the last subsection is "Typesetting the TOC as a table (the old way)" and its number is "35.6".

35. ... and tricks

35.1. Hacking framed parboxes

```
\etocdefaultlines
\renewcommand\etoctoprule{\hrule height 2pt depth 2pt}
\etocruled{\color{green}\fboxrule2pt\fboxsep1ex
            \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}
```


Specifying the toc display style . . . . **28, p. 51**

The command \etocsettocstyle . . . . 28.1, p. 51

The commands \etocmulticolstyle, \etocmulticol, and \etoclocalmulticol . . . . 28.2, p. 52

The command \etoctocstyle . . . . 28.3, p. 52

The commands \etocruledstyle, \etocruled and \etoclocalruled . . . . 28.4, p. 53

The commands \etocframedstyle, \etocframed, and \etoclocalframed . . . . 28.5, p. 53

Headings, titles, \etocoldpar, \etocinnertopsep . . . . 28.6, p. 54

The compatibility mode \etocstandarddisplaystyle . . . . . 28.7, p. 54

The command \etocinline . . . . 28.8, p. 54

Starred variants of the \tableofcontents etc... commands . . **29, p. 55**

35.2. Interverting the levels

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

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

35.3. All subsections of this document

i. Limitations in the use of list environments for tables of contents (3.1, p. 7)

ii. \etocsetstyle for the line styles (4.1, p. 7)

iii. \etocsettocstyle for the toc display (4.2, p. 7)

iv. Compatibility mode (4.3, p. 8)

v. The hyperref option *bookmarksdepth* (10.1, p. 16)

vi. The commands \etoc-obeytoctocdepth and \eto-cignoretoctocdepth (11.1, p. 17)

vii. The commands \etoc-obeydepthtags and \etoc-ignoredepthtags (12.1, p. 17)

viii. The hyperref option *hidelinks* (13.1, p. 18)

ix. A Beautiful Thesis example (14.1, p. 19)

x. A TOC with a background color (14.2, p. 20)

xi. The commands \etoc-globaldefs and \etoclo-caldefs (22.1, p. 42)

xii. The \etocchecks-emptiness command (23.1, p. 43)

xiii. The \etocnotocif-notoc command (23.2, p. 43)

xiv. The \etocifwasempty command (23.3, p. 43)

xv. The \etocname and \etocpage commands (24.1, p. 44)

xvi. The \etocskipfirst-prefix and \etociffirst commands (24.2, p. 46)

xvii. The \etocnumber command (24.3, p. 46)

xviii. The \etocifnum-bered switch (24.4, p. 46)

xix. The \etocthename, \etocthenumber, and \etoc-thepage commands (24.5, p. 47)

xx. The \etoclink command (24.6, p. 47)

xxi. The \etocthelinked-name, \etocthelinked-number, \etocthelinked-page and \etocthelink commands (24.7, p. 47)

xxii. Testing the scope (26.1, p. 49)

xxiii. This is a (pale) red subsection for illustrative purposes (26.2, p. 50)

xxiv. The command \etoc-settocstyle (28.1, p. 51)

xxv. The commands \etoc-multicolstyle, \etoc-multicol, and \etoclo-calmulticol (28.2, p. 52)

xxvi. The command \etoc-tocstyle (28.3, p. 52)

35.4. 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 (using the TeX primitive \setbox rather than the LaTeX \sbox would allow to change the color of the un-boxed saved box). 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 20. 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}{\etocname}%
        \sbox{\firstnumberi}{\etocnumber}%
        \def\thispartstatsauxi{}}%
  \def\thispartstatsauxii{%
        \sbox{\firstnameii}{\etocname}%
        \sbox{\firstnumberii}{\etocnumber}%
        \def\thispartstatsauxii{}}%
  \begingroup
  \etocsetstyle{subsection} {} {}
    {\thispartstatsauxii
```

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

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

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

35.5. 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 V 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 LaTeX2e'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:

```
\etocsetnexttocdepth {all}
\begingroup
\parindent 0pt \leftskip 0cm \rightskip .75cm \parfillskip -\rightskip
\newcommand*{\EndParWithPagenoInMargin}
    {\nobreak\hfill
     \nobreak\makebox[0.75cm][r]{\mdseries\normalsize\etocpage}%
     \par}
\renewcommand*\etoctoclineleaders
    {\hbox{\normalfont\normalsize\hbox to .75ex {\hss.\hss}}}
\newcommand*{\EndParWithPagenoInMarginAndLeaders}
    {\nobreak\leaders\etoctoclineleaders\hfill
     \nobreak\makebox[0.75cm][r]{\mdseries\normalsize\etocpage}%
     \par }
\etocsetstyle {part}
              {}
              {\leavevmode\leftskip 1cm\relax}
              {\bfseries\large\llap{\makebox[1cm][r]{\etocnumber\ \ }}%
               \etocname\EndParWithPagenoInMargin\smallskip}
              {}
\etocsetstyle {section}
              {}
              {\leavevmode\leftskip 1.75cm\relax}
              {\bfseries\normalsize\llap{\makebox[.75cm][l]{\etocnumber}}%
               \etocname\EndParWithPagenoInMarginAndLeaders}
              {}
\etocsetstyle {subsection}
              {}
              {\leavevmode\leftskip 2.75cm\relax }
              {\mdseries\normalsize\llap{\makebox[1cm][l]{\etocnumber}}%
               \etocname\EndParWithPagenoInMarginAndLeaders}
              {}
\etocsetstyle {subsubsection}
              {}
              {\leavevmode\leftskip 4cm\relax }
              {\mdseries\normalsize\llap{\makebox[1.25cm][l]{\etocnumber}}%
               \etocname\EndParWithPagenoInMarginAndLeaders}
              {}
\etocsetstyle {paragraph}
              {}
              {\leavevmode\leftskip 5.5cm\relax }
              {\mdseries\normalsize\llap{\makebox[1.5cm][l]{\etocnumber}}%
               \etocname\EndParWithPagenoInMarginAndLeaders}
              {}
\etocsettagdepth {preamble}   {none}
\etocsettagdepth {overview}   {part}
\etocsettagdepth {arbitrarily}{part}
\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
                    \fbox{\makebox[8cm]{A TOC using depth tags}}}
\sloppy
\tableofcontents
\endgroup
```

35.6. 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 19 of using the execution of \tableofcontents as a way to store data which was then displayed later.

For the record, here is how the TOC from section 22 was coded in the old days. We don't have here the problems with the positioning of \hline's we face with the newer method; on the other hand we must manipulate token registers which are not familiar to most LaTeX users (macros could be used, but would be more cumbersome, except perhaps if using the ε-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 ε-TeX \unexpanded primitive. See for example section 21.

```
\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
                 \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
```

Part VIII.
etoc and the outside world

36. Compatibility with other packages

The contents of the `.toc` file (if it already exists) are read into memory by **etoc** once, at the time of `\begin{document}`.[46]

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

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.

etoc loads the package `multicol`.

etoc requires the `.toc` file to use the `\contentsline` macro (possibly patched by other packages). It is thus 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.

etoc is `hyperref` aware and hopefully `hyperref` compatible: the macros `\etocname`, `\etocnumber`, and `\etocpage` contain the `hyperref` links, if present (note that the *link-toc=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.

Release `1.07k` adds compatibility with package `tocloft`:[47] 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.

Release `1.07l` 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

---

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

[47] http://ctan.org/pkg/tocloft

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.

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\mystuff}`
as this will make the `\label/\ref` mechanism impossible. Rather you may do
> `\renewcommand\etocaftertochook{\mystuff}`
and there is also `\etocaftercontentshook` which is executed a bit earlier[48] just before the closing part of the toc display style (and thus still within a group.)

---

etoc used to be incompatible with package `tocvsec2`[49]; it now cohabits, sort of, as it deactivates `tocvsec2`'s modification of `\tableofcontents` (done in the style above) and also cancels its other `toc`-related macros, but reimplements partially their functionality with `\etocsettocdepth.toc`. By the way, at least two latex runs are necessary for new uses of this command in a document to have an effect in tables of contents.

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

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

It is possible to use simultaneously **etoc** and `tableof`[50]. 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.

37. T_EXnical matters

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

---

[48]contrarily to `\etocaftertochook`, `\etocaftercontentshook` is not executed if the `tocdepth` did not allow the printing of the TOC.
[49]http://ctan.org/pkg/tocvsec2
[50]http://ctan.org/pkg/tableof

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

The commands such as `\etocsetstyle`, `\etocsetlevel`, `\etocsettocstyle`, `\etocmulticolstyle`, `\etocruledstyle`, `\etocframedstyle` obey LaTeX's groups. All TOCs are typeset inside groups.

38. Errors and catastrophes

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

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

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

Part IX.
The code

39. Timestamp

This is the documentation as of `2015/04/28`, printed from the source file with the time stamp `28-04-2015 at 22:27:41` CEST. The package version is `v1.08f`, of `2015/04/28`.

40. Change history

v1.08f [2015/04/28]

Minor changes to the documentation. \etocsetlevel more economical.

v1.08e [2015/04/17]

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

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

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

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

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

v1.08d [2015/04/09]

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

Thanks to Christine Römer!

v1.08c [2015/03/30]

- removed a few unneeded \long from the code.

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

v1.08b [2015/03/18]

Bug fixes:

- extra space token removed from '\localtableofcontents' (showed only for inline TOCs.)

- \etocpartname (a macro used by the package own default line styles) was defined to be \partname, but this is not compatible at least with babel+french context. Now simply expands to Part.

- some problems fixed in the German documentation.

- [2015/03/28] some more problems fixed in the documentation. Added mention of \etocarticlestyle and \etocbookstyle.

v1.08a [2015/03/13]

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

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

v1.08 [2015/03/10]

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

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

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

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

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

v1.07n [2015/03/05]

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

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

v1.07m [2015/01/23]

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

Improved global toc display emulation under KOMA-script classes.

New command \etocbeforetitlehook. New command \etocdisplay.

v1.07l [doc of 2014/04/29]

Added to the documentation an example of use of \etocthelinkedname together with an enumitem inline itemize* environment; moved main TOC to immediately after the title, and license to the first pages.

Incorporation of the translation into German done on the initiative of Christine Römer by Felix Baral-Weber, Jenny Rothkrämer-Vogt, Daniel Büttner, Claudia Dahl, Christian Otto and Christine Römer (FSU Jena). My grateful thanks to all!

v1.07l [2014/04/22]

Fixes a bug with the 1.07k compatibility layer with tocloft which had broken the 1.07k (sic) compatibility with memoir (yes, memoir class 1.07k testing had been done before adding the tocloft thing to the source code . . .). Also, etoc when detecting tocvsec2 now checks if this is under the memoir class, as then nothing special needs to be done to rescue \tableofcontents, contrarily to the situation with the native tocvsec2.

v1.07k [2014/03/06]

Compatibility with package tocloft; and improved compatibility with class memoir. Novel TOC example in Overview.

v1.07j [2013/12/03]

Some issues with the documentation formatting (now two-sided) have been addressed, and a novel documentation section "Typesetting the TOC as a table" has been added. Very minor code change (\Etoc@readtoc).

v1.07i [2013/10/21]

Changes to the \etocmulticolstyle and \etocruledstyle codes to lessen the risk of a page break after the title (in the one-column case).

v1.07h [2013/10/16]

New commands \etocdepthtag.toc, \etocsettagdepth, \etocobeydepthtags, \etocignoredepthtags.

v1.07g [2013/10/13]

New commands \etocsettocdepth, \etocsettocdepth.toc, \etocobeytoctocdepth, \etocignoretoctocdepth which emulate part of tocvsec2 functionality ; measures to make tocvsec2 partially compatible with etoc.

New commands \etocsetnexttocdepth, \invisibletableofcontents, \invisiblelocaltableofcontents.

Switched from tikz-qtree to forest for the first 'toc as tree' example.

Command names are linked to their descriptions, and many other changes in the documentation.

Removed printing of temporary message when the local toc id is not yet stabilized; indeed \localtableofcontents can have many uses, such as filling up some token list register and one may wish to not have anything typeset, even in an intermediate run.

All of tex etoc.dtx, etex etoc.dtx, xetex etoc.dtx, latex etoc.dtx, pdflatex etoc.dtx are now possible, and the extracted file etoc.tex allows easy customization of compilation options for the documentation (default is via dvipdfmx which produces the smallest file).

v1.07f [2013/03/07]

New macros \etocthelinkedname, \etocthelinkednumber, \etocthelinkedpage, and \etocthelink.

v1.07e [2013/03/01]

Improvements in the package own line styles with regards to penalties and vertical spaces.

Addition to the documentation of an example of a tree-like table of contents (uses tikz).

More such examples added 2013/03/03.

v1.07d [2013/02/24]

Minor code improvements and new documentation section "Another compatibility mode".

v1.07b [2013/02/02]

Removal of the \xspace from the macros \etoc-name, \etocnumber, \etocpage.

Additional examples in the documentation.

v1.07 [2013/01/29]

New commands:

\etocthename, \etocthenumber, \etocthe-page, \etoclink,

\etoctoccontentsline, \etoctoccontentsline*

\etocnopar, \etocaftercontentshook

Modified command: \etocmulticolstyle

New documentation section "Surprising uses of etoc" which explains how to do "Lists of arbitrary things", in addition to the tables of contents.

v1.06 [2012/12/07]

The standard macros \l@section etc... are modified only during the calls to \tableofcontents; they can thus be customized as will by

the user (with the help of a package like tocloft) and this will be taken into account by etoc for the TOCs typeset in compatibility mode.

v1.05 [2012/12/01]

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

Compatibility with KOMA-script and memoir document classes.

v1.04 [2012/11/24]

A (possibly local) table of contents can be labeled:

\tableofcontents \label{toc:1}

and reproduced elsewhere in the document (with a possibly completely different layout):

\tableofcontents \ref{toc:1}

v1.02 [2012/11/18]

Initial version.

41. Implementation

```
1 \NeedsTeXFormat{LaTeX2e}
2 \ProvidesPackage{etoc}
3  [2015/04/28 v1.08f Completely customisable TOCs (jfB)]
```

```
4 \RequirePackage{multicol}
5 \DeclareOption*{\PackageWarning{etoc}{Option '\CurrentOption' is unknown.}}
6 \ProcessOptions\relax
```

1.08 suppresses \ifEtoc@part and defines \ifEtoc@skipprefix. It also defines \etocglobaldefs, \etoclocaldefs and \Etoc@global.

```
7 \newtoks\Etoc@toctoks
8 \def\Etoc@par{\par}
9 \def\etocinline{\def\Etoc@par{}}
10 \let\etocnopar\etocinline
11 \def\etocdisplay{\def\Etoc@par{\par}}% 1.07m 2015/01/23
```

\etocglobaldefs should be used only for special things such as TOC as a table; it should be put in a group to limit its scope. If used in the preamble, it must come *after* tableof if the latter is loaded too.

```
12 \let\Etoc@global\@empty          % 1.08 2015/03/10
13 \def\etocglobaldefs{\let\Etoc@global\global\let\tof@global\global}
14 \def\etoclocaldefs {\let\Etoc@global\@empty\let\tof@global\@empty}
15 \newif\ifEtoc@jj  % book
16 \newif\ifEtoc@j   % part
17 \newif\ifEtoc@    % chapter
18 \newif\ifEtoc@i   % section
19 \newif\ifEtoc@ii  % subsection
20 \newif\ifEtoc@iii % subsubsection
21 \newif\ifEtoc@iv  % paragraph
22 \newif\ifEtoc@v   % subparagraph
23 \newif\ifEtoc@number
24 \newif\ifEtoc@hyperref
```

```
25 \newif\ifEtoc@parskip    % 1.07d
26 \newif\ifEtoc@tocwithid
27 \newif\ifEtoc@standard
28 \newif\ifEtoc@skipprefix % 1.08
29 \newif\ifEtoc@isfirst    % 1.08
30 \newif\ifEtoc@localtoc
31 \newif\ifEtoc@skipthisone
32 \newif\ifEtoc@stoptoc
33 \newif\ifEtoc@notactive
34 \newif\ifEtoc@mustclosegroup
35 \newif\ifEtoc@emptytoc       % 1.08e
36 \newif\ifEtoc@checksemptiness % 1.08e. Default is 'no checks'.
37 \newcommand*\etocchecksemptiness {\Etoc@checksemptinesstrue }
38 \newif\ifEtoc@notocifnotoc % 1.08e
39 \newcommand*\etocnotocifnotoc {\Etoc@checksemptinesstrue\Etoc@notocifnotoctrue }
40 \def\etoc@{\etoc@}
```

1.07g uses a second counter; this could be avoided, but ok, let's not be that strict.

```
41 \newcounter{etoc@tocid}
42 \newcounter{etoc@tocdepth}% 1.07g
```

2015/03/08 Why do I use jj in the flags and minustwo here? and is there an entity full of forgiveness out there to save my soul even if taking into consideration my use of single and doubled @'s ? Although I desperately need to waste time I miraculously decided not to change everything today and thus the somewhat dispendious set-up below is maintained.

```
43 \@ifclassloaded{memoir}{\def\Etoc@minf{-\thr@@}}{\def\Etoc@minf{-\tw@}}
44 \def\Etoc@@minustwo@@{-\tw@}
45 \let\Etoc@@minusone@@\m@ne
46 \let\Etoc@@zero@@    \z@
47 \let\Etoc@@one@@     \@ne
48 \let\Etoc@@two@@     \tw@
49 \let\Etoc@@three@@   \thr@@
50 \chardef\Etoc@@four@@ 4
51 \chardef\Etoc@@five@@ 5
52 \chardef\Etoc@@six@@  6
53 %%\let\Etoc@localtop\Etoc@@minustwo@@ % 1.08e does it at each local TOC.
54 \def\Etoc@@minustwo@{minustwo}
55 \def\Etoc@@minusone@{minusone}
56 \def\Etoc@@zero@    {zero}
57 \def\Etoc@@one@     {one}
58 \def\Etoc@@two@     {two}
59 \def\Etoc@@three@   {three}
60 \def\Etoc@@four@    {four}
61 \def\Etoc@@five@    {five}
```

1.07g.

```
62 \expandafter\def\csname Etoc@-3@@\endcsname {-\thr@@}
63 \expandafter\let\csname Etoc@-2@@\endcsname \Etoc@@minustwo@@
64 \expandafter\let\csname Etoc@-1@@\endcsname \Etoc@@minusone@@
65 \expandafter\let\csname Etoc@0@@\endcsname  \Etoc@@zero@@
66 \expandafter\let\csname Etoc@1@@\endcsname  \Etoc@@one@@
67 \expandafter\let\csname Etoc@2@@\endcsname  \Etoc@@two@@
68 \expandafter\let\csname Etoc@3@@\endcsname  \Etoc@@three@@
69 \expandafter\let\csname Etoc@4@@\endcsname  \Etoc@@four@@
70 \expandafter\let\csname Etoc@5@@\endcsname  \Etoc@@five@@
71 \expandafter\let\csname Etoc@6@@\endcsname  \Etoc@@six@@
72 \let\Etoc@all@@    \Etoc@@five@@
73 \let\Etoc@none@@   \Etoc@minf
```

Versions earlier than `1.08` did `\Etoc@newlevel` in a quite embarrassing way with an `\edef`.

`1.08f`'s `\etocsetlevel` avoids multiple insertion of the the same level in the `\Etoc@levellist` list.

```
74 \let\Etoc@levellist\@empty
75 \def\Etoc@newlevel #1{\expandafter\def\expandafter\Etoc@levellist\expandafter
76         {\Etoc@levellist\Etoc@levellist@elt #1}}
77 \def\etocsetlevel#1#2{%
78 \@ifundefined{Etoc@#1@@}{\expandafter\Etoc@newlevel\csname l@#1\endcsname}{}%
79 \ifcase#2\relax
80       \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@zero@@
81       \expandafter\let \csname Etoc@#1@\endcsname\Etoc@@zero@
82   \or
83       \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@one@@
84       \expandafter\let \csname Etoc@#1@\endcsname\Etoc@@one@
85   \or
86       \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@two@@
87       \expandafter\let \csname Etoc@#1@\endcsname\Etoc@@two@
88   \or
89       \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@three@@
90       \expandafter\let \csname Etoc@#1@\endcsname\Etoc@@three@
91   \or
92       \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@four@@
93       \expandafter\let \csname Etoc@#1@\endcsname\Etoc@@four@
94   \or
95       \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@five@@
96       \expandafter\let \csname Etoc@#1@\endcsname\Etoc@@five@
97   \or
98       \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@six@@
99   \else
100    \ifnum#2=\m@ne
101       \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@minusone@@
102       \expandafter\let \csname Etoc@#1@\endcsname\Etoc@@minusone@
103    \else
104    \ifnum#2=-\tw@
105       \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@minustwo@@
106       \expandafter\let \csname Etoc@#1@\endcsname\Etoc@@minustwo@
107    \else
108       \PackageWarning{etoc}
109         {unexpected value '#2' in \string\etocsetlevel.^^J%
110          Should be -2,-1, 0, 1, 2, 3, 4, 5, or 6. Set to 6 (=ignored)}%
111       \expandafter\let\csname Etoc@#1@@\endcsname\Etoc@@six@@
112 \fi\fi\fi}
113 \etocsetlevel{book}{-2}
114 \etocsetlevel{part}{-1}
115 \etocsetlevel{chapter}{0}
116 \etocsetlevel{section}{1}
117 \etocsetlevel{subsection}{2}
118 \etocsetlevel{subsubsection}{3}
119 \etocsetlevel{paragraph}{4}
120 \etocsetlevel{subparagraph}{5}
```

The first two parameters of `\Etoc@et@hop` are exchanged in `1.08` compared to earlier versions; this is in relation to changes done in `\Etoc@etoccontentsline`.

```
121 \AtBeginDocument{%
122 \@ifpackageloaded{parskip}{\Etoc@parskiptrue}{}%
123 \@ifpackageloaded{hyperref}{\Etoc@hyperreftrue
124                             \def\Etoc@et@hop#1#2#3#4#5{#2{#3}{#4}{#5}#1}%
```

```
125                              \long\def\Etoc@gobblesixorfive#1#2#3#4#5#6{}}
126                              {\def\Etoc@et@hop#1#2#3#4{#2{#3}{#4}#1}%
127                              \long\def\Etoc@gobblesixorfive#1#2#3#4#5{}}%
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.

1.08 uses `\Etoc@level` rather than a `#1` which used to be anyhow always the `\Etoc@tmp`. Now `\Etoc@level` is what was called formerly `\Etoc@tmp` in `\Etoc@etoccontentsline`.

```
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@itrue
143   \or
144       \global\Etoc@vfalse
145       \global\Etoc@ivfalse
146       \global\Etoc@iiifalse
147       \global\Etoc@iitrue
148   \or
149       \global\Etoc@vfalse
150       \global\Etoc@ivfalse
151       \global\Etoc@iiitrue
152   \or
153       \global\Etoc@vfalse
154       \global\Etoc@ivtrue
155   \or
156       \global\Etoc@vtrue
157   \else
```

2015/03/08 I move up the common assignments.

```
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}
```

1.08 This was originally in `\Etoc@etoccontentsline`, intermixed with the definitions of `\Etoc@contents` and `\Etoc@prefix`. Separated here for clarity of code. One has to be careful that this may close groups. The conditionals are all global. The `\Etoc@level` has been `\global\let` to a `\chardef` variable. The final `\else` of the `\ifcase` assumes none of the `\Etoc@end@..` routines

modify it, naturally.

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

Heart of **etoc** hack into execution of .toc file. The .toc file must contain (at least indirectly via expansion) \contentsline {<unit>}... instructions, with the standard meaning of executing \l@unit (the hyperref modifications are taken into account). It is the various \l@unit's which are (locally) \let to \Etoc@lxyz and it is \Etoc@lxyz which does the parsing of the arguments in order to extract name, number and page number. Once they are obtained, via the execution of this indirectly hacked \contentsline, the code executes the \Etoc@prefix and \Etoc@contents corresponding to that unit level, according to the user \etocsetstyle declarations. Before that the begin part is executed when **etoc** first encounters that level coming from a higher one and the end part will be executed when hitting a more important level.

\etocskipfirstprefix was already part of the very first release of **etoc**. Up to 1.07n it was just an elegant \@thirdofthree possibly executed just before a \@firstoftwo{\Etoc@swa{#1}} {\Etoc@swb{#1}} line. But this made it impossible for me to execute things after the begin parts.

Executing stuff before begin is now a potential problem because the user can close some group in the begin code (this was in fact impossible in versions <1.08 because this would have erased the definitions of \Etoc@contents and \Etoc@prefix). Thus I am not so free now to define things before execution of begin, if I allow people to close a group therein, except if I make them global.

The mechanism of \etocskipfirstprefix made it impossible to add anything at the end of the begin codes. Thus I changed it in 1.08 to simply work with a (global) boolean flag. This gives user the freedom to use \etocskipfirstprefix anywhere in the begin code and it made room for a delimited macro I called \etoconlyonfirst. But then I dropped it in favor of definition of

the `\etociffirst` conditional.

Macros `\Etoc@swa` and `\Etoc@swb` which did the `\Etoc@et@hop` thing have been suppressed, their effect is incorporated to the end of `\Etoc@etoccontentsline`.

```
205 \def\etocskipfirstprefix {\global\Etoc@skipprefixtrue }
```

`2015/03/08` let's record that `\Etoc@level` used to be called `\Etoc@tmp` in versions earlier than `1.08` (and `\Etoc@tmp` is used in the completely different ulterior `\Etoc@lxyz` context). The code sets it for example to be `\Etoc@section@@` which has been let by `\etocsetlevel` to `\@ne`, thus is always self-terminating in `\ifnum..\fi` contexts.

```
206 \def\Etoc@etoccontentsline #1{%
207   \global\expandafter\let\expandafter\Etoc@level\csname Etoc@#1@@\endcsname
208   \Etoc@skipthisonefalse
209   \ifnum\Etoc@level=\Etoc@@six@@
210       \Etoc@skipthisonetrue
211   \else
212     \ifEtoc@localtoc
213       \let\Etoc@next\relax
214       \ifEtoc@stoptoc
215         \Etoc@skipthisonetrue
216       \fi
217       \ifnum\Etoc@level<\Etoc@localtop
218         \def\Etoc@next{\global\Etoc@stoptoctrue}%
219         \Etoc@skipthisonetrue
220       \fi
221       \ifEtoc@notactive
222         \def\Etoc@next{\Etoc@setflags}%
223         \Etoc@skipthisonetrue
224       \fi
225       \Etoc@next
226     \fi
227   \fi
228   \let\Etoc@next\Etoc@gobblesixorfive
229   \ifnum\c@tocdepth<\Etoc@level
230   \else
231   \ifEtoc@skipthisone
232   \else
```

`1.08` New way to handle the compatibility mode. One has to worry only for the local tables of contents (in inactive state, which still must be parsed for global level flags) but in this case `\Etoc@setflags` was done above already. Earlier method proceeded with dummy empty line styles.

```
233   \ifEtoc@standard\let\Etoc@next\@thirdofthree
234   \else
```

`2015/03/08` Ever since the first release of **etoc**, the code has to be careful that the `\Etoc@end@<level>` user defined macros may close groups. This is the reason why some assignments have to be done globally. The `\let\Etoc@next\Etoc@gobblesixorfive` above could be local, because it is used only in case the stuff below is skipped, inclusive of `\Etoc@doends` which may close groups.

Versions `1.08` extracts to a separate macro `\Etoc@doends`. Earlier code did `\global\let\Etoc@next\relax` but we can now give a default `\relax` value to `\Etoc@next` with no need of a `\global` prefix.

```
235       \Etoc@doends
```

`1.08` the `\Etoc@next` will (possibly) execute the begin code, *before* `\Etoc@contents` and `\Etoc@prefix` get defined, contrarily to what I did in earlier versions. I still reset the flags before execution of the begin codes (although the flags are not supposed to be used therein anyhow).

Also the ⟨*begin*⟩ code does `\Etoc@isfirsttrue`.

```
236       \let\Etoc@next\@empty
237       \Etoc@global\Etoc@isfirstfalse
```

`2015/03/08` If my naming scheme had not been so baroque, I could easily do things in a more concise way here...

```
238     \ifcase\Etoc@level
239         \ifEtoc@\else     \def\Etoc@next{\Etoc@begin@zero}\fi
240     \or
241         \ifEtoc@i\else    \def\Etoc@next{\Etoc@begin@one}\fi
242     \or
243         \ifEtoc@ii\else  \def\Etoc@next{\Etoc@begin@two}\fi
244     \or
245         \ifEtoc@iii\else \def\Etoc@next{\Etoc@begin@three}\fi
246     \or
247         \ifEtoc@iv\else  \def\Etoc@next{\Etoc@begin@four}\fi
248     \or
249         \ifEtoc@v\else   \def\Etoc@next{\Etoc@begin@five}\fi
250     \else
251       \ifnum\Etoc@level=\m@ne
252         \ifEtoc@j\else    \def\Etoc@next{\Etoc@begin@minusone}\fi
253       \else
254         \ifEtoc@jj\else  \def\Etoc@next{\Etoc@begin@minustwo}\fi
255       \fi
256     \fi
257     \Etoc@setflags
258     \Etoc@next
259     \let\Etoc@next\@empty
```

`1.08`:

1. separates the definition of `\Etoc@prefix` and `\Etoc@contents` from the execution of the end parts. And it does the definition with some `\csname`'s.

2. implements `\etocskipfirstprefix` with a boolean, this is less slick but more flexible than the earlier method.

3. defines the `\etociffirst` conditional.

4. has `\Etoc@level` rather than `\Etoc@tmp` and thus we can dispense with the `\ifEtoc@part` switch as we will be able to test the equality `\Etoc@level=-1` inside of `\Etoc@lxyz`.

5. adds the `\Etoc@global` prefix.

```
260     \Etoc@global\expandafter\def\expandafter\Etoc@prefix\expandafter
261         {\csname Etoc@prefix@\csname Etoc@#1@\endcsname\endcsname }%
262     \Etoc@global\expandafter\def\expandafter\Etoc@contents\expandafter
263         {\csname Etoc@contents@\csname Etoc@#1@\endcsname\endcsname }%
264     \ifEtoc@skipprefix \Etoc@global\let\Etoc@prefix\@empty\fi
265     \global\Etoc@skipprefixfalse
266   \fi\fi\fi
```

`1.08` modifies the ending to not use the `\Etoc@swa` and `\Etoc@swb` things. With this version, the `\Etoc@next` here is only to gobble everything if it has to (or gobble only the **etoc** things, when in compatibility mode).

For the record we don't do `\csname l@#1\endcsname` mainly because hyperref patches `\contentsline`.

The location where `\Etoc@savedcontentsline` is `\let` to `\contentsline` has been moved with release `1.08` to before the title as set by `\etocsettoctyle`, thus it is now possible to open a tabular in this part. And with the `\Etoc@global` prefix the user may opt for global definitions of `\etocname` and the like, thus use freely & and other group closing things. Also, some care has been put into doing only expandable things after expansion of the ⟨*prefix*⟩ and ⟨*contents*⟩ parts of a sectioning unit style.

```
267    \Etoc@next
268    \Etoc@et@hop{\Etoc@prefix\Etoc@contents}{\Etoc@savedcontentsline{#1}}%
269 }
```

2013/03/07: up to 1.06 `etoc` defined only \etocname, \etocnumber and \etocpage. The
hyperref added data is recycled in the simplest manner, prefixing it with \leavevmode. The
included \Hy@tocdestname is (was, see below 1.07f and 1.08a) left unexpanded. We have to
spend some time with delimited macros to dis-entangle the .toc data, and reconstruct the possi-
ble hyperref data. If the page number is not hyperlinked, \etocpage does *not* add the link found
possibly in the name.

   1.07 adds \etocthename, \etocthenumber, \etocthepage which are left fragile and do not
have the links data, and \etoclink{⟨*linkname*⟩} which is robust and reconstructs an arbitrarily
named link.

   A need (for things like building up a token list to be used in a tikzpicture) arose later
to have some form of the link which could be saved by a simple command like one can do
\global\let\lastname\etocthename, and avoid having to manipulate \Hy@tocdestname. So
1.07f adds \etocthelinkedname, \etocthelinkednumber, \etocthelinkedpage, \etocthe-
link: they use \hyperlink with an expanded \Hy@tocdestname.

   1.07f also adds \leavevmode to \etoclink which should have been done earlier, as it was in-
cluded in \etocname etc... attention, \@namedef{A}{B} and not \@namedef{A} {B} !! on the other
hand this gives a simple way to insert a space as the first token in the paramaters. For \Etoc@again
(which appears later in the code), a \@firstofone construct is however the simplest of all.

   1.08 adds \Etoc@global prefix (2015/03/09).

   1.08a drops the original constructions of \etocname etc... The obvious decision to convert
\etocname, etc ... to be the robust versions of \etocthelinkedname, etc ... was delayed two years
(from 1.07f 2013/03/07 to 1.08a 2015/03/13) for sentimental reason. But the original versions
contained the hyperref destination only in unexpanded form, this forced to use \etocthelinked-
name, etc... in cases when the thing had to be stored for delayed use. Now the user can do this
directly with \etocname, \etocnumber, \etocpage.

```
270 \def\Etoc@lxyz #1#2{%
271    \Etoc@global\@namedef {etoclink }{\leavevmode}%          fall-back
272    \Etoc@global\let\etocthelink      \@empty  %       fall-back
273    \Etoc@global\def\etocthename       {#1}%  (if link, will be removed later)
274    \Etoc@global\def\etocthelinkedname {#1}%  will probably get redefined
275    \Etoc@getthepage #2\etoc@ % defines \etocthelinkedpage (and \etoclink)
276    \Etoc@getnb #1\relax\relax\etoc@  % gets number *and* name, and \etoclink
277    \ifEtoc@number\else
278      \ifnum\Etoc@level=\m@ne
279        \Etoc@getit #1\hspace\relax\etoc@   % additional job for parts
280      \fi
281    \fi
282    \Etoc@global\expandafter\let\csname etocname \endcsname\etocthelinkedname
283    \Etoc@global\expandafter\let\csname etocnumber \endcsname\etocthelinkednumber
284    \Etoc@global\expandafter\let\csname etocpage \endcsname\etocthelinkedpage
285    }
```

1.08 adds the \Etoc@global's.

```
286 \def\Etoc@getthepage #1{%
287   \let\Etoc@next\Etoc@getthepage@nohyp
288   \ifEtoc@hyperref\ifx #1\hyper@linkstart
289     \let\Etoc@next\Etoc@getthepage@hyp
290   \fi\fi
291   \Etoc@next #1%
292 }
293 \def\Etoc@getthepage@nohyp #1\etoc@ {%
294    \Etoc@global\def\etocthepage       {#1}%
```

```
295      \Etoc@global\def\etocthelinkedpage {#1}%
296 }
```

1.07k corrects a bug here with an extra space before ##1 in \@namedef{etoclink } ##1. In 99.99% of use cases however, the macro gets redefined at a later stage, thus the bug did not show.

   1.07n replaces use of \toks@ by macro \Etoc@tmp. Thus, the user has the (not so recommended) possibility to use \toks@ in the styles, even to store information whose scope extends beyond the execution of each \contentsline.

   1.08 adds the \Etoc@global's.

```
297 \def\Etoc@getthepage@hyp #1#2#3#4#5\etoc@ {%
298   \Etoc@global\@namedef{etoclink }##1{\leavevmode #1{#2}{#3}{##1}#5}%
299   \Etoc@global\def\etocthepage {#4}%
300   \edef\Etoc@tmp {{#3}}%
301   \Etoc@global\expandafter\def\expandafter\etocthelinkedpage\expandafter
302     {\expandafter\hyperlink \Etoc@tmp{#4}}%
303 }
304 %
305 \def\Etoc@getnb #1{%
306   \let\Etoc@next\Etoc@getnb@nohyp
307   \ifEtoc@hyperref\ifx #1\hyper@linkstart
308     \let\Etoc@next\Etoc@getnb@hyp
309   \fi\fi
310   \Etoc@next #1%
311 }
312 %
313 \def\Etoc@getit #1{%
314   \let\Etoc@next\Etoc@getit@nohyp
315   \ifEtoc@hyperref\ifx #1\hyper@linkstart
316     \let\Etoc@next\Etoc@getit@hyp
317   \fi\fi
318   \Etoc@next #1%
319 }
```

1.08 adds the \Etoc@global's.

```
320 \def\Etoc@getnb@nohyp #1#2#3\etoc@ {%
321     \def\Etoc@getname ##1\relax\relax\etoc@ {%
322       \Etoc@global\def\etocthename        {##1}%
323       \Etoc@global\def\etocthelinkedname {##1}%
324     }%
325   \ifx #1\numberline
326     \Etoc@global\def\etocthenumber        {#2}%
327     \Etoc@global\def\etocthelinkednumber {#2}%
328     \Etoc@global\Etoc@numbertrue
329     \Etoc@getname #3\etoc@
330   \else % then \etocthename and \etocthelinkedname already defined
331     \Etoc@global\let\etocthenumber        \@empty
332     \Etoc@global\let\etocthelinkednumber \@empty
333     \Etoc@global\Etoc@numberfalse
334   \fi
335 }
```

1.07n replaces use of \toks@ by macro \Etoc@tmp. 1.08 adds the \Etoc@global's.

```
336 \def\Etoc@getnb@hyp #1#2#3#4#5#6\etoc@ {%
337         \def\Etoc@getname ##1\relax\relax\etoc@ {%
338         \Etoc@global\def\etocthename {##1}%
339         \edef\Etoc@tmp    {{#3}}%
340         \Etoc@global\expandafter\def\expandafter\etocthelinkedname\expandafter
341                        {\expandafter\hyperlink \Etoc@tmp{##1}}%
```

```
342              }%
343          \def\Etoc@getnbr ##1##2##3\etoc@ {%
344           \ifx ##1\numberline
345             \Etoc@global\def\etocthenumber {##2}%
346             \edef\Etoc@tmp    {{#3}}%
347             \Etoc@global\expandafter\def\expandafter\etocthelinkednumber
348                       \expandafter{\expandafter\hyperlink \Etoc@tmp{##2}}%
349             \Etoc@global\Etoc@numbertrue
350             \Etoc@getname ##3\etoc@
351           \else
352             \Etoc@global\let\etocthenumber         \@empty
353             \Etoc@global\let\etocthelinkednumber \@empty
354             \Etoc@global\Etoc@numberfalse
355             \Etoc@global\def\etocthename {#4}%
356             \edef\Etoc@tmp    {{#3}}%
357             \Etoc@global\expandafter\def\expandafter\etocthelinkedname
358                       \expandafter {\expandafter\hyperlink \Etoc@tmp{#4}}%
359           \fi
360          }%
361     \Etoc@global\@namedef {etoclink }##1{\leavevmode #1{#2}{#3}{##1}#5}%
362     \Etoc@global\edef\etocthelink ##1{\noexpand\hyperlink {#3}{##1}}%
363     \Etoc@getnbr #4\relax\relax\etoc@
364 }
```

 `1.08` adds the `\Etoc@global`'s.

```
365 \def\Etoc@getit@nohyp #1\hspace#2#3\etoc@ {%
366     \def\Etoc@getname ##1\hspace\relax\etoc@ {%
367        \Etoc@global\def\etocthename        {##1}%
368        \Etoc@global\def\etocthelinkedname {##1}%
369     }%
370   \ifx\relax#2\else
371        \Etoc@global\def\etocthenumber        {#1}%
372        \Etoc@global\def\etocthelinkednumber {#1}%
373        \Etoc@global\Etoc@numbertrue
374        \Etoc@getname #3\etoc@
375   \fi
376 }
```

 `1.07n` replaces use of `\toks@` by macro `\Etoc@tmp`. `1.08` adds the `\Etoc@global`'s.

```
377 \def\Etoc@getit@hyp #1#2#3#4#5#6\etoc@ {%
378     \def\Etoc@getname ##1\hspace\relax\etoc@ {%
379        \Etoc@global\def\etocthename {##1}%
380        \edef\Etoc@tmp    {{#3}}%
381        \Etoc@global\expandafter\def\expandafter\etocthelinkedname
382                  \expandafter{\expandafter\hyperlink \Etoc@tmp{##1}}%
383     }%
384     \def\Etoc@getnbr ##1\hspace##2##3\etoc@ {%
385        \ifx\relax##2\else
386          \Etoc@global\def\etocthenumber {##1}%
387          \edef\Etoc@tmp    {{#3}}%
388          \Etoc@global\expandafter\def\expandafter\etocthelinkednumber
389                  \expandafter{\expandafter\hyperlink \Etoc@tmp{##1}}%
390          \Etoc@global\Etoc@numbertrue
391          \Etoc@getname ##3\etoc@
392        \fi
393     }%
394   \Etoc@getnbr #4\hspace\relax\etoc@
395 }
```

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

```
396 \let\etocthename    \@empty
397 \let\etocthenumber \@empty
398 \let\etocthepage    \@empty
399 \let\etocthelinkedname    \@empty
400 \let\etocthelinkednumber \@empty
401 \let\etocthelinkedpage    \@empty
402 \let\etocthelink    \@empty
403 \DeclareRobustCommand*{\etocname}  {}
404 \DeclareRobustCommand*{\etocnumber}{}
405 \DeclareRobustCommand*{\etocpage}  {}
406 \DeclareRobustCommand*{\etoclink}  {}
407 \DeclareRobustCommand*{\etocifnumbered}
408    {\ifEtoc@number\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi}
409 \expandafter\let\expandafter\etocxifnumbered\csname etocifnumbered \endcsname
```

1.08 defines `\etociffirst`

```
410 \DeclareRobustCommand*{\etociffirst}
411    {\ifEtoc@isfirst\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi}
412 \expandafter\let\expandafter\etocxiffirst\csname etociffirst \endcsname
```

1.07j modifies `\Etoc@readtoc`.

```
413 \def\Etoc@readtoc {%
414   \ifeof \Etoc@tf
415   \else
416      \read \Etoc@tf to \Etoc@buffer
417      \Etoc@toctoks=\expandafter\expandafter\expandafter
418        {\expandafter\the\expandafter\Etoc@toctoks\Etoc@buffer}%
419      \expandafter\Etoc@readtoc
420   \fi
421 }
```

1.07m moves the reading of the toc file At Begin Document. Needed for Babel activated characters.

```
422 \Etoc@toctoks {}% (superfluous, but for clarity)
423 \AtBeginDocument{\IfFileExists{\jobname.toc}
424    {{\endlinechar=\m@ne
425      \makeatletter
426      \newread\Etoc@tf
427      \openin\Etoc@tf\@filef@und
428      \Etoc@readtoc
429      \global\Etoc@toctoks=\expandafter{\the\Etoc@toctoks}%
430      \closein\Etoc@tf}}
431    {\typeout{No file \jobname.toc.}}}
```

1.07d: parskip and `\@nobreakfalse` stuff moved to `\Etoc@tableofcontents`

```
432 \def\Etoc@openouttoc{%
433   \ifEtoc@hyperref
434     \ifx\hyper@last\@undefined
435     \IfFileExists{\jobname .toc}
436       {\Hy@WarningNoLine
437          {old toc file detected, not used; run LaTeX again (cheers from etoc)}%
438        \global\Etoc@toctoks={}%
439       }
440       {}%
441     \fi
442   \fi
443   \if@filesw
444    \newwrite \tf@toc
```

```
445     \immediate \openout \tf@toc \jobname .toc\relax
446   \fi
447   \global\let\Etoc@openouttoc\empty
448 }
```

`1.08` has a new way to handle `\Etoc@standardtrue`.

Also the expansion of `\Etoc@levelist` is moved to inside `\Etoc@tableofcontents` as set-up by `\etocsettocstyle`. This opens up the possibility for the user to open a group in the title (for example start a tabular) which would be closed (by the first &) sometime later. In combination with `\etocglobaldefs` it will be now much easier to typeset TOC as tables with **etoc**!

```
449 \def\Etoc@toctoc{%
450     \global\Etoc@vfalse
451     \global\Etoc@ivfalse
452     \global\Etoc@iiifalse
453     \global\Etoc@iifalse
454     \global\Etoc@ifalse
455     \global\Etoc@false
456     \global\Etoc@jfalse
457     \global\Etoc@jjfalse
458 \the\Etoc@toctoks
459 \ifEtoc@notactive\else
460   \ifEtoc@v    \Etoc@end@five\fi
461   \ifEtoc@iv   \Etoc@end@four\fi
462   \ifEtoc@iii  \Etoc@end@three\fi
463   \ifEtoc@ii   \Etoc@end@two\fi
464   \ifEtoc@i    \Etoc@end@one\fi
465   \ifEtoc@     \Etoc@end@zero\fi
466   \ifEtoc@j    \Etoc@end@minusone\fi
467   \ifEtoc@jj   \Etoc@end@minustwo\fi
468 \fi}
```

Slightly better coded in `1.08b`. Had some bad experiences with TeX conditionals back in 2011/2012 when I started LaTeX programming, and the code was a remnant of extra cautious attitude. Also, now `\etoc@@startlocaltoc` is expandable if it does nothing.

`2015/04/16: 1.08e` removes a `\ifEtoc@notactive` test as `\ifnum` can be positive only once anyhow; although the `\ifEtoc@notactive` boolean could be faster than an `\ifnum`, it adds to it as long as the local toc id is not yet found.

I replace (`1.08e`) the `\Etoc@global`'s for `\Etoc@localtop` by `\global`'s.

```
469 \def\etoc@@startlocaltoc#1#2{%
470     \ifEtoc@localtoc
471       \ifnum #1=#2\relax
472         \ifEtoc@jj  \global\let\Etoc@localtop\Etoc@@minusone@@ \fi
473         \ifEtoc@j   \global\let\Etoc@localtop\Etoc@@zero@@    \fi
474         \ifEtoc@    \global\let\Etoc@localtop\Etoc@@one@@     \fi
475         \ifEtoc@i   \global\let\Etoc@localtop\Etoc@@two@@     \fi
476         \ifEtoc@ii  \global\let\Etoc@localtop\Etoc@@three@@   \fi
477         \ifEtoc@iii \global\let\Etoc@localtop\Etoc@@four@@    \fi
478         \ifEtoc@iv  \global\let\Etoc@localtop\Etoc@@five@@    \fi
479         \ifEtoc@v   \global\let\Etoc@localtop\Etoc@@six@@     \fi
480         \global\Etoc@notactivefalse
481         \global\Etoc@vfalse
482         \global\Etoc@ivfalse
483         \global\Etoc@iiifalse
484         \global\Etoc@iifalse
485         \global\Etoc@ifalse
486         \global\Etoc@false
487         \global\Etoc@jfalse
```

```
488              \global\Etoc@jjfalse
489          \fi
490      \fi
491 }
492 \let\etoc@startlocaltoc\@gobble
```

`2015/03/16`: `1.08e` tests if a local table of contents turns out empty. The `\Etoc@localtop` here does not have the same meaning as when set-up by the regular `\etoc@@startlocaltoc`. It is off by 1 (in general).

After some hesitation I opted for making and obeying the test always (*) for local tables of contents, and do it also for global tables of contents but for the latter the user must issue `\etocno-tocifnotoc` for the emptiness test to be obeyed. The emptiness tests are to be trusted only when the compilations have stabilized.

(*) finally the user must issue `\etocchecksemptiness` once.

`\Etoc@tocid` is the number of the toc (possibly gotten via a `\ref` following a `\tableofcon-tents`), or it is `\z@` if the emptiness test is from a global toc. Until the compilations stabilize, some local TOCs can get printed at wrong locations naturally and emptiness tests can not be trusted either. Releases earlier than `1.08e` authorized `\tableofcontents\ref{foo}` with `foo` the label of a *global* toc. But as it was impossible (without much complications) to discriminate, until auxiliary files have stabilized, between a `\ref` to a local toc whose id is not yet in the `.toc` file and a `\ref` to a global toc whose id will in the end definitely not be in the `.toc` file, I decided that this silly possibility should not be supported anymore. This allowed to remove a few tokens from `\Etoc@tableofcontents` as well. And it also allowed improvements to `\Etoc@localtableofcontents`.

```
493 \def\Etoc@setemptytocbool {%
494     \global\Etoc@stoptocfalse
495     \global\Etoc@emptytoctrue
496     \global\let\Etoc@level\Etoc@minf
497     \begingroup
498       \def\etoc@startlocaltoc##1{%
499           \ifnum##1=\Etoc@tocid\relax
500               \global\let\Etoc@localtop\Etoc@level
501               \global\Etoc@notactivefalse
502           \fi }%
503       \let\contentsline\Etoc@testingcontentsline
504       \Etoc@storetocdepth
505       \the\Etoc@toctoks
506       \Etoc@resettocdepth
507     \endgroup
508 }
509 \DeclareRobustCommand*\etocifwasempty
510   {\ifEtoc@emptytoc\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi }
511 \expandafter
512 \let\expandafter\etocxifwasempty\csname etocifwasempty \endcsname
513 \def\Etoc@testingcontentsline #1{%
514   \ifEtoc@stoptoc
515   \else
516   \ifnum\csname Etoc@#1@@\endcsname=\Etoc@@six@@
517   \else
518     \global\expandafter\let\expandafter\Etoc@level\csname Etoc@#1@@\endcsname
519     \ifEtoc@notactive
520     \else
521     \ifnum\Etoc@level>\c@tocdepth
522     \else
523       \global\Etoc@stoptoctrue
524       \ifnum\Etoc@level>\Etoc@localtop\global\Etoc@emptytocfalse\fi
```

```
525    \fi\fi
526  \fi\fi
527  \Etoc@gobblesixorfive\relax\relax\relax
528 }
```

`1.07g` suppresses the printing of `--unknown etoc ref: run latex again--` as sometimes one uses `\localtableofcontents` to prepare something else, and one does not want any text output even in intermediate runs.

Also `1.07g` adds `\etocaftertochook` to help with `\invisiblelocaltableofcontents` (and then I did `\etocsetnexttocdepth`). `1.07h` makes the hook used by `\etocsetnexttocdepth` private.

`2015/03/14`: `1.08b` suppresses a space token from an unprotected end of line at the end after the replacement text of `\etoc@startlocaltoc` (bug dating back to `1.03 2012/11/23`). I didn't see it due to the originally badly formatted source of `\Etoc@next`.

`2015/03/16`: `1.08e` lets `\localtableofcontents` do a first scan of the `.toc` file (as stored in `\Etoc@toctoks`) to determine if the table of contents will in fact end up empty. In that case, nothing is typeset. The command `\etocaftertochook` is still executed though. Other ways were envisioned (like delimited macros) to determine this potential emptiness, but in the end I opted for execution of the `.toc` file with suitable definitions for `\contentsline` and `\etoc@startlocaltoc`. Notice though that if emptiness would result from empty line styles, this can not be detected. Emptiness means "no executed `\contentsline`".

For this detection of emptiness, assignments (here and in `\Etoc@testingcontentsline`) are made globally, I think this is the best (just in case some portions of the `.toc` file turn out to be inside some groups — perhaps for some silly color assignments, etc... — whose boundaries do not necessarily respect unit levels).

The flag `\Etoc@tocwithid` discriminates between a `\localtableofcontents` and a `\table-ofcontents \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.

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.

```
529 \def\Etoc@localtableofcontents#1{%
530    \edef\Etoc@tocid{#1}%
531    \global\Etoc@emptytocfalse
532    \ifnum\Etoc@tocid<\@ne
533      \PackageWarning{etoc}{Unknown toc reference \@secondoftwo#1. Run LaTeX again}%
534      % versions < 1.08e did not do any printing, 1.08e prints the toc head-
   ing.
535      \global\Etoc@notactivetrue
536      \global\Etoc@stoptoctrue % do only heading
537      \Etoc@localtoctrue % needed for stoptoc obedience
538      \Etoc@tableofcontents
```

```
539        \expandafter\Etoc@gobbletoetoc@ % skip all the rest
540     \fi
541     \global\Etoc@notactivetrue
542     \ifEtoc@checksemptiness \Etoc@setemptytocbool \fi
543     \ifEtoc@emptytoc
544        \ifEtoc@notactive
545           \PackageWarning{etoc}{Unknown toc ID \number\Etoc@tocid. Run LaTeX again}%
546           % emptiness undecided. Declare it non empty.
547           \global\Etoc@emptytocfalse
548           \global\Etoc@stoptoctrue % do only heading
549           \Etoc@localtoctrue
550           \Etoc@tableofcontents
551           \expandafter\expandafter\expandafter\Etoc@gobbletoetoc@
552        \else
553        %%  \PackageWarning{etoc}{This toc is apparently empty}%
554        \fi
555     \else
556        \Etoc@localtoctrue
557        \global\Etoc@stoptocfalse
558        \global\Etoc@notactivetrue
559        \global\let\Etoc@localtop\Etoc@@minustwo@@
560        \edef\etoc@startlocaltoc##1{\noexpand\etoc@@startlocaltoc{##1}{\Etoc@tocid}}%
561        \Etoc@tableofcontents
562     \fi
563     \@gobble\etoc@
564     \endgroup\ifEtoc@mustclosegroup\endgroup\fi
565     \Etoc@aftertochook % 1.07h
566     \etocaftertochook % 1.07g
567 }
```

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`.

```
568 \def\Etoc@getrefno #1#2\etoc@ {#1}
569 \def\Etoc@getref #1{\@ifundefined{r@#1}{0}{\expandafter\expandafter\expandafter
570         \Etoc@getrefno\csname r@#1\endcsname\relax\etoc@}}
571 \def\Etoc@ref#1{\Etoc@localtableofcontents{\Etoc@getref{#1}}}
572 \def\Etoc@label#1{\label{#1}\futurelet\Etoc@nexttoken\Etoc@t@bleofcontents}
573 \@firstofone{\def\Etoc@again} {\futurelet\Etoc@nexttoken\Etoc@t@bleofcontents}
```

`1.08e` adds test for emptiness. And uses another coding style for conditional branching. Easier to read. Not tested for efficiency.

   Also, `\ref{foo}` expects foo to be a label to a *local* TOC. Earlier versions accepted a reference to a global TOC, I have removed the support from `\Etoc@tableofcontents`.

   The syntax `\localtableofcontents\ref{foo}` is now accepted.

```
574 \def\Etoc@dothis #1#2\etoc@ {\fi #1}
575 \def\Etoc@t@bleofcontents{%
576     \ifx\Etoc@nexttoken\label
577        \Etoc@dothis{\expandafter\Etoc@label\@gobble}%
578     \fi
579     \ifx\Etoc@nexttoken\@sptoken
580        \Etoc@dothis{\Etoc@again}%
581     \fi
582     \ifx\Etoc@nexttoken\ref
583        \Etoc@dothis{\Etoc@tocwithidfalse\expandafter\Etoc@ref\@gobble}%
```

```
584        \fi
585        \ifEtoc@tocwithid
586          \Etoc@dothis{\Etoc@localtableofcontents{\c@etoc@tocid}}%
587        \fi
588        \global\Etoc@notactivefalse
589        \global\Etoc@emptytocfalse
590        \ifEtoc@checksemptiness
591           \let\Etoc@tocid\z@
592           \global\let\Etoc@localtop\Etoc@minf
593           \Etoc@setemptytocbool
594        \fi
595        \ifEtoc@emptytoc
596          \ifEtoc@notocifnotoc\else
597             \Etoc@localtocfalse
598             \Etoc@tableofcontents
599          \fi
600        \else
601           \Etoc@localtocfalse
602           \Etoc@tableofcontents
603        \fi
604        \endgroup\ifEtoc@mustclosegroup\endgroup\fi
605        \Etoc@aftertochook % 1.07h
606        \etocaftertochook % 1.07g
607        \@gobble\etoc@
608        }
```

1.07g for consistency Etoc@ prefix added.

    1.08c does not use \arabic in the \addtocontents since I have seen that in some circumstances (for some right to left languages with polyglossia or babel), one can not rely on \arabic having its default definition. As the number written here will be used later in an \ifnum, I should not have used it in the first place (done 2015/03/30).

```
609 \def\Etoc@table@fcontents{%
610     \refstepcounter{etoc@tocid}%
611     \Etoc@tocwithidfalse
612     \futurelet\Etoc@nexttoken\Etoc@t@bleofcontents}
613 \def\Etoc@localtable@fcontents{%
614     \refstepcounter{etoc@tocid}%
615     \addtocontents{toc}{\string\etoc@startlocaltoc{\the\c@etoc@tocid}}%
616     \Etoc@tocwithidtrue
617     \futurelet\Etoc@nexttoken\Etoc@t@bleofcontents}
```

1.07g defines \etoctableofcontents to be able to undo the evil and brutal doings of some packages with \tableofcontents.

    1.08 patches tableof's influence. Version 1.4a or later of tableof is needed.

```
618 \let\etocaftertitlehook      \@empty
619 \let\etocaftercontentshook \@empty
620 \def\etoctableofcontents{%
621     \Etoc@openouttoc
622     \begingroup % closed in \Etoc@t@bleofcontents or \Etoc@localtableofcontents
```

No need for the tableof extra group level, even it is better to get rid of it. I don't globally cancel \tof@begingroup and \tof@endgroup to leave open for the user the (dubious) possibility of using directly \tableof, \tablenotof (rather than only the reasonable thing which is \nexttocwithtags{}{}).

```
623        \let\tof@begingroup\@empty
624        \let\tof@endgroup  \@empty
625        \def\etoc@startlocaltoc##1{\etoc@@startlocaltoc{##1}{\c@etoc@tocid}}%
626        \@ifstar
```

```
627        {\def\Etoc@aftertitlehook{}\Etoc@table@fcontents}
628        {\let\Etoc@aftertitlehook\etocaftertitlehook\Etoc@table@fcontents}}
629 \let\tableofcontents\etoctableofcontents
630 \newcommand*\localtableofcontents{%
631    \Etoc@openouttoc
632    \begingroup % closed in \Etoc@t@bleofcontents or \Etoc@localtableofcontents
633        \let\tof@begingroup\@empty
634        \let\tof@endgroup   \@empty
635        \@ifstar
636        {\def\Etoc@aftertitlehook{}\Etoc@localtable@fcontents}
637        {\let\Etoc@aftertitlehook\etocaftertitlehook\Etoc@localtable@fcontents}}
```

`1.08` moves earlier the definitions of `\Etoc@savedcontentsline` as well as `\Etoc@levellist`. It will thus not be operative to do `\etocsetlevel` from inside the title now. But perhaps it will be easier to deal with tables.

Also I remove a big `\@firstofone`, and use rather `\Etoc@gobbletoetoc@` for the gobbling case.

Also, `1.08` modifies the code to do only expandable things after `\Etoc@toctoc`, in order to allow opening of an alignment in the first argument and closing in the second. Earlier versions had an `\Etoc@@next` initially set to relax after the first `\Etoc@toctoc`. I was in 2012/2013 quite uneasy with the TeX conditionals hence wanted for safety to move the things for the second invocation of `\Etoc@toctoc` after the `\fi\fi\fi`. I could define an `\afterfififi` macro, but it should be ok as it is done here.

Let's note for the record that `1.08e` moved the `\Etoc@par` to `\Etoc@tableofcontents`. This was needed for the implementation of the emptiness test. It might change some things as the `\ref-stepcounter` is now done before the (possible) `\par`. These must be fringe cases though. Most sane people will have left an empty line above their `\tableofcontents` anyhow in the source. And inline toc have a no-op `\Etoc@par`.

`1.08e` suppresses the possibility to do a `\tableofcontents\ref{foo}` with foo the label of a global TOC. Just do `\tableofcontents`.

```
638 \long\def\Etoc@gobbletoetoc@ #1\etoc@ {}
639 \newcommand\etocsettocstyle[2]{%
640 \def\Etoc@tableofcontents
641 {%
642    \ifnum\c@tocdepth>\Etoc@minf
643    \else\expandafter
644     \Etoc@gobbletoetoc@
645    \fi
646    \etocbeforetitlehook % 1.07m
647    \Etoc@par % here 1.08e/1.08f
648    \Etoc@storetocdepth % 1.07g
649    \let\Etoc@savedcontentsline\contentsline % moved up here 1.08
650    \let\contentsline\Etoc@etoccontentsline
651    \ifEtoc@standard
652    \else % moved here from \Etoc@toctoc (1.08)
653        \def\Etoc@levellist@elt####1{\let####1\Etoc@lxyz}%
654        \Etoc@levellist
655        \let\booknumberline\numberline
656        \let\partnumberline\numberline
657        \let\chapternumberline\numberline
658    \fi
659    #1%
660    \ifEtoc@parskip\parskip\z@skip\fi %1.07d
661    \Etoc@aftertitlehook
662    \Etoc@toctoc
663    \etocaftercontentshook
```

```
664      #2\@nobreakfalse % 1.07d: \@nobreakfalse moved here
665      \Etoc@resettocdepth  % 1.07g. moved here from before #2 by 1.08
```
1.08: necessary if `tableof` 1.4a has been used in global mode and thus was forced to mod-
ify globally \contentsline. The collateral effect is that **etoc** is now forced to reset globally
\contentsline.
```
666      \ifx\Etoc@global\global\ifx\tof@finish\@empty\else
667          \global\let\contentsline\Etoc@savedcontentsline
668      \fi\fi
669      \@gobble\etoc@
670 }% end of \Etoc@tableofcontents definition by \etocsettocstyle
671 }
```
1.08 hacks the begin parts to update therein the \ifEtoc@isfirst flag (2015/03/09).

   1.08c removes unneeded \long from the \def's inside \etoc@setstyle (2015/03/29).
```
672 \def\etocsetstyle{\Etoc@standardfalse\etoc@setstyle}
673 \long\def\etoc@setstyle#1#2#3#4#5{%
674      \expandafter\def
675      \csname Etoc@begin@\csname Etoc@#1@\endcsname\endcsname
676                    {#2\Etoc@global\Etoc@isfirsttrue}%
677      \expandafter\def
678      \csname Etoc@prefix@\csname Etoc@#1@\endcsname\endcsname    {#3}%
679      \expandafter\def
680      \csname Etoc@contents@\csname Etoc@#1@\endcsname\endcsname {#4}%
681      \expandafter\def
682      \csname Etoc@end@\csname Etoc@#1@\endcsname\endcsname        {#5}%
683 }
```
placeholder for comments
```
684 \def\etocfontminustwo {\normalfont \LARGE \bfseries}
685 \def\etocfontminusone {\normalfont \large \bfseries}
686 \def\etocfontzero      {\normalfont \large \bfseries}
687 \def\etocfontone       {\normalfont \normalsize \bfseries}
688 \def\etocfonttwo       {\normalfont \normalsize}
689 \def\etocfontthree     {\normalfont \footnotesize}
```
placeholder for comments
```
690 \def\etocsepminustwo  {4ex \@plus .5ex \@minus .5ex}
691 \def\etocsepminusone  {4ex \@plus .5ex \@minus .5ex}
692 \def\etocsepzero      {2.5ex \@plus .4ex \@minus .4ex}
693 \def\etocsepone       {1.5ex \@plus .3ex \@minus .3ex}
694 \def\etocseptwo       {.5ex \@plus .1ex \@minus .1ex}
695 \def\etocsepthree     {.25ex \@plus .05ex \@minus .05ex}
```
placeholder for comments
```
696 \def\etocbaselinespreadminustwo {1}
697 \def\etocbaselinespreadminusone {1}
698 \def\etocbaselinespreadzero      {1}
699 \def\etocbaselinespreadone       {1}
700 \def\etocbaselinespreadtwo       {1}
701 \def\etocbaselinespreadthree     {.9}
```
placeholder for comments
```
702 \def\etocminustwoleftmargin  {1.5em plus 0.5fil}
703 \def\etocminustworightmargin {1.5em plus -0.5fil}
704 \def\etocminusoneleftmargin  {1em}
705 \def\etocminusonerightmargin {1em}
706 \def\etoctoclineleaders
707        {\hbox{\normalfont\normalsize\hb@xt@2ex {\hss.\hss}}}
708 \def\etocabbrevpagename {p.~}
```

Versions earlier than 1.08b (and since v1.05 2012/12/01) defined \etocpartname (for use by
etoc's own line styles) to expand to \partname. But this didn't make sense in the context for
example of babel and frenchb, because \frenchpartname does things depending on the current
value of the counter part. The code in recent frenchb (but not yet v2.5a when \etocpartname
was introduced) constructs control sequences \ordinali, etc... If the part counter is zero, this gives
\ordinal. Usually this is not defined, hence no error happens (as it is constructed via \csname),
but under class memoir the bug showed up. All this to explain that I found out about this long lasting
problem only on 2015/03/14. Probably a sign that **etoc**'s own line styles are rarely used...

```
709 \def\etocpartname        {Part}% modified 1.08b
710 \def\etocbookname        {Book}
```

placeholder for comments The macro \etocdefaultlines was initially called \etoctoclines.
Now \etoctoclines just does \Etoc@standardfalse. Version 1.07e has rewritten entirely the
stuff related to penalties and \addvspace, as this was not satisfactory in the earlier versions, which
were written at a early stage in the development of the package. Actually I am not fully satisfied
with these line styles.

```
711 \def\etocdefaultlines{\Etoc@standardfalse %
712 %% 'book':
713 \etoc@setstyle{@minustwo}
714   {\addpenalty\@M\etocskipfirstprefix}
715   {\addpenalty\@secpenalty}
716   {\begingroup
717    \etocfontminustwo
718    \addvspace{\etocsepminustwo}%
719    \parindent \z@
720    \leftskip  \etocminustwoleftmargin
721    \rightskip \etocminustworightmargin
722    \parfillskip \@flushglue
723    \vbox{\etocifnumbered{\etocbookname\enspace\etocnumber:\quad}{}\etocname
724        \baselineskip\etocbaselinespreadminustwo\baselineskip
725        \par}%
726    \addpenalty\@M\addvspace{\etocsepminusone}%
727    \endgroup}
728   {}%
729 %% 'part':
730 \etoc@setstyle{@minusone}
731   {\addpenalty\@M\etocskipfirstprefix}
732   {\addpenalty\@secpenalty}
733   {\begingroup
734    \etocfontminusone
735    \addvspace{\etocsepminusone}%
736    \parindent \z@
737    \leftskip  \etocminusoneleftmargin
738    \rightskip \etocminusonerightmargin
739    \parfillskip \@flushglue
740    \vbox{\etocifnumbered{\etocpartname\enspace\etocnumber.\quad}{}\etocname
741        \baselineskip\etocbaselinespreadminusone\baselineskip
742        \par}%
743    \addpenalty\@M\addvspace{\etocsepzero}%
744    \endgroup}
745   {}%
746 %% 'chapter':
747 \etoc@setstyle{@zero}
748   {\addpenalty\@M\etocskipfirstprefix}
749   {\addpenalty\@itempenalty}
750   {\begingroup
```

```
751    \etocfontzero
752    \addvspace{\etocsepzero}%
753    \parindent \z@ \parfillskip \@flushglue
754    \vbox{\etocifnumbered{\etocnumber.\enspace}{}\etocname
755          \baselineskip\etocbaselinespreadzero\baselineskip
756          \par}%
757    \endgroup}
758  {\addpenalty{-\@highpenalty}\addvspace{\etocsepminusone}}%
759 %% 'section':
760 \etoc@setstyle{@one}
761  {\addpenalty\@M\etocskipfirstprefix}
762  {\addpenalty\@itempenalty}
763  {\begingroup
764   \etocfontone
765   \addvspace{\etocsepone}%
766   \parindent \z@ \parfillskip \z@
767   \setbox\z@\vbox{\parfillskip\@flushglue
768                  \etocname\par
769                  \setbox\tw@\lastbox
770                  \global\setbox\@ne\hbox{\unhbox\tw@\ }}%
771   \dimen\z@=\wd\@ne
772   \setbox\z@=\etoctoclineleaders
773   \advance\dimen\z@\wd\z@
774   \etocifnumbered
775     {\setbox\tw@\hbox{\etocnumber, \etocabbrevpagename\etocpage}}
776     {\setbox\tw@\hbox{\etocabbrevpagename\etocpage}}%
777   \advance\dimen\z@\wd\tw@
778   \ifdim\dimen\z@ < \linewidth
779       \vbox{\etocname~%
780             \leaders\box\z@\hfil\box\tw@
781             \baselineskip\etocbaselinespreadone\baselineskip
782             \par}% 1.08b adds the % here
783   \else
784       \vbox{\etocname~%
785             \leaders\copy\z@\hfil\break
786             \hbox{}\leaders\box\z@\hfil\box\tw@
787             \baselineskip\etocbaselinespreadone\baselineskip
788             \par}%
789   \fi
790   \endgroup}
791  {\addpenalty\@secpenalty\addvspace{\etocsepzero}}%
792 %% 'subsection':
793 \etoc@setstyle{@two}
794  {\addpenalty\@medpenalty\etocskipfirstprefix}
795  {\addpenalty\@itempenalty}
796  {\begingroup
797   \etocfonttwo
798   \addvspace{\etocseptwo}%
799   \parindent \z@ \parfillskip \z@
800   \setbox\z@\vbox{\parfillskip\@flushglue
801                  \etocname\par\setbox\tw@\lastbox
802                  \global\setbox\@ne\hbox{\unhbox\tw@}}%
803   \dimen\z@=\wd\@ne
804   \setbox\z@=\etoctoclineleaders
805   \advance\dimen\z@\wd\z@
806   \etocifnumbered
807     {\setbox\tw@\hbox{\etocnumber, \etocabbrevpagename\etocpage}}
```

```
808        {\setbox\tw@\hbox{\etocabbrevpagename\etocpage}}%
809     \advance\dimen\z@\wd\tw@
810     \ifdim\dimen\z@ < \linewidth
811        \vbox{\etocname~%
812              \leaders\box\z@\hfil\box\tw@
813              \baselineskip\etocbaselinespreadtwo\baselineskip
814              \par}%
815     \else
816        \vbox{\etocname~%
817              \leaders\copy\z@\hfil\break
818              \hbox{}\leaders\box\z@\hfil\box\tw@
819              \baselineskip\etocbaselinespreadtwo\baselineskip
820              \par}%
821     \fi
822     \endgroup}
823   {\addpenalty\@secpenalty\addvspace{\etocsepone}}%
824 %% `subsubsection':
825 \etoc@setstyle{@three}
826   {\addpenalty\@M
827    \etocfontthree
828    \vspace{\etocsepthree}%
829    \noindent
830    \etocskipfirstprefix}
831   {\allowbreak\,--\,}
832   {\etocname}
833   {.\hfil
834     \begingroup
835     \baselineskip\etocbaselinespreadthree\baselineskip
836     \par
837     \endgroup
838   \addpenalty{-\@highpenalty}}%
```

placeholder for comments

```
839 \etoc@setstyle{@four}{}{}{}{}%
840 \etoc@setstyle{@five}{}{}{}{}%
841 } % end of \etocdefaultlines!
```

The \etocinnertopsep default value is too big as well as \etocbelowtocskip and \etocabovetocskip, I guess, but if I am remember correctly I chose them to mimick the standard TOC spacings in article class.

```
842 \def\etocabovetocskip{3.5ex \@plus 1ex \@minus .2ex}
843 \def\etocbelowtocskip{3.5ex \@plus 1ex \@minus .2ex}
844 \def\etoccolumnsep{2em}
845 \def\etocmulticolsep{0ex}
846 \def\etocmulticolpretolerance{-1}
847 \def\etocmulticoltolerance{200}
848 \def\etocdefaultnbcol{2}
849 \def\etocinnertopsep{2ex}
```

1.07i adds a \nobreak before the \etocinnertopsep and a test of vertical mode to see if truly adding a \par is a reasonable idea, in the case of single-column mode.

1.08b revisits this old code written perhaps at a time I didn't know about \expandafter !

```
850 \newcommand\etocmulticolstyle[2][\etocdefaultnbcol]{%
851 \etocsettocstyle
852    {\let\etocoldpar\par
853     \addvspace{\etocabovetocskip}%
854     \ifnum #1>\@ne
855         \expandafter\@firstoftwo
```

```
856      \else \expandafter\@secondoftwo
857      \fi
858      {\multicolpretolerance\etocmulticolpretolerance
859      \multicoltolerance\etocmulticoltolerance
860      \setlength{\columnsep}{\etoccolumnsep}%
861      \setlength{\multicolsep}{\etocmulticolsep}%
862      \begin{multicols}{#1}[#2\etocoldpar\addvspace{\etocinnertopsep}]]
863      {#2\ifvmode\else\begingroup\interlinepenalty\@M\parskip\z@skip
864                     \@@par\endgroup
865        \fi
866      \nobreak\addvspace{\etocinnertopsep}%
867      \pretolerance\etocmulticolpretolerance
868      \tolerance\etocmulticoltolerance}%
869    }%
870    {\ifnum #1>\@ne
871          \expandafter\@firstofone
872      \else \expandafter\@gobble
873      \fi
874    {\end{multicols}}%
875    \addvspace{\etocbelowtocskip}}%
876 }
```

placeholder for comments

```
877 \def\etocinnerbottomsep{3.5ex}
878 \def\etocinnerleftsep{2em}
879 \def\etocinnerrightsep{2em}
880 \def\etoctoprule{\hrule}
881 \def\etocleftrule{\vrule}
882 \def\etocrightrule{\vrule}
883 \def\etocbottomrule{\hrule}
884 \def\etoctoprulecolorcmd{\relax}
885 \def\etocbottomrulecolorcmd{\relax}
886 \def\etocleftrulecolorcmd{\relax}
887 \def\etocrightrulecolorcmd{\relax}
```

`1.07i` moves the `\nobreak` to before the `\vskip\etocinnertopsep` (especially important for the single column case).

```
888 \def\etoc@ruledheading #1{%
889    \hb@xt@\linewidth{\color@begingroup
890          \hss #1\hss\hskip-\linewidth
891          \etoctoprulecolorcmd\leaders\etoctoprule\hss
892          \phantom{#1}%
893          \leaders\etoctoprule\hss\color@endgroup}%
894          \nointerlineskip\nobreak\vskip\etocinnertopsep}
895 \newcommand*\etocruledstyle[2][\etocdefaultnbcol]{%
896 \etocsettocstyle
897    {\addvspace{\etocabovetocskip}%
898     \ifnum #1>\@ne\expandafter\@firstoftwo
899         \else \expandafter\@secondoftwo
900     \fi
901       {\multicolpretolerance\etocmulticolpretolerance
902        \multicoltolerance\etocmulticoltolerance
903        \setlength{\columnsep}{\etoccolumnsep}%
904        \setlength{\multicolsep}{\etocmulticolsep}%
905        \begin{multicols}{#1}[\etoc@ruledheading{#2}]]
906       {\etoc@ruledheading{#2}%
907        \pretolerance\etocmulticolpretolerance
908        \tolerance\etocmulticoltolerance}}
```

```
909    {\ifnum #1>\@ne\expandafter\@firstofone
910        \else \expandafter\@gobble
911    \fi
912    {\end{multicols}}%
913    \addvspace{\etocbelowtocskip}}}
```

1.07k defines \Etoc@relax and \etocbkgcolorcmd as \long: the user manual says to use \renewcommand\etocbkgcolorcmd, and an \ifx test is used in the framed style. It was thus a bug to have non-long definitions before. Or I could have said in the user manual to use \renewcommand*, or in \etocframedstyle I should test for the two, or I should pause to try to remember about this code and think about it.

```
914 \def\etocframedmphook{\relax}
915 \long\def\etocbkgcolorcmd{\relax}
916 \long\def\Etoc@relax{\relax}
```

placeholder for comments

```
917 \newbox\etoc@framed@titlebox
918 \newbox\etoc@framed@contentsbox
919 \newcommand*\etocframedstyle[2][\etocdefaultnbcol]{%
920 \etocsettocstyle{%
921     \addvspace{\etocabovetocskip}%
922     \sbox\z@{#2}%
923     \dimen\z@\dp\z@
924         \ifdim\wd\z@<\linewidth \dp\z@\z@ \else \dimen\z@\z@ \fi
925     \setbox\etoc@framed@titlebox=\hb@xt@\linewidth{\color@begingroup
926         \hss
927         \ifx\etocbkgcolorcmd\Etoc@relax\else
928             \sbox\tw@{\color{white}%
929             \vrule\@width\wd\z@\@height\ht\z@\@depth\dimen\z@}%
930             \ifdim\wd\z@<\linewidth \dp\tw@\z@\fi
931             \box\tw@
932             \hskip-\wd\z@
933         \fi
934         \copy\z@
935         \hss
936         \hskip-\linewidth
937         \etoctoprulecolorcmd\leaders\etoctoprule\hss%
938         \hskip\wd\z@
939         \etoctoprulecolorcmd\leaders\etoctoprule\hss\color@endgroup}%
940     \setbox\z@\hbox{\etocleftrule\etocrightrule}%
941     \dimen\tw@\linewidth\advance\dimen\tw@-\wd\z@
942         \advance\dimen\tw@-\etocinnerleftsep
943         \advance\dimen\tw@-\etocinnerrightsep
944     \setbox\etoc@framed@contentsbox=\vbox\bgroup
945         \hsize\dimen\tw@
946         \kern\dimen\z@
947         \vskip\etocinnertopsep
948         \hbox\bgroup
949         \begin{minipage}{\hsize}%
950         \etocframedmphook
951     \ifnum #1>\@ne\expandafter\@firstoftwo
952     \else \expandafter\@secondoftwo
953     \fi
954         {\multicolpretolerance\etocmulticolpretolerance
955         \multicoltolerance\etocmulticoltolerance
956         \setlength{\columnsep}{\etoccolumnsep}%
957         \setlength{\multicolsep}{\etocmulticolsep}%
958         \begin{multicols}{#1}}
```

```
959        {\pretolerance\etocmulticolpretolerance
960         \tolerance\etocmulticoltolerance}}
961     {\ifnum #1>\@ne\expandafter\@firstofone
962         \else \expandafter\@gobble
963      \fi
964      {\end{multicols}\unskip }%
965     \end{minipage}%
966     \egroup
967     \vskip\etocinnerbottomsep
968     \egroup
969     \vbox{\hsize\linewidth
970         \ifx\etocbkgcolorcmd\Etoc@relax\else
971             \kern\ht\etoc@framed@titlebox
972             \kern\dp\etoc@framed@titlebox
973             \hb@xt@\linewidth{\color@begingroup
974             \etocleftrulecolorcmd\etocleftrule
975             \etocbkgcolorcmd
976             \leaders\vrule
977                 \@height\ht\etoc@framed@contentsbox
978                 \@depth\dp\etoc@framed@contentsbox
979             \hss
980             \etocrightrulecolorcmd\etocrightrule
981             \color@endgroup}\nointerlineskip
982             \vskip-\dp\etoc@framed@contentsbox
983             \vskip-\ht\etoc@framed@contentsbox
984             \vskip-\dp\etoc@framed@titlebox
985             \vskip-\ht\etoc@framed@titlebox
986         \fi
987     \box\etoc@framed@titlebox\nointerlineskip
988     \hb@xt@\linewidth{\color@begingroup
989     {\etocleftrulecolorcmd\etocleftrule}%
990     \hss\box\etoc@framed@contentsbox\hss
991     \etocrightrulecolorcmd\etocrightrule\color@endgroup}
992     \nointerlineskip
993     \vskip\ht\etoc@framed@contentsbox
994     \vskip\dp\etoc@framed@contentsbox
995     \hb@xt@\linewidth{\color@begingroup\etocbottomrulecolorcmd
996         \leaders\etocbottomrule\hss\color@endgroup}}
997     \addvspace{\etocbelowtocskip}}}
```

placeholder for comments

```
998 \newcommand\etoc@multicoltoc[2][\etocdefaultnbcol]{%
999     \etocmulticolstyle[#1]{#2}%
1000    \tableofcontents}
1001 \newcommand\etoc@multicoltoci[2][\etocdefaultnbcol]{%
1002    \etocmulticolstyle[#1]{#2}%
1003    \tableofcontents*}
1004 \newcommand\etoc@local@multicoltoc[2][\etocdefaultnbcol]{%
1005    \etocmulticolstyle[#1]{#2}%
1006    \localtableofcontents}
1007 \newcommand\etoc@local@multicoltoci[2][\etocdefaultnbcol]{%
1008    \etocmulticolstyle[#1]{#2}%
1009    \localtableofcontents*}
```

placeholder for comments

```
1010 \newcommand*\etoc@ruledtoc[2][\etocdefaultnbcol]{%
1011     \etocruledstyle[#1]{#2}%
1012     \tableofcontents}
```

```
1013 \newcommand*\etoc@ruledtoci[2][\etocdefaultnbcol]{%
1014     \etocruledstyle[#1]{#2}%
1015     \tableofcontents*}
1016 \newcommand*\etoc@local@ruledtoc[2][\etocdefaultnbcol]{%
1017     \etocruledstyle[#1]{#2}%
1018     \localtableofcontents}
1019 \newcommand*\etoc@local@ruledtoci[2][\etocdefaultnbcol]{%
1020     \etocruledstyle[#1]{#2}%
1021     \localtableofcontents*}
```

  placeholder for comments

```
1022 \newcommand*\etoc@framedtoc[2][\etocdefaultnbcol]{%
1023     \etocframedstyle[#1]{#2}%
1024     \tableofcontents}
1025 \newcommand*\etoc@framedtoci[2][\etocdefaultnbcol]{%
1026     \etocframedstyle[#1]{#2}%
1027     \tableofcontents*}
1028 \newcommand*\etoc@local@framedtoc[2][\etocdefaultnbcol]{%
1029     \etocframedstyle[#1]{#2}%
1030     \localtableofcontents}
1031 \newcommand*\etoc@local@framedtoci[2][\etocdefaultnbcol]{%
1032     \etocframedstyle[#1]{#2}%
1033     \localtableofcontents*}
```

  placeholder for comments

```
1034 \def\etocmulticol{\begingroup
1035     \Etoc@mustclosegrouptrue
1036     \@ifstar
1037     {\etoc@multicoltoci}
1038     {\etoc@multicoltoc}}
1039 \def\etocruled{\begingroup
1040     \Etoc@mustclosegrouptrue
1041     \@ifstar
1042     {\etoc@ruledtoci}
1043     {\etoc@ruledtoc}}
1044 \def\etocframed{\begingroup
1045     \Etoc@mustclosegrouptrue
1046     \@ifstar
1047     {\etoc@framedtoci}
1048     {\etoc@framedtoc}}
1049 \def\etoclocalmulticol{\begingroup
1050     \Etoc@mustclosegrouptrue
1051     \@ifstar
1052     {\etoc@local@multicoltoci}
1053     {\etoc@local@multicoltoc}}
1054 \def\etoclocalruled{\begingroup
1055     \Etoc@mustclosegrouptrue
1056     \@ifstar
1057     {\etoc@local@ruledtoci}
1058     {\etoc@local@ruledtoc}}
1059 \def\etoclocalframed{\begingroup
1060     \Etoc@mustclosegrouptrue
1061     \@ifstar
1062     {\etoc@local@framedtoci}
1063     {\etoc@local@framedtoc}}
```

  placeholder for comments

```
1064 \def\etocarticlestyle{%
1065     \etocsettocstyle
```

```
1066     {\section *{\contentsname
1067                 \@mkboth {\MakeUppercase \contentsname}
1068                          {\MakeUppercase \contentsname}}}
1069     {}}
1070 \def\etocarticlestylenomarks{%
1071     \etocsettocstyle
1072     {\section *{\contentsname}}
1073     {}}
```
placeholder for comments
```
1074 \def\etocbookstyle{%
1075     \etocsettocstyle
1076     {\if@twocolumn \@restonecoltrue \onecolumn \else \@restonecolfalse \fi
1077      \chapter *{\contentsname
1078                 \@mkboth {\MakeUppercase \contentsname}
1079                          {\MakeUppercase \contentsname}}}
1080     {\if@restonecol \twocolumn \fi}}
1081 \def\etocbookstylenomarks{%
1082     \etocsettocstyle
1083     {\if@twocolumn \@restonecoltrue \onecolumn \else \@restonecolfalse \fi
1084      \chapter *{\contentsname}}
1085     {\if@restonecol \twocolumn \fi}}
1086 \let\etocreportstyle\etocbookstyle
1087 \let\etocreportstylenomarks\etocbookstylenomarks
1088 \def\etocmemoirtoctotocfmt #1#2{%
1089     \def\Etoc@addsuitablecontentsline{\addcontentsline {toc}{#1}{#2}}%
1090     \renewcommand*\etocaftertitlehook{%
1091      \ifmem@em@starred@listof
1092      \else\phantomsection\aftergroup\Etoc@addsuitablecontentsline\fi}}
1093 \def\etocmemoirstyle{%
1094     \etocsettocstyle
1095        {\ensureonecol \par \begingroup \@nameuse {@tocmaketitle}
1096         \Etoc@aftertitlehook\let\Etoc@aftertitlehook\relax
1097         \parskip \cftparskip \@nameuse {cfttocbeforelisthook}}
1098        {\@nameuse {cfttocafterlisthook}\endgroup\restorefromonecol}}
```
placeholder for comments
```
1099 \def\etocscrartclstyle{%
1100     \etocsettocstyle
1101        {\let\if@dynlist\if@tocleft
1102         \def\@currext{toc}% added 1.07m 2015/01/23
1103         \iftocfeature {toc}{onecolumn}
1104            {\iftocfeature {toc}{leveldown}
1105             {}
1106             {\if@twocolumn \aftergroup \twocolumn \onecolumn \fi }}
1107            {}%
```
1.07k: next line to do as within current scrartcl 2013/12/19 v3.12 KOMA-Script article class.
```
1108         \iftocfeature {toc}{numberline}{\def \nonumberline {\numberline {}}}{}%
1109         \tocbasic@listhead {\listoftocname}%
1110         \begingroup \expandafter \expandafter \expandafter
1111         \endgroup \expandafter
1112         \ifx
1113             \csname microtypesetup\endcsname \relax
1114         \else
1115             \iftocfeature {toc}{noprotrusion}{}
1116                {\microtypesetup {protrusion=false}%
1117                 \PackageInfo {tocbasic}%
1118                 {character protrusion at toc deactivated}}%
```

```
1119          \fi
1120          \setlength {\parskip }{\z@ }%
1121          \setlength {\parindent }{\z@ }%
1122          \setlength {\parfillskip }{\z@ \@plus 1fil}%
1123          \csname tocbasic@@before@hook\endcsname
1124          \csname tb@toc@before@hook\endcsname}
1125        {\csname tb@toc@after@hook\endcsname
1126          \csname tocbasic@@after@hook\endcsname}}
1127 \let\etocscrbookstyle\etocscrartclstyle
1128 \let\etocscrreprtstyle\etocscrartclstyle
```

placeholder for comments

```
1129 \def\etocstandarddisplaystyle{\etocarticlestyle}
1130 \newcommand*\etocmarkboth[1]{%
1131      \@mkboth{\MakeUppercase{#1}}{\MakeUppercase{#1}}}
1132 \newcommand*\etocmarkbothnouc[1]{\@mkboth{#1}{#1}}
1133 \newcommand\etoctocstyle[3][section]{\etocmulticolstyle[#2]%
1134      {\csname #1\endcsname *{#3}}}
1135 \newcommand\etoctocstylewithmarks[4][section]{\etocmulticolstyle[#2]%
1136      {\csname #1\endcsname *{#3\etocmarkboth{#4}}}}
1137 \newcommand\etoctocstylewithmarksnouc[4][section]{\etocmulticolstyle[#2]%
1138      {\csname #1\endcsname *{#3\etocmarkbothnouc{#4}}}}
```

placeholder for comments

```
1139 \def\Etoc@redefetocstyle#1{%
1140      \renewcommand\etoctocstylewithmarks[4][#1]
1141      {\etocmulticolstyle[##2]%
1142          {\csname ##1\endcsname *{##3\etocmarkboth{##4}}}}
1143      \renewcommand\etoctocstylewithmarksnouc[4][#1]
1144      {\etocmulticolstyle[##2]%
1145          {\csname ##1\endcsname *{##3\etocmarkbothnouc{##4}}}}
1146      \renewcommand\etoctocstyle[3][#1]{%
1147       \etocmulticolstyle[##2]{\csname ##1\endcsname *{##3}}}}
1148 \@ifclassloaded{scrartcl}
1149      {\renewcommand*\etocstandarddisplaystyle{\etocscrartclstyle}}{}
1150 \@ifclassloaded{book}
1151      {\renewcommand*\etocfontone{\normalfont\normalsize}
1152       \renewcommand*\etocstandarddisplaystyle{\etocbookstyle}
1153       \Etoc@redefetocstyle{chapter}}{}
1154 \@ifclassloaded{report}
1155      {\renewcommand*\etocfontone{\normalfont\normalsize}
1156       \renewcommand*\etocstandarddisplaystyle{\etocreportstyle}
1157       \Etoc@redefetocstyle{chapter}}{}
1158 \@ifclassloaded{scrbook}
1159      {\renewcommand*\etocfontone{\normalfont\normalsize}
1160       \renewcommand*\etocstandarddisplaystyle{\etocscrbookstyle}
1161       \Etoc@redefetocstyle{chapter}}{}
1162 \@ifclassloaded{scrreprt}
1163      {\renewcommand*\etocfontone{\normalfont\normalsize}
1164       \renewcommand*\etocstandarddisplaystyle{\etocscrreprtstyle}
1165       \Etoc@redefetocstyle{chapter}}{}
```

 1.07k (2014/03/06) adds the appendix to the list of known levels if class memoir is detected.

```
1166 \@ifclassloaded{memoir}
1167      {\etocsetlevel{appendix}{0}%
1168       \renewcommand*\etocfontone{\normalfont\normalsize}
1169       \etocmemoirtoctotocfmt{chapter}{\contentsname}%
1170       \renewcommand*\etocstandarddisplaystyle{\etocmemoirstyle}
1171       \Etoc@redefetocstyle{chapter}}{}
```

`1.07k (2014/03/06)` adds the compatibility with the `tocloft` package; in compatibility mode etoc will obey the `tocloft` customisation for the division headings as well as for the toc title.

`1.07l (2014/04/02)` fixes the bug from the `tocloft` compatibility layer which was `memoir` incompatible: `memoir` has its version of `tocloft` which doesn't have the `\if@cftnctoc` boolean.

```
1172 \def\etoctocloftstyle {%
1173     \etocsettocstyle
1174     {\@cfttocstart\par\begingroup
1175       \parindent\z@ \parskip\cftparskip \@cftmaketoctitle
1176       \if@cfttocbibind\@cftdobibtoc\fi }%
1177     {\endgroup\@cfttocfinish }%
1178 }
1179 \@ifclassloaded{memoir}{}
1180 {\@ifpackageloaded{tocloft}
1181     {\if@cftnctoc\else
1182      \renewcommand*\etocstandarddisplaystyle{\etoctocloftstyle}
1183      \AtBeginDocument{\let\tableofcontents\etoctableofcontents}
1184      \fi }
1185     {\AtBeginDocument
1186       {\@ifpackageloaded{tocloft}
1187        {\if@cftnctoc\else
1188         \PackageWarning {etoc}
1189         {Package 'tocloft' was loaded after 'etoc'!^^J%
1190 **** to prevent it from overwriting \protect\tableofcontents, it will be tricked^^J%
1191 **** into believing to have been loaded with option 'titles'. For better^^J%
1192 **** compatibility, please load 'tocloft' before 'etoc'}%
1193         \AtEndDocument{\PackageWarning{etoc}
1194                       {Please load 'tocloft' /before/ 'etoc'!}}\fi
1195        \@cftnctoctrue }%
1196       {}%
1197      }%
1198     }%
1199 }
```

placeholder for comments

```
1200 \def\Etoc@addtocontents #1#2%
1201     {\ifEtoc@hyperref
1202        \addtocontents {toc}{\protect\contentsline
1203                 {#1}{#2}%
1204                 {\thepage }{\@currentHref }}%
1205      \else
1206        \addtocontents {toc}{\protect\contentsline
1207           {#1}{#2}{\thepage }}%
1208      \fi}
1209 \def\Etoc@addcontentsline@ #1#2#3%
1210     {\@namedef{toclevel@#1}{#3}%
1211      \addcontentsline {toc}{#1}{#2}}
1212 \DeclareRobustCommand*{\etoctoccontentsline}
1213     {\@ifstar{\Etoc@addcontentsline@}{\Etoc@addtocontents}}
```

All the following added for version `1.07g [2013/10/13]`. Motivated by a demand about `tovsec2`. But it is impossible to make `tocvsec2` compatible, if not re-doing all 'toc' macros in `tocvsec2` (`v1.3, 2011/08/07`). I do not retain the stack idea, which does not convince me as useful. The empty line in the `.toc` file is put there by `tocvsec2` (see its code line 143).

I take this opportunity to add to the start of the toc file `\let\etoc@startlocaltoc\@gobble` in case `etoc` is not detected, I should possibly have done that long ago (like I did in `tableof`).

In `1.07h` I decide that I should not have used `\Etoc@` but `\etoc@` for `\Etoc@settocdepth`. So I now use lowercase and add a temporary line for transition in the very improbable situation that

someone actually did already use the just released 1.07g.

In 1.07i the deactivation of \etoc@settocdepth done at the start of the .toc file (for the case this file is used after having removed etoc from the document, or imported in another document) is done via letting it to \@gobble rather than \count@ (which would not be ok, as what follows would be undefined too). The reason for the \count@ also used originally in \etocignoretoctocdepth is that when I first worked on 1.07g I possibly had more than one token after \etoc@settocdepth, but I changed that later.

```
1214 \def\Etoc@storetocdepth {\global\c@etoc@tocdepth\c@tocdepth }
1215 \def\Etoc@resettocdepth {\global\c@tocdepth\c@etoc@tocdepth }
1216 \def\etocobeytoctocdepth {\def\etoc@settocdepth
1217          {\afterassignment\Etoc@@nottoodeep \global\c@tocdepth}}
1218 \def\Etoc@@nottoodeep {\ifnum\c@tocdepth>\c@etoc@tocdepth
1219                        \global\c@tocdepth\c@etoc@tocdepth\fi }
1220 \def\etocignoretoctocdepth {\let\etoc@settocdepth\@gobble }
1221 \def\etocsettocdepth   {\futurelet\Etoc@nexttoken\Etoc@set@tocdepth }
1222 \def\Etoc@set@tocdepth {\ifx\Etoc@nexttoken\bgroup
1223                        \expandafter\Etoc@set@tocdepth@
1224                        \else\expandafter\Etoc@set@toctocdepth
1225                        \fi }
1226 \def\Etoc@set@tocdepth@ #1{\@ifundefined {Etoc@#1@@}
1227       {\PackageWarning{etoc}
1228          {Unknown sectioning unit #1, \protect\etocsettocdepth\space ignored}}
1229       {\global\c@tocdepth\csname Etoc@#1@@\endcsname}%
1230 }
1231 \def\Etoc@set@toctocdepth  #1#{\Etoc@set@toctocdepth@ }
1232 \def\Etoc@set@toctocdepth@ #1{%
1233   \@ifundefined{Etoc@#1@@}%
1234   {\PackageWarning{etoc}
1235       {Unknown sectioning depth #1, \protect\etocsettocdepth.toc ignored}}%
1236   {\addtocontents {toc}
1237       {\protect\etoc@settocdepth\expandafter\protect\csname Etoc@#1@@\endcsname}}%
1238 }
```

1.07h adds the depth tags, and an even more general mechanism could be added at some point: we could insert 'action' tags and have the user map them to arbitrary commands. For the time being we just provide \etocdepthtag.toc {name} and \etocsettagdepth {tag_name}{level} where level is numeric or alphabetical. The numeric level -3 is now again not accepted by \etocsetlevel, it can be used with \etocsettagdepth (which accepts equivalently none).

```
1239 \def\etocdepthtag    #1#{\Etoc@depthtag } % \etocdepthtag.toc {name}
1240 \def\Etoc@depthtag  #1{\addtocontents {toc}{\protect\etoc@depthtag {#1}}}
1241 \def\etocignoredepthtags {\let\etoc@depthtag \@gobble }
1242 \def\etocobeydepthtags   {\let\etoc@depthtag \Etoc@depthtag@ }
1243 \def\Etoc@depthtag@ #1{\@ifundefined{Etoc@depthof@#1}%
1244       {}% ignore in silence if tag has no associated depth
1245       {\afterassignment\Etoc@@nottoodeep
1246              \global\c@tocdepth\csname Etoc@depthof@#1\endcsname}%
1247 }
1248 \def\etocsettagdepth #1#2{\@ifundefined{Etoc@#2@@}%
1249       {\PackageWarning{etoc}
1250         {Unknown sectioning depth #2, \protect\etocsettagdepth\space ignored}}%
1251       {\@namedef{Etoc@depthof@#1}{\@nameuse{Etoc@#2@@}}}%
1252 }
```

We must cancel all tocvsec2 toc-related actions. But a check must be done for the memoir class, as its tocvsec2 emulation does not have the incompatible things etoc needs to revert. This extra check added with etoc 1.07l.

```
1253 \def\Etoc@tocvsec@err #1{\PackageError {etoc}
1254        {the command \protect#1\space is incompatible with etoc}
1255        {use \protect\etocsettocdepth.toc as replacement (check the manual)}%
1256 }%
1257 \AtBeginDocument {%
1258    \@ifclassloaded{memoir}
1259    {\PackageWarning{etoc}{You may consider using \protect\etocsettocdepth.toc
1260      as a drop-in^^J
1261      replacement to the memoir own \protect\settocdepth\space
1262      if you wish to exploit etoc to its^^J
1263      fullest. Similarly consider using
1264     \protect\etocsettocdepth\space and
1265     \protect\etocsetnexttocdepth^^J rather
1266     than memoir's \protect\maxtocdepth.}}
1267    {\@ifpackageloaded {tocvsec2}
1268    {\def\maxtocdepth  #1{\Etoc@tocvsec@err \maxtocdepth    }%
1269     \def\settocdepth  #1{\Etoc@tocvsec@err \settocdepth    }%
1270     \def\resettocdepth  {\@ifstar {\Etoc@tocvsec@err \resettocdepth }%
1271                                   {\Etoc@tocvsec@err \resettocdepth }%
1272                        }%
1273     \def\save@tocdepth #1#2#3{}%    % if etoc is added to file previously
1274     \let\reset@tocdepth\relax        % using tocvsec2
1275     \let\remax@tocdepth\relax
1276     \let\tableofcontents\etoctableofcontents
1277     \PackageWarning {etoc}
1278     {package tocvsec2 detected and its modification of^^J
1279      \protect\tableofcontents\space reverted. Use
1280      \protect\etocsettocdepth.toc as a replacement^^J
1281      for the tocvsec2 toc-related commands}}{}%
1282    }%
1283 }%

  1.07g

1284 \def\invisibletableofcontents {\etocsetnexttocdepth {-3}\tableofcontents }%
1285 \def\invisiblelocaltableofcontents
1286                      {\etocsetnexttocdepth {-3}\localtableofcontents }%
1287 \def\etocsetnexttocdepth #1{%
1288    \@ifundefined{Etoc@#1@@}
1289    {\PackageWarning{etoc}
1290     {Unknown sectioning unit #1, \protect\etocsetnextocdepth\space ignored}}
1291    {\edef\Etoc@aftertochook {\global\c@tocdepth\the\c@tocdepth\space
1292                             \let\noexpand\Etoc@aftertochook\noexpand\@empty }%
1293     \global\c@tocdepth\csname Etoc@#1@@\endcsname}%
1294 }%
```

1.07h. In order for the .toc file to be usable without error even without etoc. In 1.07i use of \@gobble for \etoc@settocdepth.

```
1295 \addtocontents {toc}{\protect\@ifundefined{etoctocstyle}%
1296          {\let\protect\etoc@startlocaltoc\protect\@gobble
1297           \let\protect\etoc@settocdepth\protect\@gobble
1298           \let\protect\etoc@depthtag\protect\@gobble }{}}%
```

Initializations.

```
1299 \def\etocstandardlines {\Etoc@standardtrue}
1300 \def\etoctoclines       {\Etoc@standardfalse}
1301 \etocdefaultlines
1302 \etocstandardlines
1303 \etocstandarddisplaystyle
```

```
1304 \etocobeytoctocdepth
1305 \etocobeydepthtags
1306 \let\Etoc@aftertochook\@empty % private for \etocsetnexttocdepth
1307 \let\etocaftertochook \@empty % public for end user.
1308 \let\etocbeforetitlehook \@empty % new with 1.07m
1309 \endinput
```