

The **etoc** package

JEAN-FRANÇOIS BURNOL

jfbu (at) free (dot) fr

Package version: v1.07j (2013/12/03)

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.

TABLE OF CONTENTS

Foreword	1, p. 1
-----------------	---------

Part I. Overview

Do I need to be a geek to use <code>etoc</code>?	2, p. 4
Limitations in the use of list environments for tables of contents.	
Line styles and toc display style	3, p. 5
<code>\etocsetstyle</code> for the line styles— <code>\etocsettocstyle</code> for the toc display—Compatibility mode.	
A first example	4, p. 6
Linked list of the main package commands	5, p. 8

Part II. Arbitrarily many TOCs, and local ones too

Labeling and reusing elsewhere	6, p. 9
A powerful functionality of <code>etoc</code>: the re-assignment of levels with <code>\etocsetlevel</code>	7, p. 10
The <code>\etocsettocdepth</code> and <code>\etocsetnexttocdepth</code> commands	8, p. 11
The hyperref option <code>bookmarksdepth</code> .	
The command <code>\etocsettocdepth.toc</code>	9, p. 12
The commands <code>\etocobeytoctocdepth</code> and <code>\etocignoretoctocdepth</code> .	
The commands <code>\etocdepthtag.toc</code> and <code>\etocsettagdepth</code>	10, p. 12
The commands <code>\etocobeydepthtags</code> and <code>\etocignoredepthtags</code> .	
Adding commands to the <code>.toc</code> file	11, p. 13
The hyperref option <code>hidelinks</code> .	
Two Examples	12, p. 14
A Beautiful Thesis example—A TOC with a background color.	

Part III. Surprising uses of `etoc`

The TOC of TOCs	13, p. 17
Arbitrary “Lists Of...”, <code>\etoccontentsline</code>	14, p. 18
A TOC with a fancy layout	15, p. 19
Another compatibility mode	16, p. 20
The TOC as a tree	17, p. 23
The TOC as a molecule	18, p. 25

Part IV. Commands for the toc line styles

The `\etocsetstyle` command 19, p. 28

The `\etocname` and `\etocpage` commands—The `\etocskipfirstprefix` command—The `\etocnumber` command—The `\etocthenname`, `\etocthenumber`, and `\etocthepage` commands—The `\etoclink` command—The `\etocthelinkedname`, `\etocthelinkednumber`, `\etocthelinkedpage` and `\etocthelink` commands.

The `\etocsetlevel` command 20, p. 31

Scope of commands added to the `.toc` file 21, p. 32

Testing the scope—This is a (pale) red subsection for illustrative purposes.

Am I also red? 22, p. 33

Part V. Commands for the toc display style

Specifying the toc display style 23, p. 34

The command `\etocsettocstyle`—The commands `\etocmulticolstyle`, `\etocmulticol`, and `\etoclocalmulticol`—The commands `\etocruledstyle`, `\etocruled` and `\etoclocalruled`—The commands `\etocframedstyle`, `\etocframed`, and `\etoclocalframed`—Headings, titles, `\etocoldpar`, `\etocinner-topsep`—The compatibility mode `\etocstandarddisplaystyle`—The command `\etocinline`.

Starred variants of the `\tableofcontents` etc... commands 24, p. 37

Table of contents for this part 25, p. 37

Testing the compatibility mode—A framed display—A (crazy) inline display.

Part VI. Using and customizing `etoc`

Summary of the main style commands 26, p. 41

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

The package default line styles: `\etocdefaultlines` 27, p. 42

Customizing `etoc` 28, p. 45

Customizing the `etoc` pre-defined line styles—Customizing the toc display styles.

One more example of colored TOC layout 29, p. 47

Part VII. Tips

... and tricks 30, p. 50

Hacking framed parboxes—Interverting the levels—All subsections of this document—Displaying statistics—Using depth tags—Typesetting the TOC as a table—Compatibility with other packages—T_EXnical matters—Errors and catastrophes.

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

Part VIII. The code

Timestamp	31, p. 62
License, installation	32, p. 62
Change history	33, p. 63
Implementation	34, p. 63

Part I. Overview

Here are some statistics for this part: it contains 4 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 “2”. The name of the last section is “Linked list of the main package commands” and its number is “5”. The name of the first subsection is “Limitations in the use of list environments for tables of contents” and the corresponding number is “2.1”. The name of the last subsection is “Compatibility mode” and its number is “3.3”.

2. 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 \(section 4\)](#). 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 30.5](#), allows to mimick very well, if so desired, the standard looks (but one would need to add to these line styles some suitably chosen penalties to reduce the possibilities for bad page breaks in TOCs).

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](#).

¹<http://ctan.org/pkg/enumitem>

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 34 visible tables of contents (and a few invisible ones) and uses only one .toc file!² So one should think twice before adding manually extra commands to the .toc file (see [section 11](#)).

2.1. 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 (L^AT_EX's environments create groups).

Therefore the built-in “line styles” proposed by **etoc** as an example (and which are illustrated³ by the [main table of contents](#) in this document) do not make use of environments. Actually, in this user manual, only the [table of contents](#) at the start of [Part IV](#) and the [subsection 30.3](#) (which is a TOC!) have their line styles expressed in terms of enumerate environments.

3. 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`.

3.1. `\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.

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*.

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

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

4. A first example

3.2. `\etocsettocstyle` for the toc display

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

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

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

3.3. 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.⁴ 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. One re-activates the compatibility mode with `\etocstandardlines`; and the command `\etococlines` will re-activate the line styles as defined by the last uses of `\etocsetstyle`, if their scope was not limited to a group or environment.

The command `\etocdefaultlines` resets the line styles to be the package initial default ones.

Even if `\etocsetstyle` has been used, the global display style remains initially as defined by the document class; 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`.

4. 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}
```

⁴for the “toc display style” **etoc** checks if it knows the class, and if not defaults to the `article` class layout.

And as we used `\invisible...`,⁵ the local TOC will exist only through its clones elsewhere in the document.

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)

2. Do I need to be a geek to use <code>etoc</code>?	4
2.1 Limitations in the use of list environments for tables of contents	5
3. Line styles and toc display style	5
3.1 <code>\etocsetstyle</code> for the line styles	5
3.2 <code>\etocsettocstyle</code> for the toc display	6
3.3 Compatibility mode	6
4. A first example	6
5. Linked list of the main package commands	8

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. See also [subsection 30.5](#) which provides a TOC with parts and paragraphs, with multi-line headings as in the standard classes layout (and the technique for setting there the page numbers on the right is closer than here to what L^AT_EX2_ε'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 }
```

⁵this 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.

```

{\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

```

5. Linked list of the main package commands

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

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 “6”. The name of the last section is “Two Examples” and its number is “12”. The name of the first subsection is “The hyperref option *bookmarksdepth*” and the corresponding number is “8.1”. The name of the last subsection is “A TOC with a background color” and its number is “12.2”.

- | | |
|---|----|
| 6. Labeling and reusing elsewhere | 9 |
| 7. A powerful functionality of etoc : the re-assignment of levels with <code>\etoc-setlevel</code> | 10 |

8. The <code>\etocsettocdepth</code> and <code>\etocsetnexttocdepth</code> commands	11
8.1. The hyperref option <i>bookmarksdepth</i>	11
9. The command <code>\etocsettocdepth.toc</code>	12
9.1. The commands <code>\etocobeytoctocdepth</code> and <code>\etocignoretoctocdepth</code>	12
10. The commands <code>\etocdepthtag.toc</code> and <code>\etocsettagdepth</code>	12
10.1. The commands <code>\etocobeydepthtags</code> and <code>\etocignoredepthtags</code> . . .	13
11. Adding commands to the <code>.toc</code> file	13
11.1. The hyperref option <i>hidelinks</i>	14
12. Two Examples	14
12.1. A Beautiful Thesis example	14
12.2. A TOC with a background color	16

6. 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 this page.

I am from far away	
26. Summary of the main style commands	41
26.1. Setting up local styles	41
26.2. Setting up toc display styles	41
26.3. Displaying tables of contents	42
26.4. Labels and references	42
27. The package default line styles: <code>\etocdefaultlines</code>	42
28. Customizing etoc	45
28.1. Customizing the etoc pre-defined line styles	45
28.2. Customizing the toc display styles	46
29. One more example of colored TOC layout	47

We used this:

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

```
\begin{figure*}[ht!]  
  \centering  
  \begingroup  
  \etocstandardlines  
  \renewcommand{\etocbkgcolorcmd}{\color{green!5}}  
  \renewcommand{\etocbelowtocskip}{0pt\relax}  
  \fboxseplex  
  \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}`.

7. 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)⁶

8. 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 contents. Thus, `\invisibletableofcontents` is essentially `\etocsetnexttocdepth{none}\tableofcontents`.

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

8.1. The `hyperref` option `bookmarksdepth`

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

Or, there is the `bookmarksdepth=n` option of package `hyperref`, with n the desired document bookmarks maximal depth, which can be numeric or the name of a level known to `hyperref`. This documentation previously passed `bookmarksdepth=3` as option to `hyperref`, so even if `tocdepth` was left to 1 by inadvertance after printing a certain table

⁶and one should naturally not print this TOC of paragraphs in compatibility mode, which would insist on inserting a gigantic left margin.

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

of contents this did not modify the bookmark tree of the pdf file. Now that `\etocsetnexttocdepth` has been added to the package, we have used it systematically and there was no need for `bookmarksdepth=3` anymore.

9. The command `\etocsettocdepth.toc`

This command `\etocsettocdepth.toc` implements some functionality of `tocvsec2`⁷, a package which however was incompatible with `etoc` (it can still be used for its `secnumdepth`-related commands, but its `toc`-related activities will get canceled by `etoc`) and more-or-less designed for a single table of contents.

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

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

9.1. 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`.

10. 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

⁷<http://ctan.org/pkg/tocvsec2>. I thank D.B. for drawing my attention to the incompatibility of this package with `etoc`.

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

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 30.5](#).

10.1. 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.

11. 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
```

12. Two Examples

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

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

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

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

11.1. 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).

12. Two Examples

12.1. 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 2	Do I need to be a geek to use etoc?	4
2.1	<i>Limitations in the use of list environments for tables of contents</i>	5
Chapter 3	Line styles and toc display style	5
3.1	<i>\etocsetstyle for the line styles</i>	5
3.2	<i>\etocsettocstyle for the toc display</i>	6
3.3	<i>Compatibility mode</i>	6
Chapter 4	A first example	6
Chapter 5	Linked list of the main package commands	8

12.2. 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.

Contents of Part One

2 Do I need to be a geek to use `etoc`?4

2.1 Limitations in the use of list environments for tables of contents 5

3 Line styles and toc display style5

3.1 `\etocsetstyle` for the line styles 5

3.2 `\etocsettocstyle` for the toc display 6

3.3 Compatibility mode 6

4 A first example6

5 Linked list of the main package commands8

```
\tableofcontents \ref{toc:overview} (cf. subsection 12.2 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 6 sections and 0 subsection. The name of the first section is “The TOC of TOCs” and the corresponding number is “13”. The name of the last section is “The TOC as a molecule” and its number is “18”.

13. The TOC of TOCs17

14. Arbitrary “Lists Of...”, `\etoccontentsline`18

15. A TOC with a fancy layout19

16. Another compatibility mode20

13. The TOC of TOCs

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

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

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

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

There is a much better way: there is a hook `\etocaftertitlehook` and we told it to increment the `visibletoc` counter and to write a line to the `.toc` file, in a manner analogous to what sectioning commands such as `chapter`, `section`, or `subsection` do. As `etoc` increments its own private counter even before typesetting the title of a table of contents, this provides (most of the time) a better link destination than any counter manipulated from inside `\etocaftertitlehook` (for which the link would target the area just after the title). So, rather than including `\refstepcounter{visibletoc}` inside `\etocaftertitlehook`, we just put there `\stepcounter{visibletoc}` followed by the command `\etoccontentsline{visibletoc}{\thevisibletoc}`. This `etoc` command `\etoccontentsline{\<level_name>}{\<name>}` has the same effect as:

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

but its usefulness is to circumvent⁹ 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').¹⁰

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 `\etocset-`

⁸The 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...

⁹using `\addtocontents` rather than `\addcontentsline`

¹⁰ The package provides a starred variant `\etoccontentsline*`, which does allow the creation of bookmarks and has a third mandatory argument which is the Level to be used by these bookmarks; depending on the context the starred as well as the non-starred variants may be profitably preceded by `\phantomsection`.

14. Arbitrary “Lists Of...”, `\etoccontentsline`

`style{visibletoc}{\etocskipfirstprefix}{, }{\etocname}{}` which defines an inline display with the comma as separator. Finally, as **etoc** issues `\par` automatically by default just before typesetting a table of contents, we used the command `\etocinline` (also known as `\etocnopar`) which turns off this behavior.

Here are the implementation details:

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

Here is the numbered and linked list of all tables of contents which are displayed within this document: `\tableofcontents`.
`\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 30.2](#) gives another use of the shuffling of levels.

14. Arbitrary “Lists Of...”, `\etoccontentsline`

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

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

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

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

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

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

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

```
\etocsetnexttocdepth{2}      % sections are at level 1 and will show up
\begingroup
\etocsetlevel{exercise}{2}    % but:
\etocsetlevel{chapter}{3}     %      no chapters
\etocsetlevel{subsection}{3}  %      no subsections
\etocsetlevel{part}{3}        %      no parts
\etocsetstyle{exercise}{}{}   % \etocname = exercise number
    {\noindent\etocname\strut\leaders\etocclineleaders\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
```

15. 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.

16. Another compatibility mode

Another technique for doing this kind of things is explained in [subsection 30.4](#).

TABLE OF CONTENTS

PART I

			SECTIONS.	PAGE.
Introductory	1—8	5
Concord.				
LESSON.				
1.	Concord of Subject and Verb	...	9—17	7
2.	Concord of Substantive and Adjective	...	18—22	9
	Concord of Relative and its Antecedent	...	23—25	

PART II

Government.

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

16. Another compatibility mode

As explained in [subsection 3.3](#), the commands `\etocstandardlines` and `\etocstandarddisplaystyle` tell `etoc` to, essentially, act as an observer. The document class layout for the table of contents is then perfectly obeyed. There is no way to customize this standard layout (change fonts, margins, vertical spacings, etc...) from within the package. For this, use some package dedicated to this task; because `etoc` either is (temporarily perhaps)

in compatibility mode with no customization on its part possible, or the user has specified the layout in `\etocsetstyle` commands (and `\etocsettocstyle`) and is (supposedly...) in complete control.

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

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

Here is how to proceed. 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.¹¹ 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`.¹² The number and the name of each entry are

¹¹and 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.

¹²For the `memoir` class, one needs a bit more: each of the command `\booknumberline`, `\partnumberline` and `\chapternumberline` will have to be saved with a `\let`, and, one then specifies:

each separately an `hyperref` link, as is always the case with `etoc`, when not in compatibility mode. Replacing `\etocnumber` with `\etocthenumber` will give a TOC where the numbers are not links anymore, but the names still are. Or one may decide to use `\etocthename` and keep an hyperlinked number with `\etocnumber`.

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

Contents

2. Do I need to be a geek to use <code>etoc</code>?	4
2.1. Limitations in the use of list environments for tables of contents	5
3. Line styles and toc display style	5
<i>\etocsetstyle for the line styles (3.1) – \etocsettocstyle for the toc display (3.2) – Compatibility mode (3.3).</i>	
4. A first example	6
5. Linked list of the main package commands	8

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

```
\etocsetstyle{section}{}
{\ifnum\etocthenumber=3
  \etocsetstyle{subsection}
  {\par\nopagebreak\begin{group}
   \leftskip1.5em \rightskip\@tocrmarg \parfillskip\@flushglue
   \parindent 0pt \normalfont\normalsize\rmfamily\itshape
   % \columnseplem
   % \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}}
{}%

\etocsetstyle{chapter}{}{}{\savedchapterline{\savedchapternumberline
{\etocnumber}\etocname}{\etocpage}}{}
(and analogously for part, respectively book).
```

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

17. The TOC as a tree

Using `tikz`¹³ and the package `forest`¹⁴ we shall display the table of contents of this part as a tree. The technique is to use the `etoc` modified command `\tableofcontents` not for typesetting, but to prepare a macro, or rather here a *token list* variable, with all the instructions to be executed later. LESLIE LAMPORT's book has no mention whatsoever of token lists, and L^AT_EX gives the impression to not really expect the general user to ever hear about them (or delimited macros); this whole section and the next are thus for advanced users.

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

We need to store, not the macro names, but the macro contents. And also we wish to maintain the correct `hyperref` hyperlinks.¹⁵ The hyperlink information put by `etoc` inside `\etocname`, etc..., is just a pointer, not storable for later use; but the package provides `\etocthelinkednumber`, `\etocthelinkedname`, and `\etocthelinkedpage` whose contents contain the hyperlink in a form which can be stored and later reused.

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.¹⁶ 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
```

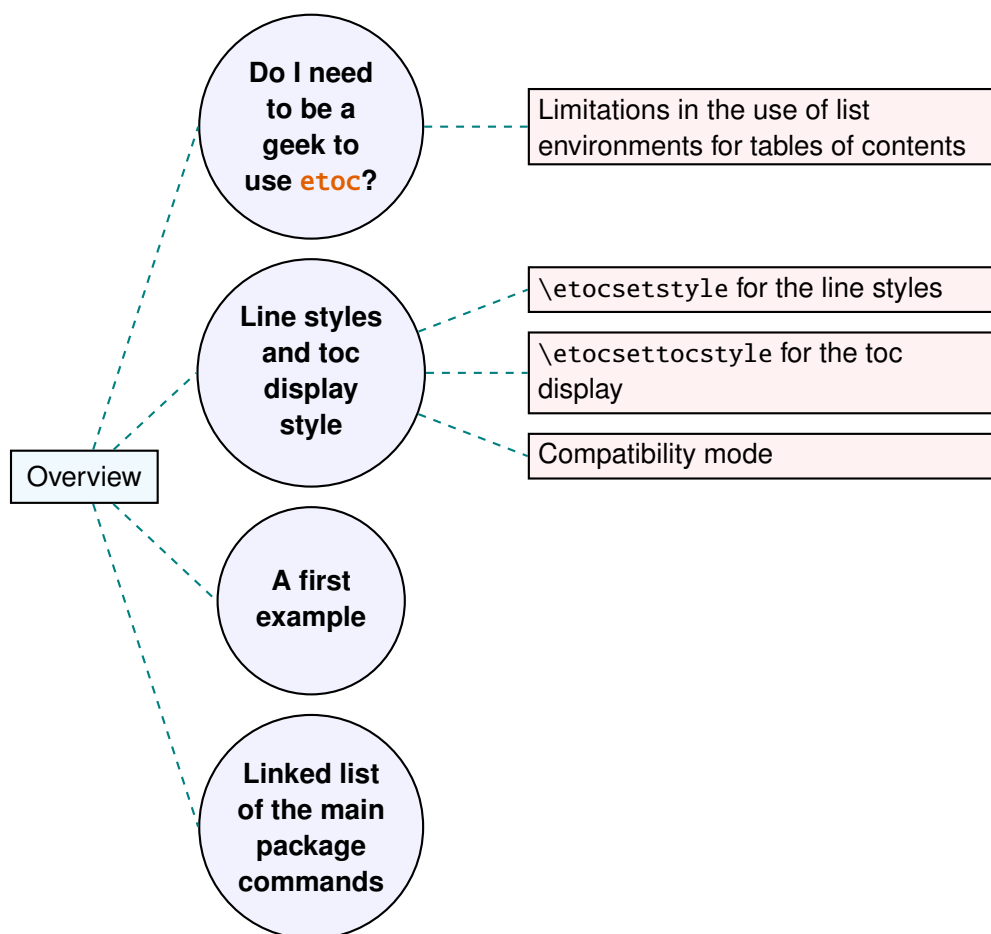
¹³<http://ctan.org/pkg/pgf>

¹⁴<http://ctan.org/pkg/forest>

¹⁵This 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).

¹⁶There 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`.

17. The TOC as a tree



A table of contents displayed as a forest tree.

```

\edef\foresttreenode{ [{\noexpand\parbox{2cm}{\the\tmptok}}}%
}

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

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

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

```



```

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

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

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

```

Why `\hypersetup{hidelinks}`? as explained in [subsection 11.1](#), 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 30.4](#)) 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 (see [subsection 30.6](#)); there we also will 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.

18. 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

18. The TOC as a molecule

handled by T_EX (they need, when filling up the token list to be always balanced at each step, else complicated tricks must be employed.)

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

```
% \newtoks\treetok % 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}
}

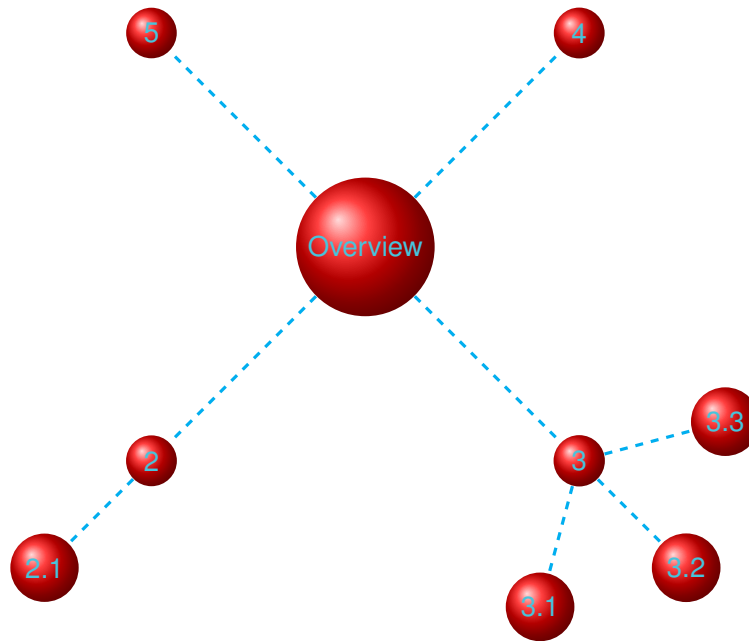
\etocsetstyle{subsection}
{
  \etocskipfirstprefix
  {\appendchildtree\sectiontok\subsectiontok}
  {\preparetreenode \subsectiontok\expandafter{\treenode}}
  {\appendchildtree\sectiontok\subsectiontok}
}

\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=90},
     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]
```



A table of contents as a “molecule.”

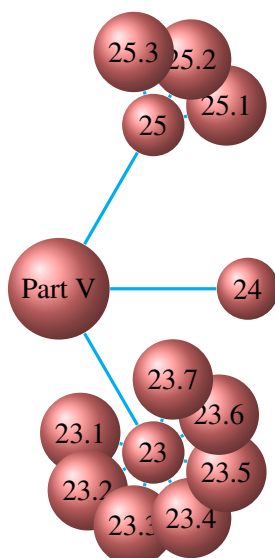
```

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

(\tikzparentnode) --(\tikzchildnode)}}

```

This [TikZ TOC](#) is fully hyperlinked, like the previous [Forest TOC](#).



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}}}%
...

```

Part IV.

Commands for the toc line styles

Here are some statistics for this part: it contains 4 sections and 8 subsections. The name of the first section is “The `\etocsetstyle` command” and the corresponding number is “19”. The name of the last section is “**Am I also red?**” and its number is “22”. The name of the first subsection is “The `\etocname` and `\etocpage` commands” and the corresponding number is “19.1”. The name of the last subsection is “**This is a (pale) red subsection for illustrative purposes**” and its number is “21.2”.

Contents of Part IV

19 The `\etocsetstyle` command (page 28)

19.1 The `\etocname` and `\etocpage` commands (p. 28)

19.2 The `\etocskipfirstprefix` command (p. 29)

19.3 The `\etocnumber` command (p. 30)

19.4 The `\etocthenname`, `\etocthenumber`, and `\etocthepage` commands (p. 30)

19.5 The `\etoclink` command (p. 31)

19.6 The `\etocthelinkedname`, `\etocthelinkednumber`, `\etocthelinkedpage` and `\etocthelink` commands (p. 31)

20 The `\etocsetlevel` command (page 31)

21 Scope of commands added to the `.toc` file (page 32)

21.1 Testing the scope (p. 32)

21.2 **This is a (pale) red subsection for illustrative purposes** (p. 32)

22 Am I also red? (page 33)

19. The `\etocsetstyle` command

19.1. 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:¹⁷

¹⁷the 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.

```

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

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

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

```

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.

Each `\etocsetstyle` command has five mandatory arguments:

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

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

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

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

19.2. The `\etocskipfirstprefix` command

The chosen `subsubsection` style also uses the command `\etocskipfirstprefix`, which, if present, *must* be the very last one in the *start* code. It instructs to not use for the first item the specified “prefix” code.

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!

19. The `\etocsetstyle` command

19.3. 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`.

19.3.1. 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).¹⁸

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

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

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

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

19.4. 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`.

¹⁸as 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`.

19.5. 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.

19.6. 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 now provides `\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.

20. 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}
```

Of course, in compatibility mode, it will be assumed here that the macros `\l@cell`, `\l@molecule`, ..., pre-exist. If no table of contents is typeset in compatibility mode, then all that matters is that the line styles have been set. If for example section is at level 1, then there is no need to do a `\etocsetstyle{molecule}` 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}
```

21. Scope of commands added to the .toc file

```
\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 \etocclines 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.

21. Scope of commands added to the .toc file

21.1. 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
```

21.2. 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.

22. 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 2.1](#).

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 10 subsections. The name of the first section is “Specifying the toc display style” and the corresponding number is “23”. The name of the last section is “Table of contents for this part” and its number is “25”. The name of the first subsection is “The command `\etocsettocstyle`” and the corresponding number is “23.1”. The name of the last subsection is “A (crazy) inline display” and its number is “25.3”.

23. Specifying the toc display style	34
23.1. The command <code>\etocsettocstyle</code>	34
23.2. The commands <code>\etocmulticolstyle</code> , <code>\etocmulticol</code> , and <code>\etoclocalmulticol</code>	34
23.3. The commands <code>\etocruledstyle</code> , <code>\etocruled</code> and <code>\etoclocalruled</code> .	35
23.4. The commands <code>\etocframedstyle</code> , <code>\etocframed</code> , and <code>\etoclocalframed</code>	35
23.5. Headings, titles, <code>\etocoldpar</code> , <code>\etocinnertopsep</code>	36
23.6. The compatibility mode <code>\etocstandarddisplaystyle</code>	36
23.7. The command <code>\etocinline</code>	36
24. Starred variants of the <code>\tableofcontents</code> etc... commands	37
25. Table of contents for this part	37
25.1. Testing the compatibility mode	37
25.2. A framed display	38
25.3. A (crazy) inline display	39

23. 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})
```

23.1. The command `\etocsettocstyle`

This is a command with two mandatory arguments:

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

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

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

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

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>}`.

23.2. 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>}`.

23.2.1. The command `\etocstyle`

```
\etocstyle[⟨kind⟩]{⟨number_of_columns⟩}{⟨title⟩}
= \etocmulticolstyle[⟨number_of_columns⟩]{⟨kind*{title⟩}
```

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

23.2.1.1. `\etocstylewithmarks`

```
=\etocmulticolstyle[⟨number_of_columns⟩]{⟨kind*{title \markboth{\MakeUppercase{mark}}}}}
```

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

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

23.2.1.3. really?

23.3. The commands `\etocruledstyle`, `\etocruled` and `\etoclocalruled`

The general format of `\etocruledstyle` is:

```
\etocruledstyle[⟨number of columns⟩]{⟨title of the toc⟩}
```

The title is horizontal material (the LR mode of *L^AT_EX*, 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}%
\fbxrule1pt\fbxseplex
\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`.

23.4. 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).

23. Specifying the toc display style

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 28.2](#)).

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`¹⁹ or `tcolorbox`²⁰ in the arguments of `\etocsettocstyle` is recommended.

Examples in this document are on pages 9, 16, 39, and 49.

23.5. 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`²¹ at the end of the heading, then does a vertical skip of value `\etocinnertopsep`. The command `\etocoldpar` can also be used explicitly if needed in the mandatory argument to `\etocmulticolstyle` (it is not allowed to insert an empty line in this argument).

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

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

23.6. 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.

23.7. 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.

¹⁹<http://ctan.org/pkg/mdframed>

²⁰<http://ctan.org/pkg/tcolorbox>

²¹this 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.

24. 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 `\etocmemoirtocformat{<kind>}{<name>}` will change the format (`<kind>` is `chapter`, `section`, `subsection`... and `<name>` can be for example `\contentsname`.) The initial set-up is with `chapter` and `\contentsname`.

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

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 also `\etocaftercontentshook`, similarly defined to do nothing. They can be used for some special effects.

25. Table of contents for this part

25.1. 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.²² 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

23. Specifying the toc display style	34
23.1. The command <code>\etocsettocstyle</code>	34

²²the present document uses the `scrartcl` class, and we check here that the `etoc` compatibility mode does respect the customizing done via the class commands.

Contents

23.2. The commands <code>\etocmulticolstyle</code> , <code>\etocmulticol</code> , and <code>\etoclocalmulticol</code>	34
23.3. The commands <code>\etocruledstyle</code> , <code>\etocruled</code> and <code>\etoclocalruled</code> . . .	35
23.4. The commands <code>\etocframedstyle</code> , <code>\etocframed</code> , and <code>\etoclocalframed</code>	35
23.5. Headings, titles, <code>\etocoldpar</code> , <code>\etocinnertopsep</code>	36
23.6. The compatibility mode <code>\etocstandarddisplaystyle</code>	36
23.7. The command <code>\etocinline</code>	36
24. Starred variants of the <code>\tableofcontents</code> etc... commands	37
25. Table of contents for this part	37
25.1. Testing the compatibility mode	37
25.2. A framed display	38
25.3. A (crazy) inline display	39

25.2. 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 facing page).²³

```
\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
\fbboxrule1pt
\renewcommand{\etocbelowtocskip}{0pt\relax}
\etocframedstyle {\normalsize\rmfamily\itshape
  \fcolorbox{red}{white}{\parbox{.8\linewidth}{\centering
    This is a table of contents \‘a la \etoc, but just for
    the sections and subsections in this part. As it is put
    in a frame, it has to be small enough to fit on
    one page. It has the label |toc:b|.}}}
\begin{figure*}[ht!]
  \centering
\normalcolor % work-around to a bug involving color and multicol (and figure)
```

²³inside the `figure*` environment and before the framed toc, which uses `multicol`s, we added `\normalcolor` as work-around to an interaction bug (related to the `\color{red}` command we added to the .toc file) between `multicol`s, `color` and the figure environment. This bug has nothing to do with **etoc**.

```
\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`.*

Specifying the toc display style

23, p. 34

The command `\etocsettocstyle`
23.1, p. 34

The commands `\etocmulticolstyle`, `\etocmulticol`, and `\etoclocalmulticol` . . . 23.2, p. 34

The commands `\etocruledstyle`, `\etocruled` and `\etoclocalruled`
23.3, p. 35

The commands `\etocframedstyle`, `\etocframed`, and `\etoclocalframed`
23.4, p. 35

Headings, titles, `\etocoldpar`, `\etocinnertopsep` 23.5, p. 36

The compatibility mode `\etocstandarddisplaystyle` . . . 23.6, p. 36

The command `\etocinline` . . . 23.7, p. 36

Starred variants of the `\tableofcontents etc...` commands . . . 24, p. 37

Table of contents for this part . . . 25, p. 37

Testing the compatibility mode . . . 25.1, p. 37

A framed display . . . 25.2, p. 38

A (crazy) inline display 25.3, p. 39

25.3. 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 taste in many matters does not seem to be widely shared).

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
```


Here is the inline table of contents. *Foreword.* **Overview:** *Do I need to be a geek to use **etoc**?*²⁴, *Line styles and toc display style*²⁵, *A first example, Linked list of the main package commands.* **Arbitrarily many TOCs, and local ones too:** *Labeling and reusing elsewhere, A powerful functionality of **etoc**: the re-assignment of levels with `\etocsetlevel`, The `\etocsettocdepth` and `\etocsetnexttocdepth` commands*²⁶, *The command `\etocsettocdepth.toc`*²⁷, *The commands `\etocdepthtag.toc` and `\etocsettagdepth`*²⁸, *Adding commands to the .toc file*²⁹, *Two Examples*³⁰. **Surprising uses of **etoc**:** *The TOC of TOCs, Arbitrary “Lists Of...”, `\etoccontentsline`, A TOC with a fancy layout, Another compatibility mode, The TOC as a tree, The TOC as a molecule.* **Commands for the toc line styles:** *The `\etocsetstyle` command*³¹, *The `\etocsetlevel` command, Scope of commands added to the .toc file*³², *Am I also red?.* **Commands for the toc display style:** *Specifying the toc display style*³³, *Starred variants of the `\tableofcontents` etc... commands, Table of contents for this part*³⁴. **Using and customizing **etoc**:** *Summary of the main style commands*³⁵, *The package default line styles: `\etocdefaultlines`, Customizing **etoc***³⁶, *One more example of colored TOC layout.* **Tips:** *... and tricks*³⁷. **The code:** *Timestamp, License, installation, Change history, Implementation.*

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

²⁴Limitations in the use of list environments for tables of contents.

²⁵`\etocsetstyle` for the line styles; `\etocsettocstyle` for the toc display; Compatibility mode.

²⁶The hyperref option `bookmarksdepth`.

²⁷The commands `\etocobeytoctocdepth` and `\etocignoretoctocdepth`.

²⁸The commands `\etocobeydepthtags` and `\etocignoredepthtags`.

²⁹The hyperref option `hidelinks`.

³⁰A Beautiful Thesis example; A TOC with a background color.

³¹The `\etocname` and `\etocpage` commands; The `\etocskipfirstprefix` command; The `\etocnumber` command; The `\etocthename`, `\etocthenumber`, and `\etocthepage` commands; The `\etoclink` command; The `\etocthelinkedname`, `\etocthelinkednumber`, `\etocthelinkedpage` and `\etocthelink` commands.

³²Testing the scope; **This is a (pale) red subsection for illustrative purposes.**

³³The command `\etocsettocstyle`; The commands `\etocmulticolstyle`, `\etocmulticol`, and `\etoclocalmulticol`; The commands `\etocruledstyle`, `\etocruled` and `\etoclocalruled`; The commands `\etocframedstyle`, `\etocframed`, and `\etoclocalframed`; Headings, titles, `\etocoldpar`, `\etocinnertopsep`; The compatibility mode `\etocstandarddisplaystyle`; The command `\etocinline`.

³⁴Testing the compatibility mode; A framed display; A (crazy) inline display.

³⁵Setting up local styles; Setting up toc display styles; Displaying tables of contents; Labels and references.

³⁶Customizing the **etoc** pre-defined line styles; Customizing the toc display styles.

³⁷Hacking framed parboxes; Interverting the levels; All subsections of this document; Displaying statistics; Using depth tags; Typesetting the TOC as a table; Compatibility with other packages; T_EXnical matters; Errors and catastrophes.

is “26”. The name of the last section is “One more example of colored TOC layout” and its number is “29”. The name of the first subsection is “Setting up local styles” and the corresponding number is “26.1”. The name of the last subsection is “Customizing the toc display styles” and its number is “28.2”.

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

Summary of the main style commands	26, p. 41	The package default line styles: <code>\etoc-defaultlines</code>	27, p. 42
Setting up local styles . . .	26.1, p. 41	Customizing <code>etoc</code>	28, p. 45
Setting up toc display styles	26.2, p. 41	Customizing the <code>etoc</code> pre-defined line styles	28.1, p. 45
Displaying tables of contents	26.3, p. 42	Customizing the toc display styles	28.2, p. 46
Labels and references . . .	26.4, p. 42	One more example of colored TOC layout	29, p. 47

26. Summary of the main style commands

26.1. Setting up local styles

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

26.2. Setting up toc display styles

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

26.3. 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
```

26.4. 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}
similarly with \etocmulticol etc . . .
```

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.

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

Activating the use of the package default line styles is done via `\etocdefaultlines`, or `\etococlines` if these 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.³⁸ 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.

³⁸and 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).

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

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

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

1. Foreword

4. A first example

Part I. Overview

5. Linked list of the main package commands

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

Limitations in the use of list environments for tables of contents 2.1, p. 5

Part II. Arbitrarily many TOCs, and local ones too

3. Line styles and toc display style

`\etocsetstyle` for the line styles 3.1, p. 5

6. Labeling and reusing elsewhere

`\etocsettocstyle` for the toc display 3.2, p. 6

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

Compatibility mode 3.3, p. 6

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

The hyperref option `bookmarksdepth` ..
..... 8.1, p. 11

9. The command `\etocsettocdepth.toc`

The commands `\etocobeytoctocdepth`
and `\etocignoretoctocdepth`
..... 9.1, p. 12

10. The commands `\etocdepth-tag.toc` and `\etocsettagdepth`

The commands `\etocobeydepthtags`
and `\etocignoredepthtags`
..... 10.1, p. 13

11. Adding commands to the .toc file

The hyperref option `hidelinks`
..... 11.1, p. 14

12. Two Examples

A Beautiful Thesis example . 12.1, p. 14
A TOC with a background color
..... 12.2, p. 16

Part III. Surprising uses of `etoc`

13. The TOC of TOCs

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

15. A TOC with a fancy layout

16. Another compatibility mode

17. The TOC as a tree

18. The TOC as a molecule

Part IV. Commands for the toc line styles

19. The `\etocsetstyle` command

The `\etocname` and `\etocpage` com-
mands 19.1, p. 28

The `\etocskipfirstprefix` command
..... 19.2, p. 29

The `\etocnumber` command
..... 19.3, p. 30

The `\etocthenname`, `\etocthenumber`,
and `\etocthepage` commands
..... 19.4, p. 30

The `\etoclink` command .. 19.5, p. 31

The `\etocthelinkedname`, `\etocthe-
linkednumber`, `\etocthelinkedpage` and
`\etocthelink` commands .. 19.6, p. 31

20. The `\etocsetlevel` command

21. Scope of commands added to the .toc file

Testing the scope 21.1, p. 32

This is a (pale) red subsection for illus-
trative purposes 21.2, p. 32

22. Am I also red?

Part V. Commands for the toc display style

23. Specifying the toc display style

The command `\etocsettocstyle`
..... 23.1, p. 34

The commands `\etocmulticolstyle`,
`\etocmulticol`, and `\etoclocalmulti-
col` 23.2, p. 34

The commands `\etocruledstyle`, `\etoc-
ruled` and `\etoclocalruled`
..... 23.3, p. 35

The commands `\etocframedstyle`, `\etoc-
framed`, and `\etoclocalframed`
..... 23.4, p. 35

Headings, titles, `\etocoldpar`, `\etocinnertopsep` 23.5, p. 36

The compatibility mode `\etocstandarddisplaystyle` 23.6, p. 36

The command `\etocinline` 23.7, p. 36

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

25. Table of contents for this part

Testing the compatibility mode 25.1, p. 37

A framed display 25.2, p. 38

A (crazy) inline display 25.3, p. 39

Part VI. Using and customizing **etoc**

26. Summary of the main style commands

Setting up local styles 26.1, p. 41

Setting up toc display styles 26.2, p. 41

Displaying tables of contents 26.3, p. 42

Labels and references 26.4, p. 42

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

28. Customizing **etoc**

Customizing the **etoc** pre-defined line styles 28.1, p. 45

28. Customizing **etoc**

28.1. 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}
```

Customizing the toc display styles 28.2, p. 46

29. One more example of colored TOC layout

Part VII. Tips

30. ... and tricks

Hacking framed parboxes .. 30.1, p. 50

Interverting the levels 30.2, p. 51

All subsections of this document 30.3, p. 51

Displaying statistics 30.4, p. 52

Using depth tags 30.5, p. 54

Typesetting the TOC as a table 30.6, p. 56

Compatibility with other packages 30.7, p. 60

\TeX nical matters 30.8, p. 61

Errors and catastrophes 30.9, p. 61

Part VIII. The code

31. Timestamp

32. License, installation

33. Change history

34. Implementation

28. Customizing **etoc**

```
\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*\etocseppone{1.5ex plus .3ex minus .3ex}
%%\newcommand*\etocseptwo{1ex plus .15ex minus .15ex} % modified in 1.07e
\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*\etocclineleaders
  {\hbox{\normalfont\normalsize\hbox to 2ex {\hss.\hss}}}
\newcommand*\etocabbrevpagename{p.~}
\newcommand*\etocpartname{\partname} % utilisateurs de frenchb: attention
                                     % car donne "partie" sans majuscule.
\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**.³⁹

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

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

28.2. 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}
```

³⁹with 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.

```
\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*\etocoprul{\hrule}
\newcommand*\etocoprulcolorcmd{\relax}
```

% for the framed style only:

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

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

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

% hooks

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

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

The `\...colorcmd` things are initially set to be `\relax`, so there is no need to do `\usepackage{color}` if the document does not use colors. If the scope of a change to a color command such as `\etocbkgcolorcmd` has not been limited to a group and one then wishes to let it again be `\relax` one must use a `\renewcommand` 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{\etocoprul}{\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.

29. 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

29. One more example of colored TOC layout

at section level, to get it *inside* the frame.

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

\renewcommand{\etocbkgcolorcmd}{\color{tocbackground}}
\renewcommand{\etoclefttrulecolorcmd}{\color{tocbackground}}
\renewcommand{\etocrightrulecolorcmd}{\color{tocbackground}}
\renewcommand{\etocbottomrulecolorcmd}{\color{tocbackground}}
\renewcommand{\etocbottomrulecolorcmd}{\color{tocbackground}}

\renewcommand{\etoclefttrule}{\vrule width 1cm}
\renewcommand{\etocrightrule}{\vrule width .5cm}
\renewcommand{\etocbottomrule}{\hrule height 12pt}
\renewcommand{\etocbottomrule}{\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}\hskip.2cm\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}{\hbox to 1.5cm{\hss\etocnumber\hskip.2cm}%
  \etocname\hfill\hbox{\etocpage\hskip.2cm}}\vskip6pt}
{}}

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

```

I am a twin of that other TOC (click me!)		
23	Specifying the toc display style	34
23.1	The command \etocsettocstyle	34
23.2	The commands \etocmulticolstyle, \etocmulticol, and \etoclocalmulticol	34
23.3	The commands \etocruledstyle, \etocruled and \etoclocalruled	35
23.4	The commands \etocframedstyle, \etocframed, and \etoclocalframed	35
23.5	Headings, titles, \etocoldpar, \etocinnertopsep	36
23.6	The compatibility mode \etocstandarddisplaystyle ...	36
23.7	The command \etocinline	36
24	Starred variants of the \tableofcontents etc... commands	37
25	Table of contents for this part	37
25.1	Testing the compatibility mode	37
25.2	A framed display	38
25.3	A (crazy) inline display	39

The TOC has been put in a `float` which appears on the current page. The coding is a bit involved⁴⁰ as it does not use any additional package. Also, it was written at some early stage and I have not revised it since.

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

⁴⁰and reveals the author's preference for the T_EX syntax...

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 9 subsections. The name of the first section is “... and tricks” and the corresponding number is “30”. The name of the last section is “... and tricks” and its number is “30”. The name of the first subsection is “Hacking framed parboxes” and the corresponding number is “30.1”. The name of the last subsection is “Errors and catastrophes” and its number is “30.9”.

30. ... and tricks	50
30.1. Hacking framed parboxes	50
30.2. Interverting the levels	51
30.3. All subsections of this document	51
30.4. Displaying statistics	52
30.5. Using depth tags	54
30.6. Typesetting the TOC as a table	56
30.7. Compatibility with other packages	60
30.8. T _E Xnical matters	61
30.9. Errors and catastrophes	61

30. ... and tricks

30.1. 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	23, p. 34	The commands <code>\etocmulticolstyle</code> , <code>\etocmulticol</code> , and <code>\etoclocalmulti-</code> <code>col</code>	23.2, p. 34
The command <code>\etocsettocstyle</code>	23.1, p. 34		

The commands `\etocruledstyle`, `\etocruled` and `\etoclocalruled` 23.3, p. 35

The commands `\etocframedstyle`, `\etocframed`, and `\etoclocalframed` 23.4, p. 35

Headings, titles, `\etocoldpar`, `\etocinnertopsep` 23.5, p. 36

The compatibility mode `\etocstandarddisplaystyle` 23.6, p. 36

The command `\etocinline` 23.7, p. 36

Starred variants of the `\tableofcontents` etc... commands . . . 24, p. 37

Table of contents for this part 25, p. 37

Testing the compatibility mode 25.1, p. 37

A framed display 25.2, p. 38

A (crazy) inline display . . . 25.3, p. 39

30.2. 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}
\etocmarkbothnouc{List of all subsections}
\etocmulticol[3]{\subsection{All subsections of this document}}
\endgroup
```

30.3. All subsections of this document

i. Limitations in the use of list environments for tables of contents (2.1, p. 5)

ii. `\etocsetstyle` for the line styles (3.1, p. 5)

iii. `\etocsettocstyle` for the toc display (3.2, p. 6)

iv. Compatibility mode (3.3, p. 6)

v. The `hyperref` option `bookmarksdepth` (8.1, p. 11)

vi. The commands `\etocobeytoctocdepth` and `\etocignoretoctocdepth` (9.1, p. 12)

vii. The commands `\etocobeydepthtags` and `\etocignoredepthtags` (10.1, p. 13)

viii. The `hyperref` option `hidelinks` (11.1, p. 14)

ix. A Beautiful Thesis example (12.1, p. 14)

x. A TOC with a background color (12.2, p. 16)

xi. The `\etocname` and `\etocpage` commands (19.1, p. 28)

xii. The `\etocskipfirstprefix` command (19.2, p. 29)

xiii. The `\etocnumber` command (19.3, p. 30)

xiv. The `\etocthename`, `\etocthenumber`, and `\etocthepage` commands (19.4, p. 30)

xv. The `\etoclink` command (19.5, p. 31)

xvi. The `\etocthelinkedname`, `\etocthelinkednumber`, `\etocthelinkedpage` and `\etocthelink` commands (19.6, p. 31)

xvii. Testing the scope (21.1, p. 32)

xviii. This is a (pale) red

subsection for illustrative purposes (21.2, p. 32)

xix. The command `\etocsettocstyle` (23.1, p. 34)
xx. The commands `\etocmulticolstyle`, `\etocmulticol`, and `\etoclocalmulticol` (23.2, p. 34)
xxi. The commands `\etocruledstyle`, `\etocruled` and `\etoclocalruled` (23.3, p. 35)
xxii. The commands `\etocframedstyle`, `\etocframed`, and `\etoclocalframed` (23.4, p. 35)
xxiii. Headings, titles, `\etocoldpar`, `\etocinnertopsep` (23.5, p. 36)
xxiv. The compatibility mode `\etocstandarddisplaystyle` (23.6, p. 36)

xxv. The command `\etocinline` (23.7, p. 36)
xxvi. Testing the compatibility mode (25.1, p. 37)
xxvii. A framed display (25.2, p. 38)
xxviii. A (crazy) inline display (25.3, p. 39)
xxix. Setting up local styles (26.1, p. 41)
xxx. Setting up toc display styles (26.2, p. 41)
xxxi. Displaying tables of contents (26.3, p. 42)
xxxii. Labels and references (26.4, p. 42)
xxxiii. Customizing the **etoc** pre-defined line styles (28.1, p. 45)
xxxiv. Customizing the toc display styles (28.2, p. 46)

xxxv. Hacking framed par-boxes (30.1, p. 50)
xxxvi. Intervverting the levels (30.2, p. 51)
xxxvii. All subsections of this document (30.3, p. 51)
xxxviii. Displaying statistics (30.4, p. 52)
xxxix. Using depth tags (30.5, p. 54)
xl. Typesetting the TOC as a table (30.6, p. 56)
xli. Compatibility with other packages (30.7, p. 60)
xlii. T_EXnical matters (30.8, p. 61)
xliii. Errors and catastrophes (30.9, p. 61)

30.4. 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 T_EX primitive `\setbox` rather than the L^AT_EX `\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 18](#). 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{%
```

```

\box{\firstnameii}{\etocname}%
\box{\firstnumberii}{\etocnumber}%
\def\thispartstatsauxii{}%
\begingroup
\etocsetstyle{subsection} {} {}
{\thispartstatsauxii
\stepcounter{mycountii}%
\box{\lastnameii}{\etocname}%
\box{\lastnumberii}{\etocnumber}} {}%
\etocsetstyle{section} {} {}
{\thispartstatsauxi
\stepcounter{mycounti}%
\box{\lastnamei}{\etocname}%
\box{\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

```

30.5. 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. So we simply added to this source various `\etocdepthtag.toc` commands, and it is now just a matter to set the levels for each one of these tags (as they were left undefined until now, they were ignored in all previous 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 4](#) (but with a way to put page numbers on the right which is more like the method used by L^AT_EX2e's `\@dottedtocline`; and multi-line headings now leave empty the area underneath the numbers contrarily to the code from [section 4](#)). And after a few minutes for choosing lengths (now that this has been done once, it can be recycled easily) we get:

A TOC using depth tags

I Overview	4
II Arbitrarily many TOCs, and local ones too	8
III Surprising uses of <code>etoc</code>	16
IV Commands for the toc line styles	28
V Commands for the toc display style	33
23 Specifying the toc display style	34
23.1 The command <code>\etocsettocstyle</code>	34
23.2 The commands <code>\etocmulticolstyle</code> , <code>\etocmulticol</code> , and <code>\etoclocalmulticol</code>	34
23.2.1 The command <code>\etocstyle</code>	35
23.2.1.1 <code>\etocstylewithmarks</code>	35
23.2.1.2 Do we really want paragraph entries in the TOC?	35
23.2.1.3 really?	35
23.3 The commands <code>\etocruledstyle</code> , <code>\etocruled</code> and <code>\etoclocalruled</code>	35
23.4 The commands <code>\etocframedstyle</code> , <code>\etocframed</code> , and <code>\etoclocalframed</code>	35
23.5 Headings, titles, <code>\etocoldpar</code> , <code>\etocinnertopsep</code>	36
23.6 The compatibility mode <code>\etocstandarddisplaystyle</code>	36
23.7 The command <code>\etocinline</code>	36
24 Starred variants of the <code>\tableofcontents</code> etc... commands	37
25 Table of contents for this part	37
25.1 Testing the compatibility mode	37

25.2 A framed display	38
25.3 A (crazy) inline display	39
VI Using and customizing <i>etoc</i>	40
VII Tips	50

```

\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*{\etoclineleaders}
{
  \hbox{\normalfont\normalsize\hbox to .75ex {\hss.\hss}}}
\newcommand*{\EndParWithPagenoInMarginAndLeaders}
{
  \nobreak\leaders\etoclineleaders\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 {code}        {none}
\renewcommand\etocdisplayrule {\hrule height 3pt\relax }
\renewcommand\etocdisplayrulecolorcmd {\color{blue}}
\renewcommand\etocaftercontentshook
  {\medskip\begin{group} \color{blue}\hrule height 3pt \end{group} }
\etocruledstyle [1]{\Large\bfseries
  \fbox{\makebox[8cm]{A TOC using depth tags}}}

\sloppy
\tableofcontents
\endgroup

```

30.6. Typesetting the TOC as a table

Tabular cells create groups; definitions of `\etocname`, `\etocnumber`, `\etocpage` made by **etoc** are local, thus if the first cell of a row is where the `.toc` file line is read and `\etocname` and `\etocnumber` get defined, these definitions will be lost after the tabulation character `&`. The source code of **etoc** does allow the “begin” and “finish” parts of the line styles to open or close groups, but the possibility of a group closed by a `&` for example in the “prefix” part of the line style is not anticipated.

Another, similar, matter regards the redefinition of `\contentsline` made by **etoc** each time it typesets a TOC. It is done after the first argument to `\etocsettocstyle` has been executed, hence if this argument opens a tabular, the **etoc** redefinition of `\contentsline` will be done in the first cell of the first row and get lost thereafter.

Thus one can not typeset a TOC directly as a tabular, with the opening and closing specified via `\etocsettocstyle`, and using `&` and `\\` inside the level styles.

The **etoc** source code could be modified to make some relevant definitions global, but I decided against it. Typesetting as a tabular (or, rather, `longtable`) will appeal only to advanced users, and advanced users can employ the technique explained in [section 17](#). Here is an example, giving again a fully hyperlinked table of contents. The code follows, after the table.

TABLE OF CONTENTS		
1	Foreword	1
I	Overview	4
2	Do I need to be a geek to use etoc ?	4
	2.1 Limitations in the use of list environments for tables of contents	5
3	Line styles and toc display style	5
	3.1 <code>\etocsetstyle</code> for the line styles	5
	3.2 <code>\etocsettocstyle</code> for the toc display	6
	3.3 Compatibility mode	6
4	A first example	6
5	Linked list of the main package commands	8
II	Arbitrarily many TOCs, and local ones too	8
6	Labeling and reusing elsewhere	9

30.6. Typesetting the TOC as a table

7	A powerful functionality of etoc : the re-assignment of levels with <code>\etocsetlevel</code>	10
8	The <code>\etocsettocdepth</code> and <code>\etocsetnext-tocdepth</code> commands	11
	8.1 The hyperref option <i>bookmarks-depth</i>	11
9	The command <code>\etocsettocdepth.toc</code>	12
	9.1 The commands <code>\etocobeytoc-tocdepth</code> and <code>\etocignoretoc-tocdepth</code>	12
10	The commands <code>\etocdepthtag.toc</code> and <code>\etocsettagdepth</code>	12
	10.1 The commands <code>\etocobeydepth-tags</code> and <code>\etocignoredepthtags</code>	13
11	Adding commands to the <code>.toc</code> file	13
	11.1 The hyperref option <i>hidelinks</i>	14
12	Two Examples	14
	12.1 A Beautiful Thesis example	14
	12.2 A TOC with a background color	16
III	Surprising uses of etoc	16
13	The TOC of TOCs	17
14	Arbitrary “Lists Of...”, <code>\etoc toc contentsline</code>	18
15	A TOC with a fancy layout	19
16	Another compatibility mode	20
17	The TOC as a tree	23
18	The TOC as a molecule	25
IV	Commands for the toc line styles	28
19	The <code>\etocsetstyle</code> command	28
	19.1 The <code>\etocname</code> and <code>\etocpage</code> commands	28
	19.2 The <code>\etocskipfirstprefix</code> command	29
	19.3 The <code>\etocnumber</code> command	30
	19.4 The <code>\etocthename</code> , <code>\etocthenumber</code> , and <code>\etocthepage</code> commands	30
	19.5 The <code>\etoclink</code> command	31
	19.6 The <code>\etocthelinkedname</code> , <code>\etocthelinkednumber</code> , <code>\etocthelinkedpage</code> and <code>\etocthelink</code> commands	31
20	The <code>\etocsetlevel</code> command	31
21	Scope of commands added to the <code>.toc</code> file	32
	21.1 Testing the scope	32
	21.2 This is a (pale) red subsection for illustrative purposes	32
22	Am I also red?	33

V	Commands for the toc display style	33
23	Specifying the toc display style	34
	23.1 The command <code>\etocsettocstyle</code>	34
	23.2 The commands <code>\etocmulticol-</code> <code>style</code> , <code>\etocmulticol</code> , and <code>\etoc-</code> <code>localmulticol</code>	34
	23.3 The commands <code>\etocruledstyle</code> , <code>\etocruled</code> and <code>\etoclocal-</code> <code>ruled</code>	35
	23.4 The commands <code>\etocframed-</code> <code>style</code> , <code>\etocframed</code> , and <code>\etoclocalframed</code>	35
	23.5 Headings, titles, <code>\etocoldpar</code> , <code>\etocinnertopsep</code>	36
	23.6 The compatibility mode <code>\etoc-</code> <code>standarddisplaystyle</code>	36
	23.7 The command <code>\etocinline</code>	36
24	Starred variants of the <code>\tableofcontents</code> etc... commands	37
25	Table of contents for this part	37
	25.1 Testing the compatibility mode	37
	25.2 A framed display	38
	25.3 A (crazy) inline display	39
VI	Using and customizing <code>etoc</code>	40
26	Summary of the main style commands	41
	26.1 Setting up local styles	41
	26.2 Setting up toc display styles	41
	26.3 Displaying tables of contents	42
	26.4 Labels and references	42
27	The package default line styles: <code>\etoc-</code> <code>defaultlines</code>	42
28	Customizing <code>etoc</code>	45
	28.1 Customizing the <code>etoc</code> pre-defined line styles	45
	28.2 Customizing the toc display styles	46
29	One more example of colored TOC layout	47
VII	Tips	50
30	... and tricks	50
	30.1 Hacking framed parboxes	50
	30.2 Inverting the levels	51
	30.3 All subsections of this document	51
	30.4 Displaying statistics	52
	30.5 Using depth tags	54
	30.6 Typesetting the TOC as a table	56
	30.7 Compatibility with other packages	60
	30.8 T _E Xnical matters	61
	30.9 Errors and catastrophes	61

VIII	The code	62
31	Timestamp	62
32	License, installation	62
33	Change history	63
34	Implementation	63

```

\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
    \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\\}%
}
\newcommand*\{PrepareSubsection}{%
  \toks0 \expandafter{\etocthelinkednumber}%
  \toks2 \expandafter{\etocthelinkedname}%
  \toks4 \expandafter{\etocthelinkedpage}%
  \edef\toctablepiece{\noexpand\centering\the\toks0
    &\the\toks2 &\noexpand\itshape\the\toks4 \noexpand\\}%
}
\etocsetstyle{part}{}{}
  {\PreparePart \appendtotok\toctabletok\toctablepiece}{}
\etocsetstyle{section}{}{}
  {\PrepareSection \appendtotok\toctabletok\toctablepiece}{}
\etocsetstyle{subsection}{}{}
% each line does its own tabular, to allow page breaks in the long table
{\toctabletok\expandafter
  {\the\toctabletok &\small\begin{tabular}{@{}p{1cm}@{}p{5cm}r@{}}}%
  \PrepareSubsection\appendtotok\toctabletok\toctablepiece
  \toctabletok\expandafter{\the\toctabletok\end{tabular}&\\}}
{}

\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

```

30.7. Compatibility with other packages

etoc loads the package `multicol`. It is `hyperref` aware and hopefully `hyperref` compatible! It doesn't matter whether **etoc** or `hyperref` is loaded first.

The contents of the `.toc` file (if it already exists) are read into memory by **etoc** once, at the time of `\usepackage{etoc}`. 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 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 earlier⁴¹ just before the closing part of the toc display style (and thus still within a group.)

etoc used to be incompatible with package `tocvsec2`⁴²; 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.

etoc expects the document sectioning units to write their data into the file having extension `.toc`, in the form of lines containing the `\contentsline` command and its arguments. The macros `\etocname`, `\etocnumber`, and `\etocpage` contain the `hyperref` links, if present (note that the `linktoc=all` option of `hyperref` tells it to put a link also in the page number corresponding to a given toc entry). For example, the tables of contents of the present document are all fully linked.

It 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`⁴³. For the advanced uses such as what is done in section 18 it is important to know that `tableof` adds one level of grouping inside the `.toc` file itself. So when one needs to make some information `\global`, one can not wait to be at the level of the second argument of `\etocsettocstyle`, as `tableof` will already have closed the group then. The `\global` things (on volatile data)

⁴¹contrarily to `\etocaftertochook`, `\etocaftercontentshook` is not executed if the `tocdepth` did not allow the printing of the TOC.

⁴²<http://ctan.org/pkg/tocvsec2>

⁴³<http://ctan.org/pkg/tableof>

must be done at the latest in the *<finish>* part of the top (or last) sectioning level used. This only applies of course to `\tableofcontents` or `\localtableofcontents` following the `\nexttocwithtags{required}{forbidden}` command from `tableof`.

And when the commands `\tableof` or `\tablenotof` of package `tableof` are used, they typeset the table of contents according to the document class defaults: to benefit from the **etoc** styles, it is mandatory to use either `\tableofcontents`, or `\localtableofcontents` or one of the other **etoc** commands, and `tableof` (v1.1) will influence the outcome only if `\nexttocwithtags{required}{forbidden}` was added before the table of contents typesetting command.

30.8. 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.

On the other hand `\etocthenname`, `\etocthenumber`, `\etocthepage` do not represent `hyperref` links, and are *not* protected against expansion.

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

30.9. 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 VIII.

The code

31. Timestamp

This is the documentation as of 2013/12/03, printed from the source file with the time stamp 03-12-2013 at 17:49:49 CET. The package version is v1.07j, of 2013/12/03.

32. License, installation

```
% Copyright (C) 2012-2013 by Jean-Francois Burnol <jfbu at free dot fr>
%
%   This Work may be distributed and/or modified under the
%   conditions of the LaTeX Project Public License, 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.
%
% This Work has the LPPL maintenance status Author-Maintained.
%
% The Author of this Work is
%       Jean-Francois Burnol <jfbu at free dot fr>.
%
% This Work consists of the main source file etoc.dtx and the
% derived files etoc.sty, etoc.dvi, etoc.pdf, etoc.ins, etoc.tex
%
% Extraction of the package (.sty) and driver (.tex) files:
% - if etoc.ins is present:   tex etoc.ins
% - without etoc.ins:       tex etoc.dtx
%
% It is also possible to run latex/pdflatex directly on etoc.dtx
%
% At least three ways to produce etoc.pdf:
% (1) latex etoc.dtx (three times), then dvips, then ps2pdf
% (2) pdflatex etoc.dtx (three times)
% (3) latex etoc.tex (three times), then dvipdfmx
%
% Method (3) produces the smallest pdf files.
% Options can be set in etoc.tex:
% - scrdoc class options (paper size, font size, ...)
% - with or without source code,
% - with dvipdfmx or with latex/pdflatex.
%
% Installation:
%
%   etoc.sty->TDS:tex/latex/etoc/etoc.sty
%   etoc.dtx->TDS:source/latex/etoc/etoc.dtx
%   etoc.pdf->TDS:doc/latex/etoc/etoc.pdf
```

```
%
% etoc.tex, etoc.ins, etoc.aux, etoc.toc, etc... may be discarded
%
```

33. Change history

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`, `\etocobeytocdepth`, `\etocignoretocdepth` which emulate part of `tocvsec2` functionality ; measures to make `tocvsec2` partially compatible with `etoc`.

new commands `\etocsetnexttocdepth`, `\invisibletableofcontents`, `\invisiblelocaltableofcontents`.

switched from `tikz-qtrees` to `forest` for the first ‘toc as tree’ example.

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

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

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

v1.07f [2013/03/07]

new macros `\etocthelinkedname`, `\etocthelinkednumber`, `\etocthelinkedpage`, and `\etocthelink`.

v1.07e [2013/03/01]

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

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

more such examples added 2013/03/03.

v1.07d [2013/02/24]

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

v1.07b [2013/02/02]

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

additional examples in the documentation.

v1.07 [2013/01/29]

new commands:

`\etocthename`, `\etocthenumber`, `\etocthepage`, `\etoclink`,
`\etococcontentsline`, `\etococcontentsline*`
`\etocnopar`, `\etocaftercontentshook`

modified command: `\etocmulticolstyle`

new documentation section “Surprising uses of `etoc`” which explains how to do “Lists of arbitrary things”, in addition to the tables of contents.

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.

34. Implementation

```
1 \ProvidesPackage{etoc}
```

34. Implementation

```
2 [2013/12/03 v1.07j Completely customisable TOCs (jfb)]
3 \NeedsTeXFormat{LaTeX2e}
4 \RequirePackage{multicol}
5 \DeclareOption*{\PackageWarning{etoc}{Option ‘\CurrentOption’ is unknown.}}
6 \ProcessOptions\relax

placeholder for comments
7 \newtoks\Etoc@toctoks
8 \def\Etoc@par{\par}
9 \def\etocinline{\def\Etoc@par{}}
10 \let\etocnopar\etocinline
11 \newif\ifEtoc@jj % book
12 \newif\ifEtoc@j % part
13 \newif\ifEtoc@ % chapter
14 \newif\ifEtoc@i % section
15 \newif\ifEtoc@ii % subsection
16 \newif\ifEtoc@iii % subsubsection
17 \newif\ifEtoc@iv % paragraph
18 \newif\ifEtoc@v % subparagraph
19 \newif\ifEtoc@number
20 \newif\ifEtoc@hyperref
21 \newif\ifEtoc@parskip % 1.07d
22 \newif\ifEtoc@tocwithid
23 \newif\ifEtoc@standard
24 \newif\ifEtoc@part

placeholder for comments
25 \newif\ifEtoc@localtoc
26 \newif\ifEtoc@skipthisone
27 \newif\ifEtoc@stoptoc
28 \newif\ifEtoc@notactive
29 \newcounter{etoc@tocid}
30 \newif\ifEtoc@mustclosegroup
31 \def\etoc@{\etoc@}

1.07g uses a second counter; this could be avoided, but ok, let’s not be that strict.
32 \newcounter{etoc@tocdepth}% 1.07g

placeholder for comments
33 \@ifclassloaded{memoir}{\def\Etoc@minf{-\thr@@}}{\def\Etoc@minf{-\tw@}}
34 \def\Etoc@@minustwo@@{-\tw@}
35 \let\Etoc@@minusone@@\m@ne
36 \let\Etoc@@zero@@ \z@
37 \let\Etoc@@one@@ \@ne
38 \let\Etoc@@two@@ \tw@
39 \let\Etoc@@three@@ \thr@@
40 \chardef\Etoc@@four@@ 4
41 \chardef\Etoc@@five@@ 5
42 \chardef\Etoc@@six@@ 6
43 \let\Etoc@localtop\Etoc@@minustwo@@
44 \def\Etoc@@minustwo@{minustwo}
45 \def\Etoc@@minusone@{minusone}
46 \def\Etoc@@zero@ {zero}
47 \def\Etoc@@one@ {one}
48 \def\Etoc@@two@ {two}
49 \def\Etoc@@three@ {three}
50 \def\Etoc@@four@ {four}
51 \def\Etoc@@five@ {five}

1.07g
```



```

52 \expandafter\def\csname Etoc@-3@@\endcsname {-\thr@@}
53 \expandafter\let\csname Etoc@-2@@\endcsname \Etoc@@minustwo@@
54 \expandafter\let\csname Etoc@-1@@\endcsname \Etoc@@minusone@@
55 \expandafter\let\csname Etoc@0@@\endcsname \Etoc@@zero@@
56 \expandafter\let\csname Etoc@1@@\endcsname \Etoc@@one@@
57 \expandafter\let\csname Etoc@2@@\endcsname \Etoc@@two@@
58 \expandafter\let\csname Etoc@3@@\endcsname \Etoc@@three@@
59 \expandafter\let\csname Etoc@4@@\endcsname \Etoc@@four@@
60 \expandafter\let\csname Etoc@5@@\endcsname \Etoc@@five@@
61 \expandafter\let\csname Etoc@6@@\endcsname \Etoc@@six@@
62 \let\Etoc@all@@ \Etoc@@five@@
63 \let\Etoc@none@@ \Etoc@minf

placeholder for comments
64 \let\Etoc@levellist\@empty
65 \def\Etoc@newlevel#1{%
66   \def\Etoc@levellist@elt{\noexpand\Etoc@levellist@elt\noexpand}%
67   \edef\Etoc@levellist{\Etoc@levellist\Etoc@levellist@elt#1}}
68 \def\etocsetlevel#1#2{%
69   \expandafter\Etoc@newlevel\csname l@#1\endcsname
70   \ifcase#2\relax
71     \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@zero@@
72     \expandafter\let \csname Etoc@#1@\endcsname\Etoc@@zero@
73   \or
74     \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@one@@
75     \expandafter\let \csname Etoc@#1@\endcsname\Etoc@@one@
76   \or
77     \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@two@@
78     \expandafter\let \csname Etoc@#1@\endcsname\Etoc@@two@
79   \or
80     \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@three@@
81     \expandafter\let \csname Etoc@#1@\endcsname\Etoc@@three@
82   \or
83     \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@four@@
84     \expandafter\let \csname Etoc@#1@\endcsname\Etoc@@four@
85   \or
86     \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@five@@
87     \expandafter\let \csname Etoc@#1@\endcsname\Etoc@@five@
88   \or
89     \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@six@@
90   \else
91     \ifnum#2=\m@ne
92       \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@minusone@@
93       \expandafter\let \csname Etoc@#1@\endcsname\Etoc@@minusone@
94     \else
95       \ifnum#2=-\tw@
96         \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@minustwo@@
97         \expandafter\let \csname Etoc@#1@\endcsname\Etoc@@minustwo@
98       \else
99         \PackageWarning{etoc}
100           {unexpected value ‘#2’ in \string\etocsetlevel.^}%
101           Should be -2,-1, 0, 1, 2, 3, 4, 5, or 6. Set to 6 (=ignored)}%
102       \expandafter\let\csname Etoc@#1@@\endcsname\Etoc@@six@@
103 \fi\fi\fi}
104 \etocsetlevel{book}{-2}
105 \etocsetlevel{part}{-1}
106 \etocsetlevel{chapter}{0}
107 \etocsetlevel{section}{1}

```

34. Implementation

```
108 \etocsetlevel{subsection}{2}
109 \etocsetlevel{subsubsection}{3}
110 \etocsetlevel{paragraph}{4}
111 \etocsetlevel{subparagraph}{5}
placeholder for comments
112 \def\Etoc@setflags #1{%
113   \ifcase #1\relax
114     \global\Etoc@vfalse
115     \global\Etoc@ivfalse
116     \global\Etoc@iiifalse
117     \global\Etoc@iifalse
118     \global\Etoc@ifalse
119     \global\Etoc@true
120   \or
121     \global\Etoc@vfalse
122     \global\Etoc@ivfalse
123     \global\Etoc@iiifalse
124     \global\Etoc@iifalse
125     \global\Etoc@itrue
126   \or
127     \global\Etoc@vfalse
128     \global\Etoc@ivfalse
129     \global\Etoc@iiifalse
130     \global\Etoc@iitrue
131   \or
132     \global\Etoc@vfalse
133     \global\Etoc@ivfalse
134     \global\Etoc@iiitrue
135   \or
136     \global\Etoc@vfalse
137     \global\Etoc@ivtrue
138   \or
139     \global\Etoc@vtrue
140   \else
141     \ifnum#1=\m@ne
142       \global\Etoc@vfalse
143       \global\Etoc@ivfalse
144       \global\Etoc@iiifalse
145       \global\Etoc@iifalse
146       \global\Etoc@ifalse
147       \global\Etoc@false
148       \global\Etoc@jtrue
149     \else
150       \global\Etoc@vfalse
151       \global\Etoc@ivfalse
152       \global\Etoc@iiifalse
153       \global\Etoc@iifalse
154       \global\Etoc@ifalse
155       \global\Etoc@false
156       \global\Etoc@jfalse
157       \global\Etoc@jjtrue
158     \fi
159   \fi}
placeholder for comments
160 \AtBeginDocument{%
161 \@ifpackageloaded{parskip}{\Etoc@parskiptrue}{}%
162 \@ifpackageloaded{hyperref}{\Etoc@hyperreftrue
```

```

163             \def\Etoc@et@hop#1#2#3#4#5{#1{#3}{#4}{#5}#2}%
164             \long\def\Etoc@gobblesixorfive#1#2#3#4#5#6{}}
165         {\def\Etoc@et@hop#1#2#3#4{#1{#3}{#4}#2}%
166         \long\def\Etoc@gobblesixorfive#1#2#3#4#5{}}%
167 }

placeholder for comments
168 \def\Etoc@swa#1{%
169     \Etoc@et@hop
170     {\Etoc@savedcontentsline{#1}}
171     {\Etoc@prefix\Etoc@contents}}
172 \def\Etoc@swb#1{%
173     \Etoc@et@hop
174     {\Etoc@savedcontentsline{#1}}
175     {\Etoc@contents}}
176 \let\etocskipfirstprefix\@thirdofthree

placeholder for comments
177 \def\Etoc@etoccontentsline#1{%
178     \global\expandafter\let\expandafter\Etoc@tmp\csname Etoc@#1@\endcsname
179     \Etoc@skipthisonefalse
180     \let\Etoc@next\Etoc@gobblesixorfive
181     \ifnum\Etoc@tmp=\Etoc@@six@@
182         \Etoc@skipthisonetrue
183     \else
184         \ifEtoc@localtoc
185             \let\Etoc@prenext\relax
186             \ifEtoc@stoptoc
187                 \Etoc@skipthisonetrue
188             \fi
189             \ifnum\Etoc@tmp<\Etoc@localtop
190                 \def\Etoc@prenext{\global\Etoc@stoptoctrue}%
191                 \Etoc@skipthisonetrue
192             \fi
193             \ifEtoc@notactive
194                 \def\Etoc@prenext{\Etoc@setflags{\Etoc@tmp}}%
195                 \Etoc@skipthisonetrue
196             \fi
197             \Etoc@prenext
198         \fi
199     \fi
200     \ifnum\c@tocdepth<\Etoc@tmp\relax\else
201     \ifEtoc@skipthisone\else
202     \global\let\Etoc@next\relax
203     \ifcase\Etoc@tmp
204         \ifEtoc@v \Etoc@end@five\fi
205         \ifEtoc@iv \Etoc@end@four\fi
206         \ifEtoc@iii \Etoc@end@three\fi
207         \ifEtoc@ii \Etoc@end@two\fi
208         \ifEtoc@i \Etoc@end@one\fi
209         \ifEtoc@ \else \def\Etoc@next{\Etoc@begin@zero}\fi
210         \def\Etoc@contents{\Etoc@contents@zero}%
211         \def\Etoc@prefix{\Etoc@prefix@zero}%
212     \or
213         \ifEtoc@v \Etoc@end@five\fi
214         \ifEtoc@iv \Etoc@end@four\fi
215         \ifEtoc@iii \Etoc@end@three\fi
216         \ifEtoc@ii \Etoc@end@two\fi
217         \ifEtoc@i \else \def\Etoc@next{\Etoc@begin@one}\fi

```

34. Implementation

```
218     \def\Etoc@contents{\Etoc@contents@one}%
219     \def\Etoc@prefix{\Etoc@prefix@one}%
220   \or
221     \ifEtoc@v \Etoc@end@five\fi
222     \ifEtoc@iv \Etoc@end@four\fi
223     \ifEtoc@iii \Etoc@end@three\fi
224     \ifEtoc@ii \else \def\Etoc@next{\Etoc@begin@two}\fi
225     \def\Etoc@contents{\Etoc@contents@two}%
226     \def\Etoc@prefix{\Etoc@prefix@two}%
227   \or
228     \ifEtoc@v \Etoc@end@five\fi
229     \ifEtoc@iv \Etoc@end@four\fi
230     \ifEtoc@iii \else \def\Etoc@next{\Etoc@begin@three}\fi
231     \def\Etoc@contents{\Etoc@contents@three}%
232     \def\Etoc@prefix{\Etoc@prefix@three}%
233   \or
234     \ifEtoc@v \Etoc@end@five\fi
235     \ifEtoc@iv \else \def\Etoc@next{\Etoc@begin@four}\fi
236     \def\Etoc@contents{\Etoc@contents@four}%
237     \def\Etoc@prefix{\Etoc@prefix@four}%
238   \or
239     \ifEtoc@v \else \def\Etoc@next{\Etoc@begin@five}\fi
240     \def\Etoc@contents{\Etoc@contents@five}%
241     \def\Etoc@prefix{\Etoc@prefix@five}%
242   \else
243     \ifnum\Etoc@tmp=\m@ne
244       \ifEtoc@v \Etoc@end@five\fi
245       \ifEtoc@iv \Etoc@end@four\fi
246       \ifEtoc@iii \Etoc@end@three\fi
247       \ifEtoc@ii \Etoc@end@two\fi
248       \ifEtoc@i \Etoc@end@one\fi
249       \ifEtoc@ \Etoc@end@zero\fi
250       \ifEtoc@j \else \def\Etoc@next{\Etoc@begin@minusone}\fi
251       \def\Etoc@contents{\Etoc@contents@minusone}%
252       \def\Etoc@prefix{\Etoc@prefix@minusone}%
253     \else
254       \ifEtoc@v \Etoc@end@five\fi
255       \ifEtoc@iv \Etoc@end@four\fi
256       \ifEtoc@iii \Etoc@end@three\fi
257       \ifEtoc@ii \Etoc@end@two\fi
258       \ifEtoc@i \Etoc@end@one\fi
259       \ifEtoc@ \Etoc@end@zero\fi
260       \ifEtoc@j \Etoc@end@minusone\fi
261       \ifEtoc@jj \else \def\Etoc@next{\Etoc@begin@minustwo}\fi
262       \def\Etoc@contents{\Etoc@contents@minustwo}%
263       \def\Etoc@prefix{\Etoc@prefix@minustwo}%
264       \fi
265     \fi
266     \ifnum\Etoc@tmp=\m@ne\Etoc@parttrue\else\Etoc@partfalse\fi
267     \Etoc@setflags{\Etoc@tmp}%
268   \fi\fi
269   \Etoc@next
270   \@firstoftwo{\Etoc@swa{#1}}{\Etoc@swb{#1}}
```

[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 left unexpanded. We have to spend some time with delimited macros to dis-

entangle the .toc data, and reconstruct the possible hyperref data. If the page number is not hyperlinked, \etocpage does *not* add the link found possibly in the name.

Then 1.07 added \etocthename, \etocthenumber, \etocthepage which are left fragile and do not have the links data, and \etoclink{\linkname} which is robust and reconstructs an arbitrarily named link. A need (for things like building up a token list to be used in a tikzpicture) arose later to have some form of the link which could be saved by a simple command like one can do \global\let\lastname\etocthename, and avoid having to manipulate \Hy@tocdestname. So 1.07f adds \etocthelinkedname, \etocthelinkednumber, \etocthelinkedpage, \etocthelink: they use \hyperlink with an expanded \Hy@tocdestname.

One could now define \etocname, etc ... to be the robust versions of \etocthelinkedname, etc ..., but the original definitions are kept by sentimentalism. 1.07f also adds \leavevmode to \etoclink which should have been done earlier, as it was included in \etocname etc...

attention, \@namedef{A}{B} and not \@namedef{A} {B} !! on the other hand this gives a simple way to insert a space as the first token in the paramaters. For \Etoc@again (which appears later in the code), a \@firstofone construct is however the simplest of all.

```

271 \def\Etoc@lxyz #1#2{%
272   \@namedef {etoclink }{\leavevmode}%           fall-back
273   \let\etocthelink      \@empty %               fall-back
274   \@namedef {etocname }{\leavevmode #1}% fall-back (perhaps linked)
275   \def\etocthename      {#1}% (if link, will be removed later)
276   \def\etocthelinkedname {#1}% will probably get redefined
277   \@namedef {etocpage }{\leavevmode #2}% (perhaps linked)
278   \Etoc@getthepage #2\etoc@ % defines also \etocthelinkedpage (and \etoclink)
279   \Etoc@getnb #1\relax\relax\etoc@ % gets number *and* name, and \etoclink
280   \ifEtoc@number\else
281     \ifEtoc@part
282       \Etoc@getit #1\hspace\relax\etoc@ % additional job for parts
283     \fi
284   \fi}

placeholder for comments
285 \def\Etoc@getthepage #1{%
286   \let\Etoc@next\Etoc@getthepage@nohyp
287   \ifEtoc@hyperref\ifx #1\hyper@linkstart
288     \let\Etoc@next\Etoc@getthepage@hyp
289   \fi\fi
290   \Etoc@next #1%
291 }
292 \def\Etoc@getthepage@nohyp #1\etoc@ {%
293   \def\etocthepage {#1}%
294   \def\etocthelinkedpage {#1}%
295 }
296 \def\Etoc@getthepage@hyp #1#2#3#4#5\etoc@ {%
297   \@namedef{etoclink } ##1{\leavevmode #1{#2}{#3}{##1}{#5}%
298   \edef\etocthelink ##1{\noexpand\hyperlink {#3}{##1}}%
299   \def\etocthepage {#4}%
300   \toks@ {#4}%
301   \edef\etocthelinkedpage {\noexpand\hyperlink {#3}{\the\toks@}}%
302 }
303 %
304 \def\Etoc@getnb #1{%
305   \let\Etoc@next\Etoc@getnb@nohyp
306   \ifEtoc@hyperref\ifx #1\hyper@linkstart
307     \let\Etoc@next\Etoc@getnb@hyp
308   \fi\fi
309   \Etoc@next #1%
310 }

```

34. Implementation

```
311 %
312 \def\Etoc@getit #1{%
313   \let\Etoc@next\Etoc@getit@nohyp
314   \ifEtoc@hyperref\ifx #1\hyper@linkstart
315     \let\Etoc@next\Etoc@getit@hyp
316   \fi\fi
317   \Etoc@next #1%
318 }

placeholder for comments
319 \def\Etoc@getnb@nohyp #1#2#3\etoc@ {%
320   \def\Etoc@getname ##1\relax\relax\etoc@ {%
321     \@namedef {etocname }{\leavevmode ##1}%
322     \def\etocthename {##1}%
323     \def\etocthelinkedname {##1}%
324   }%
325   \ifx #1\numberline
326     \@namedef {etocnumber }{\leavevmode #2}%
327     \def\etocthenumber {#2}%
328     \def\etocthelinkednumber {#2}%
329     \Etoc@numbertrue
330     \Etoc@getname #3\etoc@
331   \else % then \etocthename and \etocthelinkedname already defined
332     \@namedef {etocnumber }{\leavevmode}%
333     \let\etocthenumber \@empty
334     \let\etocthelinkednumber \@empty
335     \Etoc@numberfalse
336   \fi
337 }

placeholder for comments
338 \def\Etoc@getnb@hyp #1#2#3#4#5#6\etoc@ {%
339   \def\Etoc@getname ##1\relax\relax\etoc@ {%
340     \@namedef {etocname }{\leavevmode #1{#2}{#3}{##1}#5}%
341     \def\etocthename {##1}%
342     \toks@ {##1}%
343     \edef\etocthelinkedname {\noexpand\hyperlink {#3}{\the\toks@}}%
344   }%
345   \def\Etoc@getnbr ##1##2##3\etoc@ {%
346     \ifx ##1\numberline
347       \@namedef {etocnumber }{\leavevmode #1{#2}{#3}{##2}#5}%
348       \def\etocthenumber {##2}%
349       \toks@ {##2}%
350       \edef\etocthelinkednumber {\noexpand\hyperlink {#3}{\the\toks@}}%
351       \Etoc@numbertrue
352       \Etoc@getname ##3\etoc@
353     \else
354       \@namedef {etocnumber }{\leavevmode}%
355       \let\etocthenumber \@empty
356       \let\etocthelinkednumber \@empty
357       \Etoc@numberfalse
358       \def\etocthename {#4}%
359       \toks@ {#4}%
360       \edef\etocthelinkedname {\noexpand\hyperlink {#3}{\the\toks@}}%
361     \fi
362   }%
363   \@namedef {etoclink }##1{\leavevmode #1{#2}{#3}{##1}#5}%
364   \edef\etocthelink ##1{\noexpand\hyperlink {#3}{##1}}%
365   \Etoc@getnbr #4\relax\relax\etoc@
```

```

366 }
placeholder for comments
367 \def\Etoc@getit@nohyp #1\hspace#2#3\etoc@ {%
368   \def\Etoc@getname ##1\hspace\relax\etoc@ {%
369     \@namedef {etocname }{\leavevmode ##1}%
370     \def\etocthename      {##1}%
371     \def\etocthelinkedname {##1}%
372   }%
373   \ifx\relax#2\else
374     \@namedef {etocnumber }{\leavevmode #1}%
375     \def\etocthenumber      {#1}%
376     \def\etocthelinkednumber {#1}%
377     \Etoc@numbertrue
378     \Etoc@getname #3\etoc@
379   \fi
380 }
placeholder for comments
381 \def\Etoc@getit@hyp #1#2#3#4#5#6\etoc@ {%
382   \def\Etoc@getname ##1\hspace\relax\etoc@ {%
383     \@namedef {etocname }{\leavevmode #1{#2}{#3}{##1}#5}%
384     \def\etocthename {##1}%
385     \toks@            {##1}%
386     \edef\etocthelinkedname {\noexpand\hyperlink {#3}{\the\toks@}}%
387   }%
388   \def\Etoc@getnbr ##1\hspace##2##3\etoc@ {%
389     \ifx\relax##2\else
390       \@namedef {etocnumber }{\leavevmode #1{#2}{#3}{##1}#5}%
391       \def\etocthenumber {##1}%
392       \toks@            {##1}%
393       \edef\etocthelinkednumber {\noexpand\hyperlink {#3}{\the\toks@}}%
394       \Etoc@numbertrue
395       \Etoc@getname ##3\etoc@
396     \fi
397   }%
398   \Etoc@getnbr #4\hspace\relax\etoc@
399 }
placeholder for comments
400 \let\etocthename \@empty
401 \let\etocthenumber \@empty
402 \let\etocthepage \@empty
403 \let\etocthelinkedname \@empty
404 \let\etocthelinkednumber \@empty
405 \let\etocthelinkedpage \@empty
406 \let\etocthelink \@empty
407 \DeclareRobustCommand*\etocname {}
408 \DeclareRobustCommand*\etocnumber {}
409 \DeclareRobustCommand*\etocpage {}
410 \DeclareRobustCommand*\etoclink {}
411 \DeclareRobustCommand*\etocifnumbered
412   {\ifEtoc@number\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi}
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

```

34. Implementation

```
418      {\expandafter\the\expandafter\Etoc@toctoks\Etoc@buffer}%
419      \expandafter\Etoc@readtoc
420    \fi
421  }
422  \Etoc@toctoks {}% (superfluous, but for clarity)
423  \IfFileExists{\jobname .toc}
424    {{\endlinechar=-1
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 }

placeholder for comments
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  \ifEtoc@standard
459    \etoc@setstyle{@minustwo}{}{}{}{}%
460    \etoc@setstyle{@minusone}{}{}{}{}%
461    \etoc@setstyle{@zero}{}{}{}{}%
462    \etoc@setstyle{@one}{}{}{}{}%
463    \etoc@setstyle{@two}{}{}{}{}%
464    \etoc@setstyle{@three}{}{}{}{}%
465    \etoc@setstyle{@four}{}{}{}{}%
466    \etoc@setstyle{@five}{}{}{}{}%
467  \else
468    \def\Etoc@levellist@elt##1{\let##1\Etoc@lxyz}%
469    \Etoc@levellist
470    \let\booknumberline\numberline
471    \let\partnumberline\numberline
472    \let\chapternumberline\numberline
```



```

473 \fi
474 \the\Etoc@toctoks
475 \ifEtoc@notactive\else
476   \ifEtoc@v   \Etoc@end@five\fi
477   \ifEtoc@iv  \Etoc@end@four\fi
478   \ifEtoc@iii \Etoc@end@three\fi
479   \ifEtoc@ii  \Etoc@end@two\fi
480   \ifEtoc@i   \Etoc@end@one\fi
481   \ifEtoc@    \Etoc@end@zero\fi
482   \ifEtoc@j   \Etoc@end@minusone\fi
483   \ifEtoc@jj  \Etoc@end@minustwo\fi
484 \fi}

placeholder for comments
485 \def\Etoc@@startlocaltoc#1#2{%
486 \let\Etoc@next\relax
487   \ifEtoc@localtoc
488     \ifEtoc@notactive
489       \ifnum #1=#2\relax
490         \ifEtoc@jj \let\Etoc@localtop\Etoc@@minusone@@ \fi
491         \ifEtoc@j  \let\Etoc@localtop\Etoc@@zero@@ \fi
492         \ifEtoc@   \let\Etoc@localtop\Etoc@@one@@ \fi
493         \ifEtoc@i  \let\Etoc@localtop\Etoc@@two@@ \fi
494         \ifEtoc@ii \let\Etoc@localtop\Etoc@@three@@ \fi
495         \ifEtoc@iii \let\Etoc@localtop\Etoc@@four@@ \fi
496         \ifEtoc@iv \let\Etoc@localtop\Etoc@@five@@ \fi
497         \ifEtoc@v  \let\Etoc@localtop\Etoc@@six@@ \fi
498       \def\Etoc@next{\global\Etoc@notactivefalse
499         \global\Etoc@vfalse
500         \global\Etoc@ivfalse
501         \global\Etoc@iiifalse
502         \global\Etoc@iifalse
503         \global\Etoc@ifalse
504         \global\Etoc@false
505         \global\Etoc@jfalse
506         \global\Etoc@jjfalse}%
507     \fi
508   \fi\fi
509 \Etoc@next}
510 \let\Etoc@startlocaltoc\@gobble

```

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 at any output of text at all even in an intermediate run.

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

```

511 \def\Etoc@localtableofcontents#1{%
512   \edef\Etoc@tmp{#1}%
513   \ifnum\Etoc@tmp<\@ne
514     \PackageWarning{etoc}
515       {Unknown toc id: run LaTeX to get references right}%
516     \let\Etoc@next\@gobble\else\let\Etoc@next\@firstofone\fi
517   \Etoc@next
518   {\edef\Etoc@startlocaltoc##1{%
519     \noexpand\Etoc@@startlocaltoc{##1}{#1}}
520   \Etoc@localtoctrue
521   \let\Etoc@localtop\Etoc@@minustwo@@

```

34. Implementation

```
522 \global\Etoc@stoptocfalse
523 \global\Etoc@notactivetrue
524 \Etoc@tableofcontents}%
525 \endgroup\ifEtoc@mustclosegroup\endgroup\fi
526 \Etoc@aftertohook % 1.07h
527 \etocaftertohook % 1.07g
528 }

[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.
529 \def\Etoc@getrefno #1#2\etoc@ {#1}
530 \def\Etoc@getref #1{\@ifundefined{r@#1}{0}{\expandafter\expandafter\expandafter
531 \Etoc@getrefno\csname r@#1\endcsname\relax\etoc@}}
532 \def\Etoc@ref#1{\Etoc@localtableofcontents{\Etoc@getref{#1}}}
533 \def\Etoc@label#1{\label{#1}\futurelet\Etoc@nexttoken\Etoc@t@bleofcontents}
534 \@firstofone{\def\Etoc@again} {\futurelet\Etoc@nexttoken\Etoc@t@bleofcontents}

placeholder for comments
535 \def\Etoc@t@bleofcontents{%
536 \ifx\Etoc@nexttoken\label
537 \def\Etoc@next{\expandafter\Etoc@label\@gobble}\else
538 \ifx\Etoc@nexttoken\@sptoken
539 \let\Etoc@next\Etoc@again\else
540 \ifEtoc@tocwithid
541 \def\Etoc@next{\Etoc@localtableofcontents{\c@etoc@tocid}}%
542 \else
543 \ifx\Etoc@nexttoken\ref
544 \def\Etoc@next{\expandafter\Etoc@ref\@gobble}%
545 \else
546 \def\Etoc@next{\Etoc@localtocfalse
547 \global\Etoc@notactivefalse
548 \Etoc@tableofcontents
549 \endgroup
550 \ifEtoc@mustclosegroup\endgroup\fi
551 \Etoc@aftertohook % 1.07h
552 \etocaftertohook % 1.07g
553 }%
554 \fi
555 \fi\fi\fi\Etoc@next}

1.07g for consistency Etoc@ prefix added.
556 \def\Etoc@table@fcontents{%
557 \refstepcounter{etoc@tocid}%
558 \Etoc@tocwithidfalse
559 \futurelet\Etoc@nexttoken\Etoc@t@bleofcontents}
560 \def\Etoc@localtable@fcontents{%
561 \refstepcounter{etoc@tocid}%
562 \addtocontents{toc}
563 {\string\etoc@startlocaltoc{\arabic{etoc@tocid}}}%
564 \Etoc@tocwithidtrue
565 \futurelet\Etoc@nexttoken\Etoc@t@bleofcontents}

1.07g defines \etoc@tableofcontents to be able to undo the evil and brutal doings of some pack-
ages with \tableofcontents.
566 \let\etocaftertitlehook \@empty
567 \let\etocaftercontentshook \@empty
```

```

568 \def\etocstableofcontents{%
569   \Etoc@openouttoc
570   \Etoc@par
571   \begingroup % closed in \Etoc@t@bleofcontents or \Etoc@localtableofcontents
572   \def\etoc@startlocaltoc##1{\etoc@@startlocaltoc{##1}{\c@etoc@tocid}}%
573   \@ifstar
574   {\def\Etoc@aftertitlehook{}\Etoc@table@fcontents}
575   {\let\Etoc@aftertitlehook\etocaftertitlehook\Etoc@table@fcontents}}
576 \let\tableofcontents\etocstableofcontents
577 \newcommand*\localtableofcontents{%
578   \Etoc@openouttoc
579   \Etoc@par
580   \begingroup % closed in \Etoc@t@bleofcontents or \Etoc@localtableofcontents
581   \@ifstar
582   {\def\Etoc@aftertitlehook{}\Etoc@localtable@fcontents}
583   {\let\Etoc@aftertitlehook\etocaftertitlehook\Etoc@localtable@fcontents}}

placeholder for comments.
584 \newcommand\etocsettocstyle[2]{%
585 \def\Etoc@tableofcontents
586 {\ifnum\c@tocdepth>\Etoc@minf
587   \let\Etoc@@next\@firstofone\else
588   \let\Etoc@@next\@gobble
589 \fi
590 \Etoc@@next{\Etoc@storetocdepth % 1.07g
591             #1\ifEtoc@parskip\parskip\z@skip\fi %1.07d
592             \Etoc@aftertitlehook
593             \let\Etoc@savedcontentsline\contentsline
594             \let\contentsline\Etoc@etoccontentsline
595             \Etoc@toctoc
596             \let\Etoc@@next\relax
597             \ifEtoc@tocwithid\else
598             \ifEtoc@localtoc
599             \ifEtoc@notactive
600             \def\Etoc@@next{\Etoc@localtocfalse
601                           \global\Etoc@notactivefalse
602                           \Etoc@toctoc}%
603             \fi\fi
604             \fi
605             \Etoc@@next
606             \Etoc@resettocdepth % 1.07g
607             \etocaftercontentshook
608             #2\@nbreakfalse}} % 1.07d: \@nbreakfalse moved here

placeholder for comments
609 \def\etocsetstyle{\Etoc@standardfalse\etoc@setstyle}
610 \long\def\etoc@setstyle#1#2#3#4#5{%
611 \long\expandafter\def
612   \csname Etoc@begin@\csname Etoc@#1@\endcsname\endcsname {#2}%
613 \long\expandafter\def
614   \csname Etoc@prefix@\csname Etoc@#1@\endcsname\endcsname {#3}%
615 \long\expandafter\def
616   \csname Etoc@contents@\csname Etoc@#1@\endcsname\endcsname {#4}%
617 \long\expandafter\def
618   \csname Etoc@end@\csname Etoc@#1@\endcsname\endcsname {#5}}

placeholder for comments
619 \def\etocfontminustwo {\normalfont \LARGE \bfseries}
620 \def\etocfontminusone {\normalfont \large \bfseries}

```

34. Implementation

```
621 \def\etocfontzero      {\normalfont \large \bfseries}
622 \def\etocfontone       {\normalfont \normalsize \bfseries}
623 \def\etocfonttwo       {\normalfont \normalsize}
624 \def\etocfontthree     {\normalfont \footnotesize}

placeholder for comments
625 \def\etocsepminustwo   {4ex \@plus .5ex \@minus .5ex}
626 \def\etocsepminusone   {4ex \@plus .5ex \@minus .5ex}
627 \def\etocsepzero       {2.5ex \@plus .4ex \@minus .4ex}
628 \def\etocseppone       {1.5ex \@plus .3ex \@minus .3ex}
629 \def\etocseptwo        {.5ex \@plus .1ex \@minus .1ex}
630 \def\etocsepthree      {.25ex \@plus .05ex \@minus .05ex}

placeholder for comments
631 \def\etocbaselinespreadminustwo {1}
632 \def\etocbaselinespreadminusone {1}
633 \def\etocbaselinespreadzero     {1}
634 \def\etocbaselinespreadone      {1}
635 \def\etocbaselinespreadtwo      {1}
636 \def\etocbaselinespreadthree    {.9}

placeholder for comments
637 \def\etocminustwoleftmargin {1.5em plus 0.5fil}
638 \def\etocminustworightmargin {1.5em plus -0.5fil}
639 \def\etocminusoneleftmargin {1em}
640 \def\etocminusonerightmargin {1em}
641 \def\etoclineleaders
642     {\hbox{\normalfont\normalsize\hb@xt@2ex {\hss.\hss}}}
643 \def\etocabbrevpagename {p.~}
644 \def\etocpartname       {\partname}
645 \def\etocbookname       {Book}

placeholder for comments The macro \etocdefaultlines was initially called \etoclines.
Now \etoclines 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.
646 \def\etocdefaultlines{\Etoc@standardfalse %
647 %% 'book' in memoir class:
648 \etoc@setstyle{@minustwo}
649   {\addpenalty\@M\etocskipfirstprefix}
650   {\addpenalty\@secpenalty}
651   {\beginingroup
652     \etocfontminustwo
653     \addvspace{\etocsepminustwo}%
654     \parindent \z@
655     \leftskip \etocminustwoleftmargin
656     \rightskip \etocminustworightmargin
657     \parfillskip \@flushglue
658     \vbox{\etocifnumbered{\etocbookname\enspace\etocnumber:\quad}}{\etocname
659       \baselineskip\etocbaselinespreadminustwo\baselineskip
660       \par}%
661     \addpenalty\@M\addvspace{\etocsepminusone}%
662     \endgroup}
663   }%
664 %% 'part':
665 \etoc@setstyle{@minusone}
666   {\addpenalty\@M\etocskipfirstprefix}
667   {\addpenalty\@secpenalty}
668   {\beginingroup
```

```

669 \etocfontminusone
670 \addvspace{\etocsepminusone}%
671 \parindent \z@
672 \leftskip \etocminusoneleftmargin
673 \rightskip \etocminusonerightmargin
674 \parfillskip \@flushglue
675 \vbox{\etocifnumbered{\etocpartname\enspace\etocnumber.\quad}}{\etocname
676 \baselineskip\etocbaselinespreadminusone\baselineskip
677 \par}%
678 \addpenalty\@M\addvspace{\etocsepzero}%
679 \endgroup}
680 {}%
681 %% 'chapter':
682 \etoc@setstyle{@zero}
683 {\addpenalty\@M\etocskipfirstprefix}
684 {\addpenalty\@itempenalty}
685 {\begingroup
686 \etocfontzero
687 \addvspace{\etocsepzero}%
688 \parindent \z@ \parfillskip \@flushglue
689 \vbox{\etocifnumbered{\etocnumber.\enspace}}{\etocname
690 \baselineskip\etocbaselinespreadzero\baselineskip
691 \par}%
692 \endgroup}
693 {\addpenalty{-\@highpenalty}\addvspace{\etocsepminusone}}%
694 %% 'section':
695 \etoc@setstyle{@one}
696 {\addpenalty\@M\etocskipfirstprefix}
697 {\addpenalty\@itempenalty}
698 {\begingroup
699 \etocfontone
700 \addvspace{\etocsepone}%
701 \parindent \z@ \parfillskip \z@
702 \setbox\z@\vbox{\parfillskip\@flushglue
703 \etocname\par
704 \setbox\tw@\lastbox
705 \global\setbox\@ne\hbox{\unhbox\tw@\ }}%
706 \dimen\z@=\wd\@ne
707 \setbox\z@=\etoclineleaders
708 \advance\dimen\z@\wd\z@
709 \etocifnumbered
710 {\setbox\tw@\hbox{\etocnumber, \etocabbrevpagename\etocpage}}
711 {\setbox\tw@\hbox{\etocabbrevpagename\etocpage}}%
712 \advance\dimen\z@\wd\tw@
713 \ifdim\dimen\z@ < \linewidth
714 \vbox{\etocname~%
715 \leaders\box\z@\hfil\box\tw@
716 \baselineskip\etocbaselinespreadone\baselineskip
717 \par}
718 \else
719 \vbox{\etocname~%
720 \leaders\copy\z@\hfil\break
721 \hbox{}}\leaders\box\z@\hfil\box\tw@
722 \baselineskip\etocbaselinespreadone\baselineskip
723 \par}
724 \fi
725 \endgroup}

```

34. Implementation

```
726 {\addpenalty\@secpenalty\addvspace{\etocsepzero}}%
727 %% 'subsection':
728 \etoc@setstyle{@two}
729 {\addpenalty\@medpenalty\etocskipfirstprefix}
730 {\addpenalty\@itempenalty}
731 {\begingroup
732   \etocfonttwo
733   \addvspace{\etocseptwo}%
734   \parindent \z@ \parfillskip \z@
735   \setbox\z@\vbox{\parfillskip\@flushglue
736                     \etocname\par\setbox\tw@\lastbox
737                     \global\setbox\@ne\hbox{\unhbox\tw@}}}%
738   \dimen\z@=\wd\@ne
739   \setbox\z@=\etoclineleaders
740   \advance\dimen\z@\wd\z@
741   \etocifnumbered
742     {\setbox\tw@\hbox{\etocnumber, \etocabbrevpagename\etocpage}}
743     {\setbox\tw@\hbox{\etocabbrevpagename\etocpage}}}%
744   \advance\dimen\z@\wd\tw@
745   \ifdim\dimen\z@ < \linewidth
746     \vbox{\etocname~%
747           \leaders\box\z@\hfil\box\tw@
748           \baselineskip\etocbaselinespreadtwo\baselineskip
749           \par}
750   \else
751     \vbox{\etocname~%
752           \leaders\copy\z@\hfil\break
753           \hbox{\leaders\box\z@\hfil\box\tw@
754                 \baselineskip\etocbaselinespreadtwo\baselineskip
755                 \par}
756   \fi
757   \endgroup}
758 {\addpenalty\@secpenalty\addvspace{\etocsepone}}%
759 %% 'subsubsection':
760 \etoc@setstyle{@three}
761 {\addpenalty\@M
762   \etocfontthree
763   \vspace{\etocsepthree}%
764   \noindent
765   \etocskipfirstprefix}
766 {\allowbreak\,--\,}
767 {\etocname}
768 {\hfil
769   \begingroup
770     \baselineskip\etocbaselinespreadthree\baselineskip
771     \par
772   \endgroup
773   \addpenalty{-\@highpenalty}}%
```

placeholder for comments

```
774 \etoc@setstyle{@four}{\{\}\{\}\{\}\{\}\}%
775 \etoc@setstyle{@five}{\{\}\{\}\{\}\{\}\{\}\}%
776 } % 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.

```
777 \def\etocabovetocskip{3.5ex \@plus 1ex \@minus .2ex}
```

```

778 \def\etocbelowtocskip{3.5ex \@plus 1ex \@minus .2ex}
779 \def\etoccolumnsep{2em}
780 \def\etocmulticolsep{0ex}
781 \def\etocmulticolpretolerance{-1}
782 \def\etocmulticoltolerance{200}
783 \def\etocdefaultnbcol{2}
784 \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.
785 \newcommand\etocmulticolstyle[2][\etocdefaultnbcol]{%
786 \etocsettocstyle
787   {\let\etocoldpar\par
788    \addvspace{\etocabovetocskip}%
789    \ifnum #1>\@ne\let\Etoc@next\@firstoftwo
790     \else \let\Etoc@next\@secondoftwo\fi
791    \Etoc@next{%
792     \multicolpretolerance\etocmulticolpretolerance
793     \multicoltolerance\etocmulticoltolerance
794     \setlength{\columnsep}{\etoccolumnsep}%
795     \setlength{\multicolsep}{\etocmulticolsep}%
796     \begin{multicols}{#1}[#2\etocoldpar\addvspace{\etocinnertopsep}]}
797     {#2\ifvmode\else\begin{group}\interlinepenalty\@M\parskip\z@skip
798      \@@par\endgroup
799     \fi
800     \nobreak\addvspace{\etocinnertopsep}%
801     \pretolerance\etocmulticolpretolerance
802     \tolerance\etocmulticoltolerance}}
803   {\ifnum #1>\@ne\let\Etoc@next\@firstofone
804    \else \let\Etoc@next\@gobble\fi
805    \Etoc@next{\end{multicols}}}%
806   \addvspace{\etocbelowtocskip}}}

placeholder for comments
807 \def\etocinnerbottomsep{3.5ex}
808 \def\etocinnerleftsep{2em}
809 \def\etocinnerrightsep{2em}
810 \def\etocoprulercmd{\hrule}
811 \def\etocleftrulercmd{\vrule}
812 \def\etocrightrulercmd{\vrule}
813 \def\etocbottomrulercmd{\hrule}
814 \def\etocoprulercolorcmd{\relax}
815 \def\etocbottomrulercolorcmd{\relax}
816 \def\etocleftrulercolorcmd{\relax}
817 \def\etocrightrulercolorcmd{\relax}

1.07i moves the \nobreak to before the \vskip\etocinnertopsep (especially important for the
single column case).
818 \def\etoc@ruledheading #1{%
819   \hb@xt@\linewidth{\color@begin{group}
820     \hss #1\hss\hskip-\linewidth
821     \etocoprulercolorcmd\leaders\etocoprulercmd\hss
822     \phantom{#1}%
823     \leaders\etocoprulercmd\hss\color@end{group}}%
824   \nointerlineskip\nobreak\vskip\etocinnertopsep}
825 \newcommand*{\etocruledstyle}[2][\etocdefaultnbcol]{%
826 \etocsettocstyle
827   {\addvspace{\etocabovetocskip}%
828   \ifnum #1>\@ne\let\Etoc@next\@firstoftwo

```

34. Implementation

```

829     \else \let\Etoc@next\@secondoftwo\fi
830 \Etoc@next
831     {\multicolpretolerance\etocmulticolpretolerance
832      \multicoltolerance\etocmulticoltolerance
833      \setlength{\columnsep}{\etoccolumnsep}%
834      \setlength{\multicolsep}{\etocmulticolsep}%
835      \begin{multicols}{#1}[\etoc@ruledheading{#2}]}
836     {\etoc@ruledheading{#2}%
837      \pretolerance\etocmulticolpretolerance
838      \tolerance\etocmulticoltolerance}}
839 {\ifnum #1>\@ne\let\Etoc@next\@firstofone
840  \else \let\Etoc@next\@gobble\fi
841  \Etoc@next\end{multicols}}%
842  \addvspace{\etocbelowtocskip}}}

placeholder for comments
843 \long\def\etocframedmphook{\relax}
844 \def\etocbkgcolorcmd{\relax}
845 \def\Etoc@relax{\relax}
846 \newbox\etoc@framed@titlebox
847 \newbox\etoc@framed@contentsbox
848 \newcommand*\etocframedstyle[2][\etocdefaultnbcol]{%
849 \etocsettocstyle{%
850   \addvspace{\etocabovetocskip}%
851   \sbox\z@{#2}%
852   \dimen\z@{\dp\z@}
853   \ifdim\wd\z@<\linewidth \dp\z@\z@ \else \dimen\z@\z@ \fi
854   \setbox\etoc@framed@titlebox=\hb@xt@\linewidth{\color@begingroup
855     \hss
856     \ifx\etocbkgcolorcmd\Etoc@relax\else
857       \sbox\tw@{\color{white}%
858        \vrule\@width\wd\z@\@height\ht\z@\@depth\dimen\z@}%
859       \ifdim\wd\z@<\linewidth \dp\tw@\z@\fi
860       \box\tw@
861       \hskip-\wd\z@
862     \fi
863     \copy\z@
864     \hss
865     \hskip-\linewidth
866     \etocprulecolorcmd\leaders\etocprule\hss%
867     \hskip\wd\z@
868     \etocprulecolorcmd\leaders\etocprule\hss\color@endgroup}%
869   \setbox\z@\hbox{\etocleftrule\etocrightrule}%
870   \dimen\tw@\linewidth\advance\dimen\tw@-\wd\z@
871   \advance\dimen\tw@-\etocinnerleftsep
872   \advance\dimen\tw@-\etocinnerrightsep
873   \setbox\etoc@framed@contentsbox=\vbox\bgroup
874     \hsize\dimen\tw@
875     \kern\dimen\z@
876     \vskip\etocinnertopsep
877     \hbox\bgroup
878     \begin{minipage}{\hsize}%
879     \etocframedmphook
880   \ifnum #1>\@ne\let\Etoc@next\@firstoftwo
881   \else \let\Etoc@next\@secondoftwo\fi
882   \Etoc@next
883   {\multicolpretolerance\etocmulticolpretolerance
884    \multicoltolerance\etocmulticoltolerance

```



```

885     \setlength{\columnsep}{\etoccolumnsep}%
886     \setlength{\multicolsep}{\etocmulticolsep}%
887     \begin{multicols}{#1}}
888     {\pretolerance\etocmulticolpretolerance
889     \tolerance\etocmulticoltolerance}}
890     {\ifnum #1>\@ne\let\Etoc@next\@firstofone
891     \else \let\Etoc@next\@gobble\fi
892     \Etoc@next{\end{multicols}\unskip}%
893     \end{minipage}}%
894     \egroup
895     \vskip\etocinnerbottomsep
896     \egroup
897     \vbox{\hsize\linewidth
898         \ifx\etocbkgcolorcmd\Etoc@relax\else
899             \kern\ht\etoc@framed@titlebox
900             \kern\dp\etoc@framed@titlebox
901             \hb@xt@\linewidth{\color@begingroup
902             \etocleftrulecolorcmd\etocleftrule
903             \etocbkgcolorcmd
904             \leaders\vrule
905                 \@height\ht\etoc@framed@contentsbox
906                 \@depth\dp\etoc@framed@contentsbox
907             \hss
908             \etocrightrulecolorcmd\etocrightrule
909             \color@endgroup}\nointerlineskip
910             \vskip-\dp\etoc@framed@contentsbox
911             \vskip-\ht\etoc@framed@contentsbox
912             \vskip-\dp\etoc@framed@titlebox
913             \vskip-\ht\etoc@framed@titlebox
914         \fi
915         \box\etoc@framed@titlebox\nointerlineskip
916         \hb@xt@\linewidth{\color@begingroup
917         {\etocleftrulecolorcmd\etocleftrule}%
918         \hss\box\etoc@framed@contentsbox\hss
919         \etocrightrulecolorcmd\etocrightrule\color@endgroup}
920         \nointerlineskip
921         \vskip\ht\etoc@framed@contentsbox
922         \vskip\dp\etoc@framed@contentsbox
923         \hb@xt@\linewidth{\color@begingroup\etocbottomrulecolorcmd
924         \leaders\etocbottomrule\hss\color@endgroup}}
925     \addvspace{\etocbelowntocskip}}

placeholder for comments
926 \newcommand\etoc@multicoltoctoc[2][\etocdefaultnbcol]{%
927     \etocmulticolstyle[#1]{#2}%
928     \tableofcontents}
929 \newcommand\etoc@multicoltoctoci[2][\etocdefaultnbcol]{%
930     \etocmulticolstyle[#1]{#2}%
931     \tableofcontents*}
932 \newcommand\etoc@local@multicoltoctoc[2][\etocdefaultnbcol]{%
933     \etocmulticolstyle[#1]{#2}%
934     \localtableofcontents}
935 \newcommand\etoc@local@multicoltoctoci[2][\etocdefaultnbcol]{%
936     \etocmulticolstyle[#1]{#2}%
937     \localtableofcontents*}

placeholder for comments
938 \newcommand*\etoc@ruledtoctoc[2][\etocdefaultnbcol]{%
939     \etocruledstyle[#1]{#2}%

```

34. Implementation

```
940 \tableofcontents}
941 \newcommand*\etoc@ruledtoci[2][\etocdefaultnbcol]{%
942 \etocruledstyle[#1]{#2}%
943 \tableofcontents*}
944 \newcommand*\etoc@local@ruledtoc[2][\etocdefaultnbcol]{%
945 \etocruledstyle[#1]{#2}%
946 \localtableofcontents}
947 \newcommand*\etoc@local@ruledtoci[2][\etocdefaultnbcol]{%
948 \etocruledstyle[#1]{#2}%
949 \localtableofcontents*}

placeholder for comments
950 \newcommand*\etoc@framedtoc[2][\etocdefaultnbcol]{%
951 \etocframedstyle[#1]{#2}%
952 \tableofcontents}
953 \newcommand*\etoc@framedtoci[2][\etocdefaultnbcol]{%
954 \etocframedstyle[#1]{#2}%
955 \tableofcontents*}
956 \newcommand*\etoc@local@framedtoc[2][\etocdefaultnbcol]{%
957 \etocframedstyle[#1]{#2}%
958 \localtableofcontents}
959 \newcommand*\etoc@local@framedtoci[2][\etocdefaultnbcol]{%
960 \etocframedstyle[#1]{#2}%
961 \localtableofcontents*}

placeholder for comments
962 \def\etocmulticol{\begingroup
963 \Etoc@mustclosegrouptrue
964 \@ifstar
965 {\etoc@multicoltoctoci}
966 {\etoc@multicoltoctoc}}
967 \def\etocruled{\begingroup
968 \Etoc@mustclosegrouptrue
969 \@ifstar
970 {\etoc@ruledtoci}
971 {\etoc@ruledtoc}}
972 \def\etocframed{\begingroup
973 \Etoc@mustclosegrouptrue
974 \@ifstar
975 {\etoc@framedtoci}
976 {\etoc@framedtoc}}
977 \def\etoclocalmulticol{\begingroup
978 \Etoc@mustclosegrouptrue
979 \@ifstar
980 {\etoc@local@multicoltoctoci}
981 {\etoc@local@multicoltoctoc}}
982 \def\etoclocalruled{\begingroup
983 \Etoc@mustclosegrouptrue
984 \@ifstar
985 {\etoc@local@ruledtoci}
986 {\etoc@local@ruledtoc}}
987 \def\etoclocalframed{\begingroup
988 \Etoc@mustclosegrouptrue
989 \@ifstar
990 {\etoc@local@framedtoci}
991 {\etoc@local@framedtoc}}

placeholder for comments
992 \def\etocarticlestyle{%
```

```

993 \etocsettocstyle
994 {\section *{\contentsname
995 \mkboth {\MakeUppercase \contentsname}
996 {\MakeUppercase \contentsname}}}
997 {}
998 \def\etocarticlestylenomarks{%
999 \etocsettocstyle
1000 {\section *{\contentsname}}
1001 {}
placeholder for comments
1002 \def\etocbookstyle{%
1003 \etocsettocstyle
1004 {\if@twocolumn \@restonecoltrue \onecolumn \else \@restonecolfalse \fi
1005 \chapter *{\contentsname
1006 \mkboth {\MakeUppercase \contentsname}
1007 {\MakeUppercase \contentsname}}}
1008 {\if@restonecol \twocolumn \fi}}
1009 \def\etocbookstylenomarks{%
1010 \etocsettocstyle
1011 {\if@twocolumn \@restonecoltrue \onecolumn \else \@restonecolfalse \fi
1012 \chapter *{\contentsname}}
1013 {\if@restonecol \twocolumn \fi}}
1014 \let\etocreportstyle\etocbookstyle
1015 \let\etocreportstylenomarks\etocbookstylenomarks
1016 \def\etocmemoirtocfmt #1#2{%
1017 \def\Etoc@addsuitablecontentsline{\addcontentsline {toc}{#1}{#2}}%
1018 \renewcommand*{\etocaftertitlehook{%
1019 \ifmem@em@starred@listof
1020 \else\phantomsection\aftergroup\Etoc@addsuitablecontentsline\fi}}
1021 \def\etocmemoirstyle{%
1022 \etocsettocstyle
1023 {\ensureonecol \par \begingroup \@nameuse {@tocmaketitle}
1024 \Etoc@aftertitlehook\let\Etoc@aftertitlehook\relax
1025 \parskip \cftparskip \@nameuse {cfttocbeforelisthook}}
1026 {\@nameuse {cfttocafterlisthook}\endgroup\restorefromonecol}}
placeholder for comments
1027 \def\etocscrartclstyle{%
1028 \etocsettocstyle
1029 {\let\if@dynlist\if@tocleft
1030 \iftocfeature {toc}{onecolumn}
1031 {\iftocfeature {toc}{leveldown}
1032 {}
1033 {\if@twocolumn \aftergroup \twocolumn \onecolumn \fi }}
1034 {}}%
1035 \tocbasic@listhead {\listoftocname}%
1036 \begingroup \expandafter \expandafter \expandafter
1037 \endgroup \expandafter
1038 \ifx
1039 \csname microtypesetup\endcsname \relax
1040 \else
1041 \iftocfeature {toc}{noptrusion}}
1042 {\microtypesetup {protrusion=false}%
1043 \PackageInfo {tocbasic}%
1044 {character protrusion at toc deactivated}}%
1045 \fi
1046 \setlength {\parskip }{\z@ }%
1047 \setlength {\parindent }{\z@ }%

```

34. Implementation

```
1048      \setlength {\parfillskip }{\z@ \@plus 1fil}%
1049      \csname tocbasic@@before@hook\endcsname
1050      \csname tb@toc@before@hook\endcsname}
1051      {\csname tb@toc@after@hook\endcsname
1052      \csname tocbasic@@after@hook\endcsname}}
1053 \let\etocscrbookstyle\etocscrartclstyle
1054 \let\etocscrreprtstyle\etocscrartclstyle

placeholder for comments
1055 \def\etocstandarddisplaystyle{\etocarticlestyle}
1056 \newcommand*\etocmarkboth[1]{%
1057   \@mkboth{\MakeUppercase{#1}}{\MakeUppercase{#1}}}
1058 \newcommand*\etocmarkbothnouc[1]{\@mkboth{#1}{#1}}
1059 \newcommand\etocstyle[3][section]{\etocmulticolstyle[#2]%
1060   {\csname #1\endcsname *{#3}}}
1061 \newcommand\etocstylewithmarks[4][section]{\etocmulticolstyle[#2]%
1062   {\csname #1\endcsname *{#3}\etocmarkboth{#4}}}%
1063 \newcommand\etocstylewithmarksnouc[4][section]{\etocmulticolstyle[#2]%
1064   {\csname #1\endcsname *{#3}\etocmarkbothnouc{#4}}}%

placeholder for comments
1065 \def\Etoc@redefetocstyle#1{%
1066   \renewcommand\etocstylewithmarks[4][#1]
1067   {\etocmulticolstyle[##2]%
1068     {\csname ##1\endcsname *{##3}\etocmarkboth{##4}}}%
1069   \renewcommand\etocstylewithmarksnouc[4][#1]
1070   {\etocmulticolstyle[##2]%
1071     {\csname ##1\endcsname *{##3}\etocmarkbothnouc{##4}}}%
1072   \renewcommand\etocstyle[3][#1]{%
1073     \etocmulticolstyle[##2]{\csname ##1\endcsname *{##3}}}%
1074 \@ifclassloaded{scrartcl}
1075   {\renewcommand*\etocstandarddisplaystyle{\etocscrartclstyle}}{}
1076 \@ifclassloaded{book}
1077   {\renewcommand*\etocfontone{\normalfont\normalsize}
1078   \renewcommand*\etocstandarddisplaystyle{\etocbookstyle}
1079   \Etoc@redefetocstyle{chapter}}{}
1080 \@ifclassloaded{report}
1081   {\renewcommand*\etocfontone{\normalfont\normalsize}
1082   \renewcommand*\etocstandarddisplaystyle{\etocreportstyle}
1083   \Etoc@redefetocstyle{chapter}}{}
1084 \@ifclassloaded{scrbook}
1085   {\renewcommand*\etocfontone{\normalfont\normalsize}
1086   \renewcommand*\etocstandarddisplaystyle{\etocscrbookstyle}
1087   \Etoc@redefetocstyle{chapter}}{}
1088 \@ifclassloaded{scrreprt}
1089   {\renewcommand*\etocfontone{\normalfont\normalsize}
1090   \renewcommand*\etocstandarddisplaystyle{\etocscrreprtstyle}
1091   \Etoc@redefetocstyle{chapter}}{}
1092 \@ifclassloaded{memoir}
1093   {\renewcommand*\etocfontone{\normalfont\normalsize}
1094   \etocmemoirtocfmt{chapter}{\contentsname}%
1095   \renewcommand*\etocstandarddisplaystyle{\etocmemoirstyle}
1096   \Etoc@redefetocstyle{chapter}}{}

placeholder for comments
1097 \def\Etoc@addtocontents #1#2%
1098   {\ifEtoc@hyperref
1099     \addtocontents {toc}{\protect\contentsline
1100       {#1}{#2}%
```

```

1101          {\thepage }{\@currentHref }}%
1102      \else
1103          \addtocontents {toc}{\protect\contentsline
1104              {#1}{#2}{\thepage }}%
1105      \fi}
1106 \def\Etoc@addcontentsline@ #1#2#3%
1107     {\@namedef{toclevel@#1}{#3}%
1108      \addcontentsline {toc}{#1}{#2}}
1109 \DeclareRobustCommand*\etoc@toc@contentsline{
1110     {\@ifstar{\Etoc@addcontentsline@}{\Etoc@addtocontents}}

```

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

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

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

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

```

1111 \def\Etoc@storetocdepth {\global\c@etoc@tocdepth\c@tocdepth }
1112 \def\Etoc@resettocdepth {\global\c@tocdepth\c@etoc@tocdepth }
1113 \def\etocobeytoctocdepth {\def\etoc@settocdepth
1114     {\afterassignment\Etoc@@nottodeep \global\c@tocdepth}}
1115 \def\Etoc@@nottodeep {\ifnum\c@tocdepth>\c@etoc@tocdepth
1116     \global\c@tocdepth\c@etoc@tocdepth\fi }
1117 \def\etocignoretoctocdepth {\let\etoc@settocdepth\@gobble }
1118 \def\etocsettocdepth {\futurelet\Etoc@nexttoken\Etoc@set@tocdepth }
1119 \def\Etoc@set@tocdepth {\ifx\Etoc@nexttoken\bgroup
1120     \expandafter\Etoc@set@tocdepth@
1121     \else\expandafter\Etoc@set@toctocdepth
1122     \fi }
1123 \def\Etoc@set@tocdepth@ #1{\@ifundefined {Etoc@#1@@}
1124     {\PackageWarning{etoc}
1125         {Unknown sectioning unit #1, \protect\etocsettocdepth\space ignored}}
1126     {\global\c@tocdepth\csname Etoc@#1@@\endcsname}%
1127 }
1128 \def\Etoc@set@toctocdepth #1#{\Etoc@set@toctocdepth@ }
1129 \def\Etoc@set@toctocdepth@ #1{%
1130     \@ifundefined{Etoc@#1@@}%
1131     {\PackageWarning{etoc}
1132         {Unknown sectioning depth #1, \protect\etocsettocdepth.toc ignored}}%
1133     {\addtocontents {toc}
1134     {\protect\etoc@settocdepth\expandafter\protect\csname Etoc@#1@@\endcsname}}%
1135 }

```

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).

34. Implementation

```

1136 \def\etocdepthtag #1{\Etoc@depthtag } % \etocdepthtag.toc {name}
1137 \def\Etoc@depthtag #1{\addtocontents {toc}{\protect\etoc@depthtag {#1}}}
1138 \def\etocignoredepthtags {\let\etoc@depthtag \@gobble }
1139 \def\etocobeydepthtags {\let\etoc@depthtag \Etoc@depthtag@ }
1140 \def\Etoc@depthtag@ #1{\@ifundefined{Etoc@depthof@#1}%
1141     {}% ignore in silence if tag has no associated depth
1142     {\afterassignment\Etoc@nottoo deep
1143         \global\c@tocdepth\csname Etoc@depthof@#1\endcsname}%
1144 }
1145 \def\etocsettagdepth #1#2{\@ifundefined{Etoc@#2@@}%
1146     {\PackageWarning{etoc}
1147         {Unknown sectioning depth #2, \protect\etocsettagdepth\space ignored}}%
1148     {\@namedef{Etoc@depthof@#1}{\@nameuse{Etoc@#2@@}}}%
1149 }

```

We must cancel all tocvsec2 toc-related actions.

```

1150 \def\Etoc@tocvsec@err #1{\PackageError {etoc}
1151     {the command \protect#1\space is incompatible with etoc}
1152     {use \protect\etocsettocdepth.toc as replacement (check the manual)}}%
1153 }%
1154 \AtBeginDocument {%
1155     \@ifpackageloaded {tocvsec2}
1156     {\def\maxtocdepth #1{\Etoc@tocvsec@err \maxtocdepth }%
1157     \def\settocdepth #1{\Etoc@tocvsec@err \settocdepth }%
1158     \def\resettocdepth {\@ifstar {\Etoc@tocvsec@err \resettocdepth }%
1159         {\Etoc@tocvsec@err \resettocdepth }%
1160     }%
1161     \def\save@tocdepth #1#2#3{% % if etoc is added to file previously
1162     \let\reset@tocdepth\relax % using tocvsec2
1163     \let\remax@tocdepth\relax
1164     \let\tableofcontents\etoc\tableofcontents
1165     \PackageWarning {etoc}
1166     {package tocvsec2 detected and its modification of^^}
1167     \protect\tableofcontents\space reverted. Use
1168     \protect\etocsettocdepth.toc as a replacement^^}
1169     for the tocvsec2 toc-related commands}}}%
1170 }%

```

1.07g

```

1171 \def\invisibletableofcontents {\etocsetnexttocdepth {-3}\tableofcontents }%
1172 \def\invisiblelocaltableofcontents
1173     {\etocsetnexttocdepth {-3}\localtableofcontents }%
1174 \def\etocsetnexttocdepth #1{%
1175     \@ifundefined{Etoc@#1@@}
1176     {\PackageWarning{etoc}
1177         {Unknown sectioning unit #1, \protect\etocsetnexttocdepth\space ignored}}
1178     {\edef\Etoc@aftertochook {\global\c@tocdepth\the\c@tocdepth\space
1179         \let\noexpand\Etoc@aftertochook\noexpand\@empty }%
1180     \global\c@tocdepth\csname Etoc@#1@@\endcsname}%
1181 }%

```

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.

```

1182 \addtocontents {toc}{\protect\@ifundefined{etoc\etocstyle}%
1183     {\let\protect\etoc@startlocaltoc\protect\@gobble
1184     \let\protect\etoc@settocdepth\protect\@gobble
1185     \let\protect\etoc@depthtag\protect\@gobble }{}}%

```

Initializations.

```

1186 \def\etocstandardlines {\Etoc@standardtrue}

```

```
1187 \def\etoclines          {\Etoc@standardfalse}
1188 \etocdefaultlines
1189 \etocstandardlines
1190 \etocstandarddisplaystyle
1191 \etocobeytoctocdepth
1192 \etocobeydepthtags
1193 \let\Etoc@aftertohook \@empty % private for \etocsetnexttocdepth
1194 \let\etocaftertohook \@empty % public for end user.
1195 \endinput
```