

The **etoc** package

JEAN-FRANÇOIS BURNOL

jfbu (at) free (dot) fr

Package version: v1.07e (2013/03/01)

Documentation generated from the source file
with timestamp “01-03-2013 at 15:31:43 CET”

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.

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 TOC defined somewhere else in the document.

The formatting inherited (and possibly customized by other packages) from the document class may also be used in *compatibility mode*. Regarding the *global toc display*, **etoc** provides pre-defined styles based on a multi-column format, optionally with a frame or a ruled title.

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

Foreword

Popular packages dealing with TOCs include `tocloft`, `titletoc` and `minitoc`. Why another one? well, initially I started **etoc** for my own use, and only later found out about the above 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 is compatible with the `article`, `book`, `report`, `scrartcl`, `scrbook`, `scrreprt` and `memoir` classes.

CONTENTS

Change history

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).

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`,

`\etoccontentline`, `\etoccontentline*`

`\etocnopar`, `\etocaftercontentshook`

modified command: `\etocmulticolstyle`

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

v1.06 [2012/12/07]

the standard macros `\@section` etc... are modified only during the calls to `\tableofcontents`; they can thus be customized as will by the user (with the help of a package like `tocloft`) and this will be taken into account by `etoc` for the TOCs typeset in compatibility mode.

v1.05 [2012/12/01]

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

compatibility with KOMA-script and memoir document classes.

v1.04 [2012/11/24]

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

`\tableofcontents \label{toc:1}`

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

`\tableofcontents \ref{toc:1}`

v1.02 [2012/11/18]

initial version.

TABLE OF CONTENTS

Part I. Overview

Initial motivation: nested lists 1, p. 4

Limitations intrinsic to the use of environments.

Line styles and toc display style 2, p. 4

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

Arbitrarily many TOCs, and local ones too 3, p. 6

Labeling and reusing elsewhere – The hyperref option `bookmarksdepth` – On manually adding layout commands to the `.toc` file – Shuffling the levels with `\etocsetlevel` – Local table of contents for this part.

A simple example 4, p. 9

Surprising uses of `etoc` 5, p. 11

The TOC of TOCs – Arbitrary "Lists Of..." – A TOC with a fancy layout – Another compatibility mode – The TOC as a tree.

Part II. Package commands for line styles

The `\etocsetstyle` command 6, p. 20

The `\etocname` and `\etocpage` commands – The `\etocskipfirstprefix` command – The `\etocnumber`

CONTENTS

command—The `\etocthenname`, `\etocthenumber`, and `\etocthepage` commands—The `\etoclink` command.

The `\etocsetlevel` command 7, p. 23

Scope of commands added to the `.toc` file 8, p. 24

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

Am I also red? 9, p. 24

Part III. Package commands for toc display styles

Specifying the toc display style 10, p. 25

The command `\etocruledstyle`—The command `\etocmulticolstyle`—The command `\etocruled`—The commands `\etocframedstyle` and `\etocframed`—Headings, titles, `\etocinnertopsep`—The command `\etocsettocstyle`—The compatibility mode `\etocstandarddisplaystyle`.

Starred variants of the `\tableofcontents` etc... commands 11, p. 28

Table of contents for this part 12, p. 28

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

Part IV. Using and customizing `etoc`

Summary of the main style commands 13, p. 32

Setting up local styles—Setting up toc display styles—Displaying tables of contents—Labels and references—The package default line styles: `\etocdefaultlines`—One more example TOC layout.

Customizing `etoc` 14, p. 38

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

Part V. Tips

... and tricks 15, p. 40

Hacking framed parboxes—Interverting the levels—All subsections of this document—Compatibility with other packages—T_EXnical matters—Errors and catastrophes.

Part VI. The code

Implementation 16, p. 44

Part I.

Overview

1. Initial motivation: nested lists

The initial impetus was to feed nested list environments with the data consisting of the *name* (`\etocname`), *number* (`\etocnumber`), and *page number* (`\etocpage`) as recorded¹ in the `.toc` file. For example, typesetting the line corresponding to the first sub-section in a given section would open a list environment which would be closed only when a section, chapter, or part line entry in the `.toc` file would be encountered. `etoc` allows to do this very easily and the opening and closing may be for example `\begin{enumerate}` and `\end{enumerate}` pairs, will all the customizing allowed by packages such as `enumitem`.

1.1. Limitations intrinsic to the use of environments

There is a first limitation to this method: the `.toc` file may contain other commands, such as language changing commands, which do not expect to see their scope limited in this way inside a group (\LaTeX 's environments create groups). Therefore the package own line styles (illustrated by the main table of contents in this document) do not make use of environments to avoid that problem.

A second limitation is that one may nest at most 4 levels of `enumerate` environments, and 4 levels of `itemize` environment. I tried alternating them and did succeed to nest 6 levels (and not 8 alas . . . ²). With `\etocnumber` as the optional parameter to `\item`: `\item[\etocnumber]` one may transform the `itemize` into an enumerated list... anyway, 4 levels of sectional divisions in a TOC are generally sufficient, and again using `enumerate` environments is only a possibility provided by `etoc`, it is by no means mandatory to use them in the line styles specifications.

We will give in this manual a simple-minded example of nested use of `enumerate` environments. 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.

The built-in default “line styles” provided by the package do not make use of environments.

2. 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

¹the *number* has to be disentangled from the *name*, and in case `hyperref` is present, the hyperlink has to be redistributed around each of them.

²this is surely a well-known issue which I did not at all investigate any further.

2. Line styles and toc display style

by the command `\etocsettocstyle` (or some higher-level commands) and the former by the command `\etocsetstyle`.

2.1. `\etocsettocstyle` for the toc display

The low-level `\etocsettocstyle` command allows to decide 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 head-marks) 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 of course allowed).

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

2.2. `\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`. The four other arguments say: 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*.

2.3. 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.³ 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 the compatibility mode for line styles (but not for the global display style; for this one needs to use the command `\etocsettocstyle`).

To exit after `\etocstandardlines` from compatibility mode one uses the command `\etoc toc lines`, which re-activates the latest line styles as defined by `\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.

³for the “toc display style” `etoc` checks if it knows the class, and if not defaults to the `article` class layout.

3. Arbitrarily many TOCs, and local ones too

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 contents⁴: *i.e.* all sections and sub-units inside a given chapter, or all subsubsections and lower inside a given subsection, etc . . .

3.1. Labeling and reusing elsewhere

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

I am from far away	
13. Summary of the main style commands	32
13.1. Setting up local styles	32
13.2. Setting up toc display styles	32
13.3. Displaying tables of contents	33
13.4. Labels and references	33
13.5. The package default line styles: <code>\etocdefaultlines</code>	33
13.6. One more example TOC layout	36
14. Customizing etoc	38
14.1. Customizing the etoc pre-defined line styles	38
14.2. Customizing the toc display styles	39

We actually did something like:

```
\tableofcontents\label{toc:d}\ref{toc:c}
```

Hence [the present location](#) can itself now be referred to via `\ref{toc:d}`: it gives the id of this TOC⁵ in the sequence of document TOCs, and will be a link if package `hyperref` is used.

However one should not use elsewhere `\tableofcontents\ref{toc:d}`. Due to the way **etoc** implements the cloning, the doubly cloned TOC will be typeset as a full table of contents. So to clone again, one should use the original: `\tableofcontents\ref{toc:c}`.

3.2. The `hyperref` option `bookmarksdepth`

When modifying the counter `tocdepth` for the purposes of multiple uses of `\tableofcontents` or `\localtableofcontents`, one should consider that package `hyperref` by

⁴Up to version 1.04 we called this `\tableofcontents*`, but for reasons of compatibility with the `memoir` class, we have decided to drop this usage.

⁵*i.e.* 3, there was an invisible TOC with id 2 at the beginning of this part.

3. Arbitrarily many TOCs, and local ones too

default takes into account the *current* value of the `tocdepth` counter to decide whether the final pdf output will contain a bookmark corresponding to the used sectioning command. Thus, one often will have to reset `tocdepth` to its previous value immediately after the display of the table of contents.

Or, there is the `bookmarksdepth=n` option of package `hyperref`, with n the desired document bookmarks depth, which can be numeric or the name of a level known to `hyperref`. The present document passed `bookmarksdepth=3` as option to `hyperref`, so as to not have to reset `tocdepth` each time its value was changed.

3.3. On manually adding layout commands to the .toc file

When displaying in that way many tables of contents in the same document one should of course beware of the impact of adding manually 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 have issued rather:

```
\addtocontents{toc}{\string\myclearpage},
```

to have a `\let\myclearpage\relax` at the top level of the document and to use where needed something like:

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

The `memoir` class has the command `\settocdepth` which writes a `\changetocdepth` command inside the .toc file. This will impact the typesetting by **etoc** of *all* tables of contents, with possibly unexpected results: imagine the document has `\settocdepth{chapter}` at some point to avoid having the sections from subsequent chapters be listed in the main table of contents. Then a local table of contents in one of these chapters will print a title but will be without any entry. A solution is to do `\begingroup \renewcommand*\changetocdepth[1]{} \localtableofcontents \endgroup`, and to set the desired level for the local table of contents with the other `memoir` command `\max-tocdepth`.⁶

3.4. Shuffling the levels with \etocsetlevel

The intrinsic levels manipulated by **etoc** are numeric: from -2 (which corresponds to book in the `memoir` class) down 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>}{<n>}`. In conjunction with the use of the L^AT_EX `tocdepth` counter, this has powerful applications: `<level_name>` does not have to coincide with an actual document sectioning command, and **etoc** can be used to print arbitrary “lists of things”, using no other auxiliary file than the .toc file. This is explained further in the section 5.

⁶The `memoir` class allows multiple calls to the `\tableofcontents` command, so these issues already arise there, independently of **etoc**, see page 170 of the `memoir` manual.

3.5. Local table of contents for this part

Immediately after the line in the source file starting with the first `\part` we wrote:

```
\setcounter{tocdepth}{-3}
\localtableofcontents \label{toc:partone}
```

The `tocdepth` being set to -3, nothing at all was typeset in the page (**etoc** cancels printing even the heading of the TOC if the `tocdepth` is -2 or less, except for the `memoir` class for which it needs to be -3 or less). We can then display here this TOC with:

```
\setcounter{tocdepth}{3}
\tableofcontents \ref{toc:partone}
```

Contents of Part One		
1	Initial motivation: nested lists	4
1.1	Limitations intrinsic to the use of environments	4
2	Line styles and toc display style	4
2.1	<code>\etocsettocstyle</code> for the toc display	5
2.2	<code>\etocsetstyle</code> for the line styles	5
2.3	Compatibility mode	5
3	Arbitrarily many TOCs, and local ones too	6
3.1	Labeling and reusing elsewhere	6
3.2	The hyperref option <code>bookmarksdepth</code>	6
3.3	On manually adding layout commands to the .toc file	7
3.4	Shuffling the levels with <code>\etocsetlevel</code>	7
3.5	Local table of contents for this part	8
4	A simple example	9
5	Surprising uses of etoc	11
5.1	The TOC of TOCs	11
5.2	Arbitrary “Lists Of...”	12
5.3	A TOC with a fancy layout	13
5.4	Another compatibility mode	15
5.5	The TOC as a tree	17

The actual design is not pre-built in **etoc**; it uses the **etoc** ‘framed’ style with a background color and a dummy empty title, the actual title having been put inside the frame as part of the `<start>` code of the section ‘line style’. The frame borders have been set to

4. A simple example

have the same color as the one which serves as background for the entire thing. This design (with other colors) is in use also for [this other toc](#), and the coding used is to be found at its location.

Here is now a far simpler design.

4. A simple example

Here is a simple example of use of the package functionalities. We display again the local table of contents for this part but choose font sizes and style which would be used rather for, respectively, chapters and sections in an average length memoir.

<hr/>		
<i>My Beautiful Thesis</i>		
Chapter 1	Initial motivation: nested lists	4
1.1	<i>Limitations intrinsic to the use of environments</i>	4
Chapter 2	Line styles and toc display style	4
2.1	<i><code>\etocsettocstyle</code> for the toc display</i>	5
2.2	<i><code>\etocsetstyle</code> for the line styles</i>	5
2.3	<i>Compatibility mode</i>	5
Chapter 3	Arbitrarily many TOCs, and local ones too	6
3.1	<i>Labeling and reusing elsewhere</i>	6
3.2	<i>The hyperref option bookmarksdepth</i>	6
3.3	<i>On manually adding layout commands to the .toc file</i>	7
3.4	<i>Shuffling the levels with <code>\etocsetlevel</code></i>	7
3.5	<i>Local table of contents for this part</i>	8
Chapter 4	A simple example	9
Chapter 5	Surprising uses of <code>etoc</code>	11
5.1	<i>The TOC of TOCs</i>	11
5.2	<i>Arbitrary “Lists Of...”</i>	12
5.3	<i>A TOC with a fancy layout</i>	13
5.4	<i>Another compatibility mode</i>	15
5.5	<i>The TOC as a tree</i>	17

4. A simple example

And now, the line style specifications. They have some redundancy for clarity, and do not care about what to do at possible page breaks. Also, this does not care about potential multi-column use.

```
\begingroup % we start a group to keep the style changes local
% the changes to paragraph layout parameters done in the style
% specifications would anyhow have been kept local as the TOCs are
% always typeset by the package in a group.
\newlength{\tocleftmargin}    \setlength{\tocleftmargin}{5cm}
\newlength{\tocrightmargin}   \setlength{\tocrightmargin}{1cm}

\etocsetstyle{section}        % would be chapter in real life
{\addvspace{1ex}\parfillskip0pt
 \leftskip\tocleftmargin      % was already done in title
 \rightskip\the\tocrightmargin plus 1fil
 \parindent0pt\color{cyan}}   % again this was 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}     % would be section in real life
{}
{\mdseries\large\addvspace{.5ex}\leavevmode}
{\llap{\etocnumber\hspace{.75cm}}{\textit{\etocname}}%
 \hfill\makebox[-\tocrightmargin][l]{\makebox[0pt]{\etocpage}}\par}
{}

% \color{cyan}\parindent0pt and \leftskip\tocleftmargin
% were repeated in the <start> code of the ‘‘section style’’,
% for clarity of code.

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

\tableofcontents \ref{toc:partone}
\endgroup
```

As one can see, the code uses the basic commands from $\text{T}_{\text{E}}\text{X}$ / $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ for paragraph layouts and an efficient mix of plain $\text{T}_{\text{E}}\text{X}$ and $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ syntaxes.

Users not so knowledgeable in the $\text{T}_{\text{E}}\text{X}$ syntax (for cause of having been exposed only to $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ “introductions”) have the possibility explained earlier to use nested `enumerate` environments, of course this means mastering another syntax.

However, using as here the $\text{T}_{\text{E}}\text{X}$ primitive `\par`,⁷ and basic skip registers `\leftskip`, `\rightskip`, `\parfillskip`, ... is by far the surest way to completely understand and master what happens. Any user of $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ should learn their signification.

⁷`\par` is redefined by $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ but this is of no immediate relevance here.

5. Surprising uses of **etoc**

5.1. The TOC of TOCs

Here is the numbered and linked list of all tables of contents which are displayed within this document: [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#), [9](#), [10](#), [11](#), [12](#), [13](#), [14](#), [15](#), [16](#), [17](#). And to obtain it here we just wrote:⁸

```
Here is the numbered and linked list of all ta-
bles 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').

The package provides a starred variant `\etoccontentsline*`, which will allow the creation of bookmarks and has a third mandatory argument which is the level to be used by these bookmarks.

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

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

⁸click the '5' in the list to get confirmation...

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

5. Surprising uses of *etoc*

(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}
  \etocsetlevel{visibletoc}{0}
  \etocsetstyle{visibletoc}
    {\etocskipfirstprefix}{, }{{\color{niceone}\etocname}}{}
  \etocsettocstyle{}{} % don't set any title, rules or frame or multicol!
  \setcounter{tocdepth}{0} % 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.

Subsection 15.2 gives another use of the shuffling of levels.

5.2. Arbitrary “Lists Of...”

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}} }
```

5. Surprising uses of **etoc**

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}}
```

This example originates with question 94766 on the `TEX-StackExchange` site. 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:

```
\setcounter{tocdepth}{2}      % sections are at level 1 and will show up
\begingroup
\etocsetlevel{exercise}{2}    % but:
\etocsetlevel{chapter}{3}     %      no chapters
\etocsetlevel{subsection}{3}  %      no subsections
\etocsetlevel{part}{3}        %      no parts
\etocsetstyle{exercise}{}{}   % \etocname = exercise number
  {\noindent\etocname\strut\leaders\etoclineleaders\hfill\etocpage\par}
  {\pagebreak[2]\vskip\baselineskip}
\etocsetstyle{section}{}{}
  {\noindent\strut{\bfseries\large\etocnumber\hskip.5em\etocname}\par}
  {\nopagebreak[3]}
\etocruledstyle[4]{\Large\bfseries List of the exercises}
\setlength{\columnseprule}{.4pt}
\tableofcontents
\endgroup
```

In the above, recall that `LATEX` counters are global. The current `tocdepth` value is 2, and if not reset it will prevent `hyperref` to assign bookmarks to sub-subsections (level 3). The global `hyperref` option `bookmarksdepth` can be used to avoid having to systematically reset `tocdepth` after having changed it.

5.3. A TOC with a fancy layout

Another question (numbered 61297) on the `TEX StackExchange` site was about using `LATEX` to obtain a table of contents where the sections from a given chapter would 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). The next page is the result of my effort at using **etoc** for this problem. How this was done will be found on the above cited site.

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:

5. Surprising uses of *etoc*

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

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
5. The Causal	42—49
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

5.4. Another compatibility mode

As explained in the section 2.3, 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.¹⁰ With these commands **etoc** will construct a TOC completely identical to what would have been done by one of the document

¹⁰and if moreover one just wants to keep the same layout as in the default, one may question why

class: article, book, scrartcl, scrbook.¹¹ The number and the name of each entry are each separately an `hyperref` link, as is always the case with `etoc`, when not in compatibility mode. Replacing `\etocnumber` with `\etocthenumber` will give a TOC where the numbers are not links anymore, but the names still are. Or one may decide to use `\etocthename` and keep an hyperlinked number with `\etocnumber`.

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

Contents

1. Initial motivation: nested lists	4
1.1. Limitations intrinsic to the use of environments	4
2. Line styles and toc display style	4
2.1. <code>\etocsettocstyle</code> for the toc display	5
2.2. <code>\etocsetstyle</code> for the line styles	5
2.3. Compatibility mode	5
3. Arbitrarily many TOCs, and local ones too	6
<i>Labeling and reusing elsewhere (3.1)–The <code>hyperref</code> option <code>bookmarksdepth</code> (3.2)–On manually adding layout commands to the <code>.toc</code> file (3.3)–Shuffling the levels with <code>\etocsetlevel</code> (3.4)–Local table of contents for this part (3.5).</i>	
4. A simple example	9
5. Surprising uses of <code>etoc</code>	11
5.1. The TOC of TOCs	11
5.2. Arbitrary “Lists Of...”	12
5.3. A TOC with a fancy layout	13
5.4. Another compatibility mode	15
5.5. The TOC as a tree	17

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
```

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.

¹¹For 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:

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

(and analogously for part, respectively book).


```

\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}}
{}%

```

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 subsection entries.

Also, a fancier layout has been commented out.

5.5. The TOC as a tree

Using `tikz` and the package `tikz-qtree`¹² we shall display the table of contents of this part as a tree. The technique (this whole subsection perhaps should have double dangerous-bend signs) is to use the `etoc` modified command `\tableofcontents` not for typesetting, but to prepare a macro, or rather here a token list with name `\treetok`, with all the instructions to be executed later. Putting `\etocnumber` or `\etocname` commands in `\treetok` would be of no use: to which number or name would they refer to, in such a delayed execution? Rather the *contents* of `\etocthenumber` or `\etocthename` are added with suitable decoration to `\treetok`: the `etoc` line styles are modified to expand once `\etocthename` (for example) at the time of the execution and then add the outcome to `\treetok` for later execution in a `tikzpicture`.

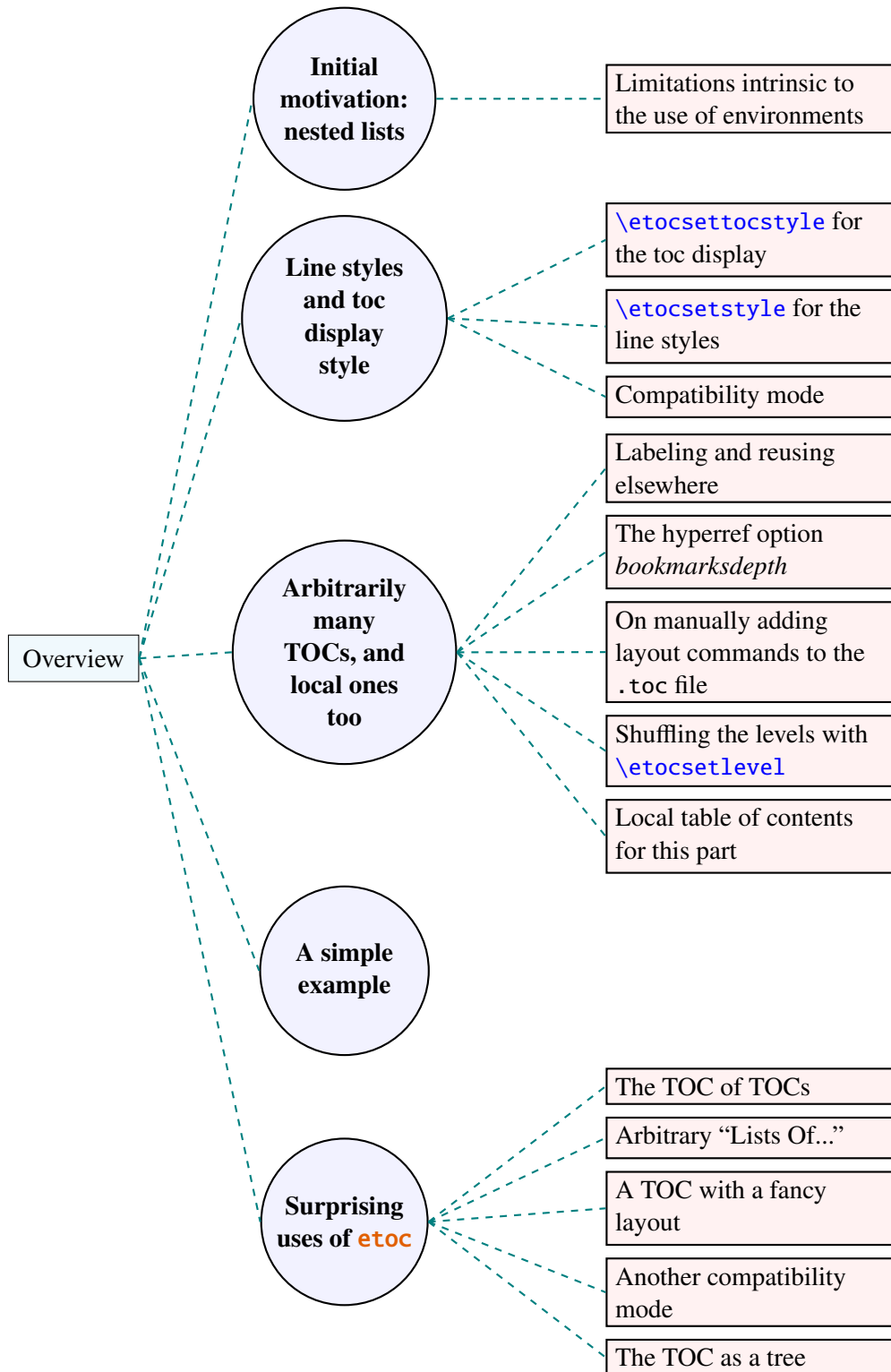
It is perfectly possible to do the same with the `hyperref` link: rather than trying to do things with `\etoclink` (which is a robust command, but one could access the contents of `\csname etoclink \endcsname`),¹³ I mention to the advanced users that the simpler method is to store `\hyperlink{A}{B}` in `\treetok` where A stands for the *expansion* of `\Hy@tocdestname`, and B stands for the (one time) *expansion* of `\etocthename`.

This being said, here in our use of `tikz-qtree` we have been confronted with the problem that the hyperlinks, even if correctly added to the `\treetok` token list, appeared in the final document in (most of the time) completely wrong locations.¹⁴

¹²<http://ctan.org/pkg/pgf>, resp. <http://ctan.org/pkg/tikz-qtree>

¹³warning: it contains `\Hy@tocdestname` *unexpanded*.

¹⁴`dvipdfmx` gives me errors: **** WARNING **** Annotation out of page boundary. Current page's MediaBox: [0 0 595.276 841.89] Annotation: [94.3997 1200.33 146.512 1209.98]



Using `hopgf.sty` [2012/12/16 v1.1] of HEIKO OBERDIEK did not fix it. I read on the internet that there is some problem with `pgf` with regards to hyperlinks, but I don't know if this is the same issue here or something specific to the `\pgftree` TikZ command, or if the way I tried to put the links in the nodes was just completely wrong (as this is my first ever use of TikZ).

Contents

As this whole thing is with double dangerous-bend signs, I shall now include the code used, with not much commenting:

```

\makeatletter
\newtoks\treetok \newtoks\tmptok

\begin{group}
\def\appendtotok #1#2{% #1=token list, #2=macro, expands once #2
  #1\expandafter\expandafter\expandafter
    {\expandafter\the\expandafter #1#2}}

\def\PrepareSectionNode{%
  \tmptok {\centering\bfseries}%
  \appendtotok\tmptok\etocthenam
  \edef\qtreenode {[. {\noexpand\parbox{2cm}{\the\tmptok}} ]}%
}

\def\PrepareSubsectionNode{%
  \tmptok {\raggedright}%
  \appendtotok\tmptok\etocthenam
  \edef\qtreenode{[. {\noexpand\parbox{4cm}{\the\tmptok}} ]}%
}

% See the tikz-qtree documentation for the syntax of trees.
% (the space at the end of \qtreenode is important)
% The whole command is constructed inside the token list \treetok

\etocsetstyle{section}
{ \treetok{\Tree [.Overview ]\etocskipfirstprefix}
  \appendtotok\treetok{ ] } % ] not active!
  {\PrepareSectionNode \appendtotok\treetok\qtreenode}
  {\appendtotok\treetok{ ] }\global\treetok\expandafter{\the\treetok}}

\etocsetstyle{subsection}
{ \etocskipfirstprefix
  \appendtotok\treetok{ ] }
  {\PrepareSubsectionNode \appendtotok\treetok\qtreenode}
  {\appendtotok\treetok{ ] } }

\setcounter{tocdepth}{2}
\etocsettocstyle{ }{ }
% the whole effect of \tableofcontents will be to fill \treetok
% according to the syntax expected by tikz-qtree.
% This is always in a group, so we used \global in the section-finish code.
% Also, there is a \par, but we are already in vertical mode.
\tableofcontents \ref{toc:partone}
\end{group}

% Time now to display the tree (see tikz-qtree documentation)
\begin{tikzpicture}[grow'=right]
\tikzset{sibling distance=1ex,
  level 1/.style={level distance=4cm},
  level 2/.style={level distance=6cm},
  every level 0 node/.style={draw,fill=cyan!5,inner sep=6pt},
  every level 1 node/.style={circle,draw,thick,fill=blue!5},
  every level 2 node/.style={draw,thick,fill=red!5},
  edge from parent/.style= {
    draw, thick, color=teal,
    edge from parent path=
      {[dashed](\tikzparentnode.east) -- (\tikzchildnode.west)}}
}
\the\treetok
\end{tikzpicture}

```

Part II.

Package commands for line styles

Contents of this second part

- 6** The `\etocsetstyle` command (page 20)
 - 6.1 The `\etocname` and `\etocpage` commands (p. 20)
 - 6.2 The `\etocskipfirstprefix` command (p. 21)
 - 6.3 The `\etocnumber` command (p. 21)
 - The `\etocifnumbered` switch.
 - 6.4 The `\etocthenname`, `\etocthenumber`, and `\etocthepage` commands (p. 22)
 - 6.5 The `\etoclink` command (p. 23)
- 7** The `\etocsetlevel` command (page 23)
- 8** Scope of commands added to the `.toc` file (page 24)
 - 8.1 Testing the scope (p. 24)
 - 8.2 This is a (pale) red subsection for illustrative purposes (p. 24)
- 9** Am I also red? (page 24)

6. The `\etocsetstyle` command

6.1. 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:¹⁵

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

```
\etocsetstyle{subsection}
{\begin{enumerate}}
```

¹⁵the 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.

6. The `\etocsetstyle` command

```
{\normalfont\item}  
{\etocname{}} (p.\etocpage)}  
{\end{enumerate}}  
  
\etocsetstyle{subsubsection}  
{\par\nobreak\begingroup\normalfont  
  \footnotesize\itshape\etocskipfirstprefix}  
{\allowbreak\,--\,}  
{\etocname}  
{.\hfil\par\endgroup\pagebreak[3]}
```

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

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 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.¹⁶

6.2. 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!

6.3. 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

¹⁶ up to version 1.07a the package put an `\xspace` in each of `\etocname`, `\etocnumber`, and `\etocpage`, but this wasn't such a great idea and has now been removed.

6. The `\etocsetstyle` command

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 protected command which explains why it can be used inside the label specification without needing an added `\protect`.

6.3.1. 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).¹⁷

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\colorbox{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}`.

6.4. The `\etocthenname`, `\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 the `\etocthe...` commands.

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

¹⁷as seen we use `\colorbox` 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`.

6.5. 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`, except that `\etocname` starts with a `\leavevmode` (this was found out to be necessary when `hyperref` has added its link data, during testing of various uses of the first versions of **etoc**).

The command `\etoclink` is protected.

7. The `\etocsetlevel` command

As already explained in the section 5, 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}
\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 the section 5 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:

```
\setcounter{tocdepth}{3}
\etocdefaultlines % use the package default line styles. At this early stage in
                  % the document they had not yet been modified by \etocsetstyle
                  % commands, so \etoclines could have been used, too.
\etocmarkboth\contentsname
\etocmulticolstyle[1] % one-column display
```

8. Scope of commands added to the .toc file

```
{\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 than what is achieved by the default L^AT_EX table of contents. But here, we have few subsub-sections and it is more interesting to drop them and print in a visually different manner sections and subsections.

8. Scope of commands added to the .toc file

8.1. 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
```

8.2. 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.

9. 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 1.1.

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 III.

Package commands for toc display styles

```
\setcounter{tocdepth}{-3}  
\localtableofcontents \label{toc:globalcmds}
```

10. 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})
```

10.1. The command `\etocruledstyle`

The general format of `\etocruledstyle` is:

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

Note that the title is horizontal material, if it does not fit on one line it should be put in a `\parbox` of a given width. We did this and even enclosed the parboxes in `\fbox`s to get frames around them. For the example with the standard formatting we did not use an `\fbox` and got rid of the horizontal rules via:

```
\renewcommand{\etocptoprul}{\hrule height 0pt}
```

The green frame for the heading of the table of contents at the [start of the second part of this document](#) was obtained with:

```
\etocruledstyle[1]{\etocfontminusone\color{green}%  
  \fboxrule1pt\fboxsep1ex  
  \framebox[\linewidth]  
    {\normalcolor\hss Contents of this second part\hss}}
```

10.2. The command `\etocmulticolstyle`

This is also 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, and from 2 on is passed to a `multicols` environment). The *<heading>* should be some ‘vertical’ material like:

10. Specifying the toc display style

$\langle heading \rangle = \backslash section* \{ \langle title \rangle \}$

[New with 1.07] An explicit $\backslash par$ not being accepted in the $\langle heading \rangle$ argument (this is actually a restriction originating in the `multicols` environment), an implicit one is automatically added by **etoc** at the end of the argument, as in this example which shows how the main table of contents of this document was configured:

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

After $\backslash etocmulticolstyle$ all future $\backslash tableofcontents$ will use the specified style, if not changed in-between. A shortcut for just one table of contents and not affecting the styles of later TOCs is:

$\backslash etocmulticol[\langle number_of_columns \rangle] \{ \langle heading \rangle \}$

And there is also $\backslash etoclocalmulticol[\langle number_of_columns \rangle] \{ \langle heading \rangle \}$.

10.2.1. The command $\backslash etoctocstyle$

$\backslash etoctocstyle[\langle kind \rangle] \{ \langle number_of_columns \rangle \} \{ \langle title \rangle \}$
= $\backslash etocmulticolstyle[\langle number_of_columns \rangle] \{ \backslash kind* \{ \langle title \rangle \} \}$

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

$\backslash etoctocstylewithmarks[\langle kind \rangle] \{ \langle number_of_columns \rangle \} \{ \langle title \rangle \} \{ \langle mark \rangle \}$
= $\backslash etocmulticolstyle[\langle number_of_columns \rangle] \{ \backslash kind* \{ \langle title \rangle \backslash markboth \{ \backslash MakeUppercase \{ \langle mark \rangle \} \} \} \}$

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

Do we really want paragraph entries in the TOC?

really?

10.3. The command $\backslash etocruled$

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

$\backslash etocruled[\langle number_of_columns \rangle] \{ \langle title \rangle \}$

And the local form will be $\backslash etoclocalruled$.

10.4. The commands $\backslash etocframedstyle$ and $\backslash etocframed$

Same mechanism:

$\backslash etocframedstyle[\langle number_of_columns \rangle] \{ \langle title \rangle \}$

and the accompanying shortcut:

$\backslash etocframed[\langle number_of_columns \rangle] \{ \langle title \rangle \}$

Here the entire table of contents is framed, hence this can only work if it fits on a page. Note

that the title itself is not framed, if one wants a frame one should set it up inside the `<title>` argument to `\etocframedstyle` or `\etocframed`. There is also `\etoclocalframedstyle` and `\etoclocalframed`.

10.5. Headings, titles, `\etocinnertopsep`

[*modified in v1.07*] There is a slight difference between `\etocmulticolstyle` and `\etocruledstyle` or `\etocframedstyle`. For `\etocmulticolstyle` the mandatory `<heading>` argument can be something like: `\section*{Table of Contents}`. On the contrary `\etocruledstyle` and `\etocframedstyle` expect an argument “in LR mode” (to use the terminology from the *LaTeX, a document preparation system*). This means that multiline contents arguments to `\etocruledstyle` or `\etocframedstyle` must be enclosed in something like a `\parbox`.

[*new in v1.07*] The command `\etocmulticolstyle` now also accepts horizontal mode material in its mandatory argument `<heading>`: it internally automatically adds a closing `\par`. So one can use for example `\etocmulticolstyle{Hello World}`. Speaking of `\par`, there is a `multicols` aspect which has nothing to do with **etoc**, the input on the left creates a compilation error:

<pre>\begin{multicols}{2}[hello\par world] someone here? \end{multicols}</pre>	<pre>\let\oldpar\par \begin{multicols}{2}[hello\oldpar world] at least me. \end{multicols}</pre>
--	--

But the version on the right does not (it disguises `\par` so as to be acceptable). **etoc** provides `\etocoldpar` as a substitute for `\par` (it does `\let\etocoldpar\par` just before the `multicols` environment and automatically adds it to close the heading, before the vertical skip of value `\etocinnertopsep`).¹⁸ The command `\etocoldpar` can also be used explicitly in the mandatory argument to `\etocmulticolstyle`.

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 by `\renewcommand*{\etocinnertopsep}{<new_value>}`, not with `\setlength`.

10.6. 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) expect to start in ‘vertical mode’.

It 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.

¹⁸this 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.

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

The previously described commands `\etocmulticolstyle`, `\etocruledstyle`, and `\etocframedstyle` actually call `\etocsettocstyle` as a lower-level routine, and start a `multicols` environment in `{\before_toc}` to close it in `{\after_toc}`.

10.7. The compatibility mode `\etocstandarddisplaystyle`

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

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

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

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

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

Independently of the situation with the `memoir` class there is generally speaking a hook macro called `\etocaftertitlehook` which is inhibited by using the starred variants of the displaying commands. Except for the `memoir` class, this hook is initially defined to do nothing. There is also `\etocaftercontentshook`, similarly defined to do nothing. They can be used for some special effects.

12. Table of contents for this part

12.1. 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.¹⁹ The original was invisibly defined with a label at the beginning

¹⁹the present document uses the `scrartcl` class, and we check here that the **etoc** compatibility mode does respect the customizing done via the class commands.

of this part III.

```
\KOMAOptions{toc=left}
\etocstandarddisplaystyle % necessary for the display to obey toc=left
\etocstandardlines
\setcounter{tocdepth}{3}
\tableofcontents \ref{toc:globalcmds}
```

Contents

10. Specifying the toc display style	25
10.1. The command <code>\etocruledstyle</code>	25
10.2. The command <code>\etocmulticolstyle</code>	25
10.2.1. The command <code>\etocstyle</code>	26
10.3. The command <code>\etocruled</code>	26
10.4. The commands <code>\etocframedstyle</code> and <code>\etocframed</code>	26
10.5. Headings, titles, <code>\etocinnertopsep</code>	27
10.6. The command <code>\etocsettocstyle</code>	27
10.7. The compatibility mode <code>\etocstandarddisplaystyle</code>	28
11. Starred variants of the <code>\tableofcontents</code> etc... commands	28
12. Table of contents for this part	28
12.1. Testing the compatibility mode	28
12.2. A framed display	29
12.3. A (crazy) inline display	30

12.2. A framed display

We now opt for a “framed” style, using the package default line styles and some colors added.

```
\etocdefaultlines
\beginngroup % we use a group to limit the scope of the next commands
\renewcommand{\etoccolumnsep}{2em}
\renewcommand{\etocinnerleftsep}{1.5em}
\renewcommand{\etocinnerrightsep}{1.5em}
% specify a background color for the toc contents
\renewcommand{\etocbkgcolorcmd}{\color{yellow!10}}
%\renewcommand{\etocbkgcolorcmd}{\relax}
% set up the top and bottom rules
\renewcommand{\etoctoprule}{\hrule height 1pt}
\renewcommand{\etocbottomrule}{\hrule height 1pt}
\renewcommand{\etocbottomrulecolorcmd}{\color{red!25}}
% set up the left and right rules
\renewcommand{\etoclefttrule}{\vrule width 5pt}
\renewcommand{\etocrighttrule}{\vrule width 5pt}
\renewcommand{\etoclefttrulecolorcmd}{\color{red!25}}
\renewcommand{\etocrighttrulecolorcmd}{\color{red!25}}
% use \fcolorbox to set up a colored frame for the title
```

```
\fboxrule1pt
\etocframedstyle{\normalsize\rmfamily\itshape
  \fcolorbox{red}{white}{\parbox{.8\linewidth}{\centering
    This is a table of contents \textit{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 the
    current page. It has the label \texttt{toc:b}|.}}}
% set up a label for future (or earlier...) reference
\setcounter{tocdepth}{3}
\tableofcontents \label{toc:b} \ref{toc:globalcmds}
\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 the current page. It has the label `toc:b`.*

Specifying the toc display style

10, p. 25

The command `\etocruledstyle`

10.1, p. 25

The command `\etocmulticolstyle`

10.2, p. 25

The command `\etocdisplaystyle`.

The command `\etocruled`

10.3, p. 26

The commands `\etocframedstyle`
and `\etocframed`

10.4, p. 26

Headings, titles, `\etocinnertop-`
`sep`

10.5, p. 27

The command `\etocsettocstyle`

10.6, p. 27

The compatibility mode `\etocstandarddisplaystyle`

10.7, p. 28

**Starred variants of the `\tableof-`
`contents` etc... commands**

11, p. 28

Table of contents for this part

12, p. 28

Testing the compatibility mode

12.1, p. 28

A framed display

12.2, p. 29

A (crazy) inline display

12.3, p. 30

12.3. 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, not for print!

Here is the code which will be used:

```
\begin{group}
\setcounter{tocdepth}{3}
\newsavebox{\forsubsection}
\etocsetstyle{part}{\etocskipfirstprefix}{. \upshape}{\bfseries\etocname:~}{~}
\etocsetstyle{section}{\itshape\etocskipfirstprefix}{, }{\mdseries\etocname}
\etocsetstyle{subsection}
  {\begin{lrbox}{\forsubsection}\upshape\etocskipfirstprefix}
  {; }
  {\etocname}
```

Contents

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

Here is the inline table of contents. **Overview:** *Initial motivation: nested lists²⁰, Line styles and toc display style²¹, Arbitrarily many TOCs, and local ones too²², A simple example, Surprising uses of **etoc**²³. **Package commands for line styles:** The `\etocsetstyle` command²⁴, The `\etocsetlevel` command, Scope of commands added to the `.toc` file²⁵, Am I also red?. **Package commands for toc display styles:** Specifying the toc display style²⁶, Starred variants of the `\tableofcontents` etc... commands, Table of contents for this part²⁷. **Using and customizing **etoc**:** Summary of the main style commands²⁸, Customizing **etoc**²⁹. **Tips:** ... and tricks³⁰. **The code:** Implementation.*

²⁰Limitations intrinsic to the use of environments.

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

²²Labeling and reusing elsewhere; The hyperref option *bookmarksdepth*; On manually adding layout commands to the `.toc` file; Shuffling the levels with `\etocsetlevel`; Local table of contents for this part.

²³The TOC of TOCs; Arbitrary “Lists Of...”; A TOC with a fancy layout; Another compatibility mode; The TOC as a tree.

²⁴The `\etocname` and `\etocpage` commands; The `\etocskipfirstprefix` command; The `\etocnumber` command (*The `\etocifnumbered` switch*); The `\etocthename`, `\etocthenumber`, and `\etocthepage` commands; The `\etoclink` command.

²⁵Testing the scope; **This is a (pale) red subsection for illustrative purposes.**

²⁶The command `\etocruledstyle`; The command `\etocmulticolstyle` (*The command `\etocdisplaystyle`*); The command `\etocruled`; The commands `\etocframedstyle` and `\etocframed`; Headings, titles, `\etocinnertopsep`; The command `\etocsettocstyle`; The compatibility mode `\etocstandarddisplaystyle`.

²⁷Testing the compatibility mode; A framed display; A (crazy) inline display.

²⁸Setting up local styles; Setting up toc display styles; Displaying tables of contents; Labels and references; The package default line styles: `\etocdefaultlines`; One more example TOC layout.

²⁹Customizing the **etoc** pre-defined line styles; Customizing the toc display styles.

³⁰Hacking framed parboxes; Interverting the levels; All subsections of this document; Compatibility with other packages; T_EXnical matters; Errors and catastrophes.

Part IV.

Using and customizing **etoc**

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	The package default line styles: <code>\etocdefaultlines</code>
. 13, p. 32 13.5, p. 33
Setting up local styles . . . 13.1, p. 32	One more example TOC layout . . .
Setting up toc display styles 13.6, p. 36
. 13.2, p. 32	Customizing etoc 14, p. 38
Displaying tables of contents	Customizing the etoc pre-defined line
. 13.3, p. 33	styles 14.1, p. 38
Labels and references . . . 13.4, p. 33	Customizing the toc display styles . .
 14.2, p. 39

13. Summary of the main style commands

13.1. Setting up local styles

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

13.2. 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>}
```


13.3. 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

```

13.4. 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.

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

Activating the use of the package default line styles³¹ 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 L^AT_EX default.³² 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 L^AT_EX default, sub-sections are not visually much different

³¹they were written at a very early stage in the development of the package, and version 1.07e has, among other things, modified the previous unsatisfactory use of penalties and vertical spacing commands. This will cause differences to documents having been compiled with earlier versions, apologies for that.

³²and 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).

from sections (especially for document classes with a `\chapter` command), so the result is less hierarchical in appearance than in the L^AT_EX default.

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

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

```
\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
    being printed as chapters, and subsections as sections.
  }}}
\etocdefaultlines
\setcounter{tocdepth}{1}
\begingroup
\etocsetlevel{section}{0}
\etocsetlevel{subsection}{1}
\renewcommand*{\etocfontone}{\normalfont \normalsize}
\tableofcontents
\endgroup
```

<i>This is the global table of contents on two columns, using etoc default line styles, but with sections being printed as chapters, and subsections as sections.</i>
--

Part I. Overview

1. Initial motivation: nested lists

Limitations intrinsic to the use of environments 1.1, p. 4

2. Line styles and toc display style

`\etocsettocstyle` for the toc display 2.1, p. 5

`\etocsetstyle` for the line styles 2.2, p. 5

Compatibility mode 2.3, p. 5

3. Arbitrarily many TOCs, and local ones too

Labeling and reusing elsewhere 3.1, p. 6

The hyperref option *bookmarksdepth* 3.2, p. 6

On manually adding layout commands to the .toc file 3.3, p. 7

Shuffling the levels with `\etocsetlevel` 3.4, p. 7

Local table of contents for this part 3.5, p. 8

4. A simple example

5. Surprising uses of **etoc**

The TOC of TOCs	5.1, p. 11
Arbitrary “Lists Of...”	5.2, p. 12
A TOC with a fancy layout	5.3, p. 13
Another compatibility mode	5.4, p. 15
The TOC as a tree	5.5, p. 17

Part II. Package commands for line styles

6. The `\etocsetstyle` command

The <code>\etocname</code> and <code>\etocpage</code> commands	6.1, p. 20
The <code>\etocskipfirstprefix</code> command	6.2, p. 21
The <code>\etocnumber</code> command	6.3, p. 21
The <code>\etocthenname</code> , <code>\etocthenumber</code> , and <code>\etocthepage</code> commands	6.4, p. 22
The <code>\etoclink</code> command	6.5, p. 23

7. The `\etocsetlevel` command

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

Testing the scope	8.1, p. 24
This is a (pale) red subsection for illustrative purposes	8.2, p. 24

9. Am I also red?

Part III. Package commands for toc display styles

10. Specifying the toc display style

The command <code>\etocruledstyle</code>	10.1, p. 25
--	-------------

The command <code>\etocmulticolstyle</code>	10.2, p. 25
---	-------------

The command <code>\etocruled</code>	10.3, p. 26
---	-------------

The commands <code>\etocframedstyle</code> and <code>\etocframed</code>	10.4, p. 26
---	-------------

Headings, titles, <code>\etocinnertopsep</code>	10.5, p. 27
---	-------------

The command <code>\etocsettocstyle</code>	10.6, p. 27
---	-------------

The compatibility mode <code>\etocstandard-displaystyle</code>	10.7, p. 28
--	-------------

11. Starred variants of the `\table-ofcontents` etc... commands

12. Table of contents for this part

Testing the compatibility mode	12.1, p. 28
A framed display	12.2, p. 29
A (crazy) inline display	12.3, p. 30

Part IV. Using and customizing **etoc**

13. Summary of the main style commands

Setting up local styles	13.1, p. 32
Setting up toc display styles	13.2, p. 32
Displaying tables of contents	13.3, p. 33
Labels and references	13.4, p. 33
The package default line styles: <code>\etocdefaultlines</code>	13.5, p. 33
One more example TOC layout	13.6, p. 36

14. Customizing `etoc`

Customizing the `etoc` pre-defined line styles 14.1, p. 38

Customizing the toc display styles 14.2, p. 39

All subsections of this document 15.3, p. 41

Compatibility with other packages 15.4, p. 42

T_EXnical matters 15.5, p. 43

Errors and catastrophes 15.6, p. 43

Part V. Tips**15. ... and tricks**

Hacking framed parboxes 15.1, p. 40

Interverting the levels 15.2, p. 41

Part VI. The code**16. Implementation****13.6. One more example TOC layout**

I got motivated by a question³³ I saw on the T_EX StackExchange site. I copied the color RGB specifications from an answer which had been provided to the question. The `\etoc-framedstyle` puts the title on the top rule in a centered position. This is not very convenient for this example so we included the title as part of the `\start` code at section level, to get it *inside* the frame.

```
\setcounter{tocdepth}{3}
\begingroup
\definecolor{subsecnum}{RGB}{13,151,225}
\definecolor{secbackground}{RGB}{0,177,235}
\definecolor{tocbackground}{RGB}{212,237,252}

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

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

³³tex.stackexchange.com/questions/83184

13. Summary of the main style commands

```

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

\etocsetstyle{section}
{\coloredstuff{white}
{\hfil \hyperref[toc:b]{\bfseries\large I am a twin of
that other TOC (click me!)}\hfil}}
{\vskip3pt\sffamily\small}
{\coloredstuff{white}{\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

```

The coding is a bit involved³⁴ as it does not use any additional package. Also, it was written at some early stage and I have not revised it since.

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

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

³⁴and reveals the author's preference for the T_EX syntax...

I am a twin of that other TOC (click me!)		
10	Specifying the toc display style	25
10.1	The command <code>\etocruledstyle</code>	25
10.2	The command <code>\etocmulticolstyle</code>	25
10.2.1	The command <code>\etocstyle</code>	26
10.3	The command <code>\etocruled</code>	26
10.4	The commands <code>\etocframedstyle</code> and <code>\etocframed</code> . . .	26
10.5	Headings, titles, <code>\etocinnertopsep</code>	27
10.6	The command <code>\etocsettocstyle</code>	27
10.7	The compatibility mode <code>\etocstandarddisplaystyle</code> . . .	28
11	Starred variants of the <code>\tableofcontents</code> etc... commands	28
12	Table of contents for this part	28
12.1	Testing the compatibility mode	28
12.2	A framed display	29
12.3	A (crazy) inline display	30

14. Customizing **etoc**

14.1. Customizing the **etoc** pre-defined line styles

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

```

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

\newcommand*\etocsepminustwo{4ex plus .5ex minus .5ex}
\newcommand*\etocsepminusone{4ex plus .5ex minus .5ex}
\newcommand*\etocsepzero{2.5ex plus .4ex minus .4ex}
\newcommand*\etocsepone{1.5ex plus .3ex minus .3ex}
%%\newcommand*\etocseptwo{1ex \@plus .15ex \@minus .15ex} % modified in 1.07e
\newcommand*\etocseptwo{.5ex plus .1ex minus .1ex}
\newcommand*\etocseptthree{.25ex plus .05ex minus .05ex}

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

```

14. Customizing *etoc*

```
\newcommand*\etocminusoneleftmargin{1em}
\newcommand*\etocminusonerightmargin{1em}

\newcommand*\etocbaselinespreadminus-two{1}
\newcommand*\etocbaselinespreadminus-one{1}
\newcommand*\etocbaselinespreadzero{1}
\newcommand*\etocbaselinespreadone{1}
\newcommand*\etocbaselinespreadtwo{1}
\newcommand*\etocbaselinespreadthree{.9}
\newcommand*\etoclineleaders
  {\hbox{\normalfont\normalsize\hbox to 2ex {\hss.\hss}}}
\newcommand*\etocabbrevpagename{p.~}
\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*.³⁵

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.

14.2. Customizing the toc display styles

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

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

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

% for the framed style only:

```
\newcommand*\etocinnerleftsep{2em}
\newcommand*\etocinnerrightsep{2em}
```

³⁵with 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*\etocinnerbottomsep{3.5ex}

\newcommand*\etoclefttrule{\vrule}
\newcommand*\etocrightrule{\vrule}
\newcommand*\etocbottomrule{\hrule}
\newcommand*\etoclefttrulecolorcmd{\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{\etoctoprule}{\hrule height 1ex}
```

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

The top and bottom rules do not have to be rules and can be horizontal *leaders* (of a specified height) in the general \TeX sense. However the left and right rules are not used as (horizontal) leaders but as objects of a given specified width. Note that *only* the Plain \TeX syntax for rules is accepted here.

Part V.

Tips

15. ... and tricks

15.1. Hacking framed parboxes

```

\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}`

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

Specifying the toc display style	The command <code>\etocsettocstyle</code>
10, p. 25	10.6, p. 27
The command <code>\etocruledstyle</code>	The compatibility mode <code>\etocstandard-</code>
10.1, p. 25	<code>displaystyle</code>
The command <code>\etocmulticolstyle</code>	Starred variants of the <code>\tableofcon-</code>
10.2, p. 25	tents etc... commands
The command <code>\etocdisplaystyle</code> .	Table of contents for this part
The command <code>\etocruled</code> 10.3, p. 26	12, p. 28
The commands <code>\etocframedstyle</code> and	Testing the compatibility mode
<code>\etocframed</code>	12.1, p. 28
10.4, p. 26	A framed display
Headings, titles, <code>\etocinnertopsep</code>	12.2, p. 29
10.5, p. 27	A (crazy) inline display
	12.3, p. 30

15.2. Interverting the levels

Let us display and count all subsections occurring in this document (see section 5 for other uses of this technique):

```
\setcounter{tocdepth}{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.\sim\etocpage)}
  {\end{enumerate}}
\renewcommand{\etoccolumnsep}{2.75em}
\renewcommand{\columnseprule}{1pt}
\etocmarkbothnouc{List of all subsections}
\etocmulticol[3]{\subsection{All subsections of this document}}
\endgroup
```

15.3. All subsections of this document

i. Limitations intrinsic to the use of environments (1.1, p. 4)	iii. <code>\etocsetstyle</code> for the line styles (2.2, p. 5)	elsewhere (3.1, p. 6)
ii. <code>\etocsettocstyle</code> for the toc display (2.1, p. 5)	iv. Compatibility mode (2.3, p. 5)	vi. The hyperref option <code>bookmarksdepth</code> (3.2, p. 6)
	v. Labeling and reusing	vii. On manually adding layout commands to the

.toc file (3.3, p. 7)	xx. Testing the scope (8.1, p. 24)	xxxiii. Setting up toc display styles (13.2, p. 32)
viii. Shuffling the levels with <code>\etocsetlevel</code> (3.4, p. 7)	xxi. This is a (pale) red subsection for illustrative purposes (8.2, p. 24)	xxxiv. Displaying tables of contents (13.3, p. 33)
ix. Local table of contents for this part (3.5, p. 8)	xxii. The command <code>\etocruledstyle</code> (10.1, p. 25)	xxxv. Labels and references (13.4, p. 33)
x. The TOC of TOCs (5.1, p. 11)	xxiii. The command <code>\etocmulticolstyle</code> (10.2, p. 25)	xxxvi. The package default line styles: <code>\etocdefaultlines</code> (13.5, p. 33)
xi. Arbitrary “Lists Of...” (5.2, p. 12)	xxiv. The command <code>\etocruled</code> (10.3, p. 26)	xxxvii. One more example TOC layout (13.6, p. 36)
xii. A TOC with a fancy layout (5.3, p. 13)	xxv. The commands <code>\etocframedstyle</code> and <code>\etocframed</code> (10.4, p. 26)	xxxviii. Customizing the <code>etoc</code> pre-defined line styles (14.1, p. 38)
xiii. Another compatibility mode (5.4, p. 15)	xxvi. Headings, titles, <code>\etocinnertopsep</code> (10.5, p. 27)	xxxix. Customizing the toc display styles (14.2, p. 39)
xiv. The TOC as a tree (5.5, p. 17)	xxvii. The command <code>\etocsettocstyle</code> (10.6, p. 27)	xl. Hacking framed par-boxes (15.1, p. 40)
xv. The <code>\etocname</code> and <code>\etocpage</code> commands (6.1, p. 20)	xxviii. The compatibility mode <code>\etocstandarddisplaystyle</code> (10.7, p. 28)	xli. Interverting the levels (15.2, p. 41)
xvi. The <code>\etocskipfirstprefix</code> command (6.2, p. 21)	xxix. Testing the compatibility mode (12.1, p. 28)	xlii. All subsections of this document (15.3, p. 41)
xvii. The <code>\etocnumber</code> command (6.3, p. 21)	xxx. A framed display (12.2, p. 29)	xliii. Compatibility with other packages (15.4, p. 42)
xviii. The <code>\etocthename</code> , <code>\etocthenumber</code> , and <code>\etocthepage</code> commands (6.4, p. 22)	xxxi. A (crazy) inline display (12.3, p. 30)	xliv. T _E Xnical matters (15.5, p. 43)
xix. The <code>\etoclink</code> command (6.5, p. 23)	xxxii. Setting up local styles (13.1, p. 32)	xlv. Errors and catastrophes (15.6, p. 43)

15.4. Compatibility with other packages

`etoc` loads the package `multicol`.³⁶ 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 exists) are read into memory by `etoc` once, at the time of `\usepackage{etoc}`. The `.toc` file will be opened for write operations only at the time of the first TOC displaying command.

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

³⁶ up to version 1.07a it also used package `xspace`, and but this has been removed.

of the present document are fully linked.³⁷

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.

15.5. 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.

15.6. 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 styles expect to start 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 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).

³⁷except the tree TOC ...

Part VI.

The code

This source file `etoc.dtx` produces `etoc.sty` when one does `latex etoc.dtx` or `pdflatex etoc.dtx` (an `etoc.ins` file is also produced, for distributions expecting it for installation). Two more runs are necessary to finish producing the documentation. The `etoc.sty` file should be moved to a suitable location within the T_EX installation.

16. Implementation

Writing-up source code comments is hopefully for a future release.

```
1 \ProvidesPackage{etoc}
2 [2013/03/01 v1.07e easily customizable TOCs (jfb)]
3 \NeedsTeXFormat{LaTeX2e}
4 \RequirePackage{multicol}
5 %% \RequirePackage{xspace} %% REMOVED (1.07b)
6 \DeclareOption*{\PackageWarning{etoc}{Option ‘\CurrentOption’ is unknown.}}
7 \ProcessOptions\relax

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

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

placeholder for comments
32 \ifclassloaded{memoir}{\def\Etoc@minf{-\thr@@}}{\def\Etoc@minf{-\tw@}}
```

16. Implementation

```

33 \def\Etoc@@minustwo@{-\tw@}
34 \let\Etoc@@minusone@@\m@ne
35 \chardef\Etoc@@zero@@ 0
36 \let\Etoc@@one@@ \@ne
37 \let\Etoc@@two@@ \tw@
38 \let\Etoc@@three@@ \thr@@
39 \chardef\Etoc@@four@@ 4
40 \chardef\Etoc@@five@@ 5
41 \chardef\Etoc@@six@@ 6
42 \let\Etoc@localtop\Etoc@@minustwo@@
43 \def\Etoc@@minustwo@{minustwo}
44 \def\Etoc@@minusone@{minusone}
45 \def\Etoc@@zero@{zero}
46 \def\Etoc@@one@{one}
47 \def\Etoc@@two@{two}
48 \def\Etoc@@three@{three}
49 \def\Etoc@@four@{four}
50 \def\Etoc@@five@{five}
51 %\def\Etoc@@six@{six}

placeholder for comments

52 \def\Etoc@levellist{}
53 \def\Etoc@newlevel#1{%
54   \def\Etoc@levellist@elt{\noexpand\Etoc@levellist@elt\noexpand}%
55   \edef\Etoc@levellist{\Etoc@levellist\Etoc@levellist@elt#1}}
56 \def\etocsetlevel#1#2{%
57   \expandafter\Etoc@newlevel\csname l@#1\endcsname
58   \ifcase#2\relax
59     \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@zero@@
60     \expandafter\let \csname Etoc@#1@\endcsname\Etoc@@zero@
61   \or
62     \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@one@@
63     \expandafter\let \csname Etoc@#1@\endcsname\Etoc@@one@
64   \or
65     \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@two@@
66     \expandafter\let \csname Etoc@#1@\endcsname\Etoc@@two@
67   \or
68     \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@three@@
69     \expandafter\let \csname Etoc@#1@\endcsname\Etoc@@three@
70   \or
71     \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@four@@
72     \expandafter\let \csname Etoc@#1@\endcsname\Etoc@@four@
73   \or
74     \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@five@@
75     \expandafter\let \csname Etoc@#1@\endcsname\Etoc@@five@
76   \or
77     \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@six@@
78   \else
79     \ifnum#2=\m@ne
80       \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@minusone@@
81       \expandafter\let \csname Etoc@#1@\endcsname\Etoc@@minusone@
82     \else
83       \ifnum#2=-\tw@

```

16. Implementation

```
84 \expandafter\let \csname Etoc@#1@@\endcsname\Etoc@@minustwo@@
85 \expandafter\let \csname Etoc@#1@\endcsname\Etoc@@minustwo@@
86 \else
87 \PackageWarning{etoc}
88 {unexpected value '#2' in \string\etocsetlevel.^}%
89 Should be -2,-1, 0, 1, 2, 3, 4, 5, or 6. Set to 6 (=ignored)}%
90 \expandafter\let\csname Etoc@#1@@\endcsname\Etoc@@six@@
91 \fi\fi\fi}
92 \etocsetlevel{book}{-2}
93 \etocsetlevel{part}{-1}
94 \etocsetlevel{chapter}{0}
95 \etocsetlevel{section}{1}
96 \etocsetlevel{subsection}{2}
97 \etocsetlevel{subsubsection}{3}
98 \etocsetlevel{paragraph}{4}
99 \etocsetlevel{subparagraph}{5}

placeholder for comments
100 \def\Etoc@setflags #1{%
101 \ifcase #1\relax
102 \global\Etoc@vfalse
103 \global\Etoc@ivfalse
104 \global\Etoc@iiifalse
105 \global\Etoc@iifalse
106 \global\Etoc@iffalse
107 \global\Etoc@true
108 \or
109 \global\Etoc@vfalse
110 \global\Etoc@ivfalse
111 \global\Etoc@iiifalse
112 \global\Etoc@iifalse
113 \global\Etoc@itrue
114 \or
115 \global\Etoc@vfalse
116 \global\Etoc@ivfalse
117 \global\Etoc@iiifalse
118 \global\Etoc@iitrue
119 \or
120 \global\Etoc@vfalse
121 \global\Etoc@ivfalse
122 \global\Etoc@iiitrue
123 \or
124 \global\Etoc@vfalse
125 \global\Etoc@ivtrue
126 \or
127 \global\Etoc@vtrue
128 \else
129 \ifnum#1=\m@ne
130 \global\Etoc@vfalse
131 \global\Etoc@ivfalse
132 \global\Etoc@iiifalse
133 \global\Etoc@iifalse
134 \global\Etoc@iffalse
```

16. Implementation

```

135     \global\Etoc@false
136     \global\Etoc@jtrue
137   \else
138     \global\Etoc@vfalse
139     \global\Etoc@ivfalse
140     \global\Etoc@iiifalse
141     \global\Etoc@iifalse
142     \global\Etoc@ifalse
143     \global\Etoc@false
144     \global\Etoc@jfalse
145     \global\Etoc@jjtrue
146   \fi
147 \fi}

placeholder for comments
148 \AtBeginDocument{%
149 \@ifpackageloaded{parskip}{\Etoc@parskiptrue}{}%
150 \@ifpackageloaded{hyperref}{\Etoc@hyperreftrue
151                               \def\Etoc@et@hop#1#2#3#4#5{#1{#3}{#4}{#5}#2}%
152                               \long\def\Etoc@gobblesixorfive#1#2#3#4#5#6{}}
153                               {\def\Etoc@et@hop#1#2#3#4{#1{#3}{#4}#2}%
154                               \long\def\Etoc@gobblesixorfive#1#2#3#4#5{}}}%
155 }

placeholder for comments
156 \def\Etoc@swa#1{%
157   \Etoc@et@hop
158     {\Etoc@savedcontentsline{#1}}
159     {\Etoc@prefix\Etoc@contents}}
160 \def\Etoc@swb#1{%
161   \Etoc@et@hop
162     {\Etoc@savedcontentsline{#1}}
163     {\Etoc@contents}}
164 \let\etocskipfirstprefix\@thirdofthree

placeholder for comments
165 \def\Etoc@etoccontentsline#1{%
166   \global\expandafter\let\expandafter\Etoc@tmp\csname Etoc@#1@@\endcsname
167   \Etoc@skipthisonefalse
168   \let\Etoc@next\Etoc@gobblesixorfive
169   \ifnum\Etoc@tmp=\Etoc@@six@@
170     \Etoc@skipthisonetrue
171   \else
172     \ifEtoc@localtoc
173       \let\Etoc@prenext\relax
174       \ifEtoc@stoptoc
175         \Etoc@skipthisonetrue
176       \fi
177       \ifnum\Etoc@tmp<\Etoc@localtop
178         \def\Etoc@prenext{\global\Etoc@stoptoctrue}%
179         \Etoc@skipthisonetrue
180       \fi
181       \ifEtoc@notactive
182         \def\Etoc@prenext{\Etoc@setflags{\Etoc@tmp}}}%

```

16. Implementation

```
183     \Etoc@skipthisonetrue
184     \fi
185     \Etoc@prenext
186     \fi
187 \fi
188 \ifnum\c@tocdepth<\Etoc@tmp\relax\else
189 \ifEtoc@skipthisone\else
190 \global\let\Etoc@next\relax
191 \ifcase\Etoc@tmp
192     \ifEtoc@v \Etoc@end@five\fi
193     \ifEtoc@iv \Etoc@end@four\fi
194     \ifEtoc@iii \Etoc@end@three\fi
195     \ifEtoc@ii \Etoc@end@two\fi
196     \ifEtoc@i \Etoc@end@one\fi
197     \ifEtoc@ \else \def\Etoc@next{\Etoc@begin@zero}\fi
198     \def\Etoc@contents{\Etoc@contents@zero}%
199     \def\Etoc@prefix{\Etoc@prefix@zero}%
200 \or
201     \ifEtoc@v \Etoc@end@five\fi
202     \ifEtoc@iv \Etoc@end@four\fi
203     \ifEtoc@iii \Etoc@end@three\fi
204     \ifEtoc@ii \Etoc@end@two\fi
205     \ifEtoc@i \else \def\Etoc@next{\Etoc@begin@one}\fi
206     \def\Etoc@contents{\Etoc@contents@one}%
207     \def\Etoc@prefix{\Etoc@prefix@one}%
208 \or
209     \ifEtoc@v \Etoc@end@five\fi
210     \ifEtoc@iv \Etoc@end@four\fi
211     \ifEtoc@iii \Etoc@end@three\fi
212     \ifEtoc@ii \else \def\Etoc@next{\Etoc@begin@two}\fi
213     \def\Etoc@contents{\Etoc@contents@two}%
214     \def\Etoc@prefix{\Etoc@prefix@two}%
215 \or
216     \ifEtoc@v \Etoc@end@five\fi
217     \ifEtoc@iv \Etoc@end@four\fi
218     \ifEtoc@iii \else \def\Etoc@next{\Etoc@begin@three}\fi
219     \def\Etoc@contents{\Etoc@contents@three}%
220     \def\Etoc@prefix{\Etoc@prefix@three}%
221 \or
222     \ifEtoc@v \Etoc@end@five\fi
223     \ifEtoc@iv \else \def\Etoc@next{\Etoc@begin@four}\fi
224     \def\Etoc@contents{\Etoc@contents@four}%
225     \def\Etoc@prefix{\Etoc@prefix@four}%
226 \or
227     \ifEtoc@v \else \def\Etoc@next{\Etoc@begin@five}\fi
228     \def\Etoc@contents{\Etoc@contents@five}%
229     \def\Etoc@prefix{\Etoc@prefix@five}%
230 \else
231     \ifnum\Etoc@tmp=\m@ne
232     \ifEtoc@v \Etoc@end@five\fi
233     \ifEtoc@iv \Etoc@end@four\fi
234     \ifEtoc@iii \Etoc@end@three\fi
```


16. Implementation

```
235 \ifEtoc@ii \Etoc@end@two\fi
236 \ifEtoc@i \Etoc@end@one\fi
237 \ifEtoc@ \Etoc@end@zero\fi
238 \ifEtoc@j \else \def\Etoc@next{\Etoc@begin@minusone}\fi
239 \def\Etoc@contents{\Etoc@contents@minusone}%
240 \def\Etoc@prefix{\Etoc@prefix@minusone}%
241 \else
242 \ifEtoc@v \Etoc@end@five\fi
243 \ifEtoc@iv \Etoc@end@four\fi
244 \ifEtoc@iii \Etoc@end@three\fi
245 \ifEtoc@ii \Etoc@end@two\fi
246 \ifEtoc@i \Etoc@end@one\fi
247 \ifEtoc@ \Etoc@end@zero\fi
248 \ifEtoc@j \Etoc@end@minusone\fi
249 \ifEtoc@jj \else \def\Etoc@next{\Etoc@begin@minustwo}\fi
250 \def\Etoc@contents{\Etoc@contents@minustwo}%
251 \def\Etoc@prefix{\Etoc@prefix@minustwo}%
252 \fi
253 \fi
254 \ifnum\Etoc@tmp=\m@ne\Etoc@parttrue\else\Etoc@partfalse\fi
255 \Etoc@setflags{\Etoc@tmp}%
256 \fi\fi
257 \Etoc@next
258 \@firstoftwo{\Etoc@swa{#1}}{\Etoc@swb{#1}}

placeholder for comments
259 \def\Etoc@lxyz #1#2{%
260 \ifnum\dimexpr\etoclink > 0\relax\leavevmode\fi fall-back
261 \ifnum\dimexpr\etocpage > 0\relax\leavevmode\fi #2}% fall-back
262 \Etoc@getthepage #2\etoc@
263 \ifnum\dimexpr\etocname > 0\relax\leavevmode\fi #1}% fall-back
264 \def\Etoc@thename{#1}%
265 \Etoc@getnb #1\relax\relax\etoc@
266 \ifEtoc@number\else
267 \ifEtoc@part
268 \Etoc@getit #1\hspace\relax\etoc@
269 \fi
270 \fi}

placeholder for comments
271 \def\Etoc@getnb #1{\let\Etoc@next\Etoc@getnb@nohyp
272 \ifEtoc@hyperref\ifx #1\hyper@linkstart
273 \let\Etoc@next\Etoc@getnb@hyp
274 \fi\fi
275 \Etoc@next #1}
276 %
277 \def\Etoc@getit #1{\let\Etoc@next\Etoc@getit@nohyp
278 \ifEtoc@hyperref\ifx #1\hyper@linkstart
279 \let\Etoc@next\Etoc@getit@hyp
280 \fi\fi
281 \Etoc@next #1}
282 %
283 \def\Etoc@getthepage #1{\let\Etoc@next\Etoc@getthepage@nohyp
```

16. Implementation

```

284 \ifEtoc@hyperref\ifx #1\hyper@linkstart
285   \let\Etoc@next\Etoc@getthepage@hyp
286 \fi\fi
287 \Etoc@next #1}
288 \def\Etoc@getthepage@nohyp #1\etoc@{\def\etocthepage{#1}}
289 \def\Etoc@getthepage@hyp #1#2#3#4#5\etoc@{%
290   \@namedef{etoclink }##1{#1{#2}{#3}{##1}#5}%
291   \def\etocthepage{#4}}

placeholder for comments

292 %
293 \def\Etoc@getnb@nohyp #1#2#3\etoc@{%
294   \def\Etoc@getname ##1\relax\relax\etoc@{%
295     \@namedef{etocname }{\leavevmode ##1}%
296     \def\etocthename{##1}%
297   }%
298   \ifx #1\numberline
299     \@namedef{etocnumber }{\leavevmode #2}%
300     \def\etocthenumber{#2}%
301     \Etoc@numbertrue
302     \Etoc@getname #3\etoc@
303   \else
304     \@namedef{etocnumber }{\leavevmode}%
305     \def\etocthenumber{}%
306     \Etoc@numberfalse
307   \fi
308 }

placeholder for comments

309 \def\Etoc@getnb@hyp #1#2#3#4#5#6\etoc@{%
310   \@namedef{etoclink }##1{#1{#2}{#3}{##1}#5}%
311   \def\Etoc@getname ##1\relax\relax\etoc@{%
312     \@namedef{etocname }{\leavevmode #1{#2}{#3}{##1}#5}%
313     \def\etocthename{##1}%
314   }%
315   \def\Etoc@getnbr ##1##2##3\etoc@{%
316     \ifx ##1\numberline
317       \@namedef{etocnumber }{\leavevmode #1{#2}{#3}{##2}#5}%
318       \def\etocthenumber{##2}%
319       \Etoc@numbertrue
320       \Etoc@getname ##3\etoc@
321     \else
322       \@namedef{etocnumber }{\leavevmode}%
323       \def\etocthenumber{}%
324       \Etoc@numberfalse
325       \def\etocthename{#4}%
326     \fi}%
327   \Etoc@getnbr #4\relax\relax\etoc@
328 }

placeholder for comments

329 \def\Etoc@getit@nohyp #1\hspace#2#3\etoc@{%
330   \def\Etoc@getname ##1\hspace\relax\etoc@{%
331     \@namedef{etocname }{\leavevmode ##1}%

```

16. Implementation

```

332     \def\etocthenname{##1}%
333   }%
334   \ifx\relax#2\else
335     \@namedef{etocnumber }{\leavevmode #1}%
336     \def\etocthenumber{#1}%
337     \Etoc@numbertrue
338     \Etoc@getname #3\etoc@
339   \fi
340 }

placeholder for comments
341 \def\Etoc@getit@hyp #1#2#3#4#5#6\etoc@{%
342   \def\Etoc@getname ##1\hspace\relax\etoc@{%
343     \@namedef{etocname }{\leavevmode #1{#2}{#3}{##1}#5}%
344     \def\etocthename{##1}%
345   }%
346   \def\Etoc@getnbr ##1\hspace##2##3\etoc@{%
347     \ifx\relax##2\else
348       \@namedef{etocnumber }{\leavevmode #1{#2}{#3}{##1}#5}%
349       \def\etocthenumber{##1}%
350       \Etoc@numbertrue
351       \Etoc@getname ##3\etoc@
352     \fi}%
353   \Etoc@getnbr #4\hspace\relax\etoc@}

placeholder for comments
354 \DeclareRobustCommand*\etocname{} {}
355 \DeclareRobustCommand*\etocnumber{} {}
356 \DeclareRobustCommand*\etocpage{} {}
357 \DeclareRobustCommand*\etoclink{} {}
358 \DeclareRobustCommand*\etocifnumbered{}
359   {\ifEtoc@number\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi}

placeholder for comments
360 \def\Etoc@readtoc#1{%
361   \ifeof #1
362     \let\Etoc@nextread\@gobble
363     \global\Etoc@toctoks=\expandafter{\the\Etoc@toctoks}%
364   \else
365     \let\Etoc@nextread\Etoc@readtoc
366     \read #1 to \Etoc@buffer
367     \Etoc@toctoks=\expandafter\expandafter\expandafter
368       {\expandafter\the\expandafter\Etoc@toctoks\Etoc@buffer}%
369   \fi
370   \Etoc@nextread{#1}%
371 }
372 \IfFileExists{\jobname .toc}
373   {\endlinechar=-1
374     \makeatletter
375     \newread\Etoc@tf
376     \openin\Etoc@tf\@filef@und
377     \Etoc@readtoc\Etoc@tf
378     \closein\Etoc@tf}}
379   {\typeout{No file \jobname .toc.}}
```

16. Implementation

placeholder for comments

```
380 \def\Etoc@openouttoc{% formerly \Etoc@starttoc
381 %% 1.07d: parskip and \@nobreakfalse stuff moved to \Etoc@tableofcontents
382 \ifEtoc@hyperref
383   \ifx\hyper@last\@undefined
384     \IfFileExists{\jobname .toc}
385       {\Hy@WarningNoLine
386         {old toc file detected, not used; run LaTeX again (cheers from etoc)}}%
387       \global\Etoc@toctoks={}%
388     }
389   {}%
390 \fi
391 \fi
392 \if@filesw
393   \newwrite \tf@toc
394   \immediate \openout \tf@toc \jobname .toc\relax
395 \fi
396 \gdef\Etoc@openouttoc{}% 1.07d, rather than using a boolean
397 }
```

placeholder for comments

```
398 \def\Etoc@toctoc{%
399   \global\Etoc@vfalse
400   \global\Etoc@ivfalse
401   \global\Etoc@iiifalse
402   \global\Etoc@iifalse
403   \global\Etoc@ifalse
404   \global\Etoc@false
405   \global\Etoc@jfalse
406   \global\Etoc@jjfalse
407 \ifEtoc@standard
408   \etoc@setstyle{@minustwo}{}{}{}{}%
409   \etoc@setstyle{@minusone}{}{}{}{}%
410   \etoc@setstyle{@zero}{}{}{}{}%
411   \etoc@setstyle{@one}{}{}{}{}%
412   \etoc@setstyle{@two}{}{}{}{}%
413   \etoc@setstyle{@three}{}{}{}{}%
414   \etoc@setstyle{@four}{}{}{}{}%
415   \etoc@setstyle{@five}{}{}{}{}%
416 \else
417   \def\Etoc@levellist@elt##1{\let##1\Etoc@lxyz}%
418   \Etoc@levellist
419   \let\booknumberline\numberline
420   \let\partnumberline\numberline
421   \let\chapternumberline\numberline
422 \fi
423 \the\Etoc@toctoks
424 \ifEtoc@notactive\else
425   \ifEtoc@v \Etoc@end@five\fi
426   \ifEtoc@iv \Etoc@end@four\fi
427   \ifEtoc@iii \Etoc@end@three\fi
428   \ifEtoc@ii \Etoc@end@two\fi
429   \ifEtoc@i \Etoc@end@one\fi
```

16. Implementation

```

430 \ifEtoc@ \Etoc@end@zero\fi
431 \ifEtoc@j \Etoc@end@minusone\fi
432 \ifEtoc@jj \Etoc@end@minustwo\fi
433 \fi}

placeholder for comments
434 \def\etoc@@startlocaltoc#1#2{%
435 \let\Etoc@next\relax
436 \ifEtoc@localtoc
437 \ifEtoc@notactive
438 \ifnum #1=#2\relax
439 \ifEtoc@jj \let\Etoc@localtop\Etoc@@minusone@@ \fi
440 \ifEtoc@j \let\Etoc@localtop\Etoc@@zero@@ \fi
441 \ifEtoc@ \let\Etoc@localtop\Etoc@@one@@ \fi
442 \ifEtoc@i \let\Etoc@localtop\Etoc@@two@@ \fi
443 \ifEtoc@ii \let\Etoc@localtop\Etoc@@three@@ \fi
444 \ifEtoc@iii \let\Etoc@localtop\Etoc@@four@@ \fi
445 \ifEtoc@iv \let\Etoc@localtop\Etoc@@five@@ \fi
446 \ifEtoc@v \let\Etoc@localtop\Etoc@@six@@ \fi
447 \def\Etoc@next{\global\Etoc@notactivefalse
448 \global\Etoc@vfalse
449 \global\Etoc@ivfalse
450 \global\Etoc@iiifalse
451 \global\Etoc@iifalse
452 \global\Etoc@ifalse
453 \global\Etoc@false
454 \global\Etoc@jfalse
455 \global\Etoc@jjfalse}%
456 \fi
457 \fi\fi
458 \Etoc@next}
459 \let\etoc@startlocaltoc\@gobble

placeholder for comments
460 \def\Etoc@localtableofcontents#1{%
461 \edef\Etoc@tmp{#1}%
462 \ifnum\Etoc@tmp<\@ne
463 \PackageWarning{etoc}
464 {Unknown toc id: run LaTeX to get references right}%
465 \leavevmode --unknown etoc ref: run latex again--\par
466 \let\Etoc@next\@gobble\else\let\Etoc@next\@firstofone\fi
467 \Etoc@next
468 {\edef\etoc@startlocaltoc##1{%
469 \noexpand\etoc@@startlocaltoc{##1}{#1}}
470 \Etoc@localtoctrue
471 \let\Etoc@localtop\Etoc@@minustwo@@
472 \global\Etoc@stoptocfalse
473 \global\Etoc@notactivetrue
474 \Etoc@tableofcontents}%
475 \endgroup\ifEtoc@mustclosegroup\endgroup\fi}

placeholder for comments
476 \def\Etoc@getrefno #1#2\etoc@{#1}
477 \def\Etoc@getref #1{\@ifundefined{r@#1}{0}{\expandafter\expandafter\expandafter

```

16. Implementation

```

478      \Etoc@getrefno\csname r@#1\endcsname\relax\etoc@}}
479 \def\Etoc@ref#1{\Etoc@localtableofcontents{\Etoc@getref{#1}}}
480 \def\Etoc@label#1{\label{#1}%
481      \futurelet\Etoc@nexttoken\Etoc@t@bleofcontents}
482 {\def\1{\Etoc@again}\expandafter
483 \gdef\1 {\futurelet\Etoc@nexttoken\Etoc@t@bleofcontents}}

placeholder for comments
484 \def\Etoc@t@bleofcontents{%
485   \ifx\Etoc@nexttoken\label
486     \def\Etoc@next{\expandafter\Etoc@label\@gobble}\else
487     \ifx\Etoc@nexttoken\@sptoken
488       \let\Etoc@next\Etoc@again\else
489       \ifEtoc@tocwithid
490         \def\Etoc@next{\Etoc@localtableofcontents{\c@etoc@tocid}}%
491       \else
492         \ifx\Etoc@nexttoken\ref
493           \def\Etoc@next{\expandafter\Etoc@ref\@gobble}%
494         \else
495           \def\Etoc@next{\Etoc@localtocfalse
496             \global\Etoc@notactivefalse
497             \Etoc@tableofcontents
498             \endgroup
499             \ifEtoc@mustclosegroup\endgroup\fi}%
500         \fi
501       \fi\fi\fi\Etoc@next}

placeholder for comments
502 \def\table@fcontents{%
503   \refstepcounter{etoc@tocid}%
504   \Etoc@tocwithidfalse
505   \futurelet\Etoc@nexttoken\Etoc@t@bleofcontents}
506 \def\localtable@fcontents{%
507   \refstepcounter{etoc@tocid}%
508   \addtocontents{toc}
509     {\string\etoc@startlocaltoc{\arabic{etoc@tocid}}}%
510   \Etoc@tocwithidtrue
511   \futurelet\Etoc@nexttoken\Etoc@t@bleofcontents}

placeholder for comments
512 \newcommand*\etocaftertitlehook{}
513 \newcommand*\etocaftercontentshook{}
514 \renewcommand*\tableofcontents{%
515   \Etoc@openouttoc
516   \Etoc@par
517   \begingroup % closed in \Etoc@t@bleofcontents or \Etoc@localtableofcontents
518     \def\etoc@startlocaltoc##1{\etoc@@startlocaltoc{##1}{\c@etoc@tocid}}%
519     \@ifstar
520     {\def\Etoc@aftertitlehook{}\table@fcontents}
521     {\let\Etoc@aftertitlehook\etocaftertitlehook\table@fcontents}}
522 \newcommand*\localtableofcontents{%
523   \Etoc@openouttoc
524   \Etoc@par
525   \begingroup % closed in \Etoc@t@bleofcontents or \Etoc@localtableofcontents

```

16. Implementation

```

526 \ifstar
527 {\def\Etoc@aftertitlehook{}\localtable@fcontents}
528 {\let\Etoc@aftertitlehook\etocaftertitlehook\localtable@fcontents}}

placeholder for comments

529 \newcommand\etocsettocstyle[2]{%
530 \def\Etoc@tableofcontents
531 {\ifnum\c@tocdepth>\Etoc@minf
532 \let\Etoc@@next\@firstofone\else
533 \let\Etoc@@next\@gobble
534 \fi
535 \Etoc@@next{#1\ifEtoc@parskip\parskip\z@skip\fi %1.07d
536 \Etoc@aftertitlehook
537 \let\Etoc@savedcontentsline\contentsline
538 \let\contentsline\Etoc@etoccontentsline
539 \Etoc@toctoc
540 \let\Etoc@@next\relax
541 \ifEtoc@tocwithid\else
542 \ifEtoc@localtoc
543 \ifEtoc@notactive
544 \def\Etoc@@next{\Etoc@localtocfalse
545 \global\Etoc@notactivefalse
546 \Etoc@toctoc}%
547 \fi\fi\fi
548 \Etoc@@next
549 \etocaftercontentshook
550 #2\@nobreakfalse}} % 1.07d: \@nobreakfalse moved here

placeholder for comments

551 \newcommand*\etocsetstyle{\Etoc@standardfalse\etoc@setstyle}
552 \long\def\etoc@setstyle#1#2#3#4#5{%
553 \long\expandafter\def
554 \csname Etoc@begin@\csname Etoc@#1@\endcsname\endcsname {#2}%
555 \long\expandafter\def
556 \csname Etoc@prefix@\csname Etoc@#1@\endcsname\endcsname {#3}%
557 \long\expandafter\def
558 \csname Etoc@contents@\csname Etoc@#1@\endcsname\endcsname {#4}%
559 \long\expandafter\def
560 \csname Etoc@end@\csname Etoc@#1@\endcsname\endcsname {#5}}

placeholder for comments

561 \newcommand*\etocfontminustwo{\normalfont \LARGE \bfseries}
562 \newcommand*\etocfontminusone{\normalfont \large \bfseries}
563 \newcommand*\etocfontzero{\normalfont \large \bfseries}
564 \newcommand*\etocfontone{\normalfont \normalsize \bfseries}
565 \newcommand*\etocfonttwo{\normalfont \normalsize}
566 \newcommand*\etocfontthree{\normalfont \footnotesize}

placeholder for comments

567 \newcommand*\etocsepminustwo{4ex \@plus .5ex \@minus .5ex}
568 \newcommand*\etocsepminusone{4ex \@plus .5ex \@minus .5ex}
569 \newcommand*\etocsepzero{2.5ex \@plus .4ex \@minus .4ex}
570 \newcommand*\etocseppone{1.5ex \@plus .3ex \@minus .3ex}
571 %%\newcommand*\etocseptwo{1ex \@plus .15ex \@minus .15ex} % modified in 1.07e

```

16. Implementation

```

572 \newcommand*\etocsepttwo{.5ex \@plus .1ex \@minus .1ex}
573 \newcommand*\etocsepthree{.25ex \@plus .05ex \@minus .05ex}

placeholder for comments
574 \newcommand*\etocbaselinespreadminustwo{1}
575 \newcommand*\etocbaselinespreadminusone{1}
576 \newcommand*\etocbaselinespreadzero{1}
577 \newcommand*\etocbaselinespreadone{1}
578 \newcommand*\etocbaselinespreadtwo{1}
579 \newcommand*\etocbaselinespreadthree{.9}

placeholder for comments
580 \newcommand*\etocminustwoleftmargin{1.5em plus 0.5fil}
581 \newcommand*\etocminustworightmargin{1.5em plus -0.5fil}
582 \newcommand*\etocminusedleftmargin{1em}
583 \newcommand*\etocminusedrightmargin{1em}
584 \newcommand*\etoclineleaders
585     {\hbox{\normalfont\normalsize\hb@xt@2ex {\hss.\hss}}}
586 \newcommand*\etocabbrevpagename{p.~}
587 \newcommand*\etocpartname{\partname}
588 \newcommand*\etocbookname{Book}

placeholder for comments The macro \etocdefaultlines was initially called \etoclines. Now \etoclines just does \Etoc@standardfalse.
589 \def\etocdefaultlines{\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 de-
velopment of the package.

590 %% 'book' in memoir class:
591 \etoc@setstyle{@minustwo}
592     {\addpenalty\@M\etocskipfirstprefix}
593     {\addpenalty\@secpenalty}
594     {\beginingroup
595         \etocfontminustwo
596         \addvspace{\etocsepminustwo}%
597         \parindent \z@
598         \leftskip \etocminustwoleftmargin
599         \rightskip \etocminustworightmargin
600         \parfillskip \@flushglue
601         \vbox{\etocifnumbered{\etocbookname\enspace\etocnumber:\quad}}{\etocname
602             \baselineskip\etocbaselinespreadminustwo\baselineskip
603             \par}%
604         \addpenalty\@M\addvspace{\etocsepminusone}%
605         \endgroup}
606     {}%
607 %% 'part':
608 \etoc@setstyle{@minusone}
609     {\addpenalty\@M\etocskipfirstprefix}
610     {\addpenalty\@secpenalty}
611     {\beginingroup
612         \etocfontminusone
613         \addvspace{\etocsepminusone}%
614         \parindent \z@

```


16. Implementation

```

615 \leftskip \etocminusoneleftmargin
616 \rightskip \etocminusonerightmargin
617 \parfillskip \@flushglue
618 \vbox{\etocifnumbered{\etocpartname\enspace\etocnumber.\quad}}{\etocname
619 \baselineskip\etocbaselinespreadminusone\baselineskip
620 \par}%
621 \addpenalty\@M\addvspace{\etocsepzero}%
622 \endgroup}
623 {}%
624 %% 'chapter':
625 \etoc@setstyle{@zero}
626 {\addpenalty\@M\etocskipfirstprefix}
627 {\addpenalty\@itempenalty}
628 {\begingroup
629 \etocfontzero
630 \addvspace{\etocsepzero}%
631 \parindent \z@ \parfillskip \@flushglue
632 \vbox{\etocifnumbered{\etocnumber.\enspace}}{\etocname
633 \baselineskip\etocbaselinespreadzero\baselineskip
634 \par}%
635 \endgroup}
636 {\addpenalty{-\@highpenalty}\addvspace{\etocsepminusone}}}%
637 %% 'section':
638 \etoc@setstyle{@one}
639 {\addpenalty\@M\etocskipfirstprefix}
640 {\addpenalty\@itempenalty}
641 {\begingroup
642 \etocfontone
643 \addvspace{\etocsepone}%
644 \parindent \z@ \parfillskip \z@
645 \setbox\z@\vbox{\parfillskip\@flushglue
646 \etocname\par
647 \setbox\tw@\lastbox
648 \global\setbox\@ne\hbox{\unhbox\tw@ }}%
649 \dimen\z@=\wd\@ne
650 \setbox\z@=\etoclineleaders
651 \advance\dimen\z@\wd\z@
652 \etocifnumbered
653 {\setbox\tw@\hbox{\etocnumber, \etocabbrevpagename\etocpage}}
654 {\setbox\tw@\hbox{\etocabbrevpagename\etocpage}}}%
655 \advance\dimen\z@\wd\tw@
656 \ifdim\dimen\z@ < \linewidth
657 \vbox{\etocname~%
658 \leaders\box\z@\hfil\box\tw@
659 \baselineskip\etocbaselinespreadone\baselineskip
660 \par}
661 \else
662 \vbox{\etocname~%
663 \leaders\copy\z@\hfil\break
664 \hbox{} \leaders\box\z@\hfil\box\tw@
665 \baselineskip\etocbaselinespreadone\baselineskip
666 \par}

```

16. Implementation

```

667 \fi
668 \endgroup}
669 {\addpenalty\@secpenalty\addvspace{\etocsepzzero}}%
670 %% 'subsection':
671 \etoc@setstyle{@two}
672 {\addpenalty\@medpenalty\etocskipfirstprefix}
673 {\addpenalty\@itempenalty}
674 {\begingroup
675 \etocfonttwo
676 \addvspace{\etocsepttwo}%
677 \parindent \z@ \parfillskip \z@
678 \setbox\z@\vbox{\parfillskip\@flushglue
679 \etocname\par\setbox\tw@\lastbox
680 \global\setbox\@ne\hbox{\unhbox\tw@}}}%
681 \dimen\z@=\wd\@ne
682 \setbox\z@=\etoclineleaders
683 \advance\dimen\z@\wd\z@
684 \etocifnumbered
685 {\setbox\tw@\hbox{\etocnumber, \etocabbrevpagename\etocpage}}
686 {\setbox\tw@\hbox{\etocabbrevpagename\etocpage}}%
687 \advance\dimen\z@\wd\tw@
688 \ifdim\dimen\z@ < \linewidth
689 \vbox{\etocname~%
690 \leaders\box\z@\hfil\box\tw@
691 \baselineskip\etocbaselinespreadtwo\baselineskip
692 \par}
693 \else
694 \vbox{\etocname~%
695 \leaders\copy\z@\hfil\break
696 \hbox{} \leaders\box\z@\hfil\box\tw@
697 \baselineskip\etocbaselinespreadtwo\baselineskip
698 \par}
699 \fi
700 \endgroup}
701 {\addpenalty\@secpenalty\addvspace{\etocseppone}}%
702 %% 'subsubsection':
703 \etoc@setstyle{@three}
704 {\addpenalty\@M
705 \etocfontthree
706 \vspace{\etocsepthree}%
707 \noindent
708 \etocskipfirstprefix}
709 {\allowbreak\,--\,}
710 {\etocname}
711 {\.\hfil
712 \begingroup
713 \baselineskip\etocbaselinespreadthree\baselineskip
714 \par
715 \endgroup
716 \addpenalty{-\@highpenalty}}%
placeholder for comments
717 \etoc@setstyle{@four}{\{\}\{\}\{\}\{\}\}%

```

16. Implementation

```

718 \etoc@setstyle{@five}{}{}{}{}%
719 }

placeholder for comments

720 \newcommand*\etocabovetocskip{3.5ex \@plus 1ex \@minus .2ex}
721 \newcommand*\etocbelowtocskip{3.5ex \@plus 1ex \@minus .2ex}
722 \newcommand*\etoccolumnsep{2em}
723 \newcommand*\etocmulticolsep{0ex}
724 \newcommand*\etocmulticolpretolerance{-1}
725 \newcommand*\etocmulticoltolerance{200}
726 \newcommand*\etocdefaultnbcol{2}
727 \newcommand*\etocinnertopsep{2ex}

placeholder for comments

728 \newcommand\etocmulticolstyle[2][\etocdefaultnbcol]{%
729 \etocsettocstyle
730   {\let\etocoldpar\par
731    \addvspace{\etocabovetocskip}%
732    \ifnum #1>\@ne\let\Etoc@next\@firstoftwo
733     \else \let\Etoc@next\@secondoftwo\fi
734    \Etoc@next{%
735     \multicolpretolerance\etocmulticolpretolerance
736     \multicoltolerance\etocmulticoltolerance
737     \setlength{\columnsep}{\etoccolumnsep}%
738     \setlength{\multicolsep}{\etocmulticolsep}%
739     \begin{multicols}{#1}[#2\etocoldpar\addvspace{\etocinnertopsep}]}
740 % 2013/01/29: erroneous \etocsepminusone at last replaced by \etocinnertopsep
741 % and definition of \etocoldpar added as multicols chokes on \par as part of #2
742     {#2\par\addvspace{\etocinnertopsep}%
743      \pretolerance\etocmulticolpretolerance
744      \tolerance\etocmulticoltolerance}}
745   {\ifnum #1>\@ne\let\Etoc@next\@firstofone
746    \else \let\Etoc@next\@gobble\fi
747    \Etoc@next{\end{multicols}}}%
748   \addvspace{\etocbelowtocskip}}

placeholder for comments

749 \newcommand*\etocinnerbottomsep{3.5ex}
750 \newcommand*\etocinnerleftsep{2em}
751 \newcommand*\etocinnerrightsep{2em}
752 \newcommand*\etoctoprule{\hrule}
753 \newcommand*\etocleftrule{\vrule}
754 \newcommand*\etocrightrule{\vrule}
755 \newcommand*\etocbottomrule{\hrule}
756 \newcommand*\etoctoprulecolorcmd{\relax}
757 \newcommand*\etocbottomrulecolorcmd{\relax}
758 \newcommand*\etocleftrulecolorcmd{\relax}
759 \newcommand*\etocrightrulecolorcmd{\relax}

placeholder for comments

760 \def\etoc@ruledheading #1{%
761   \hb@xt@\linewidth{\color@begingroup
762     \hss #1\hss\hskip-\linewidth
763     \etoctoprulecolorcmd\leaders\etoctoprule\hss

```

16. Implementation

```

764      \phantom{#1}%
765      \leaders\etoptoprule\hss\color@endgroup}%
766      \nointerlineskip\vskip\etocinnertopsep}

placeholder for comments
767 \newcommand*\etocruledstyle[2][\etocdefaultnbcol]{%
768 \etocsettocstyle
769   {\addvspace{\etocabovetocskip}}%
770   \ifnum #1>\@ne\let\Etoc@next\@firstoftwo
771   \else \let\Etoc@next\@secondoftwo\fi
772   \Etoc@next
773   {\multicolpretolerance\etocmulticolpretolerance
774    \multicoltolerance\etocmulticoltolerance
775    \setlength{\columnsep}{\etoccolumnsep}%
776    \setlength{\multicolsep}{\etocmulticolsep}%
777    \begin{multicols}{#1}[\etoc@ruledheading{#2}]}
778   {\etoc@ruledheading{#2}\nobreak
779    \pretolerance\etocmulticolpretolerance
780    \tolerance\etocmulticoltolerance}}
781   {\ifnum #1>\@ne\let\Etoc@next\@firstofone
782    \else \let\Etoc@next\@gobble\fi
783    \Etoc@next\end{multicols}}}%
784   \addvspace{\etocbelowtocskip}}}}

placeholder for comments
785 \newcommand\etocframedmphook{\relax}
786 \newcommand*\etocbkgcolorcmd{\relax}
787 \def\Etoc@relax{\relax}
788 \newbox\etoc@framed@titlebox
789 \newbox\etoc@framed@contentsbox
790 \newcommand*\etocframedstyle[2][\etocdefaultnbcol]{%
791 \etocsettocstyle{%
792   \addvspace{\etocabovetocskip}%
793   \sbox\z@{#2}%
794   \dimen\z@=\dp\z@
795   \ifdim\wd\z@<\linewidth \dp\z@=\z@ \else \dimen\z@=\z@ \fi
796   \setbox\etoc@framed@titlebox=\hb@xt@\linewidth{\color@begingroup
797     \hss
798     \ifx\etocbkgcolorcmd\Etoc@relax\else
799       \sbox\tw@{\color{white}}%
800       \vrule\@width\wd\z@\@height\ht\z@\@depth\dimen\z@}%
801       \ifdim\wd\z@<\linewidth \dp\tw@=\z@\fi
802       \box\tw@
803       \hskip-\wd\z@
804     \fi
805     \copy\z@
806     \hss
807     \hskip-\linewidth
808     \etoptoprulecolorcmd\leaders\etoptoprule\hss%
809     \hskip\wd\z@
810     \etoptoprulecolorcmd\leaders\etoptoprule\hss\color@endgroup}%
811   \setbox\z@=\hbox{\etocleftrule\etocrightrule}}%
812   \dimen\tw@=\linewidth\advance\dimen\tw@-\wd\z@

```

16. Implementation

```

813      \advance\dimen\tw@-\etocinnerleftsep
814      \advance\dimen\tw@-\etocinnerrightsep
815      \setbox\etoc@framed@contentsbox=\vbox\bgroup
816      \hsize\dimen\tw@
817      \kern\dimen\z@
818      \vskip\etocinnertopsep
819      \hbox\bgroup
820      \begin{minipage}{\hsize}%
821      \etocframedmphook
822      \ifnum #1>\@ne\let\Etoc@next\@firstoftwo
823      \else \let\Etoc@next\@secondoftwo\fi
824      \Etoc@next
825      {\multicolpretolerance\etocmulticolpretolerance
826      \multicoltolerance\etocmulticoltolerance
827      \setlength{\columnsep}{\etoccolumnsep}%
828      \setlength{\multicolsep}{\etocmulticolsep}%
829      \begin{multicols}{#1}}
830      {\pretolerance\etocmulticolpretolerance
831      \tolerance\etocmulticoltolerance}}
832      {\ifnum #1>\@ne\let\Etoc@next\@firstofone
833      \else \let\Etoc@next\@gobble\fi
834      \Etoc@next\end{multicols}\unskip}%
835      \end{minipage}%
836      \egroup
837      \vskip\etocinnerbottomsep
838      \egroup
839      \vbox{\hsize\linewidth
840      \ifx\etocbkgcolorcmd\Etoc@relax\else
841      \kern\ht\etoc@framed@titlebox
842      \kern\dp\etoc@framed@titlebox
843      \hb@xt@\linewidth{\color@begingroup
844      \etocleftrulecolorcmd\etocleftrule
845      \etocbkgcolorcmd
846      \leaders\vrule
847      \@height\ht\etoc@framed@contentsbox
848      \@depth\dp\etoc@framed@contentsbox
849      \hss
850      \etocrightrulecolorcmd\etocrightrule
851      \color@endgroup}\nointerlineskip
852      \vskip-\dp\etoc@framed@contentsbox
853      \vskip-\ht\etoc@framed@contentsbox
854      \vskip-\dp\etoc@framed@titlebox
855      \vskip-\ht\etoc@framed@titlebox
856      \fi
857      \box\etoc@framed@titlebox\nointerlineskip
858      \hb@xt@\linewidth{\color@begingroup
859      {\etocleftrulecolorcmd\etocleftrule}%
860      \hss\box\etoc@framed@contentsbox\hss
861      \etocrightrulecolorcmd\etocrightrule\color@endgroup}
862      \nointerlineskip
863      \vskip\ht\etoc@framed@contentsbox
864      \vskip\dp\etoc@framed@contentsbox

```

16. Implementation

```

865 \hb@xt@\linewidth{\color@begingroup\etocbottomrulecolorcmd
866 \leaders\etocbottomrule\hss\color@endgroup}}
867 \addvspace{\etocbelowtocskip}}
placeholder for comments
868 \newcommand\etoc@multicoltoc[2][\etocdefaultnbcol]{%
869 \etocmulticolstyle[#1]{#2}%
870 \tableofcontents}
871 \newcommand\etoc@multicoltoi[2][\etocdefaultnbcol]{%
872 \etocmulticolstyle[#1]{#2}%
873 \tableofcontents*}
874 \newcommand\etoc@local@multicoltoc[2][\etocdefaultnbcol]{%
875 \etocmulticolstyle[#1]{#2}%
876 \localtableofcontents}
877 \newcommand\etoc@local@multicoltoi[2][\etocdefaultnbcol]{%
878 \etocmulticolstyle[#1]{#2}%
879 \localtableofcontents*}
placeholder for comments
880 \newcommand*\etoc@ruledtoc[2][\etocdefaultnbcol]{%
881 \etocruledstyle[#1]{#2}%
882 \tableofcontents}
883 \newcommand*\etoc@ruledtoi[2][\etocdefaultnbcol]{%
884 \etocruledstyle[#1]{#2}%
885 \tableofcontents*}
886 \newcommand*\etoc@local@ruledtoc[2][\etocdefaultnbcol]{%
887 \etocruledstyle[#1]{#2}%
888 \localtableofcontents}
889 \newcommand*\etoc@local@ruledtoi[2][\etocdefaultnbcol]{%
890 \etocruledstyle[#1]{#2}%
891 \localtableofcontents*}
placeholder for comments
892 \newcommand*\etoc@framedtoc[2][\etocdefaultnbcol]{%
893 \etocframedstyle[#1]{#2}%
894 \tableofcontents}
895 \newcommand*\etoc@framedtoi[2][\etocdefaultnbcol]{%
896 \etocframedstyle[#1]{#2}%
897 \tableofcontents*}
898 \newcommand*\etoc@local@framedtoc[2][\etocdefaultnbcol]{%
899 \etocframedstyle[#1]{#2}%
900 \localtableofcontents}
901 \newcommand*\etoc@local@framedtoi[2][\etocdefaultnbcol]{%
902 \etocframedstyle[#1]{#2}%
903 \localtableofcontents*}
placeholder for comments
904 \def\etocmulticol{\begingroup
905 \Etoc@mustclosegrouptrue
906 \@ifstar
907 {\etoc@multicoltoi}
908 {\etoc@multicoltoc}}
909 \def\etocruled{\begingroup
910 \Etoc@mustclosegrouptrue

```

16. Implementation

```

911 \@ifstar
912 {\etoc@ruledtoci}
913 {\etoc@ruledtoc}}
914 \def\etocframed{\begingroup
915 \Etoc@mustclosegrouptrue
916 \@ifstar
917 {\etoc@framedtoci}
918 {\etoc@framedtoc}}
919 \def\etoclocalmulticol{\begingroup
920 \Etoc@mustclosegrouptrue
921 \@ifstar
922 {\etoc@local@multicoltoci}
923 {\etoc@local@multicoltoc}}
924 \def\etoclocalruled{\begingroup
925 \Etoc@mustclosegrouptrue
926 \@ifstar
927 {\etoc@local@ruledtoci}
928 {\etoc@local@ruledtoc}}
929 \def\etoclocalframed{\begingroup
930 \Etoc@mustclosegrouptrue
931 \@ifstar
932 {\etoc@local@framedtoci}
933 {\etoc@local@framedtoc}}

placeholder for comments
934 \def\etocarticlestyle{%
935 \etocsettocstyle
936 {\section *{\contentsname
937 \mkboth {\MakeUppercase \contentsname}
938 \MakeUppercase \contentsname}}}
939 {}

940 \def\etocarticlestylenomarks{%
941 \etocsettocstyle
942 {\section *{\contentsname}}
943 {}

placeholder for comments
944 \def\etocbookstyle{%
945 \etocsettocstyle
946 {\if@twocolumn \@restonecoltrue \onecolumn \else \@restonecolfalse \fi
947 \chapter *{\contentsname
948 \mkboth {\MakeUppercase \contentsname}
949 \MakeUppercase \contentsname}}}
950 {\if@restonecol \twocolumn \fi}}
951 \def\etocbookstylenomarks{%
952 \etocsettocstyle
953 {\if@twocolumn \@restonecoltrue \onecolumn \else \@restonecolfalse \fi
954 \chapter *{\contentsname}}
955 {\if@restonecol \twocolumn \fi}}
956 \let\etocreportstyle\etocbookstyle
957 \let\etocreportstylenomarks\etocbookstylenomarks
958 \def\etocmemoirtotocfmt #1#2{%
959 \def\Etoc@addsuitablecontentsline{\addcontentsline {toc}{#1}{#2}}%

```

16. Implementation

```

960 \renewcommand*\etocaftertitlehook{%
961 \ifmem@em@starred@listof
962 \else\phantomsection\aftergroup\Etoc@addsuitablecontentsline\fi}}
963 \def\etocmemoirstyle{%
964 \etocsettocstyle
965 {\ensureonecol \par \begingroup \@nameuse {@tocmaketitle}
966 \Etoc@aftertitlehook\let\Etoc@aftertitlehook\relax
967 \parskip \cftparskip \@nameuse {cfttocbeforelisthook}}
968 {\@nameuse {cfttocafterlisthook}\endgroup\restorefromonecol}}

placeholder for comments
969 \def\etocscrartclstyle{%
970 \etocsettocstyle
971 {\let\if@dynlist\if@tocleft
972 \iftocfeature {toc}{onecolumn}
973 {\iftocfeature {toc}{leveldown}
974 {}
975 {\if@twocolumn \aftergroup \twocolumn \onecolumn \fi }}
976 {}}%
977 \tocbasic@listhead {\listoftocname}%
978 \begingroup \expandafter \expandafter \expandafter
979 \endgroup \expandafter
980 \ifx
981 \csname microtypesetup\endcsname \relax
982 \else
983 \iftocfeature {toc}{noptrusion}{%
984 {\microtypesetup {protrusion=false}%
985 \PackageInfo {tocbasic}%
986 {character protrusion at toc deactivated}}}%
987 \fi
988 \setlength {\parskip }{\z@ }%
989 \setlength {\parindent }{\z@ }%
990 \setlength {\parfillskip }{\z@ \@plus 1fil}%
991 \csname tocbasic@@before@hook\endcsname
992 \csname tb@toc@before@hook\endcsname}
993 {\csname tb@toc@after@hook\endcsname
994 \csname tocbasic@@after@hook\endcsname}}
995 \let\etocscrbookstyle\etocscrartclstyle
996 \let\etocscrreprtstyle\etocscrartclstyle

placeholder for comments
997 \newcommand*\etocstandarddisplaystyle{\etocarticlestyle}
998 \newcommand*\etocmarkboth[1]{%
999 \@mkboth{\MakeUppercase{#1}}{\MakeUppercase{#1}}}
1000 \newcommand*\etocmarkbothnouc[1]{\@mkboth{#1}{#1}}
1001 \newcommand\etoctocstyle[3][section]{\etocmulticolstyle[#2]%
1002 {\csname #1\endcsname *{#3}}}
1003 \newcommand\etoctocstylewithmarks[4][section]{\etocmulticolstyle[#2]%
1004 {\csname #1\endcsname *{#3\etocmarkboth{#4}}}}
1005 \newcommand\etoctocstylewithmarksnouc[4][section]{\etocmulticolstyle[#2]%
1006 {\csname #1\endcsname *{#3\etocmarkbothnouc{#4}}}}

placeholder for comments
1007 \def\Etoc@redefetocstyle#1{%

```


16. Implementation

```

1008 \renewcommand\etocstylewithmarks[4][#1]
1009 {\etocmulticolstyle[#2]%
1010   {\csname ##1\endcsname *{##3\etocmarkboth{##4}}}}
1011 \renewcommand\etocstylewithmarksnouc[4][#1]
1012 {\etocmulticolstyle[#2]%
1013   {\csname ##1\endcsname *{##3\etocmarkbothenouc{##4}}}}
1014 \renewcommand\etocstyle[3][#1]{%
1015   \etocmulticolstyle[#2]{\csname ##1\endcsname *{##3}}}}
1016 \ifclassloaded{scrartcl}
1017   {\renewcommand*\etocstandarddisplaystyle{\etocscrartclstyle}}{}
1018 \ifclassloaded{book}
1019   {\renewcommand*\etocfontone{\normalfont\normalsize}
1020     \renewcommand*\etocstandarddisplaystyle{\etocbookstyle}
1021     \Etoc@redefetocstyle{chapter}}{}
1022 \ifclassloaded{report}
1023   {\renewcommand*\etocfontone{\normalfont\normalsize}
1024     \renewcommand*\etocstandarddisplaystyle{\etocreportstyle}
1025     \Etoc@redefetocstyle{chapter}}{}
1026 \ifclassloaded{scrbook}
1027   {\renewcommand*\etocfontone{\normalfont\normalsize}
1028     \renewcommand*\etocstandarddisplaystyle{\etocscrbookstyle}
1029     \Etoc@redefetocstyle{chapter}}{}
1030 \ifclassloaded{scrreprt}
1031   {\renewcommand*\etocfontone{\normalfont\normalsize}
1032     \renewcommand*\etocstandarddisplaystyle{\etocscrreprtstyle}
1033     \Etoc@redefetocstyle{chapter}}{}
1034 \ifclassloaded{memoir}
1035   {\renewcommand*\etocfontone{\normalfont\normalsize}
1036     \etocmemoirtotocfmt{chapter}{\contentsname}%
1037     \renewcommand*\etocstandarddisplaystyle{\etocmemoirstyle}
1038     \Etoc@redefetocstyle{chapter}}{}

placeholder for comments
1039 \def\Etoc@addtocontents #1#2%
1040   {\ifEtoc@hyperref
1041     \addtocontents {toc}{\protect\contentsline
1042       {#1}{#2}%
1043       {\thepage }{\@currentHref }}%
1044   \else
1045     \addtocontents {toc}{\protect\contentsline
1046       {#1}{#2}{\thepage }}%
1047   \fi}
1048 \def\Etoc@addcontentsline@ #1#2#3%
1049   {\@namedef{toclevel@#1}{#3}%
1050     \addcontentsline {toc}{#1}{#2}}
1051 \DeclareRobustCommand*\etoccontentsline{
1052   {\@ifstar{\Etoc@addcontentsline@}{\Etoc@addtocontents}}
1053 \newcommand*\etocstandardlines{\Etoc@standardtrue}
1054 \newcommand*\etoclines{\Etoc@standardfalse} % 1.07b
1055 \etocdefaultlines % for initialization
1056 \etocstandardlines % removed silly AtBeginDocument temporarily added for 1.07
1057 \etocstandarddisplaystyle
1058 \endinput

```