

The **etoc** package

JEAN-FRANÇOIS BURNOL

jfbu (at) free (dot) fr

Package version: v1.07d (2013/02/24)

Documentation generated from the source file
with timestamp “24-02-2013 at 12:23:39 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 L^AT_EX layout for the Table of Contents is buried in the class definitions. In particular, most of the lengths therein are hardcoded, and the only way for the user to change them is to recopy the class 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

v1.07d [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`,

`\etoccontentsline`, `\etoccontentsline*`

`\etocnopar`, `\etocaftercontentshook`

modified command: `\etocmulticolstyle`

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

v1.06 [2012/12/07]

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

v1.05 [2012/12/01]

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

compatibility with KOMA-script and memoir document classes.

v1.04 [2012/11/24]

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

`\tableofcontents \label{toc:1}`

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

`\tableofcontents \ref{toc:1}`

v1.02 [2012/11/18]

initial version.

TABLE OF CONTENTS

Part I. Overview

Initial motivation: nested lists	1, p. 4
Limitations intrinsic to the use of environments	1.1, p. 4
Line styles and toc display style	2, p. 4
<code>\etocsettocstyle</code> for the toc display	2.1, p. 5
<code>\etocsetstyle</code> for the line styles	2.2, p. 5
Compatibility mode	2.3, p. 5
Arbitrarily many TOCs, and local ones too	3, p. 6
Labeling and reusing elsewhere	3.1, p. 6
The hyperref option <code>bookmarksdepth</code>	3.2, p. 6

On manually adding layout commands to the <code>.toc</code> file	3.3, p. 7
Shuffling the levels with <code>\etocsetlevel</code>	3.4, p. 7
Local table of contents for this part	3.5, p. 8
A simple example	4, p. 9
Surprising uses of <code>etoc</code>	5, p. 11
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

Part II. Package commands for line styles

The <code>\etocsetstyle</code> command	6, p. 18
---	-----------------

CONTENTS

The <code>\etocname</code> and <code>\etocpage</code> commands	6.1, p. 18
The <code>\etocskipfirstprefix</code> command	6.2, p. 19
The <code>\etocnumber</code> command	6.3, p. 19
The <code>\etocifnumbered</code> switch.	
The <code>\etocthenname</code> , <code>\etocthenumber</code> , and <code>\etocthepage</code> commands	6.4, p. 20
The <code>\etoclink</code> command	6.5, p. 20
The <code>\etocsetlevel</code> command	6.6, p. 20
<i>This is a (pale) red subsubsection for illustrative purposes.</i>	
Am I also red?	7, p. 21
A comparison	8, p. 21

Part III. Package commands for toc display styles

Specifying the toc display style	9, p. 25
The command <code>\etocruledstyle</code>	9.1, p. 26
The command <code>\etocmulticolstyle</code>	9.2, p. 26
The command <code>\etocstyle</code> .	
The command <code>\etocruled</code>	9.3, p. 27
The commands <code>\etocframedstyle</code> and <code>\etocframed</code>	9.4, p. 27
Headings, titles, <code>\etocinnertopsep</code>	9.5, p. 27
The command <code>\etocsettocstyle</code>	9.6, p. 28
The compatibility mode <code>\etocstandard-displaystyle</code>	9.7, p. 28
Starred variants of the <code>\tableofcontents</code> etc... commands	10, p. 28
Table of contents for this part	11, p. 29
Testing the compatibility mode	11.1, p. 29

A framed display	11.2, p. 30
A (crazy) inline display	11.3, p. 31

Part IV. Using and customizing `etoc`

Summary of the main style commands	12, p. 32
Setting up local styles	12.1, p. 32
Setting up toc display styles	12.2, p. 33
Displaying tables of contents	12.3, p. 33
Labels and references	12.4, p. 33
An example	12.5, p. 33
Customizing <code>etoc</code>	13, p. 36
Customizing the <code>etoc</code> pre-defined line styles	13.1, p. 36
Customizing the toc display styles	13.2, p. 37

Part V. Tips

... and tricks	14, p. 38
Hacking framed parboxes	14.1, p. 38
Interverting the levels	14.2, p. 38
All subsections of this document	14.3, p. 39
Compatibility with other packages	14.4, p. 40
T _E Xnical matters	14.5, p. 40
Errors and catastrophes	14.6, p. 40

Part VI. The code

Implementation	15, p. 41
---------------------------------	------------------

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 by the

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

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`, `\etocdisplaystyle`, `\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 `\etocdisplaystyle`, 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	
12. Summary of the main style commands	32
12.1. Setting up local styles	32
12.2. Setting up toc display styles	33
12.3. Displaying tables of contents	33
12.4. Labels and references	33
12.5. An example	33
13. Customizing etoc	36
13.1. Customizing the etoc pre-defined line styles	36
13.2. Customizing the toc display styles	37

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 `\begin{group} \renewcommand*\changetocdepth[1]{} \localtableofcontents \end{group}`, and to set the desired level for the local table of contents with the other `memoir` command `\maxtocdepth`.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

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. Arbitrarily many TOCs, and local ones too

things”, using no other auxiliary file than the .toc file. This is explained further in the section 5.

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	\etocsettocstyle for the toc display	5
2.2	\etocsetstyle 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 bookmarksdepth	6
3.3	On manually adding layout commands to the .toc file	7
3.4	Shuffling the levels with \etocsetlevel	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

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

4. A simple example

as part of the `<start>` code of the section ‘line style’. The frame borders have been set to 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 <code>hyperref</code> option <code>bookmarksdepth</code></i>	6
3.3	<i>On manually adding layout commands to the <code>.toc</code> 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

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
tables of contents which are displayed within
this document: \tableofcontents.
```

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

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

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

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

but its usefulness is to circumvent the automatic creation of bookmarks by the `hyperref` package, as they don't make much sense here (and 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 but 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 had to use the command `\etocnoper` to turn off this behavior.

Here are the implementation details:

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

5. Surprising uses of **etoc**

```
. . . in the preamble:
\newcounter{visibletoc}
\renewcommand{\etocaftertitlehook}
  {\stepcounter{visibletoc}\etoccontentsline{visibletoc}{\thevisibletoc}}
\etocsetlevel{visibletoc}{6}
. . .
\begin{document}
. . . document body
\subsection{Surprising uses of etoc}
\begingroup
  \etocnopar
  \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
```

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 14.2 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}}}
```

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**

5. Surprising uses of **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:

```
\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	23
5.	The Causal	42—49	25
6.	The Instrumental Case	27
	General uses	50—54	
	Particular uses	55—59	
7.	The Dative Case	33
	General uses	60—65	
	Particular uses	66—71	
8.	The Ablative Case	39
	General uses	72—75	
	Particular uses	76—86	
9.	The Locative Case	45
	General uses	87—92	
	Particular uses	93—100	

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.⁹ With these commands **etoc** will

⁹and if moreover one just wants to keep the same layout as in the default, one may question why

construct a TOC completely identical to what would have been done by one of the document class: `article`, `book`, `scrartcl`, `scrbook`.¹⁰ 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

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

```
\etocsetstyle{section}{}
  {\ifnum\etocthenumber=3
    \etocsetstyle{subsection}
      {\par\nopagebreak\begingroup
        \leftskip1.5em \rightskip\@tocmarg \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.

¹⁰For 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.

Part II.

Package commands for line styles

Contents of this second part

6 The `\etocsetstyle` command (page 18)

6.1 The `\etocname` and `\etocpage` commands (p. 18)

6.2 The `\etocskipfirstprefix` command (p. 19)

6.3 The `\etocnumber` command (p. 19)

The `\etocifnumbered` switch.

6.4 The `\etocthename`, `\etocthenumber`, and `\etocthepage` commands (p. 20)

6.5 The `\etoclink` command (p. 20)

6.6 The `\etocsetlevel` command (p. 20)

This is a (pale) red subsubsection for illustrative purposes.

7 Am I also red? (page 21)

8 A comparison (page 21)

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:¹¹

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

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

\etocsetstyle{subsubsection}
{\par\nobreak\begin{group}\normalfont
\footnotesize\itshape\etocskipfirstprefix}
{\allowbreak\,--\,}
{\etocname}
{.\hfil\par\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.¹²

¹¹the 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.

¹²up 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.

6.2. The `\etocskipfirstprefix` command

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:

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

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

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`.

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).¹³

```

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

```

¹³as seen we use `\fcolorbox` rather than `\fbox`. Due to some redefinition made by package `xcolor`, had we used `\fbox` (and not used `hyperref`) we would have needed `\protect\fbox`.

6. The `\etocsetstyle` command

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 `\etocthename`, `\etocthenumber`, and `\etocthepage` commands

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

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.

6.6. 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}
```

7. Am I also red?

```
\etocsetlevel{section}{1}  
\etocsetlevel{subsection}{2}  
\etocsetlevel{subsubsection}{3}  
\etocsetlevel{paragraph}{4}  
\etocsetlevel{subparagraph}{5}
```

Its own line styles are activated by `\etocdefaultlines`. They are illustrated by the main table of contents of this document. The level assignments can be modified at anytime, as has been explained in the section 5.

6.6.1. This is a (pale) red subsubsection for illustrative purposes

Just before this subsubsection we did:

```
\color{red!50}  
\addtocontents{toc}{\string\color{red!50}}
```

The title here would be red in the `article` class, but is black in the `scrartcl` class. In both classes the corresponding entry in the main table of contents at the beginning of this document is red; it is not red in the table of contents at the beginning of this second document part. The reason is that the `<finish>` code for the subsubsection level closed a group; the definition was:

```
\etocsetstyle{subsubsection}  
{\par\nobreak\begingroup\normalfont  
  \footnotesize\itshape\etocskipfirstprefix}  
\allowbreak\,--\,}  
\etocname}  
\hfil\par\endgroup\pagebreak[3]} <--- here is the \endgroup
```

This illustrates the discussion from subsection 1.1. The default package line styles do not contain group opening and closing instructions, so the influence (within TOCs such as the main one in this document) of the added line in the `.toc` file will propagate until it is cancelled by another explicit such command inserted in the `.toc` file.

7. Am I also red?

Well, actually no in this `scrartcl` class but I assure you I was in the `article` class!

```
\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. There is no easy general solution¹⁴ to this, but it will probably concern only a minority of cases. Again, the line styles defined by **etoc** do not use groups.

8. A comparison

The main table of contents was done in the **etoc** style. Let us issue here again two `\table-ofcontents` commands, one with the document class style (with some special centered title in place of the standard Contents heading) and the other one with the style we defined in the previous part on the basis of `enumerate` environments.

```
\setcounter{tocdepth}{4}
```

¹⁴one can insert by hand the language changes in each concerned sectioning command.

8. A comparison

Let us use the standard formatting of entries, as is provided by the document class, and a two-column layout:

```
\etocstandardlines
\etocruledstyle{% 2 is the default number of columns
  \renewcommand{\etocstoprule}{\hrule height0pt}%
  \normalfont\normalsize\rmfamily\itshape
  \parbox{.8\linewidth}{\centering
    This is the global table of contents with standard
    (to the {\ttfamily scrartcl} class) entries
    (and on two columns). Contrarily to
    \hyperref[toc:test]{\color{niceone}the next TOC}
    it contains paragraph entries.}}
\tableofcontents
```

This is the global table of contents with standard (to the `scrartcl` class) entries (and on two columns). Contrarily to [the next TOC](#) it contains paragraph entries.

I. Overview	4	5. Surprising uses of <code>etoc</code>	11
1. Initial motivation: nested lists	4	5.1. The TOC of TOCs	11
1.1. Limitations intrinsic to the use of environments	4	5.2. Arbitrary “Lists Of...”	12
2. Line styles and toc display style	4	5.3. A TOC with a fancy layout	13
2.1. <code>\etocsettocstyle</code> for the toc display	5	5.4. Another compatibility mode	15
2.2. <code>\etocsetstyle</code> for the line styles	5	II. Package commands for line styles	17
2.3. Compatibility mode	5	6. The <code>\etocsetstyle</code> command	18
3. Arbitrarily many TOCs, and local ones too	6	6.1. The <code>\etocname</code> and <code>\etocpage</code> commands	18
3.1. Labeling and reusing elsewhere	6	6.2. The <code>\etocskipfirstprefix</code> command	19
3.2. The <code>hyperref</code> option <code>bookmarksdepth</code>	6	6.3. The <code>\etocnumber</code> command	19
3.3. On manually adding layout commands to the <code>.toc</code> file	7	6.3.1. The <code>\etocifnumbered</code> switch	19
3.4. Shuffling the levels with <code>\etocsetlevel</code>	7	6.4. The <code>\etocthename</code> , <code>\etocthenumber</code> , and <code>\etocthepage</code> commands	20
3.5. Local table of contents for this part	8	6.5. The <code>\etoclink</code> command	20
4. A simple example	9	6.6. The <code>\etocsetlevel</code> command	20
		6.6.1. This is a (pale) red subsubsection for illustrative purposes	21
		7. Am I also red?	21

8. A comparison	21	11. Table of contents for this part	29
		11.1. Testing the compatibility mode	29
III. Package commands for toc display styles	25	11.2. A framed display	30
		11.3. A (crazy) inline display . .	31
9. Specifying the toc display style	25	IV. Using and customizing etoc	32
9.1. The command <code>\etocruled-style</code>	26	12. Summary of the main style commands	32
9.2. The command <code>\etocmulticolstyle</code>	26	12.1. Setting up local styles . . .	32
9.2.1. The command <code>\etoc-tocstyle</code>	26	12.2. Setting up toc display styles	33
9.2.1.1. <code>\etoc-tocstyle-with-marks</code> .	27	12.3. Displaying tables of contents	33
9.2.1.2. Do we really want paragraph entries in the TOC? .	27	12.4. Labels and references . . .	33
9.2.1.3. really? .	27	12.5. An example	33
9.3. The command <code>\etocruled</code>	27	13. Customizing etoc	36
9.4. The commands <code>\etoc-framedstyle</code> and <code>\etoc-framed</code>	27	13.1. Customizing the <code>etoc</code> predefined line styles	36
9.5. Headings, titles, <code>\etoc-innertopsep</code>	27	13.2. Customizing the toc display styles	37
9.6. The command <code>\etocset-tocstyle</code>	28	V. Tips	38
9.7. The compatibility mode <code>\etocstandarddisplay-style</code>	28	14. ... and tricks	38
10. Starred variants of the <code>\table-ofcontents</code> etc... commands	28	14.1. Hacking framed parboxes .	38
		14.2. Interverting the levels . . .	38
		14.3. All subsections of this document	39
		14.4. Compatibility with other packages	40
		14.5. T _E Xnical matters	40
		14.6. Errors and catastrophes . .	40
		VI. The code	41
		15. Implementation	41

We then switch to our set-up from the previous part and add to it:

```

\etocsetstyle{part}
{\begin{enumerate}[label=\etocnumber, leftmargin=1.5em]}
{\etocfontminusone\item}
{\etocname}
{\end{enumerate}}
\tableofcontents

```

This is the global table of contents using quickly defined (and easily improvable) enumerate environments for part, section and subsection, and an ad hoc in-line italic style for subsubsection.

I Overview

1 Initial motivation: nested lists (page 4)

- 1.1 Limitations intrinsic to the use of environments (p. 4)

2 Line styles and toc display style (page 4)

- 2.1 `\etocsettocstyle` for the toc display (p. 5)
- 2.2 `\etocsetstyle` for the line styles (p. 5)
- 2.3 Compatibility mode (p. 5)

3 Arbitrarily many TOCs, and local ones too (page 6)

- 3.1 Labeling and reusing elsewhere (p. 6)
- 3.2 The hyperref option *bookmarks-depth* (p. 6)
- 3.3 On manually adding layout commands to the `.toc` file (p. 7)
- 3.4 Shuffling the levels with `\etocsetlevel` (p. 7)
- 3.5 Local table of contents for this part (p. 8)

4 A simple example (page 9)

5 Surprising uses of `etoc` (page 11)

- 5.1 The TOC of TOCs (p. 11)
- 5.2 Arbitrary “Lists Of...” (p. 12)
- 5.3 A TOC with a fancy layout (p. 13)
- 5.4 Another compatibility mode (p. 15)

II Package commands for line styles

6 The `\etocsetstyle` command (page 18)

- 6.1 The `\etocname` and `\etocpage` commands (p. 18)

- 6.2 The `\etocskipfirstprefix` command (p. 19)

- 6.3 The `\etocnumber` command (p. 19)
The `\etocifnumbered` switch.

- 6.4 The `\etocthenname`, `\etocthenumber`, and `\etocthepage` commands (p. 20)

- 6.5 The `\etoclink` command (p. 20)

- 6.6 The `\etocsetlevel` command (p. 20)

This is a (pale) red subsubsection for illustrative purposes.

7 Am I also red? (page 21)

8 A comparison (page 21)

III Package commands for toc display styles

9 Specifying the toc display style (page 25)

- 9.1 The command `\etocruledstyle` (p. 26)

- 9.2 The command `\etocmulticolstyle` (p. 26)
The command `\etoc tocstyle`.

- 9.3 The command `\etocruled` (p. 27)

- 9.4 The commands `\etocframedstyle` and `\etocframed` (p. 27)

- 9.5 Headings, titles, `\etocinner-topsep` (p. 27)

- 9.6 The command `\etocsettocstyle` (p. 28)

- 9.7 The compatibility mode `\etocstandarddisplaystyle` (p. 28)

10 Starred variants of the `\table-ofcontents` etc... commands (page 28)

11	Table of contents for this part (page 29)	13.1	Customizing the etoc pre-defined line styles (p. 36)
11.1	Testing the compatibility mode (p. 29)	13.2	Customizing the toc display styles (p. 37)
11.2	A framed display (p. 30)		
11.3	A (crazy) inline display (p. 31)	V	Tips
IV	Using and customizing etoc	14	... and tricks (page 38)
12	Summary of the main style commands (page 32)	14.1	Hacking framed parboxes (p. 38)
12.1	Setting up local styles (p. 32)	14.2	Interverting the levels (p. 38)
12.2	Setting up toc display styles (p. 33)	14.3	All subsections of this document (p. 39)
12.3	Displaying tables of contents (p. 33)	14.4	Compatibility with other packages (p. 40)
12.4	Labels and references (p. 33)	14.5	T _E Xnical matters (p. 40)
12.5	An example (p. 33)	14.6	Errors and catastrophes (p. 40)
13	Customizing etoc (page 36)	VI	The code
		15	Implementation (page 41)

Part III.

Package commands for toc display styles

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

9. 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})
```

9. Specifying the toc display style

9.1. 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 `\fboxes` 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{\etocstoprule}{\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}}
```

9.2. 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:

<heading> = `\section*{<title>}`

[New with 1.07] An explicit `\par` not being accepted in the *<heading>* 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 `\etocmulticolstyle` all future `\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:

`\etocmulticol[<number of columns>]{<heading>}`

And there is also `\etoclocalmulticol[<number of columns>]{<heading>}`.

9.2.1. The command `\etoc tocstyle`

`\etoc tocstyle[<kind>]{<number of columns>}{<title>}`
= `\etocmulticolstyle[number of columns]{\kind*{title}}`

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

9. Specifying the toc display style

9.2.1.1. `\etocstylewithmarks` [*<kind>*] {<number_of_columns>} {<title>} {<mark>}
=`\etocmulticolstyle` [*number_of_columns*] {<kind>*{<title> \markboth{\MakeUppercase{<mark>}}}}
where *kind* is one of chapter, section, ... The actual display of the marks depends on the settings of the page style. There is variant `\etocstylewithmarksnouc` which does not uppercase.

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

9.2.1.3. really?

9.3. The command `\etocruled`

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

`\etocruled` [*number_of_columns*] {<title>}

And the local form will be `\etoclocalruled`.

9.4. The commands `\etocframedstyle` and `\etocframed`

Same mechanism:

`\etocframedstyle` [*number_of_columns*] {<title>}

and the accompanying shortcut:

`\etocframed` [*number_of_columns*] {<title>}

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`.

9.5. 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:

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

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

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`).¹⁵ 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{\langle new_value \rangle}`, not with `\setlength`.

9.6. The command `\etocsettocstyle`

This is a command with two mandatory arguments:

`\etocsettocstyle{\langle before_toc \rangle}{\langle after_toc \rangle}`

The `{\langle before_toc \rangle}` 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 `\langle after_toc \rangle` part would then reenact the two-column mode.

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

9.7. 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.

10. 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{\langle kind \rangle}{\langle name \rangle}` will change the format (`\langle kind \rangle` is

¹⁵this command `\etocoldpar` (= working `\par` in the argument to `\etocmulticolstyle`) is not related to the switch `\etocnopar` whose purpose is to tell **etoc** not to do a `\par` before the table of contents.

chapter, section, subsection... and $\langle name \rangle$ 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 `\etocmemoirtoc` to `\etocmemoirtoc` to `\etocfmt`.

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.

11. Table of contents for this part

11.1. 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.¹⁶ The original was invisibly defined with a label at the beginning 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

9. Specifying the toc display style	25
9.1. The command <code>\etocruledstyle</code>	26
9.2. The command <code>\etocmulticolstyle</code>	26
9.2.1. The command <code>\etocstyle</code>	26
9.3. The command <code>\etocruled</code>	27
9.4. The commands <code>\etocframedstyle</code> and <code>\etocframed</code>	27
9.5. Headings, titles, <code>\etocinnertopsep</code>	27
9.6. The command <code>\etocsettocstyle</code>	28
9.7. The compatibility mode <code>\etocstandarddisplaystyle</code>	28
10. Starred variants of the <code>\tableofcontents</code> etc... commands	28

¹⁶the present document uses the `scrartcl` class, and we check here that the `etoc` compatibility mode does respect the customizing done via the class commands.

11. Table of contents for this part	29
11.1. Testing the compatibility mode	29
11.2. A framed display	30
11.3. A (crazy) inline display	31

11.2. A framed display

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

*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	The compatibility mode <code>\etocstandarddisplaystyle</code> . . . 9.7, p. 28
. 9, p. 25	
The command <code>\etocruledstyle</code>	Starred variants of the <code>\tableofcontents</code> etc... commands 10, p. 28
. 9.1, p. 26	
The command <code>\etocmulticolstyle</code>	
. 9.2, p. 26	
The command <code>\etococstyle</code> .	
The command <code>\etocruled</code> 9.3, p. 27	Table of contents for this part 11, p. 29
The commands <code>\etocframedstyle</code> and <code>\etocframed</code> 9.4, p. 27	Testing the compatibility mode 11.1, p. 29
Headings, titles, <code>\etocinnertopsep</code> 9.5, p. 27	A framed display 11.2, p. 30
The command <code>\etocsettocstyle</code> 9.6, p. 28	A (crazy) inline display 11.3, p. 31

```

\etocdefaultlines
\begingroup % 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{\etocoprerule}{\hrule height 1pt}
\renewcommand{\etocoprerulecolorcmd}{\color{red!25}}
\renewcommand{\etocbottomrule}{\hrule height 1pt}
\renewcommand{\etocbottomrulecolorcmd}{\color{red!25}}
% set up the left and right rules
\renewcommand{\etocleftrule}{\vrule width 5pt}
\renewcommand{\etocrightrule}{\vrule width 5pt}
\renewcommand{\etocleftrulecolorcmd}{\color{red!25}}
\renewcommand{\etocrightrulecolorcmd}{\color{red!25}}

```

```
% use \fcolorbox to set up a colored frame for the title
\fbboxrule1pt
\etocframedstyle{\normalsize\rmfamily\itshape
  \fcolorbox{red}{white}{\parbox{.8\linewidth}{\centering
    This is a table of contents \‘a la \etoc, but just for
    the sections and subsections in this part. As it is put
    in a frame, it has to be small enough to fit on the
    current page. It has the label |toc:b|.}}}}
% set up a label for future (or earlier...) reference
\setcounter{tocdepth}{3}
\tableofcontents \label{toc:b} \ref{toc:globalcmds}
\endgroup
```

11.3. 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 inline table of contents. **Overview:** *Initial motivation: nested lists¹⁷, Line styles and toc display style¹⁸, Arbitrarily many TOCs, and local ones too¹⁹, A simple example, Surprising uses of **etoc**²⁰. **Package commands for line styles:** The **\etocsetstyle** command²¹, Am I also red?, A comparison. **Package commands for toc display styles:** Specifying the toc display style²², Starred variants of the **\tableofcontents** etc... commands, Table of contents for this part²³. **Using and customizing etoc:** Summary of the main style commands²⁴, Customizing **etoc**²⁵. **Tips:** ... and tricks²⁶. **The code:** Implementation.*

¹⁷Limitations intrinsic to the use of environments.

¹⁸**\etocsettocstyle** for the toc display; **\etocsetstyle** for the line styles; Compatibility mode.

¹⁹Labeling 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.

²⁰The TOC of TOCs; Arbitrary “Lists Of...”; A TOC with a fancy layout; Another compatibility mode.

²¹The **\etocname** and **\etocpage** commands; The **\etocskipfirstprefix** command; The **\etocnumber** command (*The \etocifnumbered switch*); The **\etocthenname**, **\etocthenumber**, and **\etocthepage** commands; The **\etoclink** command; The **\etocsetlevel** command (*This is a (pale) red subsubsection for illustrative purposes*).

²²The command **\etocruledstyle**; The command **\etocmulticolstyle** (*The command \etoc tocstyle*); The command **\etocruled**; The commands **\etocframedstyle** and **\etocframed**; Headings, titles, **\etocinnertopsep**; The command **\etocsettocstyle**; The compatibility mode **\etocstandarddisplaystyle**.

²³Testing the compatibility mode; A framed display; A (crazy) inline display.

²⁴Setting up local styles; Setting up toc display styles; Displaying tables of contents; Labels and references; An example.

²⁵Customizing the **etoc** pre-defined line styles; Customizing the toc display styles.

²⁶Hacking framed parboxes; Interverting the levels; All subsections of this document; Compatibility with other packages; T_EXnical matters; Errors and catastrophes.

And the code which was used:

```
\begingroup
\setcounter{tocdepth}{3}
\newsavebox{\forsubsections}
\etocsetstyle{part}{\etocskipfirstprefix}{. \upshape}{\bfseries\etocname:~}{ }
\etocsetstyle{section}{\itshape\etocskipfirstprefix}{, }{\mdseries\etocname}{ }
\etocsetstyle{subsection}
{ \begin{lrbox}{\forsubsections}\upshape\etocskipfirstprefix
{; }
{\etocname}
{\end{lrbox}\footnote{\unhbox\forsubsections.}}
\etocsetstyle{subsubsection}{ (\itshape\etocskipfirstprefix
{, }{\etocname}{\backslash\upshape})
\etocsettocstyle{Here is the inline table of contents. }{.\par}
\tableofcontents
\endgroup
```

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	Labels and references . . . 12.4, p. 33
. 12, p. 32	An example 12.5, p. 33
Setting up local styles . . . 12.1, p. 32	Customizing etoc 13, p. 36
Setting up toc display styles	Customizing the etoc pre-defined line
. 12.2, p. 33	styles 13.1, p. 36
Displaying tables of contents	Customizing the toc display styles . . .
. 12.3, p. 33 13.2, p. 37

12. Summary of the main style commands

12.1. Setting up local styles

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

12.2. 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⟩}
```

12.3. 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
```

12.4. 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.

12.5. An example

I got motivated by a question²⁷ 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 `\etocframedstyle` puts the title on the top rule in a centered position. This is not very conve-

²⁷tex.stackexchange.com/questions/83184

12. Summary of the main style commands

nient 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{\etoclefttrulecolorcmd}{\color{tocbackground}}
\renewcommand{\etocrighttrulecolorcmd}{\color{tocbackground}}
\renewcommand{\etocbottomrulecolorcmd}{\color{tocbackground}}
\renewcommand{\etocoprerulecolorcmd}{\color{tocbackground}}

\renewcommand{\etoclefttrule}{\vrule width 1cm}
\renewcommand{\etocrighttrule}{\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
    \leftskip2.5cm\hangindent1cm\rightskip1cm\noindent
    \hbox to 1cm{\color{subsecnum}\etocnumber\hss}%
    \color{black}\etocname\leaders\hbox to .2cm{\hss.}\hfill
    \rlap{\hbox to 1cm{\hss\etocpage\hskip.2cm}}\par
    \nointerlineskip\vskip3pt}
  {}

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

\newcommand{\coloredstuff}[2]{%
  \leftskip0pt\rightskip0pt\parskip0pt
  \fboxsep0pt % \colorbox uses \fboxsep also when no frame!
  \noindent\colorbox{secbackground}
    {\parbox{\linewidth}{%
      \vskip5pt
      {\noindent\color{#1}#2\par\nointerlineskip}
      \vskip3pt}}%
}
```

12. Summary of the main style commands

```

\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 involved²⁸ as it does not use any additional package. 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.

I am a twin of that other TOC (click me!)		
9	Specifying the toc display style	25
9.1	The command <code>\etocruledstyle</code>	26
9.2	The command <code>\etocmulticolstyle</code>	26
9.2.1	The command <code>\etoctocstyle</code>	26
9.3	The command <code>\etocruled</code>	27
9.4	The commands <code>\etocframedstyle</code> and <code>\etocframed</code> ..	27
9.5	Headings, titles, <code>\etocinnertopsep</code>	27
9.6	The command <code>\etocsettocstyle</code>	28
9.7	The compatibility mode <code>\etocstandarddisplaystyle</code> ..	28
10	Starred variants of the <code>\tableofcontents</code> etc... commands	28
11	Table of contents for this part	29
11.1	Testing the compatibility mode	29
11.2	A framed display	30
11.3	A (crazy) inline display	31

²⁸and reveals the author's preference for the T_EX syntax...

13. Customizing **etoc**

13.1. 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}
\newcommand*\etocsepthree{.25ex plus .05ex minus .05ex}

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

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

\newcommand*\etocclineleaders
{\hbox{\normalfont\normalsize\hbox to 2ex {\hss.\hss}}}

\newcommand*\etocabbrevpagename{p.\sim}
\newcommand*\etocpartname{\partname}
\newcommand*\etocbookname{Book}
```

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**.²⁹

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.

²⁹with 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.

13.2. Customizing the toc display styles

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

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

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

% for the framed style only:

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

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

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

% 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 T_EX 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 T_EX syntax for rules is accepted here.

Part V.

Tips

14. ... and tricks

14.1. Hacking framed parboxes

```
\renewcommand\etocptrule{\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>
. 9, p. 25 9.6, p. 28
The command <code>\etocruledstyle</code>	The compatibility mode <code>\etocstandard-</code>
. 9.1, p. 26	<code>displaystyle</code> 9.7, p. 28
The command <code>\etocmulticolstyle</code>	Starred variants of the <code>\tableofcon-</code>
. 9.2, p. 26	tents etc... commands 10, p. 28
The command <code>\etocstyle</code> .	
The command <code>\etocruled</code> 9.3, p. 27	Table of contents for this part
The commands <code>\etocframedstyle</code> and 11, p. 29
<code>\etocframed</code> 9.4, p. 27	Testing the compatibility mode
Headings, titles, <code>\etocinnertopsep</code> 11.1, p. 29
. 9.5, p. 27	A framed display 11.2, p. 30
	A (crazy) inline display 11.3, p. 31

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

14.3. All subsections of this document

- | | | |
|--|--|--|
| <p>i. Limitations intrinsic to the use of environments (1.1, p. 4)</p> <p>ii. <code>\etocsettocstyle</code> for the toc display (2.1, p. 5)</p> <p>iii. <code>\etocsetstyle</code> for the line styles (2.2, p. 5)</p> <p>iv. Compatibility mode (2.3, p. 5)</p> <p>v. Labeling and reusing elsewhere (3.1, p. 6)</p> <p>vi. The <code>hyperref</code> option <code>bookmarksdepth</code> (3.2, p. 6)</p> <p>vii. On manually adding layout commands to the <code>.toc</code> file (3.3, p. 7)</p> <p>viii. Shuffling the levels with <code>\etocsetlevel</code> (3.4, p. 7)</p> <p>ix. Local table of contents for this part (3.5, p. 8)</p> <p>x. The TOC of TOCs (5.1, p. 11)</p> <p>xi. Arbitrary “Lists Of...” (5.2, p. 12)</p> <p>xii. A TOC with a fancy layout (5.3, p. 13)</p> <p>xiii. Another compatibility mode (5.4, p. 15)</p> <p>xiv. The <code>\etocname</code> and <code>\etocpage</code> commands (6.1, p. 18)</p> | <p>xv. The <code>\etocskipfirst-prefix</code> command (6.2, p. 19)</p> <p>xvi. The <code>\etocnumber</code> command (6.3, p. 19)</p> <p>xvii. The <code>\etocthename</code>, <code>\etocthenumber</code>, and <code>\etocthepage</code> commands (6.4, p. 20)</p> <p>xviii. The <code>\etoclink</code> command (6.5, p. 20)</p> <p>xix. The <code>\etocsetlevel</code> command (6.6, p. 20)</p> <p>xx. The command <code>\etoc-ruledstyle</code> (9.1, p. 26)</p> <p>xxi. The command <code>\etoc-multicolstyle</code> (9.2, p. 26)</p> <p>xxii. The command <code>\etoc-ruled</code> (9.3, p. 27)</p> <p>xxiii. The commands <code>\etoc-framedstyle</code> and <code>\etoc-framed</code> (9.4, p. 27)</p> <p>xxiv. Headings, titles, <code>\etoc-innertopsep</code> (9.5, p. 27)</p> <p>xxv. The command <code>\etoc-settocstyle</code> (9.6, p. 28)</p> <p>xxvi. The compatibility mode <code>\etocstandard-displaystyle</code> (9.7, p. 28)</p> <p>xxvii. Testing the compatibility mode (11.1, p. 29)</p> | <p>xxviii. A framed display (11.2, p. 30)</p> <p>xxix. A (crazy) inline display (11.3, p. 31)</p> <p>xxx. Setting up local styles (12.1, p. 32)</p> <p>xxxi. Setting up toc display styles (12.2, p. 33)</p> <p>xxxii. Displaying tables of contents (12.3, p. 33)</p> <p>xxxiii. Labels and references (12.4, p. 33)</p> <p>xxxiv. An example (12.5, p. 33)</p> <p>xxxv. Customizing the <code>etoc</code> pre-defined line styles (13.1, p. 36)</p> <p>xxxvi. Customizing the toc display styles (13.2, p. 37)</p> <p>xxxvii. Hacking framed parboxes (14.1, p. 38)</p> <p>xxxviii. Interverting the levels (14.2, p. 38)</p> <p>xxxix. All subsections of this document (14.3, p. 39)</p> <p>xl. Compatibility with other packages (14.4, p. 40)</p> <p>xli. T_EXnical matters (14.5, p. 40)</p> <p>xlii. Errors and catastrophes (14.6, p. 40)</p> |
|--|--|--|

14.4. Compatibility with other packages

etoc loads the package `multicol`.³⁰ 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 of the present document are fully linked.

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

14.5. 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.

14.6. 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).

³⁰ up to version 1.07a it also used package `xspace`, and but this has been removed.

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 \TeX installation.

15. Implementation

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

```
1 \ProvidesPackage{etoc}
2 [2013/02/24 v1.07d 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*{\etocnopar}{\def\Etoc@par{}}
11 \newif\ifEtoc@jj % book
12 \newif\ifEtoc@j % part
13 \newif\ifEtoc@ % chapter
14 \newif\ifEtoc@i % section
15 \newif\ifEtoc@ii % subsection
16 \newif\ifEtoc@iii % subsubsection
17 \newif\ifEtoc@iv % paragraph
18 \newif\ifEtoc@v % subparagraph
19 \newif\ifEtoc@number
20 \newif\ifEtoc@hyperref
21 \newif\ifEtoc@parskip % 1.07d
22 \newif\ifEtoc@tocwithid
23 \newif\ifEtoc@standard
24 \newif\ifEtoc@part

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

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

15. Implementation

```

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

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

```

15. Implementation

```

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

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

```

15. Implementation

```

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

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

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

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

```

15. Implementation

```

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

```

15. Implementation

```

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

```

placeholder for comments

```

258 \def\Etoc@lxyz #1#2{%
259     \@namedef{etoclink }{\leavevmode}% fall-back
260     \@namedef{etocpage }{\leavevmode #2}% fall-back
261     \Etoc@getthepage #2\etoc@
262     \@namedef{etocname }{\leavevmode #1}% fall-back
263     \def\etocthename{#1}%
264     \Etoc@getnb #1\relax\relax\etoc@
265     \ifEtoc@number\else
266         \ifEtoc@part
267             \Etoc@getit #1\hspace\relax\etoc@
268         \fi
269     \fi}

```

placeholder for comments

```

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

```

15. Implementation

```

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

placeholder for comments

291 %
292 \def\Etoc@getnb@nohyp #1#2#3\etoc@{%
293 \def\Etoc@getname ##1\relax\relax\etoc@{%
294 \@namedef{etocname }{\leavevmode ##1}%
295 \def\etocthename{##1}%
296 }%
297 \ifx #1\numberline
298 \@namedef{etocnumber }{\leavevmode #2}%
299 \def\etocthenumber{#2}%
300 \Etoc@numbertrue
301 \Etoc@getname #3\etoc@
302 \else
303 \@namedef{etocnumber }{\leavevmode}%
304 \def\etocthenumber{}%
305 \Etoc@numberfalse
306 \fi
307 }

placeholder for comments

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

placeholder for comments

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

```

15. Implementation

```

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

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

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

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

15. Implementation

placeholder for comments

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

placeholder for comments

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

15. Implementation

```

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

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

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

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

```

15. Implementation

```

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

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

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

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

```

15. Implementation

```

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

placeholder for comments

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

placeholder for comments

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

placeholder for comments

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

placeholder for comments

566 \newcommand*\etocsepminustwo{4ex \@plus .5ex \@minus .5ex}
567 \newcommand*\etocsepminusone{4ex \@plus .5ex \@minus .5ex}
568 \newcommand*\etocsepzero{2.5ex \@plus .4ex \@minus .4ex}
569 \newcommand*\etocseppone{1.5ex \@plus .3ex \@minus .3ex}
570 \newcommand*\etocseptwo{1ex \@plus .15ex \@minus .15ex}

```

15. Implementation

```
571 \newcommand*\etocsepthree{.25ex \@plus .05ex \@minus .05ex}
placeholder for comments
572 \newcommand*\etocbaselinespreadadminustwo{1}
573 \newcommand*\etocbaselinespreadminusone{1}
574 \newcommand*\etocbaselinespreadzero{1}
575 \newcommand*\etocbaselinespreadone{1}
576 \newcommand*\etocbaselinespreadtwo{1}
577 \newcommand*\etocbaselinespreadthree{.9}
placeholder for comments
578 \newcommand*\etocminustwoleftmargin{1.5em plus 0.5fil}
579 \newcommand*\etocminustworightmargin{1.5em plus -0.5fil}
580 \newcommand*\etocminusoneleftmargin{1em}
581 \newcommand*\etocminusonerightmargin{1em}
582 \newcommand*\etoclineleaders
583     {\hbox{\normalfont\normalsize\hb@xt@2ex {\hss.\hss}}}
584 \newcommand*\etocabbrevpagename{p.~}
585 \newcommand*\etocpartname{\partname}
586 \newcommand*\etocbookname{Book}
placeholder for comments The macro \etocdefaultlines was initially called \etoc-
toclines. Now \etoclines just does \Etoc@standardfalse.
587 \def\etocdefaultlines{\Etoc@standardfalse}
placeholder for comments
588 \etoc@setstyle{@minustwo}{\etocskipfirstprefix}{\pagebreak[3]}
589 {\begingroup \etocfontminustwo
590   \addvspace{\etocsepminustwo}%
591   \parindent \z@
592   \leftskip \etocminustwoleftmargin
593   \rightskip \etocminustworightmargin
594   \parfillskip \@flushglue
595   \vbox{\etocifnumbered{\etocbookname~\etocnumber:~}{ }\etocname
596     \baselineskip\etocbaselinespreadadminustwo\baselineskip\par}%
597   \addvspace{\etocsepminusone}\nobreak
598 \endgroup}
599 {}%
600 \etoc@setstyle{@minusone}{\etocskipfirstprefix}{\pagebreak[3]}
601 {\begingroup \etocfontminusone
602   \addvspace{\etocsepminusone}%
603   \parindent \z@
604   \leftskip \etocminusoneleftmargin
605   \rightskip \etocminusonerightmargin
606   \parfillskip \@flushglue
607   \vbox{\etocpartname\etocifnumbered{ \etocnumber.~}{ }\etocname
608     \baselineskip\etocbaselinespreadadminusone\baselineskip\par}%
609   \addvspace{\etocsepzero}\nobreak
610 \endgroup}
611 {}%
612 \etoc@setstyle{@zero}
613 {\addpenalty\@highpenalty\etocskipfirstprefix}{\pagebreak[2]}
614 {\begingroup
615   \etocfontzero
```

15. Implementation

```

616 \addvspace{\etocsepzero}%
617 \parindent \z@ \parfillskip \@flushglue
618 \vbox{\etocifnumbered{\etocnumber\space}{\etocname
619 \baselineskip\etocbaselinespreadzero\baselineskip\par}%
620 \addvspace{\etocsepone}\addpenalty\@itempenalty
621 \endgroup}
622 {\addpenalty\@secpenalty}%

placeholder for comments
623 \etoc@setstyle{@one}
624 {\addpenalty\@medpenalty\etocskipfirstprefix}{\pagebreak[1]}
625 {\begingroup\etocfontone
626 \addvspace{\etocsepone}%
627 \parindent \z@ \parfillskip \z@
628 \setbox\z@\vbox{\parfillskip\@flushglue
629 \etocname\par\setbox\tw@\lastbox
630 \global\setbox\@ne\hbox{\unhbox\tw@ }}%
631 \dimen\z@=\wd\@ne
632 \setbox\z@=\etoclineleaders
633 \advance\dimen\z@\wd\z@
634 \etocifnumbered
635 {\setbox\tw@\hbox{\etocnumber, \etocabbrevpagename\etocpage}}
636 {\setbox\tw@\hbox{\etocabbrevpagename\etocpage}}}%
637 \advance\dimen\z@\wd\tw@
638 \ifdim\dimen\z@ < \linewidth
639 \vbox{\etocname~%
640 \leaders\box\z@\hfil\box\tw@
641 \baselineskip\etocbaselinespreadone\baselineskip\par}
642 \else
643 \vbox{\etocname~%
644 \leaders\copy\z@\hfil\break
645 \hbox{} \leaders\box\z@\hfil\box\tw@
646 \baselineskip\etocbaselinespreadone\baselineskip\par}
647 \fi
648 \addvspace{\etocseptwo}\addpenalty\@itempenalty
649 \endgroup}
650 {\addpenalty\@secpenalty}%

placeholder for comments
651 \etoc@setstyle{@two}
652 {\addpenalty\@medpenalty\etocskipfirstprefix}
653 {\pagebreak[1]}
654 {\begingroup\etocfonttwo
655 \addvspace{\etocseptwo}%
656 \parindent \z@ \parfillskip \z@
657 \setbox\z@\vbox{\parfillskip\@flushglue
658 \etocname\par\setbox\tw@\lastbox
659 \global\setbox\@ne\hbox{\unhbox\tw@}}}%
660 \dimen\z@=\wd\@ne
661 \setbox\z@=\etoclineleaders
662 \advance\dimen\z@\wd\z@
663 \etocifnumbered
664 {\setbox\tw@\hbox{\etocnumber, \etocabbrevpagename\etocpage}}

```

15. Implementation

```

665 {\setbox\tw@\hbox{\etocabbrevpagename\etocpage}}%
666 \advance\dimen\z@\wd\tw@
667 \ifdim\dimen\z@ < \linewidth
668     \vbox{\etocname~%
669     \leaders\box\z@\hfil\box\tw@
670     \baselineskip\etocbaselinespreadtwo\baselineskip\par}
671 \else
672     \vbox{\etocname~%
673     \leaders\copy\z@\hfil\break
674     \hbox{}}\leaders\box\z@\hfil\box\tw@
675     \baselineskip\etocbaselinespreadtwo\baselineskip\par}
676 \fi
677 \addvspace{\etocsepthree}\addpenalty\@itempenalty
678 \endgroup}
679 {\addpenalty\@secpenalty}%

placeholder for comments
680 \etoc@setstyle{@three}
681 {\nopagebreak
682 \etocfontthree
683 \addvspace{\etocsepthree}%
684 \nointerlineskip\noindent
685 \etocskipfirstprefix}
686 {\allowbreak\,--\,}
687 {\etocname}
688 {\hfil\beginingroup
689 \baselineskip\etocbaselinespreadthree\baselineskip
690 \par\endgroup
691 \addpenalty{-\@highpenalty}}%

placeholder for comments
692 \etoc@setstyle{@four}{}{}{}{}%
693 \etoc@setstyle{@five}{}{}{}{}%
694 }

placeholder for comments
695 \newcommand*\etocabovetocskip{3.5ex \@plus 1ex \@minus .2ex}
696 \newcommand*\etocbelowtocskip{3.5ex \@plus 1ex \@minus .2ex}
697 \newcommand*\etoccolumnsep{2em}
698 \newcommand*\etocmulticolsep{0ex}
699 \newcommand*\etocmulticolpretolerance{-1}
700 \newcommand*\etocmulticoltolerance{200}
701 \newcommand*\etocdefaultnbcol{2}
702 \newcommand*\etocinnertopsep{2ex}

placeholder for comments
703 \newcommand\etocmulticolstyle[2][\etocdefaultnbcol]{%
704 \etocsettocstyle
705     {\let\etocoldpar\par
706     \addvspace{\etocabovetocskip}%
707     \ifnum #1>\@ne\let\Etoc@next\@firstoftwo
708     \else \let\Etoc@next\@secondoftwo\fi
709     \Etoc@next}%
710     \multicolpretolerance\etocmulticolpretolerance

```

15. Implementation

```

711 \multicoltolerance\etocmulticoltolerance
712 \setlength{\columnsep}{\etoccolumnsep}%
713 \setlength{\multicolsep}{\etocmulticolsep}%
714 \begin{multicols}{#1}[#2\etocoldpar\addvspace{\etocinnertopsep}]]
715 % 2013/01/29: erroneous \etocsepmminusone at last replaced by \etocinnertop-
    sep
716 % and definition of \etocoldpar added as multicols chokes on \par as part of #2
717 {#2\par\addvspace{\etocinnertopsep}%
718 \pretolerance\etocmulticolpretolerance
719 \tolerance\etocmulticoltolerance}}
720 {\ifnum #1>\@ne\let\Etoc@next\@firstofone
721 \else \let\Etoc@next\@gobble\fi
722 \Etoc@next{\end{multicols}}}%
723 \addvspace{\etocbelowtocskip}}}

placeholder for comments
724 \newcommand*\etocinnerbottomsep{3.5ex}
725 \newcommand*\etocinnerleftsep{2em}
726 \newcommand*\etocinnerrightsep{2em}
727 \newcommand*\etoctoprule{\hrule}
728 \newcommand*\etocleftrule{\vrule}
729 \newcommand*\etocrightrule{\vrule}
730 \newcommand*\etocbottomrule{\hrule}
731 \newcommand*\etoctoprulecolorcmd{\relax}
732 \newcommand*\etocbottomrulecolorcmd{\relax}
733 \newcommand*\etocleftrulecolorcmd{\relax}
734 \newcommand*\etocrightrulecolorcmd{\relax}

placeholder for comments
735 \def\etoc@ruledheading #1{%
736 \hb@xt@\linewidth{\color@begingroup
737 \hss #1\hss\hskip-\linewidth
738 \etoctoprulecolorcmd\leaders\etoctoprule\hss
739 \phantom{#1}%
740 \leaders\etoctoprule\hss\color@endgroup}}%
741 \nointerlineskip\vskip\etocinnertopsep}

placeholder for comments
742 \newcommand*\etocruledstyle[2][\etocdefaultnbcoll]{%
743 \etocsettocstyle
744 {\addvspace{\etocabovetocskip}%
745 \ifnum #1>\@ne\let\Etoc@next\@firstoftwo
746 \else \let\Etoc@next\@secondoftwo\fi
747 \Etoc@next
748 {\multicolpretolerance\etocmulticolpretolerance
749 \multicoltolerance\etocmulticoltolerance
750 \setlength{\columnsep}{\etoccolumnsep}%
751 \setlength{\multicolsep}{\etocmulticolsep}%
752 \begin{multicols}{#1}[\etoc@ruledheading{#2}]]
753 {\etoc@ruledheading{#2}\nobreak
754 \pretolerance\etocmulticolpretolerance
755 \tolerance\etocmulticoltolerance}}
756 {\ifnum #1>\@ne\let\Etoc@next\@firstofone
757 \else \let\Etoc@next\@gobble\fi

```

15. Implementation

```

758 \Etoc@next{\end{multicols}}}%
759 \addvspace{\etocbelowtocskip}}}

placeholder for comments
760 \newcommand\etocframedmphook{\relax}
761 \newcommand*\etocbkgcolorcmd{\relax}
762 \def\Etoc@relax{\relax}
763 \newbox\etoc@framed@titlebox
764 \newbox\etoc@framed@contentsbox
765 \newcommand*\etocframedstyle[2][\etocdefaultnbcoll]{%
766 \etocsettocstyle{%
767 \addvspace{\etocabovetocskip}%
768 \sbox\z@{\#2}%
769 \dimen\z@{\dp\z@
770 \ifdim\wd\z@<\linewidth \dp\z@\z@ \else \dimen\z@\z@ \fi
771 \setbox\etoc@framed@titlebox=\hb@xt@\linewidth{\color@begingroup
772 \hss
773 \ifx\etocbkgcolorcmd\Etoc@relax\else
774 \sbox\tw@{\color{white}%
775 \vrule\@width\wd\z@\@height\ht\z@\@depth\dimen\z@}%
776 \ifdim\wd\z@<\linewidth \dp\tw@\z@\fi
777 \box\tw@
778 \hskip-\wd\z@
779 \fi
780 \copy\z@
781 \hss
782 \hskip-\linewidth
783 \etocoprulcolorcmd\leaders\etocoprul\hss%
784 \hskip\wd\z@
785 \etocoprulcolorcmd\leaders\etocoprul\hss\color@endgroup}%
786 \setbox\z@\hbox{\etocleftrule\etocrightrule}%
787 \dimen\tw@\linewidth\advance\dimen\tw@-\wd\z@
788 \advance\dimen\tw@-\etocinnerleftsep
789 \advance\dimen\tw@-\etocinnerrightsep
790 \setbox\etoc@framed@contentsbox=\vbox\bgroup
791 \hsize\dimen\tw@
792 \kern\dimen\z@
793 \vskip\etocinnertopsep
794 \hbox\bgroup
795 \begin{minipage}{\hsize}%
796 \etocframedmphook
797 \ifnum #1>\@ne\let\Etoc@next\@firstoftwo
798 \else \let\Etoc@next\@secondoftwo\fi
799 \Etoc@next
800 {\multicolpretolerance\etocmulticolpretolerance
801 \multicoltolerance\etocmulticoltolerance
802 \setlength{\columnsep}{\etoccolumnsep}%
803 \setlength{\multicolsep}{\etocmulticolsep}%
804 \begin{multicols}{#1}}
805 {\pretolerance\etocmulticolpretolerance
806 \tolerance\etocmulticoltolerance}}
807 {\ifnum #1>\@ne\let\Etoc@next\@firstofone
808 \else \let\Etoc@next\@gobble\fi

```

15. Implementation

```

809 \Etoc@next{\end{multicols}\unskip}%
810 \end{minipage}%
811 \egroup
812 \vskip\etocinnerbottomsep
813 \egroup
814 \vbox{\hsize\linewidth
815     \ifx\etocbkgcolorcmd\Etoc@relax\else
816         \kern\ht\etoc@framed@titlebox
817         \kern\dp\etoc@framed@titlebox
818         \hb@xt@\linewidth{\color@begingroup
819             \etocleftrulecolorcmd\etocleftrule
820             \etocbkgcolorcmd
821             \leaders\vrule
822                 \@height\ht\etoc@framed@contentsbox
823                 \@depth\dp\etoc@framed@contentsbox
824             \hss
825             \etocrightrulecolorcmd\etocrightrule
826             \color@endgroup}\nointerlineskip
827             \vskip-\dp\etoc@framed@contentsbox
828             \vskip-\ht\etoc@framed@contentsbox
829             \vskip-\dp\etoc@framed@titlebox
830             \vskip-\ht\etoc@framed@titlebox
831         \fi
832     \box\etoc@framed@titlebox\nointerlineskip
833     \hb@xt@\linewidth{\color@begingroup
834     {\etocleftrulecolorcmd\etocleftrule}%
835     \hss\box\etoc@framed@contentsbox\hss
836     \etocrightrulecolorcmd\etocrightrule\color@endgroup}
837     \nointerlineskip
838     \vskip\ht\etoc@framed@contentsbox
839     \vskip\dp\etoc@framed@contentsbox
840     \hb@xt@\linewidth{\color@begingroup\etocbottomrulecolorcmd
841         \leaders\etocbottomrule\hss\color@endgroup}}
842     \addvspace{\etocbelowtocskip}}

placeholder for comments

843 \newcommand\etoc@multicoltoc[2][\etocdefaultnbcoll]{%
844     \etocmulticolstyle[#1]{#2}%
845     \tableofcontents}
846 \newcommand\etoc@multicoltoctoci[2][\etocdefaultnbcoll]{%
847     \etocmulticolstyle[#1]{#2}%
848     \tableofcontents*}
849 \newcommand\etoc@local@multicoltoc[2][\etocdefaultnbcoll]{%
850     \etocmulticolstyle[#1]{#2}%
851     \localtableofcontents}
852 \newcommand\etoc@local@multicoltoctoci[2][\etocdefaultnbcoll]{%
853     \etocmulticolstyle[#1]{#2}%
854     \localtableofcontents*}

placeholder for comments

855 \newcommand*\etoc@ruledtoc[2][\etocdefaultnbcoll]{%
856     \etocruledstyle[#1]{#2}%
857     \tableofcontents}

```

15. Implementation

```
858 \newcommand*\etoc@ruledtoci[2][\etocdefaultnbcol]{%
859   \etocruledstyle[#1]{#2}%
860   \tableofcontents*}
861 \newcommand*\etoc@local@ruledtoc[2][\etocdefaultnbcol]{%
862   \etocruledstyle[#1]{#2}%
863   \localtableofcontents}
864 \newcommand*\etoc@local@ruledtoci[2][\etocdefaultnbcol]{%
865   \etocruledstyle[#1]{#2}%
866   \localtableofcontents*}

placeholder for comments

867 \newcommand*\etoc@framedtoc[2][\etocdefaultnbcol]{%
868   \etocframedstyle[#1]{#2}%
869   \tableofcontents}
870 \newcommand*\etoc@framedtoci[2][\etocdefaultnbcol]{%
871   \etocframedstyle[#1]{#2}%
872   \tableofcontents*}
873 \newcommand*\etoc@local@framedtoc[2][\etocdefaultnbcol]{%
874   \etocframedstyle[#1]{#2}%
875   \localtableofcontents}
876 \newcommand*\etoc@local@framedtoci[2][\etocdefaultnbcol]{%
877   \etocframedstyle[#1]{#2}%
878   \localtableofcontents*}

placeholder for comments

879 \def\etocmulticol{\begingroup
880   \Etoc@mustclosegrouptrue
881   \@ifstar
882   {\etoc@multicoltoctoci}
883   {\etoc@multicoltoctoc}}
884 \def\etocruled{\begingroup
885   \Etoc@mustclosegrouptrue
886   \@ifstar
887   {\etoc@ruledtoci}
888   {\etoc@ruledtoc}}
889 \def\etocframed{\begingroup
890   \Etoc@mustclosegrouptrue
891   \@ifstar
892   {\etoc@framedtoci}
893   {\etoc@framedtoc}}
894 \def\etoclocalmulticol{\begingroup
895   \Etoc@mustclosegrouptrue
896   \@ifstar
897   {\etoc@local@multicoltoctoci}
898   {\etoc@local@multicoltoctoc}}
899 \def\etoclocalruled{\begingroup
900   \Etoc@mustclosegrouptrue
901   \@ifstar
902   {\etoc@local@ruledtoci}
903   {\etoc@local@ruledtoc}}
904 \def\etoclocalframed{\begingroup
905   \Etoc@mustclosegrouptrue
906   \@ifstar
```

15. Implementation

```

907 {\etoc@local@framedtoci}
908 {\etoc@local@framedtoc}}
placeholder for comments
909 \def\etocarticlestyle{%
910   \etocsettocstyle
911   {\section *{\contentsname
912     \@mkboth {\MakeUppercase \contentsname}
913     {\MakeUppercase \contentsname}}}
914   {}}
915 \def\etocarticlestylenomarks{%
916   \etocsettocstyle
917   {\section *{\contentsname}}
918   {}}
placeholder for comments
919 \def\etocbookstyle{%
920   \etocsettocstyle
921   {\if@twocolumn \@restonecoltrue \onecolumn \else \@restonecolfalse \fi
922     \chapter *{\contentsname
923       \@mkboth {\MakeUppercase \contentsname}
924       {\MakeUppercase \contentsname}}}
925   {\if@restonecol \twocolumn \fi}}
926 \def\etocbookstylenomarks{%
927   \etocsettocstyle
928   {\if@twocolumn \@restonecoltrue \onecolumn \else \@restonecolfalse \fi
929     \chapter *{\contentsname}}
930   {\if@restonecol \twocolumn \fi}}
931 \let\etocreportstyle\etocbookstyle
932 \let\etocreportstylenomarks\etocbookstylenomarks
933 \def\etocmemoirtotocfmt #1#2{%
934   \def\Etoc@addsuitablecontentsline{\addcontentsline {toc}{#1}{#2}}%
935   \renewcommand*\etocaftertitlehook{%
936     \ifmem@em@starred@listof
937     \else\phantomsection\aftergroup\Etoc@addsuitablecontentsline\fi}}
938 \def\etocmemoirstyle{%
939   \etocsettocstyle
940   {\ensureonecol \par \begingroup \@nameuse {@tocmaketitle}
941     \Etoc@aftertitlehook\let\Etoc@aftertitlehook\relax
942     \parskip \cftparskip \@nameuse {cfttocbeforelisthook}}
943   {\@nameuse {cfttocafterlisthook}\endgroup\restorefromonecol}}
placeholder for comments
944 \def\etocscartclstyle{%
945   \etocsettocstyle
946   {\let\if@dynlist\if@tocleft
947     \iftocfeature {toc}{onecolumn}
948     {\iftocfeature {toc}{leveldown}
949       {}
950       {\if@twocolumn \aftergroup \twocolumn \onecolumn \fi }}
951     {}%
952     \tocbasic@listhead {\listoftocname}%
953     \begingroup \expandafter \expandafter \expandafter
954     \endgroup \expandafter

```

15. Implementation

```

955 \ifx
956 \csname microtypesetup\endcsname \relax
957 \else
958 \iftocfeature {toc}{noprotrusion}{}
959 {\microtypesetup {protrusion=false}%
960 \PackageInfo {tocbasic}%
961 {character protrusion at toc deactivated}}%
962 \fi
963 \setlength {\parskip }{\z@ }%
964 \setlength {\parindent }{\z@ }%
965 \setlength {\parfillskip }{\z@ \@plus 1fil}%
966 \csname tocbasic@@before@hook\endcsname
967 \csname tb@toc@before@hook\endcsname}
968 {\csname tb@toc@after@hook\endcsname
969 \csname tocbasic@@after@hook\endcsname}}
970 \let\etocscrbookstyle\etocscrartclstyle
971 \let\etocscrreprtstyle\etocscrartclstyle

placeholder for comments
972 \newcommand*\etocstandarddisplaystyle{\etocarticlestyle}
973 \newcommand*\etocmarkboth[1]{%
974 \mkboth{\MakeUppercase{#1}}{\MakeUppercase{#1}}}
975 \newcommand*\etocmarkbothnouc[1]{\mkboth{#1}{#1}}
976 \newcommand\etoc tocstyle[3][section]{\etocmulticolstyle[#2]%
977 {\csname #1\endcsname *{#3}}}
978 \newcommand\etoc tocstylewithmarks[4][section]{\etocmulticolstyle[#2]%
979 {\csname #1\endcsname *{#3}\etocmarkboth{#4}}}}
980 \newcommand\etoc tocstylewithmarksnouc[4][section]{\etocmulticolstyle[#2]%
981 {\csname #1\endcsname *{#3}\etocmarkbothnouc{#4}}}}

placeholder for comments
982 \def\Etoc@redefetocstyle#1{%
983 \renewcommand\etoc tocstylewithmarks[4][#1]
984 {\etocmulticolstyle[##2]%
985 {\csname ##1\endcsname *{##3}\etocmarkboth{##4}}}}
986 \renewcommand\etoc tocstylewithmarksnouc[4][#1]
987 {\etocmulticolstyle[##2]%
988 {\csname ##1\endcsname *{##3}\etocmarkbothnouc{##4}}}}
989 \renewcommand\etoc tocstyle[3][#1]{%
990 \etocmulticolstyle[##2]{\csname ##1\endcsname *{##3}}}}
991 \ifclassloaded{scrartcl}
992 {\renewcommand*\etocstandarddisplaystyle{\etocscrartclstyle}}{}
993 \ifclassloaded{book}
994 {\renewcommand*\etocfontone{\normalfont\normalsize}
995 \renewcommand*\etocstandarddisplaystyle{\etocbookstyle}
996 \Etoc@redefetocstyle{chapter}}{}
997 \ifclassloaded{report}
998 {\renewcommand*\etocfontone{\normalfont\normalsize}
999 \renewcommand*\etocstandarddisplaystyle{\etocreportstyle}
1000 \Etoc@redefetocstyle{chapter}}{}
1001 \ifclassloaded{scrbook}
1002 {\renewcommand*\etocfontone{\normalfont\normalsize}
1003 \renewcommand*\etocstandarddisplaystyle{\etocscrbookstyle}

```

15. Implementation

```
1004 \Etoc@redefetocstyle{chapter}}{}
1005 \ifclassloaded{scrreprt}
1006   {\renewcommand*\etocfontone{\normalfont\normalsize}
1007     \renewcommand*\etocstandarddisplaystyle{\etocscrreprtstyle}
1008     \Etoc@redefetocstyle{chapter}}{}
1009 \ifclassloaded{memoir}
1010   {\renewcommand*\etocfontone{\normalfont\normalsize}
1011     \etocmemoirtocformat{chapter}{\contentsname}%
1012     \renewcommand*\etocstandarddisplaystyle{\etocmemoirstyle}
1013     \Etoc@redefetocstyle{chapter}}{}

placeholder for comments
1014 \def\Etoc@addtocontents #1#2%
1015   {\ifEtoc@hyperref
1016     \addtocontents {toc}{\protect\contentsline
1017       {#1}{#2}%
1018       {\thepage }{\@currentHref }}%
1019   \else
1020     \addtocontents {toc}{\protect\contentsline
1021       {#1}{#2}{\thepage }}%
1022   \fi}
1023 \def\Etoc@addcontentsline@ #1#2#3%
1024   {\@namedef{toclevel@#1}{#3}%
1025     \addcontentsline {toc}{#1}{#2}}
1026 \DeclareRobustCommand*\etoccontentsline{
1027   {\@ifstar{\Etoc@addcontentsline@}{\Etoc@addtocontents}}
1028 %
1029 \newcommand*\etocstandardlines{\Etoc@standardtrue}
1030 \newcommand*\etocnoclines{\Etoc@standardfalse} % 1.07b
1031 %
1032 \etocdefaultlines % for initialization
1033 \etocstandardlines % removed silly AtBeginDocument temporarily added for 1.07
1034 \etocstandarddisplaystyle
1035 \endinput
```